CLASSIFICATION OF MULTI-SPECTRAL DATA BY JOINT SUPERVISED-UNSUPERVISED LEARNING

BEHZAD SHAHSHAHANI
Purdue University School of Electrical Engineering

DAVID LANDGREBE
Purdue University School of Electrical Engineering

Follow this and additional works at: http://docs.lib.purdue.edu/ecetr

http://docs.lib.purdue.edu/ecetr/170

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.
CLASSIFICATION OF MULTI-SPECTRAL DATA BY JOINT SUPERVISED-UNSUPERVISED LEARNING

BEHZAD SHAHSAHANI
DAVID LANDGREBE

TR-EE 94-1
JANUARY 1994

SCHOOL OF ELECTRICAL ENGINEERING
PURDUE UNIVERSITY
WEST LAFAYETTE, INDIANA 47907-1285

This work was sponsored in part by NASA under Grant NAGW-925, and by National Science Foundation Parallel Infrastructure Grant CDA-9015696
# TABLE OF CONTENTS

ABSTRACT ................................................................................................................................................. iv

CHAPTER 1: INTRODUCTION ...................................................................................................................... 1
  1.1 Statement of the problem .................................................................................................................. 1
  1.2 Definitions ........................................................................................................................................ 4
  1.3 Organization of Thesis ..................................................................................................................... 5

CHAPTER 2: EFFECT OF UNLABELED SAMPLES IN REDUCING THE SMALL SAMPLE SIZE PROBLEM AND MITIGATING THE HUGHES PHENOMENON ...................................................................................................................... 7
  2.1 Induction ........................................................................................................................................... 7
  2.2 Classification Performance Versus Classifier Type ........................................................................... 8
  2.3 Effect of Small Training Sample Size ............................................................................................... 12
  2.4 Effect of Additional Unlabeled Samples ......................................................................................... 16
  2.5 Effect of Unlabeled Samples in Reducing the Unrepresentative Training Samples Problem ......... 23
  2.6 Discussion and Concluding Remarks ............................................................................................... 31

CHAPTER 3: THE EFFECT OF LIMITED DESIGN SAMPLES IN UNSUPERVISED AND COMBINED SUPERVISED-UNSUPERVISED PARAMETRIC LEARNING .................................................................................................................. 35
  3.1 Introduction ...................................................................................................................................... 36
  3.2 Preliminaries and Background ......................................................................................................... 38
  3.3 Bounds for the Asymptotic Covariance Matrix of the ML Estimates ............................................. 40
  3.4 Effect of Limited Design Samples .................................................................................................. 48
  3.5 Combined Supervised-Unsupervised Learning ............................................................................... 66
  3.6 Conclusions ..................................................................................................................................... 70

CHAPTER 4: THE MULTIPLE CLASS PROBLEM ......................................................................................... 73
  4.1 Introduction ....................................................................................................................................... 73
  4.2 Supervised Learning ......................................................................................................................... 76
  4.3 Unsupervised Learning ..................................................................................................................... 79
  4.4 Combined Supervised-Unsupervised Learning ............................................................................... 87
  4.5 Discussion and Illustrations ............................................................................................................ 90
  4.6 Experimental Results ...................................................................................................................... 94
  4.7 Conclusions and Remarks ................................................................................................................. 100

CHAPTER 5: SEMI-PARAMETRIC COMBINED SUPERVISED-UNSUPERVISED LEARNING .......................................................... 101
  5.1 Previous Work on Normal Mixture Density Identification .............................................................. 102
  5.2 The Expectation Maximization (EM) Method .................................................................................... 105
  5.3 EM Algorithm for Semi-Parametric Mixture Density Identification ............................................ 106
  5.4 Constrained EM Algorithm ........................................................................................................... 111
  5.5 Modified EM Algorithm ................................................................................................................. 113
  5.6 Computational Complexity and Parallel Implementation of the EM Algorithm ............................. 115

- iii -
ABSTRACT

An important problem in pattern recognition is the effect of small design sample size on classification performance. When the ratio of the number of training samples to the number of feature measurements is small, the estimates of the discriminant functions are not accurate and therefore the classification results might not be satisfactory. This problem is becoming more and more important in remote sensing, as the number of available spectral bands is becoming greater and greater. In order to utilize fully the information contained in the high dimensional data, training samples are needed from all of the classes of interest. A large number of classes of interest, and a large number of features to be used, necessitate a large number of training samples. Such training samples are usually very expensive and time consuming to acquire.

In this thesis, we study the use of unlabeled samples, that are usually available in large numbers and with no extra effort, in reducing the small sample size problems. It is shown that by adding the unlabeled samples to the classifier design process, better estimates for the discriminant functions can be obtained. Therefore, the peaking phenomenon that is observed in the performance versus dimensionality curves, can be mitigated. Bounds on the expected amount of improvement in the classification performance are derived for the case of two multivariate Gaussian classes with a known common covariance matrix. These bounds, explicitly show the relationship between dimensionality and samples size for the case when parameters are estimated by simultaneously using training and unlabeled samples. A semi-parametric method for estimating the parameters of the class density functions, that uses both training and unlabeled samples, is proposed, and its parallel implementation is discussed. The problem of density model selection for classification is studied. An algorithm based on backtrack search strategy is presented for generating candidate models for the density functions. The candidate models are evaluated by several criteria that are based on both training and unlabeled samples.
CHAPTER 1: INTRODUCTION

1.1 Statement of the problem

An important problem in pattern recognition is the effect of small design sample size on classification performance. It is well known that when the ratio of the number of training samples to the number of feature measurements is small, the estimates of the discriminant functions are not accurate and therefore the classification results may not be satisfactory. This problem is becoming more and more important in remote sensing, as the number of available spectral bands is becoming greater and greater. The new generation of the remote sensing sensors that is proposed for the Earth Observing System (EOS) can produce data in a large number of spectral bands. The MODIS sensor produces data in about 50 bands [1], whereas the AVIRIS sensor produces as many as 200 spectral bands [2]. One objective of using such high spectral resolution sensors is to discriminate among more ground cover classes and hence obtain a better understanding about the nature of the materials that cover the surface of the earth. It is hypothesized that details such as differences among various types of the same plant species that were not possible to observe using the older generations of sensors, such as Thematic-Mapper of Landsat, would be discriminable by using the higher resolution sensors. In order to fully utilize the information contained in the new feature measurements, training samples are needed from all of the classes of interest. A large number of classes of interest, and a large number of spectral bands to be used, necessitate a large number of training samples. Such training samples are usually very expensive and time consuming to acquire. For remote
sensing applications, ground truth information must be gathered either by visual inspection of the scene near the same time that the data are being taken, or by using an experienced analyst for identifying the class labels of data based on their spectral responses. In either case, usually only a limited number of training samples can be obtained. These training samples are often used for deciding what features in data are useful for discriminating among the classes of interest, and for designing classifiers based on these derived features. Figure 1.1 illustrates a typical scenario for analyzing remote sensing data.

![Figure 1.1: Typical steps in the analysis of remote sensing data](image)

Usually, both the feature extraction and the classification stages of the analysis are based on optimizing a criterion that needs to be estimated using the training samples. If the number of training samples is small compared to the dimensionality of the data, both of
these stages might suffer from bad estimates; therefore, the resulting performance of the final analysis could be unsatisfactory.

An additional problem that usually exists in remote sensing applications is the unrepresentative training samples problem. Since usually training samples are obtained from spatially adjacent regions, they may not be good representatives of the samples of the entire same class that might exist in all regions in the scene. In addition, training samples from some classes might not exist. These problems further aggravate the difficulties in analyzing remote sensing data.

The purpose of this research is to explore and study some techniques for reducing the small sample size problems by utilizing unclassified observations that may be available in large numbers and with no extra cost. We refer to these unclassified samples as unlabeled data. Including the unlabeled samples in the process of designing classifiers can have the following potential advantages: 1) The large number of unlabeled samples can enhance the estimates of the parameters and therefore reduce the effect of small sample size problem. 2) The estimates of the parameters that are obtained by using the training samples may be updated by using additional unlabeled samples in order to obtain statistics that are more representative of the true class distributions. 3) The unlabeled samples may provide information about the classes from which no training samples are available. 4) The prior probabilities of the classes that can not be found by training samples alone may be estimated by using unlabeled samples. In what follows the definitions of some terms that are used frequently throughout this thesis are provided.
1.2 Definitions

"Supervised Learning": The process of designing a classifier by using training samples from the classes of interest is referred to as supervised learning. The type of classifier chosen depends on the criterion based on which the classification result is judged. For example, the maximum a posteriori (MAP) classifier is the optimal decision rule for minimizing the total classification error, whereas the maximum likelihood (ML) classifier minimizes the average classification error. Usually, the decision rule can be written in terms of the probability density functions (pdf's) of the classes or their parameters. These pdf's (or their parameters) are estimated by using the training samples in supervised learning. The common approach in remote sensing is to assume a Gaussian form for the pdf's of the classes and estimate the mean vectors and covariance matrices using the training samples. If the Gaussian assumption is correct, the regular maximum likelihood estimators for mean and covariance are consistent and therefore at limit when the number of training samples are infinite, the optimal decision rule can be constructed by supervised learning.

"Unsupervised Learning": The process of designing a classifier by using unlabeled samples is referred to as unsupervised learning. The term clustering is usually used synonymously with unsupervised learning. If the chosen decision rule can be written in terms of the pdf's of the classes, then unsupervised learning can be thought of as the process of estimating the class conditional pdf's from a set of random samples drawn from the mixture of the classes. A related topic to unsupervised learning is the problem of identifiability of mixture densities [3]. Assuming that the class-conditional pdf's all belong to a class of probability density functions, e.g., they are all multivariate Gaussian, then in order for unsupervised learning to be meaningful, it is necessary that the knowledge of the mixture density provide enough information for uniquely decomposing
it into its components. In other words, only one combination of the component densities can construct the mixture density. Such a mixture density is said to be identifiable. Necessary and sufficient conditions for identifiability of mixture densities have been studied previously [4]. It has been shown that the commonly used family of multivariate Gaussian pdfs is an identifiable family [4]. In addition to identifiability, in order for unsupervised learning to be possible, there must also exist consistent estimators for the parameters of the component pdfs of the mixture density. If such estimators exist, then at the limit when the number of unlabeled samples is infinite, true values of the class-conditional density functions can be obtained by using unlabeled samples and therefore the optimal classifier can be constructed. For the commonly used multivariate Gaussian case, consistent maximum likelihood estimates can be obtained [5]. More on this subject will be said throughout this thesis.

"Combined Supervised-Unsupervised Learning": The process of designing a classifier by using both training and unlabeled samples is referred to as combined supervised-unsupervised learning.

1.3 Organization of Thesis
In chapter 2, the effect of small training sample size in the performance of classifiers is studied. It is shown that by using additional unlabeled samples the error rate may be reduced, therefore, the Hughes phenomenon [6], which is the peaking effect in the performance versus feature size curve, can be mitigated. Chapter 3 provides a quantitative comparison between training samples and unlabeled samples based on their value in reducing the classification error. Bounds on the error rates of some classifiers are obtained when supervised, unsupervised, and combined supervised-unsupervised learning
are used for estimating the parameters. In chapter 4, the multi-class problem is studied by comparing the estimates of the mean vectors and covariance matrices obtained using training samples, unlabeled samples, and both. In chapter 5, the necessary equations for obtaining the maximum likelihood estimates are derived when classes are assumed to have multiple Gaussian components. These equations can be used for semi-parametric combined supervised-unsupervised learning. The computational complexity of the derived equations and their implementation on a massively parallel computer system are discussed. In chapter 6, the problem of density model selection based on both training and unlabeled samples is discussed. The equations derived in chapter 5 are used in a search algorithm for finding a suitable model for the feature space pdf. Experimental results are provided throughout the thesis.


2.1 Introduction

In a typical classification problem, the objective is to assign a class label, from a set of candidate labels, to an incoming observation. The minimum expected error that can be achieved in performing the classification process is referred to as the Bayes' error. A decision rule that assigns a sample to the class with maximum a posteriori probability (the MAP classifier), achieves the Bayes' error [7]. In order to design such a classifier, knowledge of the a posteriori probabilities and thus the class-conditional probability density functions is required. If such knowledge is available then by increasing the dimensionality of data one would expect to enhance the performance. In other words, the Bayes error is a non-increasing function of the dimensionality of the data. After all, a new feature can only add information about a sample and thus, one would expect to do at least as well as if such information were not available. In practice, however, class conditional probability density functions (pdfs) need to be estimated from a set of training samples. When these estimates are used in place of the true values of the pdfs the resulting decision rule is sub-optimal and hence has a higher probability of error. The expected value of the probability of error taken over all training sample sets of a particular size is, therefore, larger than the Bayes error. When a new feature is added to the data the Bayes error decreases, but at the same time the bias of the classification error increases. This
increase is due to the fact that more parameters must be estimated from the same number of samples. If the increase in the bias of the classification error is more than the decrease in the Bayes error, then the use of the additional feature degrades the performance of the decision rule. This effect is called the Hughes phenomenon [6]. The larger the number of the parameters that need to be estimated, the more severe the Hughes phenomenon can become. Therefore, when dimensionality of data and complexity of the decision rule increase, the Hughes effect becomes more severe.

2.2 Classification Performance vs. Classifier Type

The functional form of a classifier determines the shape of the decision boundaries that it can produce. Linear classifiers, such as the Minimum Euclidean Distance (MED) classifier, which is optimal when classes are Gaussian with identity covariance matrices, can produce hyper-plane boundaries, whereas quadratic classifiers, such as the Gaussian Maximum Likelihood (GML) classifier, which is optimal when the classes are Gaussian with different covariance matrices, can produce boundaries with second order surfaces. More complex classifiers can create even more complex boundaries. Obviously, the more complex the classifier, the more powerful it is in terms of its ability to discriminate among various classes of different shapes. In remote sensing, it has been observed that quadratic classifiers that take advantage of the second order statistics of the classes, e.g., GML classifiers, are very powerful for discrimination [8]. The value of the second order statistics is evidently more prominent when the dimensionality of the data is high. In high dimensions it seems that the second order variations of the classes contain more information than the first order variations [8]. In order to demonstrate this fact, the following experiments were performed (additional similar experiments are reported in [8]):
Experiment 1 (AVIRIS data):

A portion of an AVIRIS frame (consisting of 210 bands) taken over Tippecanoe county in Indiana was used in this experiment. Four ground cover classes were determined by consulting the ground truth map. The classes were bare soil (380 pixels), wheat (513 pixels), soybean (741 pixels), and corn (836 pixels). The average pair-wise Bhattacharyya distance between the classes was computed for every fifth band of the AVIRIS data. These bands were then ranked according to the average Bhattacharyya distance. The dimensionality of the data was incremented from 1 to 18 by sequentially adding more bands, i.e., for dimension 1 the first ranked band was used, for dimension two the first two ranked bands were used and so on. In other words, we always add one feature to the best previous set of features. In this way at each dimensionality all the information in the previous dimensions was present. One hundred training samples were drawn randomly from each class. The statistics of each class (mean vector and covariance matrix) were estimated by the maximum likelihood (ML) estimators. The rest of the samples were classified using the MED and GML classifiers, and the total classification accuracy (the ratio of the number of correctly classified samples to the total number of samples) was computed. Each experiment was repeated ten times independently and the average of the ten trials was obtained. The results are shown in Figure 2.1.
Experiment 2 (FLC1 Data):

A similar experiment was performed on a portion of the Flight Line C1 (FLC1) data set, which is a 12 band multispectral image taken over Indiana. Four ground cover classes were determined using the ground truth map: corn (2436 pixels), soybean (2640 pixels), wheat (2365 pixels) and red clover (2793 pixels). The bands were ranked according to the average Bhattacharyya distance and the dimensionality was incremented from 1 to 12 by sequentially adding bands as described in experiment 1. From each class 100 training
samples were drawn randomly and the statistics of each class were estimated. The rest of the samples were classified once using the MED classifier and once using the GML classifier. Each experiment was performed 10 times independently and the average of the ten trials was obtained. The results are shown in Figure 2.2.

![Figure 2.2: Classification accuracies of the MED and GML classifiers versus dimensionality for the FLCl data set based on 100 training samples per class](image)
From graphs 2.1 and 2.2 it is seen that the GML classifier that takes advantage of the second order statistics of the classes, and creates hyperbolic boundaries is more powerful in discriminating among the classes especially when the dimensionality of the data increases. However, the number of the parameters in the GML classifier is more than that in the MED classifier. As the dimensionality grows, the number of entries in the covariance matrices of the classes increases rapidly. Therefore, when the dimensionality of the data begins to approach the number of training samples, one would expect the Hughes phenomenon to effect the GML classifier more severely. This point will be discussed in more detail in the next sub-section.

2.3 Effect of Small Training Sample Size

Consider a classification problem involving m classes with prior probabilities $P_i$ and probability density functions $f_i(x)$. By $e^*$ we denote the Bayes' error achieved by using the MAP classifier when $P_i$ and $f_i(x)$ are known. Let $\theta$ denote the vector of parameters of the MAP classifier. If the pdfs are parametric (such as multivariate Gaussian), $\theta$ usually includes the parameters of each class (e.g. mean vectors and covariance matrices) and the associated prior probabilities. On the other hand, if $f_i(x)$ are not considered to be parametric, $\theta$ is assumed to contain the value of $f_i(x)$ at each particular sample $x$ under consideration. Let $\theta^*$ denote the true value of $\theta$. The error achieved by using $\hat{\theta}$ in the decision rule is $e^*$, the Bayes error. Now, assume that $\hat{\theta}$ is an estimate of $\theta^*$. If the deviation of $\hat{\theta}$ from $\theta^*$ is not large, one can approximate the error corresponding to the decision rule obtained using $\hat{\theta}$ by using a Taylor series expansion of up to the second term:
2 Mitigating The Hughes Phenomenon

\[
\hat{e} = e(\hat{\theta}) = e^* + \frac{\partial e^T(\theta)}{\partial \theta} \bigg|_{\theta = \theta^*} (\hat{\theta} - \theta^*) + \frac{1}{2} \text{tr} \left[ \frac{\partial^2 e(\theta)}{\partial \theta^2} \right]_{\theta = \theta^*} (\hat{\theta} - \theta^*)(\hat{\theta} - \theta^*)^T
\]  

(2.1)

where \( \text{tr}(A) \) denotes the trace of matrix \( A \). The term \( \frac{\partial e^T(\theta)}{\partial \theta} \bigg|_{\theta = \theta^*} \) is zero since \( \theta^* \) is an extreme point of \( e(\theta) \). If the bias of \( \hat{\theta} \) is zero or negligible \( (E(\hat{\theta}) = \theta^*) \), then the expected value of \( \hat{e} \) can be approximated as follows:

\[
E[\hat{e}] = e^* + \frac{1}{2} \text{tr} \left[ \frac{\partial^2 e(\theta)}{\partial \theta^2} \right]_{\theta = \theta^*} \text{cov}(\hat{\theta})
\]  

(2.2)

Notice that the bias term on the right hand of equation (2.2) is non-negative, because it is the trace of the product of two positive semi-definite matrices [9]. As the number of the parameters (8) increases, the number of terms in the bias increases and hence the expected value of the error increases, too. If this increase is not canceled by the decrease in the Bayes' error that the additional parameters may provide, then the Hughes phenomenon occurs. In order to demonstrate this fact, experiments 1 and 2 are repeated here, but in experiment one, instead of 100 training samples, 20 training samples per class are used. Similarly, in experiment 2, the number of training samples per class is reduced from 100 to 15. The effect of small training sample size is evident in Figures 2.3 and 2.4 that correspond to experiments 1 and 2 respectively.
Figure 2.3: Effect of small sample size in the performance of the MED and GML Classifiers for experiment 1 (AVIRIS data)
Figure 2.4: Effect of small sample size in the performance of the MED and GML Classifiers for experiment 2 (FLC1 data)

From Figures 2.3, and 2.4 it can be seen that when the number of training samples is small the GML classifier is more severely effected by the Hughes phenomenon than the MED classifier. The behavior of the MED classifier was not significantly changed when the number of training samples were reduced but the accuracy of the GML classifier started to decrease after the dimensionality passed beyond a certain point. Therefore, although the second order statistics can be invaluable for discrimination in high
dimensional spaces, if not properly estimated they can also significantly reduce the performance. As can be seen from equation 2.2, what causes the increase in the expected error is the covariance of the estimates of the parameters. Since, the sample mean and sample covariance are the minimum variance unbiased estimators for the mean and covariance matrix, it appears that not much improvement can be hoped for if only training samples are used in the learning process. However, if by using additional information, such as the information contained in unlabeled samples, estimates with lower covariance matrices can be found, then the bias in the classification error may be reduced and therefore the Hughes phenomenon may be mitigated.

2.4 Effect of Additional Unlabeled Samples

Consider the bias term in the right hand side of equation 2.2. Consider two different estimators, $\hat{\Theta}$ and $\tilde{\Theta}$, which both have negligible bias, and assume that $\text{cov}(\tilde{\Theta}) \leq \text{cov}(\hat{\Theta})$ (i.e., $\text{cov}(\tilde{\Theta}) - \text{cov}(\hat{\Theta})$ is positive semi-definite). Then one can show that:

$$\text{tr}\left(\frac{\partial^2 e(\Theta)}{\partial \Theta^2} |_{\Theta = \Theta^*} \text{cov}(\hat{\Theta})\right) \leq \text{tr}\left(\frac{\partial^2 e(\Theta)}{\partial \Theta^2} |_{\Theta = \Theta^*} \text{cov}(\tilde{\Theta})\right)$$

The above inequality is true because both the covariance matrix and the Hessian matrix at $\Theta^*$ are positive semi-definite (the Hessian is positive semi-definite at $\Theta^*$ since $\Theta^*$ is a minimum of $e(\Theta)$, so $e(\Theta)$ is convex around $\Theta^*$). Therefore one can write:

$$\text{tr}\left(\frac{\partial^2 e(\Theta)}{\partial \Theta^2} |_{\Theta = \Theta^*} \text{cov}(\tilde{\Theta})\right) - \text{tr}\left(\frac{\partial^2 e(\Theta)}{\partial \Theta^2} |_{\Theta = \Theta^*} \text{cov}(\hat{\Theta})\right)$$
where the last inequality is obtained because the trace of the product of two positive semi-definite matrices is non-negative [9]. Therefore, the expected error due to using $\hat{\theta}$ in the decision rule is less than the expected error due to using $\hat{\theta}$:

$$E(\hat{e}) \leq E(\hat{e})$$

It is possible to show that, by using additional unlabeled samples, estimates with smaller covariance matrices can be found. Therefore, better performance can be obtained without the additional cost of obtaining more training samples.

Let us assume that $\hat{\theta}$ is an estimate of $\theta^*$ obtained by using the training samples. Furthermore, assume that $\hat{\theta}$ is asymptotically unbiased and efficient (for example, maximum likelihood estimates always possess these properties [10]). In other words, for moderately large sample sizes $E(\hat{\theta}) = \theta^*$ and $\text{cov}(\hat{\theta}) = I^{-1}$, where $I$, is the Fisher information matrix [10]. The subscript "s" denotes that the Fisher information matrix corresponds to a supervised estimate obtained by using training samples that are drawn from each class separately. The Fisher information matrix is positive semi-definite and is defined as follows:

$$I = E\left[\frac{\partial}{\partial \theta} \log f(x) \mid \frac{\partial}{\partial \theta} \log f(x)^T\right]$$
Now, let us assume that $\tilde{\theta}$ is another estimate of $\theta^*$ obtained by using some unlabeled samples in addition to the training samples. The unlabeled samples are drawn randomly from the mixture of the $m$ classes. If $\hat{\theta}$ possesses the same properties of asymptotic unbiasedness and efficiency, one can approximate $\text{cov}(\tilde{\theta})$ by $I_e^{-1}$ where $I_e$ is the Fisher information matrix corresponding to the estimate that is obtained by combining training and unlabeled samples. Provided that the unlabeled and training samples are independent, one can write:

$$I_e = I_e + I_u$$

where $I_e$ is another information matrix corresponding to the information contained in the unlabeled samples for estimating $\theta^*$. Since all of the information matrices are positive semi-definite one can write $I_e \geq I_u$. Therefore, $\text{cov}(\tilde{\theta}) \leq \text{cov}(\hat{\theta})$. Therefore, one can conclude that the expected error of the decision rule that uses $\tilde{\theta}$ is less than the one that is obtained by using $\hat{\theta}$.

The implication of the above statement is that, if reasonable estimates for the required parameters can be found that use both the labeled (training) and unlabeled samples, then they should be used in the decision rule. In particular, the benefit of using such estimates over the ones obtained by training samples alone is that the Hughes phenomenon will occur at a higher dimensionality because the estimates obtained using both labeled and unlabeled samples provide lower expected classification error. Therefore, more features can be used without sacrificing the performance and in fact, the additional information in the new features may cause an improvement in the classification accuracy. Fortunately, for the commonly used case of Gaussian classes, the Maximum Likelihood estimates that incorporate both training and unlabeled samples can be obtained [5]. The details of the
derivation of these estimates is postponed to chapter 5, where more general estimates for
the case of multi-component classes are derived. Here we merely show the iterative
formulas for obtaining the ML estimates of the parameters of the classes under the
assumption of normality for each class.

Assume that there are \( m \) Gaussian classes and from the \( i^{th} \) class \( n_i \) training samples are
available. Denote these training samples by \( z_{ik} \) where \( i \) indicates the class \((i=1,\ldots,m)\), and
\( k \) is the index of each particular sample. In addition, assume that \( n \) unlabeled samples
denoted by \( x_k \) are available from the mixture density \( f(x|\theta) = \sum_{i=1}^{m} \pi_i f_i(x) \). The ML
estimates of the parameters of the mixture density can be found by the following iterative
equations [5]:

\[
\pi_i^+ = \frac{\sum_{k=1}^{n} \frac{P_i^c f_i(x_k|\mu_i^c, \Sigma_i^c)}{f(x_k|\theta^c)}}{n} \quad (2.3)
\]

\[
\mu_i^+ = \frac{\sum_{k=1}^{n} \frac{P_i^c f_i(x_k|\mu_i^c, \Sigma_i^c) x_k + \sum_{k=1}^{n_i} z_{ik}}{f(x_k|\theta^c)}}{\sum_{k=1}^{n} \frac{P_i^c f_i(x_k|\mu_i^c, \Sigma_i^c)}{f(x_k|\theta^c)} + n_i} \quad (2.4)
\]

\[
\Sigma_i^+ = \frac{\sum_{k=1}^{n} \frac{P_i^c f_i(x_k|\mu_i^c, \Sigma_i^c) (x_k - \mu_i^+)(x_k - \mu_i^+)^T + \sum_{k=1}^{n_i} (z_{ik} - \mu_i^+)(z_{ik} - \mu_i^+)^T}{f(x_k|\theta^c)}}{\sum_{k=1}^{n} \frac{P_i^c f_i(x_k|\mu_i^c, \Sigma_i^c)}{f(x_k|\theta^c)} + n_i} \quad (2.5)
\]

where \( \mu_i \) and \( \Sigma_i \) are the mean vector and the covariance matrix of class \( i \), and superscripts
"+" and "c" denote the next and current values of the parameters respectively. The
parameter set \( \theta \) contains all the prior probabilities, mean vectors and covariance matrices.
The ML estimates are obtained by starting from an initial point in the parameter space and iterating through the above equations. A reasonable starting point is the estimates obtained by using the training samples alone.

The equations 2.3, 2.4, and 2.5 were used with the data and training sets of experiments 1 and 2 in order to demonstrate the effect of unlabeled samples in enhancing the performance. Experiment 1 was repeated but with 20 training samples and an additional number of unlabeled samples used via the above equations for estimating the parameters. Subsequently the rest of the samples were classified according to the MAP decision rule (which also incorporates the second order statistics). The experiment was performed once with 500 unlabeled samples and once with 1000 unlabeled samples. Experiment 2 was also repeated but 15 training samples and a number of unlabeled samples were used simultaneously for estimating the parameters, and MAP classification was performed subsequently. Experiment 2 was performed once with 100 and once with 500 unlabeled samples. Notice also that an additional benefit of using unlabeled samples is that the prior probabilities of the classes can be obtained and therefore instead of the ML classifier, the MAP classifier can be constructed. Without the unlabeled samples, generally the prior probabilities can not be estimated, because the training samples are usually obtained separately from each class. Figures 2.5 and 2.6 show the results. In both of these figures, the curves for the case where only training samples were used are also shown for comparison and are labeled "supervised".
Figure 2.5: Effect of additional unlabeled samples in the classification performance for experiment 1 (AVIRIS data) with 20 training samples/class
From Figures 2.5 and 2.6 it can be seen that the use of additional unlabeled samples in the learning process can enhance the classification performance when the dimensionality of data begins to approach the number of training samples. In experiment 1, the Hughes phenomenon that began around dimension 8 when supervised learning is used, is delayed to dimension 16 when 500 or 1000 additional unlabeled samples are incorporated. Meanwhile, the minimum error for the supervised learning case was 5.42% and was
achieved at dimension 7. For the cases with additional 500 and 1000 unlabeled samples, the minimum errors were 3.11% and 3.78% at dimensions 13, and 16 respectively. Therefore, the use of additional unlabeled samples not only delayed the occurrence of the Hughes phenomenon but also made the information in the new features usable for decreasing the error further. Similarly, in experiment 2, the Hughes phenomenon that began at around dimension 4 was delayed to around dimension 7 by using 100 unlabeled samples and was virtually eliminated by using 500 additional unlabeled samples. The minimum error for the supervised learning case was 2.77% at dimension 4. For the cases with additional 100 and 500 unlabeled samples the minimum errors were 2.20% and 1.88% at dimensions 5 and 10 respectively.

2.5 Effect of Unlabeled Samples in Reducing the Unrepresentative Training Samples Problem

In remote sensing, the training samples are usually selected from spatially adjacent regions. Often, the spatial correlation among the neighboring pixels is high. This correlation is usually reduced rapidly as the distance between the pixels increases. This phenomenon causes a problem when training samples are used alone for estimating the class parameters. Usually, the parameters estimated in this way are only representative of the training fields and their nearby area. The rest of the multi-spectral image is, therefore, not represented well. Thus, the classification results that are based on such training fields are not robust in the sense that by changing the training fields, the results might change significantly. Consequently, the selection of "good" training fields becomes a burden on the user's shoulders. Often, training fields are added and eliminated empirically. It is desirable to be able to update the parameter estimates in a way to make them more representative of the whole image. When the unlabeled samples are added to the learning
2 Mitigating The Hughes Phenomenon

process, the parameter estimates get updated and become more representative of the whole data. This is because the new parameter estimates maximize the joint likelihood of the training and unlabeled data.

In Figure 2.7, the parts of the FLCl and AVIRIS data sets that are used in experiments 1 and 2 are shown. Here, experiments 1 and 2 are repeated but in experiment 1, 20 adjacent training samples from each class are selected, and in experiment 2, 15 adjacent training samples are selected. The training fields are highlighted in Figure 2.7. We classify the images once by using only the training samples for estimating the parameters of the GML classifier, and once by adding some randomly drawn unlabeled samples from the scene. In experiment 1, 500 and 1000 unlabeled samples were drawn and in experiment 2, 100 and 500 unlabeled samples were used. The classification accuracies are shown in Figure 2.8. In order to show how representative the estimated parameters were, the probability maps associated with each case were obtained. The probability maps are obtained by color coding the Mahalanobis distance of each pixel to the class to which it was classified. Dark pixels are the ones which are classified with low conditional probabilities. Light pixels are the ones that are classified with high conditional probabilities. Figures 2.9 and 2.10 show the probability maps for the two experiments. It is seen from these figures that when supervised learning was used the only bright spots were near the training fields. In order words, the rest of the images were not represented well. By adding unlabeled samples to the learning process, the representivness of the estimates is enhanced and thus the probability maps are brighter.
Figure 2.7: (a) The AVIRIS site with training fields highlighted, (b) the FLC1 site with the training fields highlighted.

Figure 2.8: Classification results based on adjacent training samples. (a) AVIRIS data set, (b) FLC1 data set.
Figure 2.9: Probability maps for AVIRIS data set. (a) supervised learning, (b) combined with 500 unlabeled samples, (c) combined with 1000 unlabeled samples.
Figure 2.10: Probability maps for FLCl data set. (a) supervised learning, (b) combined with 100 unlabeled samples, (c) combined with 500 unlabeled samples.
In a separate experiment performed on four bands of the FLC1 data set, eight classes were chosen and the statistics of each class were estimated by using training samples. Then, 23,458 additional unlabeled samples (every third pixel in every third row), were used to update the class statistics via the EM equations 2.3-2.5 (an additional "unknown" class was considered here for dealing with outliers). Under the assumption of normality for the classes, the Mahalanobis distance of a sample from its corresponding class has a chi-square distribution with degrees of freedom equal to the dimensionality of the data. Using a 5% threshold value from the chi-square table, test samples as well as the 23,458 unlabeled samples were rejected. Table 2.1 contains the rejection rates obtained using the two sets of estimates. It can be seen that the reject rates are much closer to the expected 5% when the unlabeled samples are used in the estimation process.

Table 2.1: Reject Rates Based On 5% Chi-Square Threshold

<table>
<thead>
<tr>
<th>class</th>
<th>% test data rejected using GML statistics</th>
<th>% test data rejected using EM based statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>soybean</td>
<td>40.0</td>
<td>3.9</td>
</tr>
<tr>
<td>corn</td>
<td>23.3</td>
<td>8.0</td>
</tr>
<tr>
<td>oats</td>
<td>35.3</td>
<td>2.6</td>
</tr>
<tr>
<td>wheat</td>
<td>5.1</td>
<td>5.0</td>
</tr>
<tr>
<td>red clover</td>
<td>27.6</td>
<td>13.7</td>
</tr>
<tr>
<td>alfalfa</td>
<td>20.2</td>
<td>4.2</td>
</tr>
<tr>
<td>rye</td>
<td>8.2</td>
<td>7.8</td>
</tr>
<tr>
<td>bare soil</td>
<td>11.0</td>
<td>9.8</td>
</tr>
<tr>
<td>unlabeled</td>
<td>47.3</td>
<td>5.3</td>
</tr>
</tbody>
</table>
The total classification accuracy of the GML classifier in this experiment was almost identical to the total accuracy when the updated statistics were used (89.8% versus 88.32%). This is because the training fields were carefully chosen so a GML classifier would perform well. However, the probability map obtained under the updated statistics was much brighter than the one corresponding to the GML classifier. These probability maps are shown in Figure 2.11 where training and test fields are highlighted.
Figure 2.11: Probability Maps (a): based on updated statistics, (b) based on training samples
2.6 Discussion and Concluding Remarks

In this chapter, the effect of additional unlabeled samples in enhancing the classification performance was studied. It was observed that by incorporating unlabeled samples into the learning process the Hughes phenomenon might be mitigated and the peak performance can be increased and shifted to a higher dimension. This phenomenon has several advantages. First, as it was shown in section 2.4, when unlabeled samples are used, the peak performance was enhanced. In other words, the information in the new feature measurements can be used to further reduce the error. Without the unlabeled samples, the peak performance might occur at a lower dimension after which no further improvement can be obtained, and hence the new feature measurements are useless.

Secondly, the mitigation of the Hughes phenomenon is important in the feature extraction process. The feature extraction process is usually based on finding features that optimize a particular criterion. For example, in discriminant analysis within class and between class scatter matrices are estimated by using the training samples, and then features that optimize a function of these matrices are obtained. The purpose is, of course, to eliminate the less informative features and thereby speed up the classification process. However, if the estimates of the within and between class scatter matrices are not reliable (due to limited numbers of training samples), then the features obtained are also unreliable. Using additional unlabeled samples can help obtain better estimates of these matrices. Similarly, in the Decision Boundary Feature Extraction method [11], training samples are used to obtain a decision boundary in the original high dimensional space and then features that are relevant to this boundary are kept. Again, if training samples are limited, then the decision boundary in the original space is not reliable.
Third, when the training samples are not good representatives of the true class distributions, the additional unlabeled samples may help update the class statistics and make them more representative.

An important practical point that needs to be kept in mind is that although in theory the additional unlabeled samples should always improve the performance, in practice this might not always be the case. For example, in Figures 2.5 and 2.6 it can be seen that when the dimensionality is small compared to the number of training samples the supervised learning process showed a slightly better performance than the combined supervised-unsupervised learning. The reason for this behavior is the deviation of the real world situations from the models that are assumed. For example, the unlabeled samples that are drawn from the scene might contain outliers, boundary pixels, mixels, or samples of unknown classes. Such samples can hurt the performance. Therefore, care must be taken when combined supervised-unsupervised learning is used in practice. Based on these issues the following steps for designing classifiers are suggested:

1) Estimate the Bayes error in order to have an understanding of the difficulty of the problem. Unlabeled samples can also be used for Bayes error estimation [12].

2) Design a classifier using the training samples alone.

3) Test the performance of the designed classifier (test samples, resubstitution, leave-one-out, etc.). Unlabeled samples can also be used for estimating the classification error of a classifier [13].
4) If the performance of the supervised classifier was not satisfactory (based on the Bayes error), draw a set of unlabeled samples and design a new classifier using both training and unlabeled samples. Test the classifier again and if necessary use more unlabeled samples.

Throughout this chapter, the use of unlabeled samples for improving the classification performance was studied without any attempt to quantify the amount of this improvement. In chapter 3, analytical results are derived concerning the amount of reduction in classification error that may be obtained by using additional unlabeled samples are derived.
In this chapter, the effect of limited design samples in unsupervised and combined supervised-unsupervised parametric learning is studied. Consideration is given to a mixture of two multivariate normal densities with a known common covariance matrix. Upper and lower bounds for the asymptotic covariance matrix of the maximum likelihood estimates of the means are derived. These bounds are used to obtain upper and lower bounds on the asymptotic bias and variance of the estimated Bhattacharyya distance and asymptotic bias of the classification error of a linear classifier for unsupervised learning. Similar bounds for the bias of the classification error are also derived for the case when both training and unlabeled samples are used in a combined supervised-unsupervised learning process. These bounds explicitly show the relationship among dimensionality, sample size and performance for these learning processes. It is seen that when the two classes are well separated, the unsupervised learning process can perform comparably to the supervised learning process. On the other hand, if the classes are highly overlapped, a large number of unclassified observations are required in order to obtain useful estimates. Additionally, adding unlabeled samples to the supervised learning process enhances the classification performance. Experimental results for testing the derived bounds are provided.
3 Unsupervised And Combined Supervised-Unsupervised Learning

3.1. Introduction

An important issue in statistical pattern recognition is the relationship between the design sample size and the performance of a classifier. The knowledge of this relationship enables a user to make intelligent decisions about the number of required design samples in relation to the dimensionality of data and the type of classifier. The purpose of this chapter is to study the relationship between the sample size and performance in unsupervised and combined supervised-unsupervised learning processes. We derive theoretical bounds that exhibit the functional form of this relationship.

Consider a classification problem involving two multivariate Gaussian classes with probability density functions (pdfs) $f_i(x|\mu_i, \Sigma_i)$, $i=1,2$, where $\mu_i$ and $\Sigma_i$ denote the mean vector and covariance matrix of class $i$. The prior probabilities associated with the two classes are denoted by $P_1$ and $P_2$. The optimal Bayes' rule for minimizing the total classification error assigns an observation to the class with highest a posteriori probability [7]:

$$P_1f_1(x) > P_2f_2(x) \Rightarrow x \in \text{class}1$$

$$P_1f_1(x) < P_2f_2(x) \Rightarrow x \in \text{class}2$$

In order to construct the optimal Bayes' classifier, the parameters of the above decision rule must be known. In practice, these parameters have to be estimated from a set of design samples. Design samples are called training samples if their class labels are known, otherwise they are called unlabeled samples. The process of estimating the parameters of the decision rule is called learning. If learning is performed by using the training samples, the process is referred to as supervised learning. On the other hand, if unlabeled samples are used for estimating the parameters, the process is called
unsupervised learning. Therefore, supervised and unsupervised learning are both estimation processes; the difference is in the sampling method used. The former is based on stratified sampling whereas the latter is based on random sampling.

The identifiability of the finite normal mixture family [4] and existence of consistent estimators [5] for the parameters of the decision rule guarantee that in the presence of unlimited unlabeled samples the Bayes' classifier can be constructed by unsupervised learning. The same can be stated for supervised learning. (Of course if the training samples are drawn separately from the two classes, the prior probabilities cannot be reliably estimated by supervised learning. In this case, additional knowledge about the prior probabilities is required. On the other hand, relative frequencies can be used as estimates of the prior probabilities if the training samples are randomly drawn from the mixture and are subsequently labeled). Therefore, in the presence of unlimited design samples, the supervised and unsupervised learning processes are theoretically equivalent in their ability to construct the optimal Bayes' classifier. In practice, however, there are limited design samples from which the parameters of the decision rule must be estimated. Often, these estimates are used in place of the true values of the parameters in the Bayes' rule. The resulting decision rule is sub-optimal and its quality depends on the properties of the design samples and the estimators used.

The effect of limited design samples in supervised learning has been given considerable attention in the pattern recognition literature. John [14], Sitgreaves [15], and Okamoto [16] provided expressions for the average probability of error for the linear classifier when $\Sigma_1 = \Sigma_2$. Fukunaga and Hayes [17] studied limited sample size effects and provided expressions for the average classification error of the linear and quadratic classifiers when...
3 Unsupervised And Combined Supervised-Unsupervised Learning

\[ \Sigma_1 = \Sigma_2. \]

Jain and Chandrasekaran [18] and Raudys and Jain [19] have provided excellent reviews of this topic.

Similarly in unsupervised learning, when the estimates of the parameters of the density functions (obtained using unlabeled samples) are used in the Bayes' rule, the resulting classifier is sub-optimal. Ganesalingam and McLachlan [20] provided expressions for the asymptotic average error in the univariate case with equal variances. Oneill [21] considered the multivariate case with equal covariance matrices. The expressions provided by these authors are very complex and require extensive numerical integrations.

In this chapter, we derive and test upper and lower bounds on the bias and variance of estimated Bhattacharyya distance and bias of classification error. In section 3.2 some preliminary issues are briefly discussed. In section 3.3, we derive upper and lower bounds for the asymptotic covariance matrix of the maximum likelihood estimates of the parameters. These bounds are used in section 3.4 for studying the effect of limited design samples in the asymptotic performance of classifiers obtained by unsupervised learning.

The Bhattacharyya distance and classification error are particularly studied. Section 3.5 extends this study to the case where both unlabeled and training samples are available and used in a combined supervised-unsupervised learning process.

### 3.2. Preliminaries and Background

Unsupervised learning, in the context of this chapter, is the process of estimating the parameters of a normal mixture density from a set of unlabeled observations. Various methods for performing this estimation have been proposed [22]. They include the method of moments [23,24], the Bayes' method [25], maximum likelihood [26,27,28], and other approximate methods [29]. We pay attention to the maximum likelihood (ML)
method here because of its implementational ease, its prevalence, and its useful asymptotic properties (consistency, efficiency, invariance). In general, however, the likelihood equations for obtaining the ML estimates do not have analytical solutions in the case of normal mixtures. The ML estimates are therefore commonly approximated by numerical methods such as the Expectation-Maximization (EM) algorithm [5,30]. The EM algorithm is an iterative process that is guaranteed to converge to a maximum of the likelihood function. It is not guaranteed to find the global maximum and therefore usually one must repeat the process from several different initial points in the parameter space and choose the largest maximum obtained. Another theoretical problem with the ML method is the possible existence of singular points in the parameter space at which the likelihood is infinite [5]. Disregarding such technicalities, one can assume that the means for obtaining the ML estimates are in general available. However, it is noteworthy to mention that if in addition to the unlabeled samples from the mixture, training samples from the individual classes are also available, the problems with the singular points and initial parameters may be resolved.[31]. In this case, the learning process is referred to as combined supervised-unsupervised learning. We study this case in section 3.5.

Since the ML estimates for the unsupervised learning case do not have closed form expressions, it is not possible to obtain their biases and covariance matrices. However, asymptotically the ML estimates are known to be consistent and efficient [32]. Therefore, for a large design sample size, the bias of an ML estimate is negligible, and its covariance matrix is equal to the inverse of the associated Fisher information matrix. When n independent unlabeled observations \( \{x_1, \ldots, x_n\} \) are drawn from a mixture density \( f(x) = P_1f_1(x) + P_2f_2(x) \), the Fisher information matrix can be written in the following form:

\[
I^u = nE\left\{ \frac{\partial}{\partial \theta} \log f(x) \left[ \frac{\partial}{\partial \theta} \log f(x) \right]^T \right\}
\]

(3.2)
where $\theta$ is the vector containing all the unknown parameters of the mixture density, superscript "T" stands for transpose of a vector, and the expectation is with respect to the mixture density. The superscript "u" in the left hand side stands for "unsupervised" and is used to distinguish the information matrix from the identity matrix. The evaluation of the above information matrix involves integrals for which no closed form expressions are available. In the univariate case, Behboodian [33] proposed several numerical methods for evaluating $I^u$ and provided extensive tables of the results. Chang [34] proposed a method for the multivariate case which involves a number of one-dimensional numerical integrations. Perlovsky [35] formulated the information matrices for the estimates of the means and covariance matrices independently in terms of variables that describe the amount of overlap between the classes. In the next section, we will provide upper and lower bounds for the information matrix under certain constraints.

### 3.3. Bounds for the Asymptotic Covariance Matrix of the ML Estimates

Consider the following normal mixture density.

$$f(x|\theta) = P_1f_1(x|\mu_1, \Sigma_1) + P_2f_2(x|\mu_2, \Sigma_2) \quad x \in \mathbb{R}^d$$  \hspace{1cm} (3.3)

where $\theta = (P_1, P_2, \mu_1, \mu_2, \Sigma_1, \Sigma_2)$ is the set of the parameters of the mixture. We pay attention to the equal covariance case where $\Sigma_1 = \Sigma_2 = \Sigma$ (the unequal covariance case is very difficult to study since the optimal classifier becomes quadratic and the error rate becomes very complex). In addition, we assume that the common covariance matrix is known. This assumption is not necessary for our study but greatly reduces the amount of algebra. The extension to the unknown covariance matrix case is possible. We also
assume that the prior probabilities of the two classes are known. In most of the studies in supervised learning this assumption is made, and therefore we do the same in order to compare the results directly. In fact, usually in supervised learning, the training samples are obtained separately from the two classes and therefore contain no information about the prior probabilities. However, the techniques developed in this chapter can be used for the unknown prior probabilities case also. Therefore, hereafter we assume that the only unknowns to be estimated through the unsupervised learning process are the means of the two classes.

Without loss of generality, consider the canonical form where \( \mu_1 = 0 \), \( \mu_2 = [A \ 0 \ldots 0]^T \), \( \Sigma_1 = \Sigma_2 = I_d \), \( A > 0 \), \( A \Sigma \) is the Mahalanobis distance between the two classes, and \( I_d \) is the \( d \times d \) identity matrix (\( d \) is the dimension of the feature space). Since the error rate and Bhattacharyya distance, which are the subjects of our study in the next section, are both invariant under nonsingular linear transformations, the canonical form can be used here without loss of generality. Any other two class problem for which \( \Sigma_1 = \Sigma_2 \) can be transformed into the above form through a linear transformation [7]. Now, consider (3.3) and recall that the vector of unknown parameters is \( \Theta = [\mu_1^T \mu_2^T]^T \), therefore:

\[
\begin{align*}
\frac{\partial}{\partial \Theta} \log f(x) &= \frac{1}{f(x)} \frac{\partial}{\partial x} f(x) = \frac{1}{f(x)} \left[ \frac{P_1 f_1(x) \Sigma_1^{-1}(x - \mu_1)}{P_2 f_2(x) \Sigma_2^{-1}(x - \mu_2)} \right] \\
\end{align*}
\]

Using (3.4) in (3.2) and substituting 0 for \( \mu_1 \) and \( I_d \) for \( \Sigma_1 \) and \( \Sigma_2 \), the Fisher information matrix can be written in the following form:

\[
\begin{align*}
\frac{1}{n} \mathbf{I}^u = E \left[ \frac{1}{f^2(x)} \left[ \begin{array}{c} P_1^2 f_1^2(x)xx^T \\
P_1 P_2 f_1(x)f_2(x)(x - \mu_2)x^T \\
P_2^2 f_2^2(x)(x - \mu_2)(x - \mu_2)^T \\
P_2 f_2^2(x)(x - \mu_2)(x - \mu_2)^T \\
\end{array} \right] \right] \\
\end{align*}
\]
Since the covariance matrices of $f_1(x)$ and $f_2(x)$ are both equal to $I_d$, the above integrals can be reduced to one dimensional integrals whose values may be found numerically by the methods suggested in [33]. However, simple upper and lower bounds can be found for $I^u$. Notice, that equation (3.5) can be written as follows:

$$\frac{1}{n} I_u = \begin{bmatrix}
\int P_1 P_1 f_1(x) f_2(x) f(x) \frac{xx^T}{f(x)} dx & \int P_1 P_2 f_1(x) f_2(x) \frac{x(x - \mu_2)^T}{f(x)} dx \\
\int P_1 P_2 f_1(x) f_2(x) \frac{(x - \mu_2)x^T}{f(x)} dx & \int P_2 f_2(x) \left(1 - \frac{P_1 f_1(x)}{f(x)}\right) (x - \mu_2)(x - \mu_2)^T dx
\end{bmatrix}$$

$$= \begin{bmatrix}
P_1 I_d & 0 \\
0 & P_2 I_d
\end{bmatrix} \begin{bmatrix}
\int P_1 P_2 f_1(x) f_2(x) \frac{xx^T}{f(x)} dx & \int P_1 P_2 f_1(x) f_2(x) \frac{x(x - \mu_2)^T}{f(x)} dx \\
\int P_1 P_2 f_1(x) f_2(x) \frac{(x - \mu_2)x^T}{f(x)} dx & \int P_1 P_2 f_1(x) f_2(x) \frac{(x - \mu_2)(x - \mu_2)^T}{f(x)} dx
\end{bmatrix}$$

$$= \begin{bmatrix}
P_1 I_d & 0 \\
0 & P_2 I_d
\end{bmatrix} \begin{bmatrix}
\int P_1 P_2 f_1(x) f_2(x) \frac{x}{f(x)} dx & \int P_1 P_2 f_1(x) f_2(x) \frac{x(x - \mu_2)^T}{f(x)} dx \\
(\mu_2 - x) & ((\mu_2 - x) \mu_2 - x)^T dx
\end{bmatrix}$$

Now, it is easily seen that the second term in the right hand side of (3.6) is a positive semi-definite matrix. Therefore, one can write:

$$I^u \leq n \begin{bmatrix}
P_1 I_d & 0 \\
0 & P_2 I_d
\end{bmatrix}$$
where the above inequality is a max inequality indicating that the right hand side minus
the left hand side is a positive semi-definite matrix. Notice that the right hand side of
equation (3.7) is the Fisher information matrix for estimating \( \theta \) if the \( n \) randomly drawn
design samples were labeled. In particular, let \( I^s \) be the information matrix for this case.
One can write:

\[
I^s = n \left\{ P_1 E \left[ \frac{\partial}{\partial \theta} \log f_1(x) \right] \left[ \frac{\partial}{\partial \theta} \log f_1(x) \right]^T \mid x \in \text{class1} \right\} \\
+ P_2 E \left[ \frac{\partial}{\partial \theta} \log f_2(x) \right] \left[ \frac{\partial}{\partial \theta} \log f_2(x) \right]^T \mid x \in \text{class2} \right\}
\]

\[
= n \begin{bmatrix} P_1 I_d & 0 \\ 0 & P_2 I_d \end{bmatrix}
\]

Therefore, inequality (3.7) yields the conceptually appealing fact that the information
content of \( n \) unlabeled observations for estimating the means is less than or equal to that
of \( n \) labeled observations. Of course, the missing information in the unsupervised
learning case is the labels of the samples. We therefore refer to the right hand side of
(3.7) as the "supervised bound" for \( I^u \). One would expect \( I^u \) to approach its supervised
bound as the separation between the two classes increases since by increasing the
separability the unlabeled samples become easier to classify.

Let us consider the Fisher information max \( I^u \) again and write it in the following form:
where the two sub-spaces Ω₁ and Ω₂ are defined as follows:

\[ x \in \Omega_1 \iff P_1 f_1(x) \geq P_2 f_2(x) \]
\[ x \in \Omega_2 \iff P_1 f_1(x) < P_2 f_2(x) \]

In the canonical case under consideration, one can write:

\[ x \in \Omega_1 \iff x_1 \leq t \]
\[ x \in \Omega_2 \iff x_1 > t \]

where:

\[ t = \frac{1}{\Delta} \log\left(\frac{P_1}{P_2}\right) + \frac{1}{2} \Delta \]  

(3.9)

Notice, that if \( x \in \Omega_1 \) then \( P_1 f_1(x) > 2 P_2 f_2(x) \) therefore one can write:

\[ x \in \Omega_1 \Rightarrow P_1 f_1(x) > 2 P_2 f_2(x) \]

Similarly:

\[ x \in \Omega_2 \Rightarrow P_2 f_2(x) \leq f(x) < 2 P_2 f_2(x) \]

Using the above inequalities and (3.8) one can easily see that the following is true:
The right hand side of (3.10) constitutes a positive semi-definite matrix. Similarly, the following is also true:

$$\frac{1}{n} \mathbf{I}^u \geq \int_{\Omega_1} \frac{1}{P_1f_1(x)} \left[ \begin{array}{c} P_2f_1(x) \\ P_2f_2(x) \end{array} \right] \left[ \begin{array}{c} P_1f_1(x)^T \\ P_2f_2(x)^T \end{array} \right] dx + \int_{\Omega_2} \frac{1}{P_2f_2(x)} \left[ \begin{array}{c} P_2f_1(x) \\ P_2f_2(x) \end{array} \right] \left[ \begin{array}{c} P_1f_1(x)^T \\ P_2f_2(x)^T \end{array} \right] dx$$

(3.10)

If the two classes are well separated, on $\Omega_1 : f(x) \approx P_1f_1(x)$, and on $\Omega_2 : f(x) \approx P_2f_2(x)$. Hence, for well separated classes, $\mathbf{I}^u$ is close to its upper bound in the right hand side of (3.11). Similarly, if the two classes are highly overlapped then $\mathbf{I}^u$ is closer to its lower bound in (3.10).

Since the right hand sides of equations (3.11) and (3.10) are similar except for a factor of two, we will only analyze the latter. Notice that equation (3.10) can be written in the following form:

$$\frac{1}{n} \mathbf{I}^u \leq \int_{\Omega_1} \frac{1}{P_1f_1(x)} \left[ \begin{array}{c} P_2f_1(x) \\ P_2f_2(x) \end{array} \right] \left[ \begin{array}{c} P_1f_1(x)^T \\ P_2f_2(x)^T \end{array} \right] dx + \int_{\Omega_2} \frac{1}{P_2f_2(x)} \left[ \begin{array}{c} P_2f_1(x) \\ P_2f_2(x) \end{array} \right] \left[ \begin{array}{c} P_1f_1(x)^T \\ P_2f_2(x)^T \end{array} \right] dx$$

(3.11)
3 Unsupervised And Combined Supervised-Unsupervised Learning

It is possible to show that all the sub-matrices in the right hand side of (3.12) are diagonal with entries which can be found by using the moments of truncated normal distributions. See appendix A for the details of the derivations. We put the results of these derivations together with (3.7) into the form of a theorem for convenience:

Theorem 3.1: For the canonical two component normal mixture with unknown means, the Fisher information matrix \( I^u \) is bounded as follows:

\[
I^u \leq \frac{1}{n} \left[ \begin{array}{cc} P_1 I_d & 0 \\ 0 & P_2 I_d \end{array} \right]
\]

(3.7)

and

\[
\frac{1}{n} \left[ \begin{array}{cccc} \alpha & 0 & \tau & 0 \\ 0 & \beta I_{d-1} & 0 & \nu I_{d-1} \\ \tau & 0 & \delta & 0 \\ 0 & \nu I_{d-1} & 0 & \epsilon I_{d-1} \end{array} \right] \leq I^u \leq 2n \left[ \begin{array}{cccc} \alpha & 0 & \tau & 0 \\ 0 & \beta I_{d-1} & 0 & \nu I_{d-1} \\ \tau & 0 & \delta & 0 \\ 0 & \nu I_{d-1} & 0 & \epsilon I_{d-1} \end{array} \right]
\]

(3.13)

where:

\[
a = \frac{P_1}{2} (\Phi(t) - t\phi(t)) + \frac{r_1^2}{2B_2} \exp(\Delta^2)[(1 + \Delta^2)\Phi(-\Delta - t) + (t - \Delta)\phi(t + A)]
\]
\[ \beta = \frac{P_1}{2} \Phi(t) + \frac{P_2^2}{2P_2} \exp(\Delta^2)\Phi(-\Delta - t) \]

\[ \tau = \frac{P_1}{2}(\Phi(-t) + (t - \Delta)\phi(t)) + \frac{P_2^2}{2}(\Phi(t - \Delta) - t\phi(t - \Delta)) \]

\[ \nu = \frac{P_1}{2} \Phi(-t) + \frac{P_2^2}{2} \Phi(t - \Delta) \]

\[ \delta = \frac{P_2^2}{2}(\Phi(\Delta - t) + (t - \Delta)\phi(t - A)) + \frac{P_2^2}{2P_1^2} \exp(\Delta^2)(1 + \Delta^2)\Phi(t - 2\Delta) - t\phi(2\Delta - t) \]

\[ \epsilon = \frac{P_2^2}{2} \Phi(\Delta - t) + \frac{P_2^2}{2P_1} \exp(\Delta^2)\Phi(t - 2\Delta) \]

and \( \Phi(t) \) and \( \phi(t) \) are the cumulative distribution function (cdf) and pdf of the standard normal distribution respectively.

By inverting the bounds of theorem 3.1, the asymptotic covariance matrix of the ML estimate of \( \theta = [\mu_1^T \mu_2^T]^T \) can be bounded from above and below. Notice that for any two positive definite matrices \( A \) and \( B \), if \( A \succeq B \), then \( A^{-1} \preceq B^{-1} \) [9]. The following lemma therefore follows:

**Lemma 3.1:** The asymptotic covariance matrix of the ML estimate of \( \theta \) (where \( \theta = [\mu_1^T \mu_2^T]^T \)) is bounded as follows:
3 Unsupervised And Combined Supervised-Unsupervised Learning

\[
\text{cov}(\hat{\theta}) \geq \frac{1}{n} \begin{bmatrix}
1 & 0 \\
0 & \frac{1}{2} I_d
\end{bmatrix}
\]  

(3.14)

and

\[
\text{cov}(\hat{\theta}) \leq \frac{1}{n} \begin{bmatrix}
\delta(\alpha \delta - \tau^2)^{-1} & 0 & \tau(\tau^2 - \alpha \delta)^{-1} & 0 \\
0 & \tau(\tau^2 - \alpha \delta)^{-1} & 0 & \alpha(\alpha \delta - \tau^2)^{-1} \\
0 & 0 & \nu(\nu^2 - \beta \delta)^{-1} & 0 \\
0 & \nu(\nu^2 - \beta \delta)^{-1} & 0 & \beta(\beta \delta - \nu^2)^{-1}
\end{bmatrix}
\]  

(3.15a)

\[
\text{cov}(\hat{\theta}) \geq \frac{1}{2n} \begin{bmatrix}
\delta(\alpha \delta - \tau^2)^{-1} & 0 & \tau(\tau^2 - \alpha \delta)^{-1} & 0 \\
0 & \tau(\tau^2 - \alpha \delta)^{-1} & 0 & \alpha(\alpha \delta - \tau^2)^{-1} \\
0 & 0 & \nu(\nu^2 - \beta \delta)^{-1} & 0 \\
0 & \nu(\nu^2 - \beta \delta)^{-1} & 0 & \beta(\beta \delta - \nu^2)^{-1}
\end{bmatrix}
\]  

(3.15b)

where \( \hat{\theta} \) is the ML estimate of \( \theta \) obtained by unsupervised learning.

The proof of the above lemma is obtained by applying the formulas for inverting partitioned matrices [9] to the inequalities of theorem 1.

3.4. Effect of Limited Design Samples

In this section the bounds derived in section 3.3 are used to study the effect of sample size in unsupervised learning. In particular, we study the estimates of the Bhattacharyya distance and error rate obtained by plugging in the estimates of the parameters found by unsupervised learning.
In [17], the authors have used a Taylor series expansion of up to the second order to approximate the bias and variance of general functions. If the value of a function $g(\theta)$ at point $\theta^*$ is estimated by replacing $\theta^*$ with its estimate $\hat{\theta}$, then provided that the deviation of $\hat{\theta}$ from $\theta^*$ is small one can write:

$$
\hat{g} = g(\hat{\theta}) \approx g(\theta^*) + \frac{\partial g}{\partial \theta} \bigg|_{\theta=\theta^*} \Delta \theta + \frac{1}{2} \text{tr} \left( \frac{\partial^2 g}{\partial \theta^2} \bigg|_{\theta=\theta^*} \Delta \theta \Delta \theta^T \right)
$$

(3.16)

where $\Delta \theta = \hat{\theta} - \theta^*$, and $\text{tr}(A)$ is the trace of matrix $A$. Assuming that the bias of $\hat{\theta}$ is negligible, one can write:

$$
E[\hat{g}] = g + \frac{1}{2} \text{tr} \left( \frac{\partial^2 g}{\partial \theta^2} \bigg|_{\theta=\theta^*} \text{cov}(\hat{\theta}) \right)
$$

(3.17)

and:

$$
\text{var}[\hat{g}] = E \left\{ \left[ \frac{\partial g}{\partial \theta} \bigg|_{\theta=\theta^*} \Delta \theta \right]^2 \right\} = E \left\{ \left[ \frac{\partial g}{\partial \theta} \bigg|_{\theta=\theta^*} \right]^2 \right\}
$$

(3.18)

In supervised learning, the bias and covariance matrix of the estimate of the parameter set have been found explicitly and used to approximate the bias and variance of some interesting functions, such as the Bhattacharyya distance and the classification error...
We use this method for studying the unsupervised learning process. We first recall the consistency property of the ML estimates to argue that for large design sample size, the estimate of the parameter set is unbiased and (3.17) and (3.18) are therefore applicable. We can use the bounds of lemma 3.1 in these equations in order to derive upper and lower bounds for the bias and variance of any function which is locally convex around $\theta^\ast$. The following results from linear algebra will be used in the subsequent derivations:

**Result 3.1:** Let $A$, $B$, and $C$ be $d \times d$ real, symmetric, positive semi-definite (p.s.d) matrices, and $A \geq B$, then $\text{tr}(CA) \geq \text{tr}(CB)$.

Result 3.1 is useful for obtaining bounds on the bias of a locally convex function. The following result is useful for obtaining bounds on the variance of such a function:

**Result 3.2:** Let $A$ and $B$ be $d \times d$ p.s.d matrices, and $A \geq B$. For any $d$-dimensional vector $v$, $v^T Av \geq v^T Bv$.

### 3.4.1 Bhattacharyya Distance

Consider the Bhattacharyya distance between the two classes:

$$
B = \frac{1}{8} \left( \mu_2 - \mu_1 \right)^T \left( \frac{\Sigma_1 + \Sigma_2}{2} \right)^{-1} \left( \mu_2 - \mu_1 \right) + \frac{1}{2} \ln \frac{2}{\sqrt{\det \Sigma_1 \det \Sigma_2}} \tag{3.19}
$$

Bhattacharyya distance provides an upper bound for the Bayes error through the following relationship [7]:

- 50 -
3 Unsupervised And Combined Supervised-Unsupervised Learning

\[ \text{err}^* \leq \sqrt{P_1 P_2 e^{-B}} \] (3.20)

Where \( \text{err}^* \) denotes the Bayes error. Because of the above inequality, \( B \) is often used for feature selection. In practice, \( B \) is estimated by replacing the parameters of the right hand side of equation (3.19) with their estimates. For the canonical case under study one can write:

\[ \frac{\partial B}{\partial \theta_{\theta = \theta^*}} = \frac{1}{4} \begin{bmatrix} -\mu_2 \\ \mu_2 \end{bmatrix} \quad \frac{\partial^2 B}{\partial \theta^2_{\theta = \theta^*}} = \frac{1}{4} \begin{bmatrix} I_d & -I_d \\ -I_d & I_d \end{bmatrix} \]

Notice that \( \frac{\partial^2 B}{\partial \theta^2} \) is a positive semi-definite matrix. Hence, the results 3.1 and 3.2 can be used together with the upper and lower bounds provided by lemma 1 in (3.17) and (3.18) to obtain the following inequalities for the bias and variance of the estimated Bhattacharyya distance:

\[ \text{bias}(\hat{B}) \geq \frac{d}{8nP_1 P_2} \quad \text{(the supervised bound)} \] (3.21)

\[ \frac{1}{8n} \left\{ \frac{\alpha + \delta + 2\tau}{\alpha \delta - \tau^2} + (d - 1) \frac{\epsilon + \beta + 2v}{\epsilon \beta - v^2} \right\} \geq \text{bias}(\hat{B}) \geq \frac{1}{16n} \left\{ \frac{\alpha + \delta + 2\tau}{\alpha \delta - \tau^2} + (d - 1) \frac{\epsilon + \beta + 2v}{\epsilon \beta - v^2} \right\} \]

(3.22)

Similarly:

\[ \text{var}(\hat{B}) \geq \frac{\Delta^2}{16nP_1 P_2} \quad \text{(the supervised bound)} \] (3.23)

\[ \frac{1}{16n} \Delta^2 \left( \frac{\alpha + \delta + 2\tau}{\alpha \delta - \tau^2} \right) \geq \text{var}(\hat{B}) \geq \frac{1}{32n} \Delta^2 \left( \frac{\alpha + \delta + 2\tau}{\alpha \delta - \tau^2} \right) \]

(3.24)
As an example, figures 3.1 and 3.2 illustrate the upper and lower bounds on the number of unlabeled design samples necessary in order to keep the bias of $\hat{B}$ less than 0.125 versus dimensionality of the feature space. In both cases, the curves are drawn assuming that $P_1 = P_2 = 0.5$. In Figure 3.1, $A = 1.0$ (corresponding to a Bayes' error $= 0.3085$), and in Figure 3.2, $A = 2.0$ (Bayes' error $= 0.1587$). Notice that when the classes are more separable, the upper and lower bounds are closer to the supervised bound (which illustrates the case when the drawn samples are labeled) indicating that unsupervised learning can perform comparably to supervised learning.

![Figure 3.1: Bounds on the number of samples required to have bias ($B$) $< 0.125$ versus dimensionality, when $A = 1.0$, $P_1 = P_2$.](image)
Tables 3.1 and 3.2 contain the results of a set of experiments performed to verify the theoretical bounds (3.21) and (3.22). The dimensionality was varied from 2 to 16 in increments of 2. The number of unlabeled samples used was \( n = kd \), where \( k \) was selected to be 10, 20, 30, 50, 100, and 200. The prior probabilities were both equal to 0.5. Samples were generated using a Gaussian random number generator. Using unsupervised learning, \( \hat{\mu}_1 \) and \( \hat{\mu}_2 \) were computed and used to estimate \( B \). The procedure was repeated 20 times independently. Table 3.1 is for the case when \( A = 1.0 \), and Table 3.2 is for the case when \( A = 2.0 \). In these tables, the first line is the supervised bound for \( \hat{B} \) (from (3.21)), the
second line is the lower-upper bounds from (3.22), the third line is the mean of the 20 trials, and the fourth line is the standard deviation of the 20 trials. Notice that for the asymptotic theory to hold, especially when the two classes are close to each other, a large number of samples needs to be used. This is why values of k less than 10 are not considered. Tables 3.1 and 3.2 show that the experimental results match the theory closely. However, in these tables, the mean values sometimes lie outside the derived bounds. This phenomenon is due to the fact that the bounds are asymptotic, and therefore a very large number of samples is required (especially when A is small) to ensure that the covariance matrix of the estimates is close to the inverse of the Fisher information matrix and the higher terms in the Taylor series are negligible. However, even when the mean of the 20 trials does not fall within the theoretical bounds, the bounds are within one standard deviation from the mean. In these experiments in order to prevent the unsupervised learning process from converging to the wrong local maxima of the likelihood function, the labels of the generated samples were first used to obtain the initial parameters for the EM iterative process and were then eliminated so that the samples would be unlabeled and be used through the EM algorithm.
### Table 3.1: Bounds On $E[\hat{B}]$: $A = 1.0$ (True value of $B = 0.125$)

(First line: supervised lower bound.
Second line: lower-upper bounds
Third line: mean of 20 trials
Fourth line: standard deviation of 20 trials)

<table>
<thead>
<tr>
<th>$d$</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.2798</td>
<td>.3001</td>
<td>.3068</td>
<td>.3086</td>
<td>.3102</td>
<td>.4954</td>
<td>.3122</td>
<td>.4994</td>
</tr>
<tr>
<td></td>
<td>.3136</td>
<td>.5021</td>
<td>.3145</td>
<td>.5041</td>
<td>.3153</td>
<td>.5055</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.1938</td>
<td>.2565</td>
<td>.2692</td>
<td>.3010</td>
<td>.3185</td>
<td>.3287</td>
<td>.3222</td>
<td>.3379</td>
</tr>
<tr>
<td></td>
<td>.1501</td>
<td>.1120</td>
<td>.1120</td>
<td>.0875</td>
<td>.0706</td>
<td>.0658</td>
<td>.0561</td>
<td>.0507</td>
</tr>
<tr>
<td>20</td>
<td>.1500</td>
<td>.1500</td>
<td>.1500</td>
<td>.1500</td>
<td>.1500</td>
<td>.1500</td>
<td>.1500</td>
<td>.1500</td>
</tr>
<tr>
<td></td>
<td>.2024</td>
<td>.2798</td>
<td>.2125</td>
<td>.3068</td>
<td>.2176</td>
<td>.3102</td>
<td>.2186</td>
<td>.3122</td>
</tr>
<tr>
<td></td>
<td>.2193</td>
<td>.3136</td>
<td>.2198</td>
<td>.3145</td>
<td>.2201</td>
<td>.3153</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.2142</td>
<td>.1846</td>
<td>.2116</td>
<td>.2174</td>
<td>.2494</td>
<td>.2550</td>
<td>.2407</td>
<td>.2473</td>
</tr>
<tr>
<td></td>
<td>.1339</td>
<td>.0803</td>
<td>.0683</td>
<td>.0555</td>
<td>.0554</td>
<td>.0500</td>
<td>.0394</td>
<td>.0285</td>
</tr>
<tr>
<td>30</td>
<td>.1412</td>
<td>.1412</td>
<td>.1412</td>
<td>.1412</td>
<td>.1412</td>
<td>.1412</td>
<td>.1412</td>
<td>.1412</td>
</tr>
<tr>
<td></td>
<td>.1766</td>
<td>.2282</td>
<td>.1834</td>
<td>.2417</td>
<td>.1856</td>
<td>.2462</td>
<td>.1867</td>
<td>.2485</td>
</tr>
<tr>
<td></td>
<td>.1874</td>
<td>.2498</td>
<td>.1879</td>
<td>.2507</td>
<td>.1882</td>
<td>.2514</td>
<td>.1884</td>
<td>.2518</td>
</tr>
<tr>
<td></td>
<td>.1833</td>
<td>.1795</td>
<td>.1918</td>
<td>.2044</td>
<td>.2006</td>
<td>.2100</td>
<td>.2068</td>
<td>.2165</td>
</tr>
<tr>
<td></td>
<td>.1102</td>
<td>.0720</td>
<td>.0773</td>
<td>.0777</td>
<td>.0844</td>
<td>.0731</td>
<td>.0725</td>
<td>.0399</td>
</tr>
<tr>
<td>50</td>
<td>.1350</td>
<td>.1350</td>
<td>.1350</td>
<td>.1350</td>
<td>.1350</td>
<td>.1350</td>
<td>.1350</td>
<td>.1350</td>
</tr>
<tr>
<td></td>
<td>.1651</td>
<td>.1642</td>
<td>.1676</td>
<td>.1746</td>
<td>.1786</td>
<td>.1747</td>
<td>.1788</td>
<td>.1746</td>
</tr>
<tr>
<td></td>
<td>.0926</td>
<td>.0605</td>
<td>.0527</td>
<td>.0285</td>
<td>.0386</td>
<td>.0313</td>
<td>.0241</td>
<td>.0319</td>
</tr>
<tr>
<td>100</td>
<td>.1300</td>
<td>.1300</td>
<td>.1300</td>
<td>.1300</td>
<td>.1300</td>
<td>.1300</td>
<td>.1300</td>
<td>.1300</td>
</tr>
<tr>
<td></td>
<td>.1405</td>
<td>.1560</td>
<td>.1425</td>
<td>.1600</td>
<td>.1432</td>
<td>.1614</td>
<td>.1435</td>
<td>.1620</td>
</tr>
<tr>
<td></td>
<td>.1457</td>
<td>.1624</td>
<td>.1439</td>
<td>.1627</td>
<td>.1440</td>
<td>.1629</td>
<td>.1440</td>
<td>.1631</td>
</tr>
<tr>
<td></td>
<td>.1568</td>
<td>.1533</td>
<td>.1496</td>
<td>.1545</td>
<td>.1547</td>
<td>.1526</td>
<td>.1517</td>
<td>.1500</td>
</tr>
<tr>
<td></td>
<td>.0547</td>
<td>.0262</td>
<td>.0346</td>
<td>.0305</td>
<td>.0243</td>
<td>.0245</td>
<td>.0211</td>
<td>.0181</td>
</tr>
<tr>
<td>200</td>
<td>.1275</td>
<td>.1275</td>
<td>.1275</td>
<td>.1275</td>
<td>.1275</td>
<td>.1275</td>
<td>.1275</td>
<td>.1275</td>
</tr>
<tr>
<td></td>
<td>.1327</td>
<td>.1405</td>
<td>.1338</td>
<td>.1425</td>
<td>.1341</td>
<td>.1432</td>
<td>.1343</td>
<td>.1435</td>
</tr>
<tr>
<td></td>
<td>.1344</td>
<td>.1437</td>
<td>.1344</td>
<td>.1439</td>
<td>.1345</td>
<td>.1440</td>
<td>.1345</td>
<td>.1440</td>
</tr>
<tr>
<td></td>
<td>.1365</td>
<td>.1410</td>
<td>.1387</td>
<td>.1365</td>
<td>.1383</td>
<td>.1361</td>
<td>.1341</td>
<td>.1346</td>
</tr>
<tr>
<td></td>
<td>.0524</td>
<td>.0269</td>
<td>.0290</td>
<td>.0200</td>
<td>.0174</td>
<td>.0183</td>
<td>.0154</td>
<td>.0171</td>
</tr>
</tbody>
</table>
### Table 3.2: Bounds On $E(\hat{B})$: $A = 2.0$ (True value of $B = 0.5$)


<table>
<thead>
<tr>
<th>$d$</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>5500</td>
<td>5500</td>
<td>5500</td>
<td>5500</td>
<td>5500</td>
<td>5500</td>
<td>5500</td>
<td>5500</td>
</tr>
<tr>
<td></td>
<td>.5699-.6397</td>
<td>.5768-.6535</td>
<td>.5791-.6581</td>
<td>.5802-.6604</td>
<td>.5809-.6618</td>
<td>.5814-.6627</td>
<td>.5817-.6634</td>
<td>.5819-.6639</td>
</tr>
<tr>
<td></td>
<td>.6216</td>
<td>.5515</td>
<td>.5671</td>
<td>.5549</td>
<td>.5793</td>
<td>.5902</td>
<td>.5938</td>
<td>.5960</td>
</tr>
<tr>
<td></td>
<td>.2997</td>
<td>.1581</td>
<td>.1553</td>
<td>.1153</td>
<td>.1122</td>
<td>.1152</td>
<td>.0874</td>
<td>.1108</td>
</tr>
<tr>
<td>20</td>
<td>.5250</td>
<td>.5250</td>
<td>.5250</td>
<td>.5250</td>
<td>.5250</td>
<td>.5250</td>
<td>.5250</td>
<td>.5250</td>
</tr>
<tr>
<td></td>
<td>.5349-.5699</td>
<td>.5384-.5768</td>
<td>.5395-.5791</td>
<td>.5401-.5802</td>
<td>.5405-.5809</td>
<td>.5407-.5814</td>
<td>.5408-.5817</td>
<td>.5410-.5819</td>
</tr>
<tr>
<td></td>
<td>.5176</td>
<td>.5224</td>
<td>.5401</td>
<td>.5439</td>
<td>.5520</td>
<td>.5370</td>
<td>.5525</td>
<td>.5470</td>
</tr>
<tr>
<td></td>
<td>.1559</td>
<td>.1070</td>
<td>.1171</td>
<td>.1091</td>
<td>.1074</td>
<td>.0780</td>
<td>.0631</td>
<td>.0662</td>
</tr>
<tr>
<td>30</td>
<td>.5167</td>
<td>.5167</td>
<td>.5167</td>
<td>.5167</td>
<td>.5167</td>
<td>.5167</td>
<td>.5167</td>
<td>.5167</td>
</tr>
<tr>
<td></td>
<td>.5233-.5466</td>
<td>.5256-.5512</td>
<td>.5264-.5527</td>
<td>.5267-.5535</td>
<td>.5270-.5539</td>
<td>.5271-.5542</td>
<td>.5272-.5545</td>
<td>.5273-.5546</td>
</tr>
<tr>
<td></td>
<td>.5180</td>
<td>.5222</td>
<td>.5383</td>
<td>.5372</td>
<td>.5350</td>
<td>.5471</td>
<td>.5449</td>
<td>.5462</td>
</tr>
<tr>
<td></td>
<td>.1494</td>
<td>.1085</td>
<td>.0767</td>
<td>.0775</td>
<td>.0692</td>
<td>.0514</td>
<td>.0594</td>
<td>.0534</td>
</tr>
<tr>
<td>50</td>
<td>.5100</td>
<td>.5100</td>
<td>.5100</td>
<td>.5100</td>
<td>.5100</td>
<td>.5100</td>
<td>.5100</td>
<td>.5100</td>
</tr>
<tr>
<td></td>
<td>.5140-.5280</td>
<td>.5153-.5307</td>
<td>.5158-.5316</td>
<td>.5160-.5321</td>
<td>.5162-.5324</td>
<td>.5163-.5325</td>
<td>.5163-.5327</td>
<td>.5164-.5328</td>
</tr>
<tr>
<td></td>
<td>.4900</td>
<td>.5213</td>
<td>.5276</td>
<td>.5302</td>
<td>.5355</td>
<td>.5248</td>
<td>.5246</td>
<td>.5266</td>
</tr>
<tr>
<td></td>
<td>.1042</td>
<td>.1035</td>
<td>.0669</td>
<td>.0432</td>
<td>.0572</td>
<td>.0471</td>
<td>.0402</td>
<td>.0394</td>
</tr>
<tr>
<td>100</td>
<td>.5050</td>
<td>.5050</td>
<td>.5050</td>
<td>.5050</td>
<td>.5050</td>
<td>.5050</td>
<td>.5050</td>
<td>.5050</td>
</tr>
<tr>
<td></td>
<td>.5070-.5140</td>
<td>.5077-.5154</td>
<td>.5079-.5158</td>
<td>.5080-.5160</td>
<td>.5081-.5162</td>
<td>.5081-.5163</td>
<td>.5082-.5164</td>
<td>.5082-.5164</td>
</tr>
<tr>
<td></td>
<td>.5103</td>
<td>.5234</td>
<td>.5178</td>
<td>.5197</td>
<td>.5208</td>
<td>.5147</td>
<td>.5123</td>
<td>.5118</td>
</tr>
<tr>
<td></td>
<td>.1037</td>
<td>.0420</td>
<td>.0479</td>
<td>.0380</td>
<td>.0317</td>
<td>.0361</td>
<td>.0283</td>
<td>.0266</td>
</tr>
<tr>
<td>200</td>
<td>.5025</td>
<td>.5025</td>
<td>.5025</td>
<td>.5025</td>
<td>.5025</td>
<td>.5025</td>
<td>.5025</td>
<td>.5025</td>
</tr>
<tr>
<td></td>
<td>.5035-.5070</td>
<td>.5038-.5077</td>
<td>.5040-.5079</td>
<td>.5040-.5080</td>
<td>.5040-.5081</td>
<td>.5041-.5081</td>
<td>.5041-.5082</td>
<td>.5041-.5082</td>
</tr>
<tr>
<td></td>
<td>.5165</td>
<td>.5156</td>
<td>.5110</td>
<td>.5069</td>
<td>.5066</td>
<td>.5071</td>
<td>.5058</td>
<td>.5080</td>
</tr>
<tr>
<td></td>
<td>.0459</td>
<td>.0378</td>
<td>.0371</td>
<td>.0270</td>
<td>.0259</td>
<td>.0237</td>
<td>.0185</td>
<td>.0213</td>
</tr>
</tbody>
</table>
3.4.2 Probability of Error

In the equal covariance case \( \Sigma_1 = \Sigma_2 = \Sigma \), the optimal classifier is linear:

\[
h(x) = (\mu_2 - \mu_1)^T \Sigma^{-1} x + \frac{1}{2} (\mu_1^T \Sigma^{-1} \mu_1 - \mu_2^T \Sigma^{-1} \mu_2) + \log \frac{P_2}{P_1} \begin{cases} < 0 \Rightarrow \text{class 1} \\ > 0 \Rightarrow \text{class 2} \end{cases}
\]  

(3.25)

When the true parameter values are used in \( h(x) \), the above linear classifier minimizes the probability of error. This probability, which is referred to as the Bayes’ error can be obtained as follows [7]:

\[
\text{err}^* = P_1 [1 - \Phi(t)] + P_2 \Phi(t - A)
\]

(3.26)

where \( t \) is as defined in (3.9) and \( \Delta^2 \) is the Mahalanobis distance between the two classes.

If the estimates of the parameters are used in place of their true values in \( h(x) \), the error increases. The probability of error is therefore a convex function of the parameters in the neighborhood of the true parameter values. Hence, by using results 3.1 and 3.2, the bounds provided by lemma 3.1 can be used to obtain upper and lower bounds on the bias and variance of the probability of error when estimates of the parameters are used in \( h(x) \).

We use the functional form of the error probability that was derived in [17]:

\[
\text{err} = \frac{1}{2} + \frac{1}{2\pi} \int_{-\infty}^{\infty} \int e^{joh(x)} [P_1(f_1(x) - P_2 f_2(x))] dx \text{d} \omega
\]

(3.27)

Using (3.17), we can write:
where $\text{cov}(\hat{\theta})$ in the right hand side of (3.28) can be replaced with its upper and lower bounds in order to determine the upper and lower bounds on $E(\epsilon^{err})$. By inserting (3.27) into (3.28) one gets the following:

$$E(\epsilon^{err}) = \epsilon^{err} + \frac{1}{2} \text{tr} \left\{ \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{\partial^2 \epsilon^{err}}{\partial \theta^2} e^{j\omega(x)} \right\} [\text{cov}(\theta)]_{\theta=\theta^*} \left[ P_1(x) - P_2(x) \right] dxd\omega \text{cov}(\theta)$$

$$= \epsilon^{err} + \frac{1}{2} \text{tr} \left\{ \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{\partial^2 \epsilon^{err}}{\partial \theta^2} e^{j\omega(x)} \right\} \left[ P_1 f_1(x) - P_2 f_2(x) \right] dxd\omega \text{cov}(\theta)$$

where:

$$\left[ \frac{\partial h(x)}{\partial \theta} \frac{\partial h(x)^T}{\partial \theta} \right]_{\theta=\theta^*} = \begin{bmatrix} x^T & -x(x-\mu_2)^T \\ -(x-\mu_2)x^T & (x-\mu_2)(x-\mu_2)^T \end{bmatrix}$$

and

$$\frac{\partial^2 h(x)}{\partial \theta^2} = \begin{bmatrix} I_d & 0 \\ 0 & -I_d \end{bmatrix}$$

The integrals in (3.29) can be evaluated easily by the method described in [17]. For the equal prior probability case ($P_1 = P_2 = 0.5$), replacing $\text{cov}(\hat{\theta})$ in (3.29) by its upper and lower bounds provided by lemma 3.1 results in the following inequalities for the bias of $\epsilon^{err}$:

$$\text{bias}(\epsilon^{err}) \geq \frac{1}{n\sqrt{2\pi\Delta}} e^{-\frac{1}{8\Delta^2}} \left[ \frac{\Delta^2}{4} + d - 1 \right]$$

(supervised lower bound) (3.30)
Notice that the supervised lower bound (3.30) is identical to the expression obtained by Okamoto [16] when supervised learning is used with \( n/2 \) training samples from each class, and the common covariance matrix is known.

Similarly, the variance of \( \hat{e} \) can be obtained by equation (3.18). However, since \( \frac{\partial \hat{e}}{\partial \hat{e}} \) is zero at \( \theta^* \), the variance and its bounds are all zero at this level of approximation. In a manner similar to the supervised learning case [7, pp 213-2171, it is possible to show that the \( \text{var}(\hat{e}) \) is \( O\left(\frac{1}{n^2}\right) \) and is therefore negligible (see Appendix B).

Figure 3.3 shows the upper and lower bounds of the bias of the probability of error (in percent) versus \( A \) (square root of the Mahalanobis distance), when \( P_1 = P_2, d=4, \) and \( n=1000. \) Notice, that as \( A \) increases the unsupervised curves approach the supervised lower bound curve indicating that when classes are far from each other, unsupervised learning can perform comparably to supervised learning. A similar conclusion was drawn in [20,21]. Figures 3.4 and 3.5 show the bounds on the number of unlabeled samples required to keep the bias of classification error to less that 1% when dimensionality changes. Figure 3.4 is for the case \( A = 1.0, \) and Figure 3.5 is for \( A = 2.0. \)
Figure 3.3: The bounds on the bias of the classification error versus $A$
$(n=1000, P_1 = P_2, d=4)$
Figure 3.4: Bounds on the number of samples required to have bias (\( \hat{e}_\tau \)) < 0.01 versus dimensionality, when \( A = 1.0, P_1 = P_2 \).
Tables 3.3 and 3.4 contain the results of the experiments performed for verifying (3.30) and (3.31). Dimensionality was changed from 2 to 16 in intervals of 2. The number of unlabeled samples was $n=kd$, where $k$ was selected to be 10, 20, 30, 50, 100, and 200. Prior probabilities were both equal to 0.5. Samples were generated, and using unsupervised learning, the ML estimates of $\mu_1$ and $\mu_2$ were computed. The probability of error for the classifier obtained by using these estimates was then found. The procedure was repeated 20 times independently. Table 3.3 is for the case $A = 1.0$, and Table 3.4 is for $A = 2.0$. The first line in the tables is the supervised lower bound for $e_{\text{fr}}$ (from (3.30)), the second line is the lower-upper bounds from (3.31a) and (3.31b), the third line is the
mean of the 20 trials and the fourth line is the standard deviation of the 20 trials. One can see through Tables 3.3 and 3.4 that the experiments match the theory closely. The discrepancies observed in the tables are due to the fact that the bounds are asymptotic, as discussed in the last sub-section concerning Tables 3.1 and 3.2.
### Table 3.3: Bounds On $E\{ e^2 \}$: $A = 1.0$ ($err = 30.85\%$)

(First line: supervised lower bound.
Second line: lower-upper bounds
Third line: mean of 20 trials
Fourth line: standard deviation of 20 trials)

<table>
<thead>
<tr>
<th>$k$</th>
<th>d</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>33.05</td>
<td>33.71</td>
<td>33.93</td>
<td>34.04</td>
<td>34.11</td>
<td>34.15</td>
<td>34.18</td>
<td>34.21</td>
<td></td>
</tr>
<tr>
<td></td>
<td>38.18-45.51</td>
<td>41.39-50.00</td>
<td>42.46-50.00</td>
<td>43.00-50.00</td>
<td>43.32-50.00</td>
<td>43.53-50.00</td>
<td>43.69-50.00</td>
<td>43.80-50.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>36.37</td>
<td>39.62</td>
<td>39.49</td>
<td>41.57</td>
<td>40.95</td>
<td>42.09</td>
<td>41.74</td>
<td>43.21</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5.71</td>
<td>6.04</td>
<td>4.45</td>
<td>5.11</td>
<td>4.68</td>
<td>4.79</td>
<td>3.79</td>
<td>5.54</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>31.95</td>
<td>32.28</td>
<td>32.39</td>
<td>32.45</td>
<td>32.48</td>
<td>32.50</td>
<td>32.52</td>
<td>32.53</td>
<td></td>
</tr>
<tr>
<td></td>
<td>34.52-38.18</td>
<td>36.12-41.39</td>
<td>36.66-42.46</td>
<td>36.92-43.00</td>
<td>37.08-43.32</td>
<td>37.19-43.53</td>
<td>37.27-43.69</td>
<td>37.33-43.80</td>
<td></td>
</tr>
<tr>
<td></td>
<td>34.73</td>
<td>35.66</td>
<td>36.95</td>
<td>38.46</td>
<td>38.57</td>
<td>38.29</td>
<td>37.10</td>
<td>39.12</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.73</td>
<td>3.55</td>
<td>4.16</td>
<td>5.97</td>
<td>5.10</td>
<td>4.41</td>
<td>3.05</td>
<td>5.11</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>31.58</td>
<td>31.80</td>
<td>31.88</td>
<td>31.91</td>
<td>31.94</td>
<td>31.95</td>
<td>31.96</td>
<td>31.97</td>
<td></td>
</tr>
<tr>
<td></td>
<td>33.29-35.74</td>
<td>34.36-37.86</td>
<td>34.70-38.49</td>
<td>34.90-38.95</td>
<td>35.01-39.16</td>
<td>35.08-39.31</td>
<td>35.13-39.41</td>
<td>35.17-39.48</td>
<td></td>
</tr>
<tr>
<td></td>
<td>33.76</td>
<td>35.19</td>
<td>35.41</td>
<td>36.71</td>
<td>37.03</td>
<td>35.08</td>
<td>35.01</td>
<td>36.27</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.52</td>
<td>3.04</td>
<td>3.79</td>
<td>5.68</td>
<td>5.20</td>
<td>2.53</td>
<td>2.74</td>
<td>4.22</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>31.29</td>
<td>31.42</td>
<td>31.47</td>
<td>31.49</td>
<td>31.50</td>
<td>31.51</td>
<td>31.52</td>
<td>31.52</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32.32-33.78</td>
<td>32.96-35.07</td>
<td>33.17-35.50</td>
<td>33.28-35.71</td>
<td>33.34-35.84</td>
<td>33.39-35.92</td>
<td>33.42-35.98</td>
<td>33.44-36.03</td>
<td></td>
</tr>
<tr>
<td></td>
<td>33.53</td>
<td>34.58</td>
<td>35.15</td>
<td>33.62</td>
<td>33.85</td>
<td>33.91</td>
<td>34.64</td>
<td>33.85</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.99</td>
<td>4.16</td>
<td>4.54</td>
<td>1.84</td>
<td>1.88</td>
<td>2.15</td>
<td>3.24</td>
<td>1.51</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>31.07</td>
<td>31.14</td>
<td>31.16</td>
<td>31.17</td>
<td>31.18</td>
<td>31.18</td>
<td>31.18</td>
<td>31.19</td>
<td></td>
</tr>
<tr>
<td></td>
<td>31.58-32.32</td>
<td>31.90-32.96</td>
<td>32.01-33.17</td>
<td>32.06-33.28</td>
<td>32.10-33.34</td>
<td>32.12-33.39</td>
<td>32.13-33.42</td>
<td>32.15-33.44</td>
<td></td>
</tr>
<tr>
<td></td>
<td>32.40</td>
<td>32.18</td>
<td>31.95</td>
<td>32.54</td>
<td>32.13</td>
<td>32.40</td>
<td>32.53</td>
<td>32.49</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.21</td>
<td>1.28</td>
<td>0.75</td>
<td>1.53</td>
<td>0.73</td>
<td>1.04</td>
<td>0.69</td>
<td>0.98</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>30.96</td>
<td>30.99</td>
<td>31.00</td>
<td>31.01</td>
<td>31.01</td>
<td>31.02</td>
<td>31.02</td>
<td>31.02</td>
<td></td>
</tr>
<tr>
<td></td>
<td>31.03</td>
<td>31.37</td>
<td>31.42</td>
<td>31.57</td>
<td>31.61</td>
<td>31.68</td>
<td>31.62</td>
<td>31.56</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.31</td>
<td>0.44</td>
<td>0.28</td>
<td>0.44</td>
<td>0.52</td>
<td>0.36</td>
<td>0.42</td>
<td>0.26</td>
<td></td>
</tr>
</tbody>
</table>
Table 3.4: Bounds On $E\{ \hat{e}r \}$: $A = 2.0$ (err = 15.87%)

(First line: supervised lower bound.
Second line: lower-upper bounds
Third line: mean of 20 trials
Fourth line: standard deviation of 20 trials)

<table>
<thead>
<tr>
<th>d</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>17.08</td>
<td>17.08</td>
<td>17.08</td>
<td>17.08</td>
<td>17.08</td>
<td>17.08</td>
<td>17.08</td>
<td>17.08</td>
</tr>
<tr>
<td></td>
<td>18.36</td>
<td>17.86</td>
<td>17.69</td>
<td>17.99</td>
<td>18.32</td>
<td>18.23</td>
<td>18.10</td>
<td>17.77</td>
</tr>
<tr>
<td></td>
<td>2.92</td>
<td>1.24</td>
<td>1.46</td>
<td>1.52</td>
<td>1.10</td>
<td>1.13</td>
<td>1.04</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>16.92-17.96</td>
<td>16.90-17.93</td>
<td>16.89-17.92</td>
<td>16.89-17.91</td>
<td>16.89-17.91</td>
<td>16.89-17.91</td>
<td>16.89-17.90</td>
<td>16.89-17.90</td>
</tr>
<tr>
<td></td>
<td>17.15</td>
<td>17.07</td>
<td>16.94</td>
<td>16.79</td>
<td>17.06</td>
<td>17.13</td>
<td>17.00</td>
<td>16.81</td>
</tr>
<tr>
<td></td>
<td>1.04</td>
<td>0.87</td>
<td>0.86</td>
<td>0.67</td>
<td>0.58</td>
<td>0.54</td>
<td>0.56</td>
<td>0.38</td>
</tr>
<tr>
<td>30</td>
<td>16.27</td>
<td>16.27</td>
<td>16.27</td>
<td>16.27</td>
<td>16.27</td>
<td>16.27</td>
<td>16.27</td>
<td>16.27</td>
</tr>
<tr>
<td></td>
<td>0.73</td>
<td>0.51</td>
<td>0.55</td>
<td>0.45</td>
<td>0.33</td>
<td>0.39</td>
<td>0.26</td>
<td>0.21</td>
</tr>
<tr>
<td></td>
<td>0.24</td>
<td>0.41</td>
<td>0.32</td>
<td>0.23</td>
<td>0.24</td>
<td>0.18</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>100</td>
<td>15.99</td>
<td>15.99</td>
<td>15.99</td>
<td>15.99</td>
<td>15.99</td>
<td>15.99</td>
<td>15.99</td>
<td>15.99</td>
</tr>
<tr>
<td></td>
<td>16.04</td>
<td>16.11</td>
<td>16.10</td>
<td>16.06</td>
<td>16.06</td>
<td>16.05</td>
<td>16.09</td>
<td>16.08</td>
</tr>
<tr>
<td></td>
<td>0.19</td>
<td>0.18</td>
<td>0.21</td>
<td>0.07</td>
<td>0.09</td>
<td>0.07</td>
<td>0.08</td>
<td>0.08</td>
</tr>
<tr>
<td>200</td>
<td>15.93</td>
<td>15.93</td>
<td>15.93</td>
<td>15.93</td>
<td>15.93</td>
<td>15.93</td>
<td>15.93</td>
<td>15.93</td>
</tr>
<tr>
<td></td>
<td>15.95</td>
<td>15.97</td>
<td>15.96</td>
<td>15.96</td>
<td>15.97</td>
<td>15.97</td>
<td>15.96</td>
<td>15.97</td>
</tr>
<tr>
<td></td>
<td>0.05</td>
<td>0.08</td>
<td>0.08</td>
<td>0.05</td>
<td>0.04</td>
<td>0.04</td>
<td>0.03</td>
<td>0.04</td>
</tr>
</tbody>
</table>
3.5. Combined Supervised-Unsupervised Learning

In a variety of applications, both training and unlabeled samples are available. In such cases, the unlabeled samples can be used to improve upon the decision rule obtained by training samples alone [21,37,38,39]. It is possible to estimate the parameters of the decision rule using both kinds of samples simultaneously. For example, the EM algorithm can be used for obtaining the ML estimates in this case [5]. This method of estimation is referred to as combined supervised-unsupervised learning. The ML estimates obtained in this way are asymptotically unbiased and have lower variances than the ones obtained by supervised (using training samples alone) or unsupervised learning alone. Assuming that the training and unlabeled samples are statistically independent, one can write the Fisher information matrix corresponding to the combined supervised-unsupervised learning as the sum of the information matrices corresponding to the training and unlabeled samples. By using the bounds obtained for the Fisher information matrix corresponding to the unlabeled samples in section 3.3, similar bounds can be obtained for the combined supervised-unsupervised learning case. These bounds can then be used to determine upper and lower bounds for bias of classification error as was done in the previous section for the unsupervised learning case.

Assume that in addition to \( n \) unlabeled samples, \( n_1 \) training samples from class 1 and \( n_2 \) training samples from class 2 are also available. If the estimate of the parameter set \( \theta = [\mu_1^T \mu_2^T]^T \) obtained by using all of these samples is used in the decision rule (3.25), the bias of the classification error, for the case \( P_1 = P_2 \), is bounded as follows:

\[
\text{bias}(err) \geq \frac{1}{n_1 + n/2} + \frac{1}{n_2 + n/2} \frac{1}{4\sqrt{2\pi}d/2} e^{-\frac{1}{8}d^2} \left[ \frac{\Delta^2}{4} + d - 1 \right] \tag{3.32}
\]

(supervised lower bound)
The variance of $\epsilon_{\text{err}}$ is again negligible since it is inversely proportional to the square of the number of design samples.

Figure 3.6 shows the bounds of the bias of the probability of error versus $A$ when $P_1=P_2$, $d=4$, $n=100$, and $n_1=n_2=20$. The no-unlabeled curve in Figure 3.6 refers to the case when only the training samples are used. It can be seen that by utilizing unlabeled samples, the bias of the classification error decreases. The amount of this reduction depends on the separation between the two classes as characterized by $A$. Table 3.5 contains the results of a set of experiments performed for verifying the equation (3.33). In addition to $n=k d$ unlabeled samples, at each trial of the experiment, 10 training samples from each class are also randomly drawn. The details of the experiments are the same as those performed for the unsupervised learning case. The first row of the Table 5 ($k=0$) contains the results
for the supervised learning case where no unlabeled samples are used. The discrepancies in the table are due to the asymptotic nature of the bounds as discussed for previous tabular results. Notice however, that even when the mean of the 20 trials does not fall within the theoretical bounds, the bounds are within one standard deviation from the mean.

Figure 3.6: The bounds on the bias of classification error versus $\Delta$ for combined supervised-unsupervised learning ($n=100, n_1=n_2=20, P_1=P_2, d=4$)
Table 3.5: Bounds On $E[\hat{e}^r]$; $A = 1.0$ (err* = 30.85%)
for combined learning with 10 training samples from each class
(First line: theoretical bound(s).
Second line: mean of 20 trials
Third line: standard deviation of 20 trials)

<table>
<thead>
<tr>
<th>k</th>
<th>d</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>32.48</td>
<td>34.49</td>
<td>35.28</td>
<td>35.49</td>
<td>36.04</td>
<td>37.13</td>
<td>38.00</td>
<td>39.04</td>
<td>39.92</td>
</tr>
<tr>
<td>1</td>
<td>35.30</td>
<td>36.45</td>
<td>38.00</td>
<td>38.90</td>
<td>39.04</td>
<td>39.76</td>
<td>39.76</td>
<td>39.92</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>35.54</td>
<td>36.38</td>
<td>37.74</td>
<td>38.52</td>
<td>38.41</td>
<td>39.82</td>
<td>39.82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>32.48</td>
<td>34.49</td>
<td>35.28</td>
<td>35.49</td>
<td>36.04</td>
<td>37.13</td>
<td>38.00</td>
<td>39.04</td>
<td>39.92</td>
</tr>
<tr>
<td>4</td>
<td>35.30</td>
<td>36.45</td>
<td>38.00</td>
<td>38.90</td>
<td>39.04</td>
<td>39.76</td>
<td>39.76</td>
<td>39.92</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>35.54</td>
<td>36.38</td>
<td>37.74</td>
<td>38.52</td>
<td>38.41</td>
<td>39.82</td>
<td>39.82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>32.48</td>
<td>34.49</td>
<td>35.28</td>
<td>35.49</td>
<td>36.04</td>
<td>37.13</td>
<td>38.00</td>
<td>39.04</td>
<td>39.92</td>
</tr>
<tr>
<td>7</td>
<td>35.30</td>
<td>36.45</td>
<td>38.00</td>
<td>38.90</td>
<td>39.04</td>
<td>39.76</td>
<td>39.76</td>
<td>39.92</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>35.54</td>
<td>36.38</td>
<td>37.74</td>
<td>38.52</td>
<td>38.41</td>
<td>39.82</td>
<td>39.82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>32.48</td>
<td>34.49</td>
<td>35.28</td>
<td>35.49</td>
<td>36.04</td>
<td>37.13</td>
<td>38.00</td>
<td>39.04</td>
<td>39.92</td>
</tr>
<tr>
<td>10</td>
<td>35.30</td>
<td>36.45</td>
<td>38.00</td>
<td>38.90</td>
<td>39.04</td>
<td>39.76</td>
<td>39.76</td>
<td>39.92</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>35.54</td>
<td>36.38</td>
<td>37.74</td>
<td>38.52</td>
<td>38.41</td>
<td>39.82</td>
<td>39.82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>32.48</td>
<td>34.49</td>
<td>35.28</td>
<td>35.49</td>
<td>36.04</td>
<td>37.13</td>
<td>38.00</td>
<td>39.04</td>
<td>39.92</td>
</tr>
<tr>
<td>13</td>
<td>35.30</td>
<td>36.45</td>
<td>38.00</td>
<td>38.90</td>
<td>39.04</td>
<td>39.76</td>
<td>39.76</td>
<td>39.92</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>35.54</td>
<td>36.38</td>
<td>37.74</td>
<td>38.52</td>
<td>38.41</td>
<td>39.82</td>
<td>39.82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>32.48</td>
<td>34.49</td>
<td>35.28</td>
<td>35.49</td>
<td>36.04</td>
<td>37.13</td>
<td>38.00</td>
<td>39.04</td>
<td>39.92</td>
</tr>
<tr>
<td>16</td>
<td>35.30</td>
<td>36.45</td>
<td>38.00</td>
<td>38.90</td>
<td>39.04</td>
<td>39.76</td>
<td>39.76</td>
<td>39.92</td>
<td></td>
</tr>
</tbody>
</table>
3.6. Conclusions

The effect of limited design samples in the performance of linear classifiers designed by unsupervised and combined supervised-unsupervised learning was studied in this chapter. Consideration was given to the case of a known common covariance matrix. Upper and lower bounds for the asymptotic covariance matrix of the ML estimate of the parameter set was obtained and used to derive bounds for the bias and variance of the estimated Bhattacharyya distance and bias of the classification error. It was noted that unsupervised learning can perform comparably to supervised learning when the classes are relatively separated. On the other hand, when the classes are highly overlapped, a large number of unlabeled samples is required for designing a classifier that matches the performance of one designed using supervised learning.

When both unlabeled samples and training samples are available, combined supervised-unsupervised learning can be used to obtain a decision rule that outperforms the one obtained by supervised learning alone. This result is important for situations where the number of training samples is small compared to the dimensionality of data (e.g. remote sensing data obtained by new high spectral resolution sensors) and at the same time plenty of unlabeled samples are available. In such cases, the so called Hughes phenomenon [6], the decrease in classification accuracy when the dimensionality of the data increases and the number of training samples is fixed, may be mitigated by utilizing the unlabeled samples, as described in chapter 2. Based on this we recommend that unlabeled samples be used in addition to the training samples for estimating the parameters of the classes. The EM equations can be used for this purpose [5, and chapter 5].
The techniques used in this chapter can be extended to the case of an unknown common covariance matrix. Similar bounds to those of theorem 3.1 can be obtained for the Fisher information matrix, with the distinction that the third and fourth moments of the truncated normal distributions need to be used. Consequently, the results would be more complex.
CHAPTER 4: THE MULTIPLE CLASS PROBLEM

In chapter 3, the supervised, unsupervised, and combined supervised-unsupervised parametric learning processes were compared for a two class problem. When the number of classes is more than two, analysis becomes more difficult. In fact, obtaining closed form equations for the probability of error in a multi-class problem is itself a challenge. In this chapter, we pay attention to the estimates of the mean vectors and covariance matrices in a multi-class problem. The asymptotic covariances of the ML estimates, which are the Cramer-Rao bounds for the estimates, are compared for the supervised, unsupervised, and combined supervised-unsupervised learning processes. Upper and lower bounds on the asymptotic covariance matrices are derived. It is shown that under a normal mixture density assumption for the probability density function of the feature space, the combined supervised-unsupervised learning is always superior to the supervised learning in achieving better estimates. Experimental results are provided.

4.1. Introduction

Consider a classification problem involving m Gaussian classes. The pdf of the total feature space can be modeled by a mixture of these Gaussian classes in the following way:
The Multiple Class Problem

\[ f(x|\theta) = \sum_{i=1}^{m} P_i f_i(x|\phi_i) \]  

where

\[ \sum_{i=1}^{m} P_i = 1, \quad P_i > 0 \]

and

\[ \phi_i = (\mu_i, \Sigma_i), \quad \theta = (P_1, \ldots, P_m, \mu_1, \ldots, \mu_m, \Sigma_1, \ldots, \Sigma_m) \]

where \( P_i \)'s are the prior probabilities of the components, and \( f_i(.) \) is the \( i^{th} \) normal component with mean \( \mu_i \) and covariance matrix \( \Sigma_i \).

In supervised learning, it is assumed that from each class \( i \) (\( i = 1, \ldots, m \)), \( n_i \) training samples are available. These samples are denoted by \( z_{ik} \) where \( i = 1, \ldots, m \) indicates the class of origin and \( k \) is the index of each particular sample (\( k = 1, \ldots, n_i \)). The parameters of the classes are estimated by the Maximum Likelihood (ML) estimators:

\[ \hat{\mu}_i = \frac{1}{n_i} \sum_{k=1}^{n_i} z_{ik} \]  

(4.2)

and

\[ \hat{\Sigma}_i = \frac{1}{n_i} \sum_{k=1}^{n_i} (z_{ik} - \hat{\mu}_i)(z_{ik} - \hat{\mu}_i)^T \]  

(4.3)

Notice that in supervised learning, since the training samples are drawn separately from each class, the prior probabilities of classes cannot be estimated from the training data.
In unsupervised learning, $n$ unlabeled samples denoted by $x_k (k=1,...,n)$ are assumed to be available from the mixture. The mixture density can then be identified by using the following iterative equations (see chapter 5 for more details) [5]:

$$
\pi_i^+ = \frac{\sum_{k=1}^{n} p_i^c f_i(x_k | \mu_i^c, \Sigma_i^c)}{n} \frac{f(x_k | \theta^c)}{f(x_k | \theta^c)}
$$

(4.4)

$$
\mu_i^+ = \frac{\sum_{k=1}^{n} p_i^c f_i(x_k | \mu_i^c, \Sigma_i^c) x_k}{\sum_{k=1}^{n} p_i^c f_i(x_k | \mu_i^c, \Sigma_i^c)} \frac{f(x_k | \theta^c)}{f(x_k | \theta^c)}
$$

(4.5)

and

$$
\Sigma_i^+ = \frac{\sum_{k=1}^{n} p_i^c f_i(x_k | \mu_i^c, \Sigma_i^c) (x_k - \mu_i^+)(x_k - \mu_i^+)^T}{\sum_{k=1}^{n} p_i^c f_i(x_k | \mu_i^c, \Sigma_i^c)} \frac{f(x_k | \theta^c)}{f(x_k | \theta^c)}
$$

(4.6)

Where superscript "c" denotes the current values of the parameters, and superscript "+" denotes the next values. Starting from any reasonable starting point the above equations are guaranteed to converge to a local maximum of the associated likelihood function [30].

In the combined supervised-unsupervised learning process, both of the above types of samples (training and unlabeled) are assumed to be available. Again, the ML estimates for the parameters of the mixture density can be approximated by using the equations 2.3, 2.4, and 2.5 of chapter 2.
In order to compare the above three learning processes, one can compare the estimates obtained in each case. The biases and covariance matrices of the ML estimators in the supervised learning case can be found easily [7]. Unfortunately, in the other two cases, since the estimators don't have closed form expressions these quantities cannot be found. It is possible, however, to use the asymptotic properties of the ML estimators to compare the different learning processes. In particular, ML estimators are known to be consistent and efficient [10]. Therefore, asymptotically, the biases of all the estimates are negligible and the covariance matrices are equal to the inverse of their corresponding Fisher information matrices. However, except for the supervised learning case, the Fisher information matrices themselves cannot be found explicitly either. Behboodian [33] has provided tables that can be used to estimate the information matrix in the one dimensional case with two normal components. For multidimensional cases, Chang [34] has proposed a method that uses several one dimensional numerical integrations. Recently, Perlovsky [35] formulated the information matrices in terms of variables that describe the amount of overlap between the components.

In this chapter, we derive upper and lower bounds for the Fisher information matrices of the parameters of a normal mixture density. Then we invert these bounds to obtain bounds for the asymptotic covariance matrices of the estimates in unsupervised and combined learning cases and use these bounds to compare the three learning processes.

42 Supervised Learning

In the supervised learning case, it is assumed that \( n_i \) training samples are available from the \( i^{th} \) component of the mixture whose pdf is \( N(x; \mu_i, \Sigma_i) \).
4.2.1 Mean Estimators

Denoting the Fisher information matrix associated with the $i^{th}$ component's mean ($\mu_i$) by $I_s(\mu_i)$ (where subscript "s" stands for supervised), one can write:

$$I_s(\mu_i) = n_i E \left[ \left( \frac{\partial}{\partial \mu_i} \log f_i(x) \right) \left( \frac{\partial}{\partial \mu_i} \log f_i(x) \right)^T \right]$$

$$= n_i \Sigma_i^{-1} E \left[ (x - \mu_i)(x - \mu_i)^T \right] \Sigma_i^{-1} = n_i \Sigma_i^{-1}$$

The asymptotic covariance matrix of the ML estimate of $\mu_i$ is therefore:

$$\text{cov}(\hat{\mu}_i) = [I_s(\mu_i)]^{-1} = \frac{1}{n_i} \Sigma_i$$

(4.7)

4.2.2 Inverse Covariance Matrix Estimators

In this subsection, the estimates of the inverse covariance matrices are studied. The inverses of the covariance matrices are studied instead of the covariance matrices for two reasons: first, mathematical simplicity; second, the fact that in practice the inverse covariance matrices are of more interest because they are used in discriminant functions. Furthermore, by using the invariance property of the ML estimators [10], it is possible to show that the ML estimate of $\Sigma_i^{-1}$ is simply the inverse of the ML estimate of $\Sigma_i$ [32].

Let $\Gamma_i$ be the vector containing the $d(d+1)/2$ unique entries of $\Sigma_i^{-1}$:

$$\Gamma_i = [\sigma^{11}(i) \ \sigma^{12}(i) \ \ldots \ \sigma^{1d}(i) \ \sigma^{22}(i) \ \ldots \ \sigma^{2d}(i) \ \ldots \ \sigma^{dd}(i)]^T$$
where $\sigma_{kl}^{(i)} = [\Sigma^{-1}]_{kl}$.

Denote the Fisher information matrix associated with $\Gamma_i$ by $I_s(\Sigma^{-1})$. Then:

$$I_s(\Sigma^{-1}) = n_i \mathbb{E} \left\{ \left[ \frac{\partial}{\partial \Gamma_i} \log f_i(x) \right] \begin{bmatrix} \frac{\partial}{\partial \Gamma_i} \log f_i(x) \end{bmatrix}^T \right\}$$  \hspace{1cm} (4.9)

but,

$$\frac{\partial \log f_i(x)}{\partial \sigma_{kl}^{(i)}} = \sigma_{kl}^{(i)} - (x_k - \mu_{ik})(x_l - \mu_{il}) \quad \text{if } k \neq l$$

and

$$\frac{\partial \log f_i(x)}{\partial \sigma_{kk}^{(i)}} = \frac{1}{2} \sigma_{kl}^{(i)} - \frac{1}{2} (x_k - \mu_{ik})^2$$

where $\sigma_{kl}^{(i)}$ is the $kl^{th}$ entry of $\Sigma$ [40, chapter 6]. Therefore:

$$I_s(\Sigma^{-1}) = n_i \mathbb{E} \left\{ [\alpha_i - \beta_i(x)] [\alpha_i - \beta_i(x)]^T \right\}$$ \hspace{1cm} (4.10)

where:

$$\alpha_i = \begin{bmatrix} 1/2 \sigma_{11}^{(i)} \\ \sigma_{12}^{(i)} \\ \sigma_{1d}^{(i)} \\ 1/2 \sigma_{22}^{(i)} \\ \sigma_{23}^{(i)} \\ 1/2 \sigma_{dd}^{(i)} \end{bmatrix} \quad \text{and} \quad \beta_i(x) = \begin{bmatrix} 1/2(x_1 - \mu_{il})^2 \\ (x_1 - \mu_{il})(x_2 - \mu_{il}) \\ (x_1 - \mu_{il})(x_d - \mu_{id}) \\ 1/2(x_2 - \mu_{il})^2 \\ (x_2 - \mu_{il})(x_3 - \mu_{i3}) \\ (x_2 - \mu_{il})(x_d - \mu_{id}) \\ 1/2(x_d - \mu_{il})^2 \end{bmatrix}$$ \hspace{1cm} (4.11)

Since $\mathbb{E}\{\beta_i(x)\} = \alpha_i$, we can write:
where \( \mathbf{C}_i \) is defined appropriately and its entries contain the fourth order statistics. Therefore the asymptotic covariance of \( \hat{\mu}_i \) is:

\[
\text{cov}(\hat{\mu}_i) = \frac{1}{n_i} \mathbf{C}_i^{-1}
\]  

(4.13)

### 4.3 Unsupervised Learning

In the unsupervised learning process, it is assumed that \( n \) unlabeled samples from the mixture are available. The ML estimates of the parameters of the classes are approximated by using the iterative equation described in section 4.1.

#### 4.3.1 Mean Estimators

Denoting the Fisher information matrix associated with the \( i^{th} \) component's mean (\( \mu_i \)) by \( I_u(\mu_i) \) (where subscript "u" stands for unsupervised), one can write:

\[
I_u(\mu_i) = nE \left[ \begin{bmatrix} \frac{\partial \log f(x)}{\partial \mu_i} & \frac{\partial \log f(x)}{\partial \mu_i} \end{bmatrix}^T \right] = nE \left\{ \frac{f_i^2(x)}{f(x)} \Sigma_i^{-1}(x - \mu_i)(x - \mu_i)^T \Sigma_i^{-1} dx \right\}
\]  

(4.14)

where the expectation is with respect to the mixture density. The above integral can not be evaluated in closed form. However, in order to compare unsupervised learning with supervised learning, upper and lower bounds for the information matrix can be obtained. In particular, the following theorem holds.
**Theorem 4.1:** The following matrix inequalities hold for the Fisher information matrix \( I_u(\mu_i) \):

\[
nP_i \Sigma_i^{-1} \geq I_u(\mu_i) \geq \frac{n \mathbb{P}_i^2}{2 \left( \sum_{j=1}^{m} \mathbb{P}_j / \sqrt{\text{det} \Sigma_j} \right)} \sqrt{2^d \text{det} \Sigma_i}^{-1} \tag{4.15}
\]

**Proof:**

Consider the mixture density \( f(x) \):

\[
f(x) = \sum_{j=1}^{m} P_j f_j(x)
\]

where

\[
f_j(x) = \frac{1}{\sqrt{(2\pi)^d \text{det} \Sigma_j}} \exp \left\{ -\frac{1}{2} (x - \mu_j)^T \Sigma_j^{-1} (x - \mu_j) \right\}
\]

One can write:

\[
P_i f_i(x) \leq f(x) \leq \sum_{j=1}^{m} \frac{P_j}{\sqrt{(2\pi)^d \text{det} \Sigma_j}} \quad \forall i \tag{4.16}
\]

Therefore, by substituting the lower bound of \( f(x) \) in the denominator of the right hand side of equation (4.14), it can be shown that the following is true:

\[
I_u(\mu_i) \leq n \int P_i f_i(x) \Sigma_i^{-1} (x - \mu_i) (x - \mu_i)^T \Sigma_i^{-1} dx = n \mathbb{P}_i \Sigma_i^{-1}
\]

Similarly, by substituting the upper bound of \( f(x) \) in the right hand side of (4.14), one gets the following inequality:
4 The Multiple Class Problem

\[ I_u(\mu_i) \geq \frac{n \pi_i^2}{\sum_{j=1}^{m} \frac{p_j}{\sqrt{2 \pi}^{d_j} \Sigma_j}} \int f_i^2(x) \Sigma_i^{-1} (x - \mu_i)(x - \mu_i)^T \Sigma_i^{-1} dx \]  
\hspace{1cm} (4.17)

However, since \( f_i(x) \) is a normal density function with mean vector \( \mu_i \) and covariance matrix \( \Sigma_i \), one can easily derive the following expression:

\[ f_i^2(x) = \frac{1}{\sqrt{2 \pi}^{d_i} \Sigma_i} N(x; \mu_i, \frac{1}{2} \Sigma_i) \]  
\hspace{1cm} (4.18)

and substituting for \( f_i^2(x) \) into the equation (4.17), the following can be obtained:

\[ I_u(\mu_i) \geq \frac{n \pi_i^2}{\sum_{j=1}^{m} \frac{p_j}{\sqrt{2 \pi}^{d_j} \Sigma_j}} \Sigma_i^{-1} \left( \sum_{j=1}^{m} \frac{p_j}{\sqrt{\Sigma_j}} \right) \sqrt{2^{d_i} \Sigma_i} \]

\(<Q.E.D>\)

The lower bound of theorem 4.1 is obviously not a very tight bound since in order to obtain it the mixture density \( f(x) \) was replaced by its absolute maximum over the entire feature space. The upper bound is tighter. In fact, if the components are well separated, the upper bound in (4.15) is very close to the true value of the Fisher information matrix \( I_u(\mu_i) \). This can be shown mathematically by partitioning the feature space into disjoint regions \( \Omega_i \), where \( \Omega_i \) is the region for which \( P_i f_i(x) \) is greater than all the other \( P_j f_j(x) \):

\[ x \in \Omega_i \iff P_i f_i(x) \geq P_j f_j(x) \quad \forall j \]
Then it is possible to write the equation (4.14) as:

\[
I_u(\mu_i) = nP_i^2 \int_{\Omega_i} \frac{f_i^2(x)}{f(x)} \Sigma_i^{-1}(x-\mu_i)(x-\mu_i)^T \Sigma_i^{-1} dx + \int_{\Omega_2} \cdots + \int_{\Omega_m}
\]

If the components are well separated, over the region \(\Omega_j\) (where \(j \neq i\),) \(f_i(x)\) is very small and therefore \(f_i^2(x)/f(x)\) is negligible, hence:

\[
I_u(\mu_i) = nP_i^2 \int_{\Omega_i} \frac{f_i^2(x)}{f(x)} \Sigma_i^{-1}(x-\mu_i)(x-\mu_i)^T \Sigma_i^{-1} dx
\]

But, for well separated components, on region \(\Omega_i\), \(f(x) \approx P_i f_i(x)\). Therefore:

\[
I_u(\mu_i) = nP_i \int_{\Omega_i} f_i(x) \Sigma_i^{-1}(x-\mu_i)(x-\mu_i)^T \Sigma_i^{-1} dx
\]

\[
\approx nP_i \int f_i(x) \Sigma_i^{-1}(x-\mu_i)(x-\mu_i)^T \Sigma_i^{-1} dx = nP_i \Sigma_i^{-1}
\]

where the above is established because for well separated components the region \(\Omega_i\) contains almost all the sampling region of \(f_i(x)\).

It is interesting to mention an intuitively appealing interpretation of the upper bound. The upper bound of theorem 4.1 is similar to the Fisher information matrix associated with \(\mu_i\) in the supervised learning case if \(nP_i\) training samples were available from the \(i^{th}\) component. In fact, from the \(n\) unlabeled samples drawn over the mixture, on average only \(nP_i\) belong to the \(i^{th}\) component. The rest belong to the other components and therefore contain no information regarding the statistics of the \(i^{th}\) component. Therefore,
it is reasonable to expect that the information in the \( n \) unlabeled samples is at most equal to the information in \( nP_i \) training (labeled) samples drawn from the \( i^{\text{th}} \) component. Of course, since the origins of the unlabeled samples are unknown this information is only an upper bound, as is shown in theorem 4.1. However, if the components are well separated, the unlabeled samples can easily be labeled. Therefore, if the components are distant, the upper bound of theorem 4.1 is close to the true value of the Fisher information matrix.

The bounds provided by theorem 4.1 can now be used to derive: bounds for the asymptotic covariance matrices of the mean estimators. The following lemma provides the bounds:

**Lemma 4.1:** The asymptotic covariance matrix of the ML estimate of \( \mu_i \) based on \( n \) unlabeled samples from the mixture is bounded as follows:

\[
\frac{1}{nP_i} \Sigma_i \leq \text{cov}(\hat{\mu}_i) = [I_u(\mu_i)]^{-1} \leq \frac{2}{nP_i^2} \sum_{j=1}^{m} \frac{P_j}{\sqrt{\Sigma_j}} \sqrt{2d |\Sigma_j|} \Sigma_i
\]  

(4.19)

**Proof:**

A theorem in linear algebra states that if \( A \) and \( B \) are positive definite matrices and \( A \succeq B \) then \( B^{-1} \succeq A^{-1} \) (see [9] for proof). Therefore, using the inequalities of theorem 4.1 and noticing that \( \Sigma_i^{-1} \) is a positive definite matrix, the lemma can be proved trivially.

\(<Q.E.D>\)
Using the above lemma, it is possible to find an upper bound for the number of unlabeled samples required in order to get the same quality estimates as obtained by the supervised learning process with a known number of training samples.

4.3.2 Inverse Covariance Matrix Estimators

It is possible to obtain results similar to those in section 4.2.1, for the estimates of the inverse covariance matrices. The Fisher information matrix associated with the estimate of $\Gamma_i$ is denoted by $I_u(\Sigma_i^{-1})$, where $\Gamma_i$ was defined in section 4.2.2 as the vector containing the $d(d+1)/2$ unique entries of $\Sigma_i^{-1}$. One can write:

$$I_u(\Sigma_i^{-1}) = nE\left[\left(\frac{\partial \log f(x)}{\partial \Gamma_i} \right)^T \left(\frac{\partial \log f(x)}{\partial \Gamma_i} \right)\right] = nP_i^2 \int \frac{f_i^2(x)}{f(x)} \left[\alpha_i - \beta_i(x)\right]\left[\alpha_i - \beta_i(x)\right]^T dx \quad (4.20)$$

where as in the case of $I_u(\mu_i)$, the above expectation cannot be put in closed form. But a theorem similar to theorem 4.1 can be stated:

Theorem 4.2: The following matrix inequalities hold for the Fisher information matrix $I_u(\Sigma_i^{-1})$:

$$nP_i C_i \geq I_u(\Sigma_i^{-1}) \geq \frac{nP_i^2}{\sum_{j=1}^{m} \frac{P_j}{|\Sigma_j|}} \left(\frac{2^d |\Sigma_i|}{\sum_{j=1}^{m} \frac{P_j}{|\Sigma_j|}}\right)^{-1} \quad (4.21)$$

where $C_i$ is as defined in section 4.2.2.
Proof:

Similar to the proof of theorem 4.1, one can substitute the lower bound of $f(x)$ from equation (4.16) in the righthand side of equation (4.20) to obtain the following inequality:

$$I_u(\Sigma^{-1}_i) \leq nP_i \int f_i(x)[\alpha_i - \beta_i(x)][\alpha_i - \beta_i(x)]^T dx = nP_iC_i$$

Similarly, substituting the upper bound of $f(x)$ from (4.16) into the right hand side of (4.20), one gets:

$$I_u(\Sigma^{-1}_i) \geq \frac{nP_i^2}{\sum_{j=1}^m \frac{P_i}{\sqrt{\Sigma_j}}} \int f_i^2(x)[\alpha_i - \beta_i(x)][\alpha_i - \beta_i(x)]^T dx$$

$$= \frac{nP_i^2}{\sum_{j=1}^m \frac{P_i}{\sqrt{\Sigma_j}}} \int N(x; \mu_i, \frac{1}{2} \Sigma_i)[\alpha_i - \beta_i(x)][\alpha_i - \beta_i(x)]^T dx$$

(4.22)

where (4.18) was used for getting the above result. Since $\beta_i(x)$ contains the second order terms one can write:

$$\int \beta_i(x)N(x; \mu_i, \frac{1}{2} \Sigma_i)dx = \frac{1}{2} \alpha_i$$

(4.23)

Hence one can write:

$$I_u(\Sigma^{-1}_i) \geq \frac{nP_i^2}{\sum_{j=1}^m \frac{P_i}{\sqrt{\Sigma_j}}} \int N(x; \mu_i, \frac{1}{2} \Sigma_i)\beta_i(x)\beta_i^T(x)dx$$

(4.24)
Now, the entries of $\beta_i(x) \beta_i^T(x)$ are fourth order terms and for a normal density $N(\mu; \Sigma)$ the fourth order moments can be written as follows [32]:

$$E_{N(\mu; \Sigma)} \{ (x_k - \mu_k) (x_l - \mu_l) (x_m - \mu_m) (x_n - \mu_n) \} = \sigma_{kl} \sigma_{mn} + \sigma_{km} \sigma_{ln} + \sigma_{kn} \sigma_{lm}$$

where $\sigma_{kl} = [\Sigma]_{kl}$.

We can write the following for a normal density $N(\mu; \frac{1}{2} \Sigma)$:

$$E_{N(\mu; \frac{1}{2} \Sigma)} \{ (x_k - \mu_k)(x_l - \mu_l)(x_m - \mu_m)(x_n - \mu_n) \}$$

$$= \frac{1}{4} E_{N(\mu; \Sigma)} \{ (x_k - \mu_k)(x_l - \mu_l)(x_m - \mu_m)(x_n - \mu_n) \}$$

hence:

$$\int N(x; \mu_i, \frac{1}{2} \Sigma_i) \beta_i(x) \beta_i^T(x) \, dx = \frac{1}{4} \int N(x; \mu_i, \Sigma_i) \beta_i(x) \beta_i^T(x) \, dx = \frac{1}{4} \left( C_i + \alpha_i \alpha_i^T \right) \quad (4.25)$$

where $C_i$ was defined in section 4.2.2 as $C_i \equiv E\{ \beta_i(x) \beta_i^T(x) \} - a_i a_i^T$. Therefore:

$$I_n(\Sigma_i^{-1}) \geq \frac{nP_i^2}{4 \left( \sum_{j=1}^{m} \frac{P_j}{\sqrt{\Sigma_j}} \right)^2 \sqrt{2d|\Sigma_i|}} \left( C_i + \alpha_i \alpha_i^T \right) \geq \frac{nP_i^2}{4 \left( \sum_{j=1}^{m} \frac{P_j}{\sqrt{\Sigma_j}} \right)^2 \sqrt{2d|\Sigma_i|}} C_i$$

The second inequality follows since $\alpha_i \alpha_i^T$ is positive semi-definite and thus, the lower bound of theorem 4.2 is established.

<Q.E.D>
As in the case with the mean estimators, it is possible to show that for well separated components, the information matrix is close to its upper bound. In addition, by inverting the bounds provided by theorem 4.2, the asymptotic covariance matrix of the ML estimate of $\Gamma_i$ can be bounded.

**Lemma 4.2:** The asymptotic covariance matrix of the ML estimate of $\Gamma_i$ based on $n$ unlabeled samples from the mixture is bounded as follows:

$$\frac{1}{nP_i} C_i^{-1} \leq \text{cov}(\hat{\Gamma}_i) = \left[I_u(\Sigma_i^{-1})\right]^{-1} \geq \frac{4}{nP_i^2} \left( \sum_{j=1}^{m} \frac{P_j}{\Sigma_j} \right) \sqrt{2^d |\Sigma_i|} C_i^{-1}$$ (4.26)

### 4.4 Combined Supervised-Unsupervised Learning

In the combined learning process, $n$ unlabeled samples and $n_i$ training samples are used simultaneously to obtain the estimates of the $i^{th}$ components’ parameters.

#### 4.4.1 Mean Estimators

For independent training and unlabeled samples, the Fisher information matrix associated with the mean of the $i^{th}$ component can be written as the sum of the information matrices associated with the training samples and unlabeled samples:

$$I_c(\mu_i) = I_s(\mu_i) + I_u(\mu_i)$$ (4.27)
where, in the left side, subscript "c" stands for "combined". Using the results of sections 4.2.1 and 4.3.1 it is possible to write:

\[
(nP_i + n_i)\Sigma_i^{-1} \geq I_c(\mu_i) \geq \left( \frac{n_P}{2} \sum_{j=1}^{m} \frac{P_j}{\sqrt{\Sigma_j}} \right) + n_i \left( \Sigma_i^{-1} \right)
\]  

(4.28)

and therefore, the asymptotic covariance matrix for the ML estimate of \( \mu_i \) can be bounded by inverting the above bounds:

**Lemma 4.3:** The asymptotic covariance matrix of the ML estimate of \( \mu_i \) based on \( n_i \) training samples and \( n \) unlabeled samples is bounded as follows:

\[
(nP_i + n_i)^{-1} \Sigma_i \leq \text{cov}(\hat{\mu}_i) = [I_c(\mu_i)]^{-1} \leq \left( \frac{n_P}{2} \sum_{j=1}^{m} \frac{P_j}{\sqrt{\Sigma_j}} \right) + n_i \left( \Sigma_i^{-1} \right)
\]  

(4.29)

Once again, it is possible to show that for distant components the covariance matrix is close to the lower bound.

A comparison of the above lemma with the asymptotic covariance matrix of the estimate of \( \mu_i \) in the supervised case, which was presented in equation (4.8) of section 4.2.1, shows that the use of unlabeled samples will always help in reducing the covariance of the estimate.
4.4.2 Inverse Covariance Matrix Estimators

As in the case of mean estimators, since the training samples and unlabeled samples are independent, the information matrices add and one can write:

\[ I_c(\Sigma_i^{-1}) = I_s(\Sigma_i^{-1}) + I_u(\Sigma_i^{-1}) \]  

(4.30)

Using the results of sections 4.2.2 and 4.3.2 it is possible to write:

\[ (nP_i + n_i)C_i \geq I_c(\Sigma_i^{-1}) \geq \left( \frac{nP_i}{4 \sum_{j=1}^{m} \frac{P_j}{\sqrt{\Sigma_i}}} \right) + n_i C_i \]  

(4.31)

and thus, the following lemma is established.

**Lemma 1.4:** The asymptotic covariance matrix of the ML estimate of \( \Gamma_i \) based on \( n_i \) training samples and \( n \) unlabeled samples is bounded as follows:

\[ (nP_i + n_i)^{-1}C_i^{-1} \leq \text{cov}(\hat{\Gamma}_i) = [I_c(\Sigma_i^{-1})]^{-1} \leq \left( \frac{nP_i}{4 \sum_{j=1}^{m} \frac{P_j}{\sqrt{\Sigma_i}}} \right) + n_i C_i^{-1} \]  

(4.32)

Once again, comparing the above lemma with the asymptotic covariance matrix of the estimate of \( \Gamma_i \) in the supervised case, which was presented in equation (4.13) of section 4.2.2, one can conclude that the use of unlabeled samples will **always** enhance the estimate by decreasing its covariance.
4.5. **Discussion and Illustrations**

**Lemmas 4.1** through **4.4** together with equations **4.8** and **4.13** can be used for comparing the supervised learning with unsupervised and combined learning processes. We perform this comparison by studying the effect of the parameters that are involved in the upper and lower bounds. Since the functional forms of the bounds for \( \hat{\mu}_i \) and \( \hat{\Sigma}_i \) are similar (except for a factor of 2 difference), we pay attention only to the results of lemma **4.3**.

We first study the effect of sample size (n) by considering a two-component mixture with equal prior probabilities and equal covariance determinants in 4 dimensional space \((m=2, \ P_1=P_2=0.5, \ |\Sigma_1|=|\Sigma_2|, \ d=4)\). Assuming that 10 training samples are available from the first component, the covariance matrix of \( \hat{\mu}_1 \) in the supervised learning case is \( \frac{1}{10} \Sigma_1 \).

**Lemma 3** provides the following bounds for \( \text{cov}(\hat{\mu}_1) \) in the combined learning case:

\[
\left(\frac{n}{2} + 10\right)^{-1} \Sigma_1 \leq \text{cov}(\hat{\mu}_1) \leq \left(\frac{n}{32} + 10\right)^{-1} \Sigma_1
\]

Figure **4.1** shows the upper and lower bounds as functions of n, the number of unlabeled samples. The vertical axis is the value of the scalar behind \( \Sigma_1 \) in the above bounds. It is clear that for large n, the covariance of \( \hat{\mu}_1 \) is significantly reduced by using combined learning instead of supervised learning.
Next, the effect of prior probabilities is studied. Consider again a mixture of two components in 4-dimensional space with $|\Sigma_1| = |\Sigma_2|$ and 10 training samples from class 1. In supervised learning case, $\text{cov}(\hat{\mu}_1) = \frac{1}{10} \Sigma_1$. Lemma 4.3 bounds the $\text{cov}(\hat{\mu}_1)$ in combined learning case:

$$(nP_1 + 10)^{-1} \Sigma_1 \leq \text{cov}(\hat{\mu}_1) \leq (\frac{nP_1^2}{8} + 10)^{-1} \Sigma_1$$
Figure 4.2 illustrates the bounds as functions of $P_1$ for various values of $n$. Notice that when $P_1=0$, component 1 is not present in the mixture and all of the unlabeled samples belong to the second component. As $P_1$ increases, more and more of the unlabeled samples belong to component 1. When $P_1=1$, the mixture only contains component 1 and therefore all of the unlabeled samples belong to that component.

Figure 4.2: Effect of prior probability in the covariance of the estimators.

$(m=2, d=4, n_1=10, |\Sigma_1|=|\Sigma_2|)$
Besides \( n \) and \( P_i \) that are present in both the upper and lower bounds, the upper bounds contain two more parameters: \( d \) and \( |\Sigma_1| \). These parameters are introduced because of the way the upper bounds are derived. Figure 4.3 shows the effect of \( d \) (dimensionality) for a two component mixture with \( P_1=P_2=0.5 \), \( |\Sigma_1|=|\Sigma_2| \), and 10 training samples from component 1. As the dimensionality increases, more and more unlabeled samples are needed to keep the upper bound constant.

Figure 4.3: Effect of dimensionality in the upper bound of the covariance of the estimators \((m=2, P_1=P_2, n_1=10, |\Sigma_1|=|\Sigma_2|)\).
4.6. Experimental Results

**Experiment 1:**

The first experiment was performed on remote sensing data in order to study the results of lemmas 4.1 through 4.4. From a multi-spectral image taken over Indiana (FLC1), two adjacent fields of corn and soybean were selected. The corn field contained 1832 pixels and the soybean field contained 3304. In other words, the prior probability of the corn was 0.3567 and that of soybean was 0.6431. Using four bands (corresponding to wavelengths 0.40-0.44, 0.52-0.55, 0.62-0.66, 0.90-1.00 ym) the true means and covariance matrices of the classes were calculated:

\[
\begin{align*}
\text{class 1: corn} & \quad \mu_1 = \begin{bmatrix} 85.40 \\ 87.69 \\ 63.18 \\ 81.53 \end{bmatrix}, \\
\Sigma_1 &= \begin{bmatrix} 15.55 & 14.80 & 18.59 & 10.99 \\ 14.80 & 19.89 & 9.93 & -0.06 \\ 18.59 & 9.93 & 26.37 & 10.26 \end{bmatrix} \\
\text{class 2: soybean} & \quad \mu_2 = \begin{bmatrix} 87.39 \\ 90.50 \\ 71.27 \\ 73.38 \end{bmatrix}, \\
\Sigma_2 &= \begin{bmatrix} 5.34 & 3.03 & 5.39 \\ 3.03 & 2.76 & 4.77 & 7.82 \\ 5.39 & 4.77 & 7.82 & 8.89 \end{bmatrix}
\end{align*}
\]

100 unlabeled samples were randomly drawn over the total field of interest (containing corn and soybean). The mean of class 1 was estimated. The experiment was repeated 50 times and the covariance matrix of \( \hat{\mu}_1 \) was calculated using the 50 observations. Using lemma 4.1:

\[
\frac{1}{35.67} \Sigma \leq \text{cov}(\hat{\mu}_1) \leq \frac{1}{0.468} \Sigma
\]
Notice that for two matrices $A$ and $B$, $A \geq B$ if all of the eigenvalues of $A - B$ are non-negative. Table 4.1 shows the eigenvalues corresponding to the above bounds.

Table 4.1: Eigenvalues associated with the bounds for $\text{cov}(\hat{\mu}_1)$

<table>
<thead>
<tr>
<th>Eigenvalues ($\lambda$'s)</th>
<th>$\frac{1}{0.468} \Sigma_1 - \text{cov}(\hat{\mu}_1)$</th>
<th>$\text{cov}(\hat{\mu}_1) - \frac{1}{35.67} \Sigma_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_1$</td>
<td>89.3944</td>
<td>0.3699</td>
</tr>
<tr>
<td>$\lambda_2$</td>
<td>37.8731</td>
<td>0.0466</td>
</tr>
<tr>
<td>$\lambda_3$</td>
<td>10.5145</td>
<td>0.0232</td>
</tr>
<tr>
<td>$\lambda_4$</td>
<td>2.8270</td>
<td>-0.0603</td>
</tr>
</tbody>
</table>

Using the same procedure, the inverse covariance matrix of class 1 was estimated. After 50 trials, the covariance matrix of $\hat{\Sigma}_1$ was calculated. Using lemma 4.2:

$$\frac{1}{35.67} C_1^{-1} \leq \text{cov}(\hat{\mu}_1) \leq \frac{1}{0.234} C_1^{-1}$$

where the entries of matrix $C_1$ were obtained using the fourth order statistics of class 1. Table 4.2 contains the associated eigenvalues.
Table 4.2: Eigenvalues associated with the bounds for $\text{cov}(\hat{\Sigma}_1)$

<table>
<thead>
<tr>
<th>Eigenvalues (λ's)</th>
<th>$\frac{1}{0.234}C_1^{-1} - \text{cov}(\hat{\Sigma}_1)$</th>
<th>$\text{cov}(\hat{\Sigma}_1) - \frac{1}{35.67}C_1^{-1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_1$</td>
<td>3.3440</td>
<td>0.0054</td>
</tr>
<tr>
<td>$\lambda_2$</td>
<td>0.8415</td>
<td>0.0041</td>
</tr>
<tr>
<td>$\lambda_3$</td>
<td>0.2736</td>
<td>0.0017</td>
</tr>
<tr>
<td>$\lambda_4$</td>
<td>0.2016</td>
<td>0.0008</td>
</tr>
<tr>
<td>$\lambda_5$</td>
<td>0.0943</td>
<td>0.0003</td>
</tr>
<tr>
<td>$\lambda_6$</td>
<td>0.0595</td>
<td>0.0002</td>
</tr>
<tr>
<td>$\lambda_7$</td>
<td>0.0271</td>
<td>0.0001</td>
</tr>
<tr>
<td>$\lambda_8$</td>
<td>0.0161</td>
<td>-0.0000</td>
</tr>
<tr>
<td>$\lambda_9$</td>
<td>0.0073</td>
<td>-0.0000</td>
</tr>
<tr>
<td>$\lambda_{10}$</td>
<td>0.0027</td>
<td>-0.0006</td>
</tr>
</tbody>
</table>

The same experiments were repeated for class 2 statistics where by lemmas 4.1 and 4.2:

$$\frac{1}{64.33} \Sigma_2 \leq \text{cov}(\hat{\mu}_2) \leq \frac{1}{6.802} \Sigma_2$$

and

$$\frac{1}{64.33} C_2^{-1} \leq \text{cov}(\hat{\Sigma}_2) \leq \frac{1}{3.401} C_2^{-1}$$

Tables 4.3 and 4.4 show the associated eigenvalues.
Table 4.3: Eigenvalues associated with the bounds for $\text{cov}(\hat{\mu}_2)$

<table>
<thead>
<tr>
<th>Eigenvalues (X’s)</th>
<th>$\frac{1}{6.802} \Sigma_2 - \text{cov}(\hat{\mu}_2)$</th>
<th>$\text{cov}(\hat{\mu}_2) - \frac{1}{64.33} \Sigma_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>h1</td>
<td>1.7767</td>
<td>0.0714</td>
</tr>
<tr>
<td>h2</td>
<td>1.2088</td>
<td>0.0066</td>
</tr>
<tr>
<td>$\lambda_3$</td>
<td>0.4421</td>
<td>-0.0052</td>
</tr>
<tr>
<td>$\lambda_4$</td>
<td>0.1649</td>
<td>-0.0491</td>
</tr>
</tbody>
</table>

Table 4.4: Eigenvalues associated with the bounds for $\text{cov}(\hat{\Gamma}_2)$

<table>
<thead>
<tr>
<th>Eigenvalues (X’s)</th>
<th>$\frac{1}{3.401} C_2^{-1} - \text{cov}(\hat{\Gamma}_2)$</th>
<th>$\text{cov}(\hat{\Gamma}_2) - \frac{1}{64.33} C_2^{-1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_1$</td>
<td>0.2463</td>
<td>0.0057</td>
</tr>
<tr>
<td>$\lambda_2$</td>
<td>0.0785</td>
<td>0.0034</td>
</tr>
<tr>
<td>h3</td>
<td>0.0364</td>
<td>0.0024</td>
</tr>
<tr>
<td>h4</td>
<td>0.0247</td>
<td>0.0012</td>
</tr>
<tr>
<td>h5</td>
<td>0.0208</td>
<td>0.0004</td>
</tr>
<tr>
<td>h6</td>
<td>0.0088</td>
<td>0.0002</td>
</tr>
<tr>
<td>$\lambda_7$</td>
<td>0.0078</td>
<td>0.0000</td>
</tr>
<tr>
<td>$\lambda_8$</td>
<td>0.0049</td>
<td>-0.0000</td>
</tr>
<tr>
<td>$\lambda_9$</td>
<td>0.0022</td>
<td>-0.0000</td>
</tr>
<tr>
<td>$\lambda_{10}$</td>
<td>0.0018</td>
<td>-0.0005</td>
</tr>
</tbody>
</table>
Tables 4.1 through 4.4 demonstrate that in all cases the covariance matrices were very close to their lower bounds. In fact, some of the eigenvalues associated with the lower bounds were small negative numbers. Besides the possible non-normality of the data and numerical problems in evaluating the eigenvalues, the fact that the Bhattacharyya distance between the two classes is 1.54 (which indicates a Bayes error of less than 0.1) indicates that the classes are relatively distant and therefore the covariance matrices were expected to be close to their lower bounds.

**Experiment 2**

The second experiment was performed to study the effect of the distance between the components. From a mixture of two normal densities with equal prior probabilities and the following statistics 1000 unlabeled samples were drawn randomly.

\[
\mu_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \Sigma_1 = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}
\]

\[
\mu_2 = \begin{bmatrix} a \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}
\]

where a (the distance along the first dimension) was varied from 0 to 10 in integer values. The mean of class 1 was estimated. For each value of a, 400 trials were performed and the covariance matrices of \( \hat{\mu}_1 \) were obtained. By lemma 4.1:

\[
\frac{1}{500} \Sigma_1 \leq \text{cov}(\hat{\mu}_1) \leq \frac{1}{31.25} \Sigma_1
\]
Notice that for any two matrices $A$ and $B$, $A \geq B$ implies that $\lambda_i \geq \delta_i$, where $\lambda_i$ and $\delta_i$ are the eigenvalues of $A$ and $B$ respectively [41]. Therefore, since the eigenvalues of $\Sigma_i$ are all equal to 5, the eigenvalues of the covariance matrix of $\hat{\mu}_i$ (denoted by $\lambda_i$) must be bounded as follows for all values of $\alpha$:

$$0.01 \leq \lambda_i \leq 0.16, \quad i=1,2,3,4$$

Figure 4.4 shows the eigenvalues, $\lambda_i$, as functions of $\alpha$, the distance between the means of the two classes. The lower bound is at 0.01 and the upper bound, which is not shown, is at 0.16. It is clear that as the distance between the two components increases, the eigenvalues approach their lower bound. In other words, the covariance matrix of the estimate converges to its lower bound.
4.7. Conclusions and Remarks

When a large number of unlabeled samples is available, supervised learning can be enhanced by incorporating these samples into the estimation process. In this chapter, upper and lower bounds for the asymptotic covariance matrices under three different learning processes were derived and verified by several experiments. The bounds in this chapter are derived independently for each parameter. For the more general case of joint unknown parameters the mathematics is more complicated.
CHAPTER 5: SEMI-PARAMETRIC COMBINED SUPERVISED-UNSUPERVISED LEARNING

So far in this thesis we have only been concerned with the case of Gaussian classes. The assumption of normality can be inappropriate in practice. For example in remote sensing, it is a common practice to consider several "spectral subclasses" within each " informational class" or ground cover type. Each of such spectral subclasses is then considered to be Gaussian and classification is performed with respect to the spectral subclasses [42]. It can be argued that differences in soil type, ground moisture, planting time, etc., cause the apparent multi-modality of the probability density functions of the informational classes, which in turn necessitates the partitioning of the informational classes into the spectral subclasses. The equations 2.3, 2.4 and 2.5 that were briefly discussed in Chapter 2 approximate the ML estimates of the class parameters if the classes are Gaussian. In this chapter, we generalize these formulas by obtaining the necessary equations for the ML estimates when the classes have multiple Gaussian components. This procedure is called semi-parametric learning to distinguish it from the parametric learning where the Gaussian assumption is used, and from nonparametric learning where no particular form is assumed.
5 Combined Learning

5.1 Previous Work on Normal Mixture Density Identification

Estimation of the parameters of a normal mixture density has been studied extensively over the past three decades. References [22,43] provide excellent discussions on this topic. In [5] the following four types of data have been introduced in the context of mixture density identification:

Type 1: Unlabeled samples: statistically independent unlabeled observations drawn from the mixture.

Type 2: Completely labeled training samples: statistically independent labeled observations drawn separately from individual components of the mixture.

Type 3: Statistically independent observations drawn from the mixture and subsequently labeled.

Type 4: Statistically independent observations drawn from the mixture, which have fallen into a particular region of the feature space, and subsequently labeled.

For the most part, the attention has been on identifying a normal mixture density from type one samples (unlabeled samples). However, combinations of different kinds of samples can be considered in the identification process. In chapter 1, we assumed that classes were Gaussian and referred to the case where both type 1 and type 2 data are available as combined supervised-unsupervised learning. If instead of type 2 samples, type 3 samples are available, then the equation 2.3 that estimates the prior probabilities of the components needs to be changed to accommodate the labeled observations [43]. Throughout this thesis, however, we assume that training samples are of type 2, which means they do not contain any information regarding the prior probabilities.
The method of moments for identification of univariate normal mixture was proposed by Karl Pearson [23]. The multivariate case was studied later by other researchers such as Fulcunaga and Flick [24]. The Bayes' method was studied in [25], and the maximum likelihood method was studied in [26,27,28]. Other approximate techniques have also been introduced in the literature [29].

The ML method has been given a significant amount of attention recently because of its relative simplicity. The ML method is based on obtaining the parameters that maximize the likelihood of the observed data. If only unlabeled data are available, the likelihood function can contain "singular points" at which the likelihood is infinite. These singular points arise because by placing the mean of one of the components on an unlabeled sample and driving its covariance to zero, the likelihood can be increased indefinitely. This problem caused some controversy over using the ML method for normal mixture identification. But, in practice the attention has been towards finding the local maxima in the likelihood function. The singular point problem can be avoided by various methods including the assumption of similar covariance matrices for all the components in the mixture [28], using some training samples from each component [31], or putting constraints on the maximization process [44]. In any case, when the likelihood function is written, the usual approach is to first include a Lagrangian multiplier for the constraint that the prior probabilities add up to one, and then take the derivative of the likelihood function with respect to the parameter set and equate it to zero. Unfortunately, the resulting equations cannot be solved analytically for the parameters. Therefore, numerical techniques need to be used. Wolfe [26], Day [27] and John [28] have proposed the iterative use of the resulting equations, but the sequence of the iteration; and convergence properties for this method were not clear until the introduction of the Expectation-Maximization (EM) [30] method. The EM method was proposed for solving ML
Combined Learning

problems involving missing data, of which the mixture identification problem is an example. The EM equations for obtaining the ML estimates of the parameters were derived in [5]. Various properties of the EM algorithm were studied in [5], and its convergence was studied in [45]. The EM algorithm is an iterative process in which the value of the likelihood function increases at each step. The convergence is only guaranteed to a local maximum, and therefore the process usually has to be repeated from various initial points. However, if training samples are available in addition to unlabeled samples, they can be used for obtaining reasonable starting points. Robust EM equations have been derived in [46] for dealing with the case where outliers may be present.

The form of the EM equations usually resembles the regular ML estimates with the distinction that to each unlabeled sample a set of weights is attached that shows the "degree of membership" of that sample to each component of the mixture. These weights are equal to the posterior probability of each component given the unlabeled sample and the current values of the parameters. Based on this, in [47] a nonparametric approach to mixture density identification is proposed that uses both training and unlabeled samples (nonparametric combined supervised-unsupervised learning). First training samples are used to obtain "weights" for the unlabeled samples, and then these weight are used with the unlabeled samples to obtain better estimates of the component densities of the mixture.

In this chapter, we consider a semi-parametric approach, in which both training and unlabeled samples are used. The pdf of each class is modeled by a mixture of Gaussian components. If only one component is considered per class, then our method will yield equations similar to 2.3, 2.4 and 2.5. However, since in practice the single Gaussian assumption may be inappropriate, we assume that more then one Gaussian component
can exist in a class. Since any smooth density function can be approximated with a mixture of Gaussian curves [48], our approach is justified. In addition to the unlabeled samples that are available from the mixture of the classes, training samples are also available from each class. However, classes themselves are mixture densities and therefore training samples are only known to belong to a class without any reference to the particular component within that class. This fact distinguishes our work from the previous studies. We use the EM method to study this semi-parametric case. In the next section, a brief overview of the EM method is given.

5.2 The Expectation Maximization (EM) Method

The Expectation Maximization (EM) method as described in [30] is a technique for obtaining maximum likelihood estimates in the presence of incomplete data. The mixture identification problem can be regarded as such a problem by considering each unlabeled sample as an incomplete observation from which a label indicating its component of origin is missing [5]. The completely labeled observations are denoted by \( Y = (y_1, \ldots, y_n) \) with the parametric pdf \( f(Y|\theta) \), and the unlabeled observations by \( X = (x_1, \ldots, x_r) \) with the pdf \( g(X|\theta) \). Therefore, let \( y_k = (x_k,i_k) \), where \( i_k \) is an integer between 1 and \( m \) (the number of components in the mixture) indicating the origin of sample \( y_k \). The EM algorithm maximizes the log-likelihood of the unlabeled data, \( L(\theta) = \log g(X|\theta) \), by iterating the following two steps:

1. **Expectation Step:** Determine \( Q(\theta|\theta^0) = E(\log f(Y|\theta) \mid X,\theta^0) \)
2. **Maximization Step:** Choose \( \theta^+ = \arg \max Q(\theta|\theta^0) \)
Thrt: next and current values of the parameters are denoted by the superscripts "$^+$" and "$^c$" respectively. The importance of the EM algorithm lies in the fact that at each step of the iteration $L(\theta)$ increases [30]:

$$L(\theta^+) \geq L(\theta^c)$$

### 5.3 EM Algorithm for Semi-parametric Mixture Density Identification

We assume that there are $J$ classes in the feature space denoted by $S_1, \ldots, S_J$. Each class can have several Gaussian components. Let $m$ denote the total number of the Gaussian components. We write $i \in S_j$ to indicate that component $i$ belongs to class $S_j$. The pdf of the feature space can then be written as a mixture of $m$ Gaussian components where the set of components can be partitioned into $J$ classes:

$$f(x|\theta) = \sum_{i=1}^{m} \alpha_i f_i(x|\phi_i)$$  \hspace{1cm} (5.1)

where $$\phi_i = (\mu_i, \Sigma_i), \; \theta = (\alpha_1, \ldots, \alpha_m, \mu_1, \ldots, \mu_m, \Sigma_1, \ldots, \Sigma_m).$$

In this chapter we denote the prior probabilities of the components by $\alpha_i$. From each class $S_j$, $N_j$ training samples are assumed to be available. We denote these samples by $z_{jk}$ where $j=1, \ldots, J$ indicates the class of origin and $k=1, \ldots, N_j$ is the index of each particular sample. The difference between these training samples and the Type 2 samples that were discussed in section 5.1, is that the training samples here are known to come from a particular class without any reference to the exact component of that class. In addition to the training samples, $N$ unlabeled samples, denoted by $x_k, k=1, \ldots, N$, are also assumed to
be available from the mixture. For each training sample $z_{jk}$, the pdf can be written in the following form:

$$g(z_{jk} | \theta) = \sum_{i \in T} \sum_{l \in S_j} \alpha_i f_l(z_{jk} | \phi_l)$$

The pdf of each unlabeled sample $x_k$ is $f(x_k | \theta)$. Therefore, under the assumption that the samples are all statistically independent, the log likelihood to be maximized is:

$$L(\theta) = \sum_{k=1}^{N} \log f(x_k | \theta) + \sum_{j=1}^{J} \sum_{k=1}^{N_j} \log \left( \sum_{i \in T} \sum_{l \in S_j} \alpha_i f_l(z_{jk} | \phi_l) \right)$$

where the maximization is subject to the following constraints:

$$\alpha_i > 0 \quad \text{and} \quad \sum_{i=1}^{m} \alpha_i = 1$$

The first term in this log likelihood function is the likelihood of the unlabeled samples with respect to the mixture density, and the second term is the likelihood of the training samples with respect to their corresponding classes of origin. We use the EM technique to obtain the maximum likelihood estimates.

**Expectation Steu**

Since the unlabeled and training samples are independent, $Q(\theta | \theta^{(t)})$ can be written as the sum of two terms corresponding to the two types of data:
where the first term is:

\[ Q_1(\Theta|\Theta^c) = E(\log f(Y|\Theta) | X, \Theta^c) \]

\( Y = (y_1, \ldots, y_N) \) is the completely labeled sample variable with pdf \( f(Y|\Theta) \) and \( X = (x_1, \ldots, x_N) \) is the unlabeled sample variable, and \( y_k = (x_k, i_k) \), where \( i_k \) is an integer between 1 and \( m \) (the number of components in the mixture) identifying the origin of sample \( y_k \). In [5], \( Q_1(\Theta|\Theta^c) \) has been obtained as:

\[ Q_1(\Theta|\Theta^c) = \sum_{i=1}^{m} \sum_{k=1}^{N} P^c(\cdot | x_k) \log \alpha_i + \sum_{i=1}^{m} \sum_{k=1}^{N} P^c(\cdot | x_k) \log f_i(x_k | \phi_i) \]  

(5.4)

where \( P^c(\cdot | .) \) is the posterior probability given the current parameter values:

\[ P^c(\cdot | x_k) = \frac{\alpha^c f_1(x_k | \mu^c, \Sigma^c)}{f(x_k | \Theta^c)} \]  

(5.5)

The second term in the right side of equation (5.3) can itself be written as the sum of \( J \) terms, each corresponding to one class:

\[ Q_2(\Theta|\Theta^c) = \sum_{j=1}^{J} Q^j_2(\Theta|\Theta^c) \]  

(5.6)

For each class \( S_j \), \( Q^j_2 \) can be written in the following from:
where $Z_j = (z_{j1}, \ldots, z_{jN_j})$ is the training sample variable, and $Y_j = (y_{j1}, \ldots, y_{jN_j})$ is the completely labeled sample variable. For each $y_{jk}$, we have $y_{jk} = (z_{jk}, i_{jk})$, where $i_{jk}$ is an indicator representing the index of the component which originated sample $y_{jk}$. Notice that this indicator can only assume the indices of those components which belong to class $S_j$. Also $f(.)$ is the pdf of $Y_j$:

$$f(Y_j | \theta) = \prod_{k=1}^{N_j} \left( \frac{\alpha_{i_{jk}} f_{i_{jk}}(z_{jk} | \phi_{i_{jk}})}{\sum_{i \in S_j} \alpha_i f_i(z_{jk} | \phi_i)} \right)$$

Denoting the pdf of $Z_j$ by $g(.)$, we can write:

$$g(Z_j | \theta) = \prod_{k=1}^{N_j} \frac{1}{\sum_{i \in S_j} \alpha_i f_i(z_{jk} | \phi_i)}$$

The density of $Y_j$ conditioned on $Z_j$ is $K(Y_j | Z_j, \theta) = f(Y_j | \theta) / g(Z_j | \theta)$. Therefore:

$$K(Y_j | Z_j, \theta) = \prod_{k=1}^{N_j} P_j(i_{jk} | z_{jk})$$

where $P_j(.)$ is the posterior probability (with respect to class $S_j$):

$$P_j(i_{jk} | z_{jk}) = \frac{\alpha_{i_{jk}} f_{i_{jk}}(z_{jk} | \phi_{i_{jk}})}{\sum_{i \in S_j} \alpha_i f_i(z_{jk} | \phi_i)}$$
Using (5.8) and (5.10) in equation (5.7), we get:

\[
Q_i(\theta; \theta^c) = \sum_{i \in S_j} \sum_{j \in S_j} \cdots \sum_{i \in S_j} \sum_{j \in S_j} \log \left( \frac{\alpha_{i,j,k} f_{i,j}(z_{j,k} | \phi_{i,j,k})}{\sum_{t \in S_j} \alpha_t} \right) \prod_{k=1}^{N_j} P_j^c(i,j,k | z_{j,k}) \tag{5.12}
\]

\[
= \sum_{i \in S_j} \sum_{j \in S_j} \log \left( \frac{\alpha_{i,j,k} f_{i,j}(z_{j,k} | \phi_{i})}{\sum_{t \in S_j} \alpha_t} \right) P_j^c(i,j,k | z_{j,k}) \tag{5.13}
\]

**Appendix C** contains the derivation of equation (5.13) from (5.12). **Substituting** (5.13) in (5.6) and the result in (5.3), we get:

\[
Q(\theta; \theta^c) = \sum_{i=1}^{m} \sum_{k=1}^{N} P_i^c(i|x_k) \log \alpha_i + \sum_{i=1}^{m} \sum_{k=1}^{N} P_i^c(i|x_k) \log f_i(x_k | \phi_i)
\]

\[
+ \sum_{j=1}^{J} \left( \sum_{i \in S_j} \sum_{k=1}^{N_j} P_j^c(i|z_{j,k}) \right) \log \alpha_i - \sum_{i \in S_j} \sum_{k=1}^{N_j} P_j^c(i|z_{j,k}) \log \left( \sum_{t \in S_j} \alpha_t \right)
\]

\[
+ \sum_{i \in S_j} \sum_{k=1}^{N_j} P_j^c(i|z_{j,k}) \log \left( f_i(z_{j,k} | \phi_i) \right) \tag{5.14}
\]

**Maximization Step**

By taking the derivatives of equation (5.14) with respect to the parameters and setting these derivatives equal to zero, the following iterative equations are obtained (Appendix D contains the details of the maximization step):
Combined Learning

Equations (5.15), (5.16) and (5.17) provide the EM formulas for obtaining the ML estimates of the parameters of the mixture density in the presence of training samples from multi-component classes. Notice that in all these equations component $i$ is assumed to be a member of class $S_j$, and $P^C(i.l.)$ and $P^C(j.l.)$ are the current values of the posterior probabilities that were defined in equations (5.5) and (5.11).

5.4 Constrained EM Algorithm

When any additional information concerning the values of the parameters is available, it is desirable to use it in order to reduce the possibility of converging to a singular point or wrong local maximum. Such information can be used either as prior probabilities for the
parameters, which would lead to a Bayesian estimation process, or as constraints based on which the maximization must be performed. In [44], for example, constraints on the prior probabilities and variances of the components are used, in a univariate case, to avoid converging to singular points. Here, we use a similar technique in constraining the estimation process.

A particular situation that has motivated the development of this subsection is in remote sensing where often there are more classes in the scene than the user is aware of. Usually, some training samples from the dominant classes are available that can be used in the estimation process and in obtaining starting points for the iterative EM equations. However, some smaller classes such as roads and farmsteads in agricultural scenes are often present too. There usually exist no good training samples from such classes. These classes need to be accounted for in the estimation process by considering an "unknown class" in the mixture density that represents the feature space; otherwise they can cause the estimation process to converge to a wrong local maximum. Sometimes the user has some idea of the maximum percentage of pixels in the scene that belong to such an unknown class. Such information can be used in the maximization step of the EM algorithm. Without loss of generality, we assume that class $S_1$ represents the "unknowns" class. Let us assume that the prior probability of this class can be bounded as follows:

$$\sum_{i \in S_1} \alpha_i < \varepsilon$$  \hspace{1cm} (5.18)

If the EM algorithm is used for estimating the parameter values, the above constraint should be imposed in the maximization part of the process. By considering the Kuhn-Tucker conditions [49], it can be shown that the constrained EM algorithm is as follows:
1) Perform one iteration of the regular EM algorithm.
2) Check the constraint (5.18). If it is satisfied go to step 1, otherwise go to step 3.
3) For all components \( i \in S_1 \) update the prior probabilities as follows:

\[
\alpha^+_i \leftarrow \frac{\alpha^+_i}{\sum_{r \in S_1} \alpha^+_r} \cdot \varepsilon
\]

For all components \( i \notin S_1 \) update the prior probabilities as follows:

\[
\alpha^+_i \leftarrow \frac{\alpha^+_i}{\sum_{r \in S_1} \alpha^+_r} \cdot (1 - \varepsilon)
\]

Go to step 1.

In addition, if it is desirable to constrain the prior probabilities of the informational classes to be larger than pre-specified numbers, the method described in [44] may be adopted.

### 5.5 Modified EM Algorithm

At each iteration of the EM algorithm (equations 5.15, 5.16, 5.17) the statistics of each component are updated based on both the training samples and unlabeled samples. When the number of unlabeled samples is much larger than the training samples, the effect of the training samples is small. In such case, the procedure is almost completely unsupervised. This fact might cause some difficulties especially if the unlabeled samples contain outliers or there are classes that are very overlapping. In such cases, the procedure
might not be robust with respect to the number of unlabeled samples, and sometimes the newly updated statistics are too different from the ones suggested by the training samples alone. Often, one of the classes dominates the other close classes. One way to approach this problem, as suggested in [50], is to include a "reliability" factor for the training data in the EM algorithm. This reliability factor essentially lets the user determine the relative importance of the training data in obtaining the next values of the parameters. This is done by counting the training samples more (or less, if necessary) than once, and thereby increasing (or decreasing) the effect of the training data. The work in [50] was presented for the case of Gaussian classes; here we extend it to the general case of multi-component classes.

Let's assume that at the nth iteration of the EM algorithm, the ratio of the total fraction of the unlabeled samples (in terms of the posterior probabilities) that belong to class \( S_j \) to the number of training samples from \( S_j \) is \( \beta_j \):

\[
\sum_{i \in S_j} \sum_{k=1}^{N} P^c(i|x_k) = \beta_j N_j
\]

Now, if one wants to modify the EM algorithm such that the training and unlabeled samples have equal effect in updating the statistics of class \( S_j \), one needs to change the number of training samples of class \( S_j \) from \( N_j \) to \( \beta_j N_j \). In fact, one could change the number of training samples to \( \lambda_j \beta_j N_j \), in order to have \( \lambda_j \) times more effect from the training samples than from the unlabeled samples. To do this, one could simply count the training samples more (or less) than once. Usually, the same \( \lambda \) is selected for all classes. The modified EM equations are therefore the following (for \( i \in S_j \)):
In each iteration of the EM algorithm, for each sample, the quadratic Mahalanobis distances to all the components need to be computed. Since the computation of a quadratic term is $O(d^2)$, where $d$ is the dimension of the space, and there are $m$ components, for each sample $O(md^2)$ operations need to be performed. Disregarding the computational cost of performing the exponential function, and assuming that a total of $N$ samples (both unlabeled and training) is available, the computational cost of a single iteration of the EM algorithm is $O(mNd^2)$.

Several possibilities exist for dealing with this complexity using parallelism. It is possible to eliminate the effect of $m$, the number of components, by dedicating a processor to the computation regarding each different component and therefore reducing this complexity.
Combined Learning

to $O(Nd^2)$. Alternatively, the quadratic computation can be performed in parallel. But, if a massively parallel computer system is available the data samples can be distributed among the processing elements (PEs) to achieve an $O(md^2(N \mod nproc)) \approx O(md^2)$ complexity, where $nproc$ is the number of PEs. In practice, since usually $N \gg md^2$, data parallel processing appears to be the most rewarding method. Because the operations performed on each sample are identical and can be done separately, a SIMD (Single Instruction Multiple Data) system is best suited for this purpose. The information about the current values of the parameters is kept at the control unit of the system. The samples $\mathbf{x}_i$'s (unlabeled), and $\mathbf{z}_{ij}$'s (training) are evenly distributed among the PEs so that each PE obtains at most $\lceil N \mod nproc \rceil$ unlabeled samples and $\lfloor N_j \mod nproc \rfloor$ training samples from each class $S_j$. Figure 5.1 illustrates the organization of the system.

The controller unit passes the current parameters of the model to all the PEs in parallel. Each PE uses the received information to compute the following quantities using the sample points in its local memory: $P^c_\mathbf{C}(i|\mathbf{x}_k)$, $P^c_\mathbf{j}(i|z_{jk})$, $P^c_\mathbf{C}(i|\mathbf{x}_k)\mathbf{x}_k$, $P^c_\mathbf{j}(i|z_{jk})z_{jk}$. These local values are then passed back to the controller unit where they are used to compute $\alpha_i^+$, and $\mu_i^+$. The controller unit then passes $\mu_i^+$ back to all the PEs in order for them to compute the local quantities $P^c_\mathbf{C}(i|\mathbf{x}_k)(\mathbf{x}_k - \mu_i^+)(\mathbf{x}_k - \mu_i^+)^T$ and $P^c_\mathbf{j}(i|z_{jk})(z_{jk} - \mu_i^+)(z_{jk} - \mu_i^+)^T$ and return them to it. The controller unit then uses the information to compute $\Sigma_i^+$. This process continues until convergence is achieved.
A MasPar MP1 computer system with 16,384 PEs was used for the implementation of the parallel EM algorithm. Some benchmark results on the cpu time required for ten iterations of the EM algorithm are listed in Table 5.1. The numbers in this table are averages over 5 independent trials. Further improvement on the parallel version of the algorithm is probably possible. However, from Table 5, it is evident that parallel computing can significantly increase the speed of the mixture density identification process.
Table 5.1: cpu times required per 10 iterations of the EM algorithm on a MasPar MP-1, Titan Ardent P3, and a Sun 4 Sparc. Numbers in parentheses indicate the ratio between the serial machine cpu time and that of the MasPar. Programs are written in C for both serial machines and in MPL-C for the MasPar. If an optimizing compiler is used for the Sparc a reduction of about 35-40% in cpu time is observed.

<table>
<thead>
<tr>
<th>dimension</th>
<th># components</th>
<th># samples</th>
<th>cpu time (sec) MasPar</th>
<th>cpu time (sec) Ardent P3</th>
<th>cpu time (sec) Sun 4 Sparc</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>1000</td>
<td>0.90</td>
<td>1.30 (1.44)</td>
<td>4.17 (4.63)</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>3000</td>
<td>0.90</td>
<td>4.46 (4.96)</td>
<td>12.59 (13.99)</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>6000</td>
<td>0.90</td>
<td>9.88 (10.98)</td>
<td>25.16 (27.96)</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>16384</td>
<td>0.89</td>
<td>27.69 (31.11)</td>
<td>69.67 (78.28)</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1000</td>
<td>1.31</td>
<td>1.79 (1.37)</td>
<td>6.39 (4.88)</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3000</td>
<td>1.30</td>
<td>6.38 (4.91)</td>
<td>18.4 (14.15)</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>6000</td>
<td>1.31</td>
<td>14.54 (11.1)</td>
<td>37.08 (28.31)</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>16384</td>
<td>1.31</td>
<td>42.03 (32.08)</td>
<td>103.52 (79.02)</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1000</td>
<td>1.75</td>
<td>2.32 (1.33)</td>
<td>8.56 (4.89)</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3000</td>
<td>1.72</td>
<td>8.47 (4.92)</td>
<td>24.47 (14.22)</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>6000</td>
<td>1.72</td>
<td>18.88 (10.98)</td>
<td>50.15 (29.16)</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>16384</td>
<td>1.72</td>
<td>53.48 (31.09)</td>
<td>138.37 (80.45)</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>1000</td>
<td>2.77</td>
<td>2.55 (0.92)</td>
<td>9.40 (3.39)</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>3000</td>
<td>2.77</td>
<td>9.51 (3.43)</td>
<td>28.72 (10.37)</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>6000</td>
<td>2.77</td>
<td>22.06 (7.96)</td>
<td>58.39 (21.08)</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>16384</td>
<td>2.77</td>
<td>58.94 (21.28)</td>
<td>160.43 (57.92)</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1000</td>
<td>4.12</td>
<td>3.87 (0.94)</td>
<td>13.93 (3.38)</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>3000</td>
<td>4.13</td>
<td>13.94 (3.38)</td>
<td>43.11 (10.44)</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>6000</td>
<td>4.13</td>
<td>32.95 (7.97)</td>
<td>87.20 (21.11)</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>16384</td>
<td>4.13</td>
<td>89.68 (21.71)</td>
<td>243.20 (58.89)</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>1000</td>
<td>5.47</td>
<td>5.05 (0.92)</td>
<td>18.18 (3.32)</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>3000</td>
<td>5.48</td>
<td>18.82 (3.43)</td>
<td>59.76 (10.91)</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>6000</td>
<td>5.48</td>
<td>44.61 (8.14)</td>
<td>117.03 (21.36)</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>16384</td>
<td>5.48</td>
<td>118.22 (21.57)</td>
<td>321.28 (58.63)</td>
</tr>
</tbody>
</table>
5.7 Concluding Remarks

In this chapter, the equations for obtaining the ML estimates of the classes were obtained under a semi-parametric assumption for the class-conditional probability density functions. Instead of assuming Gaussian classes, we allow classes to contain multiple Gaussian components. By varying the number of components in a class, different models can be obtained and the parameters of these models can be estimated by the equations 5.15, 5.16, and 5.17. Such models can then be compared based on a suitable "goodness" criterion. In the next chapter, some methods for comparing pdf models are studied. The equations derived in this chapter will be used for estimating the parameters of the models under study.
CHAPTER 6: DENSITY MODEL SELECTION FOR CLASSIFICATION

In chapter 5, a semi-parametric method for estimating the class parameters from training and unlabeled samples was proposed. In order to utilize this method, first a model must be selected for the probability density function of the feature space. By this, we mean that the number of classes, as well as the number of components within each class, must be known. Selection of a suitable model, or estimation of the number of components in a mixture density, is a difficult problem. Often, one needs to choose the best model among a set of candidate models by consulting a suitable criterion of goodness.

In a pattern recognition problem, the ultimate test of suitability is the error rate that is achieved when a particular model is used. Therefore, the best way to compare pdf models in a pattern recognition problem is to estimate the error rates obtained by using each candidate model.

If independent test samples of known origin are available, they can be used for estimating the error rates; otherwise cross validation methods, such as the leave-one-out method, usually provide reasonable estimates of the error rate. In practice, the number of labeled observations may be too small to be divided into design and test sets, and the leave-one-out method can be very tedious and time consuming. Therefore, it is desirable to be able to quantify the goodness of a model by simpler and faster methods. In this chapter,
several methods for evaluating model validity are studied. A search algorithm is then used for generating a set of candidate models systematically, and the "best" model is subsequently selected.

6.1 Introduction and Previous Work

In order to test the validity of a chosen model, goodness of fit tests may be used. However, the standard tests, such as the chi-square and the Kolmogorov-Smirnov tests, are not very powerful in high dimensions and when the number of samples is limited. Alternatively, the normality of each component of the model can be checked \cite{51} by tests of normality. However, no conclusive test of normality exists for multivariate data. The generalized likelihood ratio test has been used for assessing the hypothesis that the data is from a mixture of \( m_1 \) components versus an alternative that it is from a mixture of \( m_2 \) components \cite{52}. However, in this case due to the break down of a regularity condition, the test statistic does not have its usual chi-square distribution \cite{22}. Adjustments to the test statistic have been proposed \cite{53}, but with limited success \cite{22}. Other tests have also been proposed for this purpose \cite{40}, but their characteristics have not been studied.

In general, the likelihood of the observed data is an increasing function of the complexity of the assumed model. In other words, by assuming more and more components in the model, the likelihood of the observed data may be incremented indefinitely. For example, in Figure 6.1, 6020 four-dimensional soybean samples are used in estimating the parameters of a normal mixture density with various numbers of components. It can be seen that as the number of components increases in the model, the likelihood of the observed data increases too. This, however, does not necessarily mean that the more complex models are better choices, since by over-complicating the model it may begin to
overfit to the data and loose its generalization capability. It has been suggested that the "knee" of the curve should be considered the point at which the model best fits the data [52].

![Graph showing log-likelihood versus number of components](image)

**Figure 6.1**: Log-likelihood versus the number of normal components in the pdf model for 6020 four-dimensional soybean samples

Alternatively, penalized likelihood methods have been suggested for model selection [54]. The idea is to penalize the more complicated models by a term that describes their
6 Density Model Selection

complexity. The AIC criterion [55] and the Minimum Description Length (MDL) [56] are examples of penalized likelihood criteria. The AIC criterion was proposed by Akike [55] and has been shown to be inconsistent in a variety of applications [57]. The MDL was proposed by Rissanen [56]. The dominant term in the MDL criterion is similar to another criterion proposed by Schwarz (BIC) [58], whose consistency was studied in [59]. The MDL criterion has been used widely in various applications such as density estimation [60], tree classifier design [61], and segmentation of images [62]. We will use the MDL criterion later in this chapter for obtaining suitable pdf models.

Other techniques for model selection include methods that are based on searching for modes in the observed density of the data [29,63]. Such methods are usually sensitive to the way that the convexity of the densities is estimated. In addition, the number of modes is not necessarily equal to the number of components in a mixture density. In fact, for a univariate case with two normal components with equal variance, bimodality occurs only if \(|\mu_1 - \mu_2|/\sigma > 2|^{52}\).

Several heuristic methods for assessing the validity of models, in the context of clustering, have been proposed in the past three decades. Most of these methods are based on properties such as the compactness of the clusters, their average separation, overlap and so on. Some of these methods are described in [64]. One particular example is the use of estimates of allocation rates [65] for assessing how "fuzzy" the partitioning of the data is under a hypothesized model. Later in this chapter we will use a variation of this method for selecting suitable models.
6.2 Minimum Description Length (MDL)

Rissanen proposed the MDL method for model selection [56]. The MDL is an example of penalized likelihood criteria, and is based on the principle of parsimony: the simplest model that adequately explains the observations is the best to choose. Relying on the theory of communications and coding, Rissanen proposed that the "best" model is the one that yields the shortest code length for the data. The length of the shortest code for the data can be obtained, from Shannon's theorem, as a function of the probability model that underlies the generation of data. Different probability models provide different code lengths. Therefore, models can be compared based on the code that they generate for the data. However, as one increases the complexity of the probability models, higher likelihood for the data can be obtained, and therefore shorter codes can be constructed.

The MDL principle penalizes the more complex probability models. The argument used in the MDL method is that a probability model that is used for encoding the data must be known at the time of decoding in order to create the necessary "code book". Therefore, in a hypothetical communications system, the parameters that describe the probability model need to be transmitted prior to the transmission of the encoded data. Hence, the total code length is the summation of the code length for the data that is obtained using a probability model, plus the code for the selected probability model. In the absence of any prior probability for the parameters of the model, Rissanen proposed a universal prior based on which the code length for the model can be found. For example, assume that data is generated from a normal mixture density with m components, such as the one in equation (5.1). The Minimum Description Length for this case can be written in the following form:
\[ L(X, \text{Model}) = -\sum_{k=1}^{N} \log f(x_k | \theta) + \frac{1}{2} k \log N \]

where the first term is the length of the shortest prefix code for the \( N \) independent unlabeled samples, and the second term is the length of the code for transmitting the model, in which \( k \) is the number of the independent parameters in the model. The second term is obtained by using Rissanen's universal prior [56].

Example 6.1:
In the Flight Line C1 (FLC1) data set, eight ground truth classes were chosen. Training fields for each of the classes were selected. Bands 4, 6, 9 and 12 were used for analysis. The pdf of each class was modeled as a normal mixture density and the number of components in each class was incremented until a peak was observed in the MDL values. A total of 22 components was found in this way within the eight informational classes. Table 6.1 contains the number of training samples and components per class. After the class conditional pdf models were selected, the statistics of the classes were updated by using 23458 unlabeled samples (every third sample in every third row) drawn from the image, together with the training samples, via the EM equations of chapter 5. The data set was then classified according to the newly obtained statistics. The total classification accuracy was 89.76%.
In the above example, the training samples from each class were used alone for estimating the number of components within that class. It is desirable that the unlabeled data be part of this process as well. Let us assume again that the density of the feature space is modeled by the equation (5.1). Let us assume that the set of the m components is partitioned into J classes, each of which might have several components. Then assuming that N unlabeled samples \( (x_1, x_N) \) are available from the mixture 5.1, in addition to \( N_j \) training samples \( (z_{j1}, \ldots, z_{jN_j}) \) from each class \( S_j \), the code length for transmitting the data and the model parameters can be written in the following form:
6 Density Model Selection

\[ L(X,Z,\text{Model}) = -k \sum_{j=1}^{N} \log f(x_j|\theta) - \sum_{j=1}^{N} \sum_{k=1}^{J} \log \left( \frac{1}{T_j} \sum_{e \in S_j} \alpha_t \phi_t(x_{jk}|\phi_t) \right) + \frac{1}{2} k \log N^* \] (6.1)

Where the first term in the above equation is the length of the shortest prefix code for the unlabeled data, the second term is the length of the shortest prefix code for the training samples, and the third term is the length of the code for transmitting the parameters of the model, in which \( N^* \) denotes the total number of samples (unlabeled and training). The term \( k \) indicates the number of free parameters in the model. For a normal mixture density with \( m \) components, such as the one in equation (5.1), one can write:

\[ k = m(d + \frac{d(d+1)}{2} + 1) - 1 \] (6.2)

where \( d \) is the dimensionality of the space.

The parameter set \( \theta \) for each model under consideration must be estimated so that the description length is minimized. To do this, the first two terms in equation (6.1) must be minimized with respect to \( \theta \). The solution is the maximum likelihood estimate of \( \theta \) that can be found through the EM equations (5.15), (5.16), and (5.17).

6.3 Estimate of Total Allocation Rate

Estimation of the classification accuracies, or allocation rates, can be very useful in assessing the validity of a model. Training samples can be used for estimating the classification accuracy by the resubstitution method. Consider a \( J \)-class problem where
the classes may have multiple components as described in chapter 5. Let the total number of components be \( m \). Let \( z_{jk} \) be the \( k \)th training sample from class \( S_j \). Let \( I_j(x) \) be the indicator function for class \( S_j \); it is one if \( x \) is classified to class \( S_j \) based on its posterior probability and is zero otherwise. The correct allocation rate (percentage of correctly classified samples) for class \( S_j \) can be estimated by the count estimator:

\[
C_j = \frac{\sum_{k=1}^{N_j} I_j(z_{jk})}{N_j}
\]

(6.3)

The total correct allocation rate is therefore:

\[
C = \sum_{j=1}^{J} \left( \sum_{i \in S_j} \alpha_i \right) C_j
\]

(6.4)

where \( \alpha_i \) is the prior probability of component \( i \), and \( i \in S_j \) means that component \( i \) belongs to class \( S_j \). The above estimator for the total correct allocation rate is only based on the training samples. Alternatively, unlabeled samples can be used to estimate the allocation rates. In [12, 13], posterior probabilities of unlabeled samples are used for estimating the probability of correct classification. Based on these ideas, in [65] the use of the estimates of the allocation rates using unlabeled samples for cluster analysis is studied. Intuitively, one can argue that if the unlabeled samples are classified based on maximum a posterior probabilities, then a good mixture model is one under which all of these posterior probabilities have values near one. In other words, the clustering is not very fuzzy since each sample is classified with high probability. The estimates of class-conditional and total allocation rates based on unlabeled samples are [65]:
where $P(S_r|x_k)$ is the posterior probability of class $S_r$ given the observation $x_k$. In our case of interest, both training and unlabeled samples are available. Therefore, we propose the use of the following estimates for the allocation rates that are based on both types of samples:

$$C_j = \sum_{k=1}^{N} \frac{\max_r P(S_r|x_k)I_j(x_k)}{N \sum_{i \in S_j} \alpha_i}$$

(6.5)

$$C = \frac{\sum_{k=1}^{N} \max_r P(S_r|x_k)I_j(x_k)}{N}$$

(6.6)

We use equation (6.8) and refer to it as TAR (Total Allocation Rate) for evaluating different mixtures models. If TAR is near one, it means that under the proposed model the training samples are classified correctly and unlabeled samples are classified with high posterior probabilities. Notice that these estimates are all optimistically biased for the true allocation rates since the samples are already used for estimating the parameters. The bootstrap method can be used for bias correction [65].
5.4 Model Selection By Backtrack Search

Here we propose a systematic way for producing a set of candidate models based on the backtracking search strategy that uses both training and unlabeled samples. The procedure starts from an initial model. The parameters of the model are estimated via the EM equations of chapter 5. The validity of the initial model is tested, and if it is found unsuitable a new model is generated by adding a new component to one of the class distributions by splitting a component into two. If the new model is more suitable than its parent model the process continues from the new model, otherwise the procedure backtracks to the parent model. The same process is continued until either a suitable model is found, or "depth bounds" are reached and the search terminates. After the termination of the search, the best model among the visited models may be selected. The following is the outline of the procedure.

main program:

BackTrack ( Initial Model )

BackTrack (Model):

Identify the Model using the EM equations
if Model is Accepted save Model., return (Found)
if Depth Bound return (Fail)
Loop
if (no components left unsplit in Model) return (Fail)
split next component in the split candidate list
out = BackTrack (NewModel)
if (out = Fail) goto Loop
else return (Found)
If the procedure terminates without finding a suitable model, then among the visited models that were not rejected by the depth bounds, the "best" one is selected and returned. In the following paragraphs the details of the procedure are explained.

The variables in the procedure may be changed by the analyst to optimize the performance according to the circumstances. However, the following suggestions are made based on our experiments. The initial model may be selected to be a normal mixture density in which each informational class has a single component. Alternatively, one can use the training samples of each class to come up with an initial model for that class by a method such as the one described in example 6.1. In any case, the initial parameters for the EM equations must be chosen by using the training samples. If there exists a class with no training samples ("unknown" class), then we suggest that a small percentage (~5%) of the unlabeled data that are furthest from the informational classes be used for obtaining the initial parameters of the "unknown" class. In such case, one might consider constraining the prior probability of this class to be smaller than a small value (0.1) that reflects the knowledge of the analyst about that class. This reduces the possibility that bad initial points result in convergence to a bad local maximum where this unknown class is dominant. The procedure for the constrained estimation was discussed in section 5.4 of chapter 5.

The procedure may end when a "suitable" model is found. The criterion of suitability may be any of the aforementioned ones. In our experiments, we usually use a high value (e.g. 95%) as the threshold on the total allocation rate. If the estimate of the total allocation rate is higher than this threshold value, the model may be considered suitable. Alternatively, one may not consider any model to be suitable, in which case the
procedure simply performs an exhaustive search up to the depth bounds and then among the visited models the "best" is selected. Again, the selection of the best model among the candidate models may be based on any of the criteria that were discussed earlier. In our experiments, we select the model with the highest total allocation rate and break the ties among such models by choosing the one with the minimum description length.

A new model is generated by splitting a component into two and re-estimating the parameter values using the EM formulas. A component is split into two new components along its principal axis. The child components have equal initial prior probabilities, equal covariance matrices (equal to 0.25 times the covariance of the parent component), and means at equal distances from the mean of the parent component along its principal axis. Figure 6.2 illustrates the splitting operation.

![Figure 6.2: The splitting of a component into two](image)

At each model under consideration, the number of candidates for splitting is equal to the number of components in the mixture. It is possible to rank the candidates based on some criteria. In our algorithm, we have ranked the components based on their contribution to the asymmetry of the confusion matrix associated with the classification of the training
data. For each class, we compute the percentage of the training samples of the other classes that would be classified wrongly as that class. The class with the largest such percentage is considered to be the class for which one of the components should be split. Among the components of the selected class, again the component which contributes most to the misclassification of the training samples from the other classes is selected. In this way, a list of candidates for the split operation is formed at each state.

Several depth bounds may be used to constrain the search space. The first depth bound is based on MDL. If a new model has a larger description length than its parent, it is rejected and the process backtracks. The second depth bound is reached when a new model has a lower TAR than its parent. The third depth bound is based on the number of components in an informational class. Since the association of the components to classes is based on training samples and it is unreasonable to think that a small number of training samples can identify a large number of components, we limit the maximum number of components for an informational class as a function of the number of training samples from that class. Intuitively, we argue that at least \( d+1 \) training samples are required to identify a component as a member of a class. Therefore, the maximum number of components per informational class \( S_j \) is \( \lfloor N_j / (d+1) \rfloor \) (minimum is, of course, 1). In this way, on average at least \( d+1 \) training samples belong to each component of an informational class. This could prevent the covariance matrices of the components from ever becoming singular and therefore reduces the chances of the EM algorithm falling into singular points of the parameter space.

The bulk of the computations for performing the search procedure is in the identification of each model via the EM algorithm. This process may be performed using the MasPar parallel computer system as described in chapter 5.
Example 6.2:

A mixture of six two-dimensional normal components with statistics shown in the third column of Table 6.2 is considered for this experiment. Components 1 and 2 are assumed to belong to class 1, components 3 and 4 to class 2, component 5 to class 3 and component 6 to class 4. Five hundred unlabeled samples are drawn from this mixture. Figure 6.3 shows these samples. In addition to these unlabeled samples, training samples from classes 1, 2, and 3 are also assumed to be available. The numbers of training samples from each of the classes are listed in the fourth column of Table 6.2. Notice that no training samples from class 4 are used. The purpose of the experiment was to use the 500 unlabeled and the 24 training samples through the proposed search algorithm in order to discover the probability density function of the sample space.

The initial state of the search process is a mixture of four normal components: one for each of classes 1 through 3 (from which training samples are available) and an extra component for the missing class. Starting statistics for classes 1, 2, and 3 are obtained by using the training samples. For the missing class, class 4, the starting statistics are obtained by using the five percent of the unlabeled samples which are furthest from the initial locations of the classes 1 through 3. The measure of distance used is Mahalanobis distance.
Table 6.2: Statistics of the mixture density in example 6.2 and the result of the search process

<table>
<thead>
<tr>
<th>class</th>
<th>component</th>
<th>original statistics $\mu_i, \Sigma_i, \alpha_i$</th>
<th># training samples</th>
<th>statistics obtained by the search process $\mu_i, \Sigma_i, \alpha_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>$\begin{bmatrix} .6 \ .4 \end{bmatrix}$, .6, 0.20</td>
<td>10</td>
<td>$\begin{bmatrix} 2.05 \ 2.06 \end{bmatrix}$, 0.62, 0.43, 0.63, 0.</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>$\begin{bmatrix} -1 \ 3 \end{bmatrix}$, 1, 0.20</td>
<td></td>
<td>$\begin{bmatrix} -0.89 \ 3.18 \end{bmatrix}$, 1.17, 0.95, 0.19</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>$\begin{bmatrix} -2 \ -1 \end{bmatrix}$, 0.5, 0.20</td>
<td></td>
<td>$\begin{bmatrix} 1.90 \ 2.05 \end{bmatrix}$, 0.85, 1.01, 0.14</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>$\begin{bmatrix} 0 \ 0 \end{bmatrix}$, 0.5, 0.12</td>
<td></td>
<td>$\begin{bmatrix} 0.02 \ 0.09 \end{bmatrix}$, 0.52, 0.80, 0.12</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>$\begin{bmatrix} 4.5 \ 0 \ 0 \end{bmatrix}$, 0.12</td>
<td></td>
<td>$\begin{bmatrix} 4.39 \ 0.08 \end{bmatrix}$, 0.57, 1.11, 1.52, 0.14</td>
</tr>
</tbody>
</table>
The goal state was set to be a state with 100% total training sample classification accuracy. The reason for choosing such a goal criterion was to let the algorithm perform a complete search and find the best model for the distribution of the sampling space. The depth bounds used were MDL, and the estimate of the classification accuracy based on equation (6.4).

After visiting 21 models, the algorithm terminated without finding such a goal model. The search tree is shown in Figure 6.4. Each visited model in the search tree is denoted...
Density Model Selection

by a box in which individual classes are shown by dots connected by bars. Each dot stands for a single component, and components which belong to the same class are joined by a bar. The order of the classes in each state box is from left to right; the left-most being class 1 and the right-most being class 4. Next to each model box two numbers are written. The upper number is the total training sample classification accuracy at that state and the lower number is the MDL value. The sequence in which states were visited is indicated by the numbers that are inside circles below the model boxes. All of the leaf models are the ones that were rejected by the depth bounds. Therefore, at the time of termination of the search process the admissible models were states 1, 3, 10 and 15, among which state 15 had the highest training sample classification accuracy and therefore was selected as the best visited state. Notice that state 15 also has the smallest MDL value. Also, notice that in state 15, classes 1 and 2 each have two components. The last column of Table 6.2 contains the statistics of model 15. A comparison of these statistics with those of the true distribution listed in the third column of Table 6.2 shows that the algorithm was successful in discovering the structure of the data.
Example 6.3:

The following experiment was performed using a Landsat Thematic Mapper (TM) data set taken over Tippecanoe County, Indiana in July 1986. Ground truth data was gathered at the same time. Four informational classes were selected in the scene: corn, soybean, wheat and alfalfa/oats. From the ground truth map, the numbers of pixels from the informational classes were found to be: 9371, 8455, 1923 and 2175 respectively. All of the seven bands were used in the experiments. The criterion for accepting a model was having a total allocation rate (based on equation 6.8) more than 95%. The depth bounds used were based on MDL, TAR (equation 6.8), and based on the maximum allowable number of components per class as discussed previously.
Two different experiments were performed with two different sets of training samples. In experiment A, training samples were selected very carefully by an analyst through a trial and error procedure in order to maintain a high classification accuracy when a Gaussian maximum likelihood (GML) classifier was used. These training fields are highlighted in figure 6.5 (a). The total classification error of the GML classifier was 19.98%. Figure 6.6 (a) shows the error map for the GML classifier. Dark pixels in the error map indicate the pixels that were classified incorrectly. Using these "good" training fields and 1849 unlabeled samples drawn from the image (every fourth pixel in every fourth line), the proposed algorithm achieved a classification error of 18.37%. An extra class was used as the unknown class and its prior probability was constrained to less than 0.1. The error map for this case is illustrated in figure 6.7 (a). The same procedure was performed with 3249 unlabeled samples (every third pixel in every third line). The classification error was 18.46%. Figure 6.7 (b) shows the error map for this case. The reason for the degradation of performance when more unlabeled samples were used could be the presence of additional outliers in the unlabeled data.

In experiment B, the training fields were altered in order to illustrate the result of using small unrepresentative training samples in the classification process. The new training fields are highlighted in figure 6.5 (b). The GML classifier achieved a classification error of 27.73% in this case; a 7.75% increase in error from the last experiment. Figure 6.6 (b) shows the error map for the GML classifier. Using 1849 unlabeled samples, the proposed method achieved 20.25% classification error and using 3249 unlabeled samples the classification error was further reduced to 16.66%. Figures 6.7 (c) and 6.7 (d) illustrate the error maps for these two cases respectively. Obviously, the GML classifier is very
sensitive to the number and quality of training samples. This sensitivity was reduced by incorporating the unlabeled samples through the proposed algorithm.

Figure 6.5: Training fields for example 6.3. (a) training fields for experiment A, (b) training fields for experiment B.

Figure 6.6: Error maps for the GML classifier in example 6.3. (a) error map for experiment A, error = 19.98%. (b) error map for experiment B, error = 27.73%.
Figure 6.7: Error maps using additional unlabeled samples. (a) error map using 1849 unlabeled samples in experiment A, error = 18.37%. (b) error map using 3249 unlabeled samples in experiment A, error = 18.46%. (c) error map using 1849 unlabeled samples in experiment B, error = 20.25%. (d) error map using 3249 unlabeled samples in experiment B, error = 16.66%. 
Example 6.4:
In this example, a part of an AVIRIS image taken over Indiana was studied. Ten bands were used in the analysis. Seven ground cover classes were chosen in the scene: harvested corn, corn, soybean, red clover, bare soil, wheat, and soybean-2 with 16, 30, 42, 16, 16, 41, 16 training samples respectively. The number of test samples used for estimating the classification accuracy were: 1260, 1658, **2789**, 453, 400, 1847, and 1248. By visual inspection of the image, it was decided that the above list of the ground cover types were exhaustive and therefore no "unknown class" was used in this experiment. The scene was classified once by the GML classifier, and once by using an additional 1364 (every third pixel in every third row) through the aforementioned search procedure. The procedure was repeated again, but this time the modified EM algorithm with $\lambda = 1$, which was discussed in section 5.5, was used in the search procedure. In this way, at each iteration of the EM algorithm the training samples were forced to have the same amount of effect in estimating the new statistics as the unlabeled samples. The results are shown in Figure 6.8. In both of the cases, the search terminated with one component per class. A 95% threshold on the total allocation rate was used for terminating the procedure. The total classification accuracies for the GML classifier, EM based search algorithm, and the modified EM based search algorithm were **84.41%**, **92.27%**, and 91.27% respectively.
In order to investigate the effect of number of unlabeled samples, the same example was repeated again but with 759 unlabeled samples (every fourth pixel in every fourth row). The results are shown in Figure 6.9. It can be seen that the search procedure obtained a better total classification accuracy than the GML classifier, however in doing so, sometimes smaller classes, such as the bare soil class, were dominated by other classes. The use of the modified EM algorithm helped reduce this phenomenon. The total classification accuracies for the GML, EM, and modified EM procedures were 84.41%, 89.37%, and 87.95% respectively.
Example 6.5:
This example was performed on part of the FLC1 data set. All of the twelve bands were used. Five ground cover classes were selected: oats, red clover, soybean, wheat, and corn. Thr: number of the training samples per class were: 25, 49, 21, 16, and 20 respectively. Thr: number of the test samples used for estimating the accuracy were: 1975, 8117, 4296, 1149, and 941 respectively. The test samples were classified first using the GML classifier. Next, every second pixel in every second row of the image section (a total of 5202 pixels) were used through the search procedure. The procedure was performed once using the regular EM algorithm and once using the modified EM algorithm with $\lambda = 1$. 

Figure 6.9: Classification results for example 6.4 with 759 unlabeled samples.
An additional "unknown" class was considered here and its initial parameters were estimated by using 5% of the unlabeled data that were furthest from the informational classes. The criterion of suitability was a 95% threshold on the Total Allocation Rate estimate. The results are shown in Figure 6.10. The total classification accuracies for the GML, EM, and modified EM procedures were 88.41%, 94.85%, and 90.23% respectively. The total classification accuracy was improved by incorporating the unlabeled samples.

![Classification Results](image)

Figure 6.10: Classification results for example 6.5 with 5202 unlabeled samples.

In order to investigate the effect of the number of unlabeled samples, the same experiment was repeated again, but with 2422 unlabeled samples (every third pixel in
every third row). The results are shown in Figure 6.11. The total classification accuracies were very similar to the previous case: 88.41%, 94.85%, and 90.42% for the GML, EM based, and modified EM based procedures respectively.

![Classification results for example 6.5 with 2422 unlabeled samples.](image)

Figure 6.11: Classification results for example 6.5 with 2422 unlabeled samples.

The experiment was repeated one more time with 1326 unlabeled samples drawn from every fourth pixel in every fourth row. The results are shown in Figure 6.12. It is seen that in this case, although the search procedure obtained a better total classification accuracy, the regular EM based method obtained a very low accuracy for the corn class. Most of the corn samples in this experiment were classified as soybeans. This is due to the fact that these two classes are very close to each other and the corn class contains fewer samples.
The EM based method resulted in the soybean class dominating the corn class. When the modified EM method was used, this phenomenon was eliminated because the training samples in this case were given as much weight as the unlabeled samples. Therefore, the statistics of the corn class were not allowed to change as much. The total classification accuracies for the GML, EM, and modified EM procedures were 88.41%, 92.04%, and 89.99% respectively.

Figure 6.12: Classification results for example 6.5 with 1326 unlabeled samples.
6.5 Concluding Remarks

In this chapter, a method for obtaining probability density models for the multi-spectral data was proposed that is based on both training samples and unlabeled samples. A backtracking search strategy was used for generating candidate models, and various model validity criteria such as MDL, and estimates of allocation rates, were used for constraining the search space and selecting the most suitable candidate model. Experiments were presented based on both the regular and the modified EM algorithm. It was noticed that the modified EM algorithm can be very useful in obtaining balanced decision rules. The regular EM algorithm may cause the larger classes to dominate the smaller classes. In this case, although the total classification accuracy may be high, the per-class or average classification accuracy may be low. The selection of the \( \lambda \) factor in the modified EM algorithm remains to be studied.
CHAPTER 7: SUMMARY AND RECOMMENDATIONS FOR FUTURE WORK

7.1 Summary

In this thesis, the use of unlabeled data in improving the classification performance of statistical classifiers was studied. In Chapter 2, it was shown that the performance of classifiers can be improved by using unlabeled samples in addition to available training samples. As a result, the Hughes phenomenon can be delayed to a higher dimensionality, and, thus more feature measurements can be used. In addition, the performance of the feature extraction methods that are based on the estimates of class parameters is expected to improve. Moreover, it was shown that the newly generated statistics more accurately represent the true class distributions.

In Chapter 3, analytical results concerning the expected amount of improvement in the classification accuracy, when the additional unlabeled samples are used, were derived. The supervised, unsupervised, and combined supervised-unsupervised learning processes were compared by studying the expected error rates of a two class problem in each case. Upper and lower bounds on the bias of the classification error, and the bias and the variance of the estimated Bhattacharyya distance were derived. These bounds explicitly show the relationship between dimensionality and sample size for these learning processes.
In Chapter 4, the estimates of the class parameters in a multi-class problem were studied. The asymptotic covariance matrices of the estimates obtained by supervised, unsupervised, and combined supervised-unsupervised learning processes were compared.

A semi-parametric method for combined supervised-unsupervised learning that is based on the Expectation-Maximization method was proposed in Chapter 5. This method extends the existing parametric method by allowing the classes to have multiple Gaussian components. The iterative EM equations were derived, and their implementation on a massively parallel computer system was discussed.

In Chapter 6, the problem of density model selection for classification was studied. A backtrack search strategy was proposed for generating a set of candidate models. The Minimum Description Length, and the Total Allocation Rate estimate were used for selecting the most suitable model among the candidates.

1. The EM equations for obtaining the Maximum Likelihood estimates are based on assigning a set of weights to each unlabeled sample. These weights are equal to the posterior probabilities of the classes given each observation. The computation of the posterior probabilities involves taking the exponential of the Mahalanobis distances. In high dimensions, the distances become large and the exponential functions become hard to evaluate. It is possible to approximate the EM equations by using alternative weight functions that do not involve exponential functions. For example, normalized Mahalanobis distances can be used instead. In this way, the weights would indicate the
"degree of membership" of an unlabeled sample to each class. Similar techniques have been used in the fuzzy logic literature [66].

2. In chapter 2, it was shown that the probability maps that are obtained by using the combined supervised-unsupervised learning process are brighter than the ones obtained by using training samples alone. This is due to the fact that in the estimation process the likelihood of the total data, including both the unlabeled and training samples, is maximized. Therefore, one expects to get a higher likelihood for the whole image. However, in clustering and classification applications, another desirable property for a model and parameter estimates is that the posterior probabilities of the unlabeled samples be high. In other words, an unlabeled sample that is classified to a particular class should yield a high posterior probability for that class. The average of these maximum posterior probabilities is an estimate of the classification accuracy, as was discussed in section 6.3. In this regard, in addition to the probability maps, posterior probability maps may be created and used as a visual aid in evaluating the results of the analysis. In such a posterior probability map, each pixel would be colored according to the posterior probability (or "degree of membership") of the class that it was classified to (see Figure 7.1). In chapter 6, we did use the estimates of the allocation rates for evaluating the models. However, nothing in the EM algorithm suggests that the estimated parameters would yield a high average posterior probability. For example, in Figure 7.1 the posterior probability maps for the FLC1 data set are shown. The darker pixels are the ones which were classified with high posterior probabilities and the lighter pixels are the ones that were classified with low posterior probabilities (notice that this is opposite to the probability maps, but it created a clearer image). The training fields are highlighted.
Comparing Figure 7.1, with Figure 2.11, that contains the probability maps, shows that although the use of unlabeled samples increased the likelihood of the whole data set, it in fact reduced the average posterior probability under which a pixel is classified. It would be desirable if a method for both model selection and parameter estimation could be proposed that could obtain both a good probability map and a good posterior probability map.
3. The work in chapter 3, regarding the bounds for the error rate and Bhattacharyya distance, should be extended to the case of unknown covariance matrix.

4. The use of additional unlabeled samples for designing robust decision rules may be extended to the neural networks paradigm.

5. More work needs to be done on developing suitable techniques for density model selection for classification. In particular, the minimum description length principle should be used in conjunction with criteria that are more closely related to the error rate of the decision rule, rather than the cost of transmitting the data.

6. The feature extraction process deserves much attention as the dimensionality of data increases. The traditional feature extraction methods are based on eliminating the less informative features in order to increase the speed of the classification process. With high dimensional data and limited training samples, an additional objective of the feature extraction process should be to reduce the dimensionality in a way to avoid the Hughes phenomenon. We share the remarks made in [67], that the feature extraction process might be divided into two separate processes. The first process would be a preprocessing stage aimed at eliminating the Hughes phenomenon, and the second process would be a traditional feature extraction process aimed at reducing the computational cost. The preprocessing stage must not be affected by the curse of dimensionality; therefore it should either be very simple or be based on reliable estimates that might be obtained by using the unlabeled data.
LIST OF REFERENCES


References


References


[28] S. John, "On identifying the population of origin of each observation in a mixture of observations from two normal populations," Technometrics, vol 12, pp 553-563, 1970


References


References


[68] H. Schneider, Truncated And Censored Samples From Normal Populations, New York: Marcel Dekker, 1986
This appendix contains the derivations of the entries of the matrices in equation (3.13).

Consider the first term in the right hand side of equation (3.12).

The upper left sub-matrix:

The \((k,l)\)th entry where \(k \neq 1\):

\[
\frac{P_1}{2} \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-0.5x_1^2} dx_1 \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-0.5x_k^2} dx_k \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-0.5x_1^2} dx_1 = 0
\]

The \((1,1)\)th entry:

\[
\frac{P_1}{2} \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-0.5x_1^2} dx_1 = P_1 \Phi(t) = P_1 \Phi(t) - t\phi(t)
\]

Where the last equality is established by using the second order statistics of the right truncated normal densities [68].

The \((k,k)\)th entry \((k \neq 1)\):

\[
\frac{P_1}{2} \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-0.5x_1^2} dx_1 \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-0.5x_k^2} dx_k = \frac{P_1}{2} \Phi(t)
\]

The upper right sub-matrix:

\[
\int_{\Omega_1} \frac{1}{2} P_2 f_2(x)(x - \mu_2)^T dx = \int_{\Omega_1} \frac{1}{2} P_2 f_2(x)xx^T dx - \int_{\Omega_1} \frac{1}{2} P_2 f_2(x)dx \mu_2^T
\]

Consider the first term in the above equation first. The \((k,l)\)th entry \((k \neq l)\) is:
The \((k,k)\)th entry \((k \neq 1)\) is:

\[
\frac{P_2}{2} \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x_1-\Delta)^2} dx_1 \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_k^2} dx_k \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_i^2} dx_i = 0
\]

And the \((1,1)\)th entry is:

\[
\frac{P_2}{2} \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x_1-\Delta)^2} dx_1 \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_k^2} dx_k = \frac{P_2}{2} \Phi(t-\Delta)\left[1 - \frac{\phi(t-\Delta)}{\Phi(t-\Delta)}(t + \Delta) + \Delta^2\right]
\]

where the last equality is established by using the second order statistics of right truncated normal densities [68].

Now let us look at the second term in equation (A.1.1). Consider the integral:

\[
\int_{\Omega_1} \frac{P_2}{2} f_2(x) dx
\]

The first element of the above equation is:

\[
\frac{P_2}{2} \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x_1-\Delta)^2} x_1 dx_1 = \frac{P_2}{2} \left[ \Delta \Phi(t-A) - \Phi(t-A) \right]
\]

Where, the first moment of the right truncated normal density is used to evaluate the above equation. The kth element \((k \neq 1)\) of equation (A.1.2) can be written as:

\[
\frac{P_2}{2} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x_1-\Delta)^2} dx_1 \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_k^2} dx_k = 0
\]

Therefore, the second term of (A.1.1) can be written in the following form:
Putting all these together, the upper right sub-matrix of the first term in equation (3.12) can be written as a diagonal matrix whose entry \((1,1)\) is:

\[
\frac{p_2}{2} [\Phi(t - A) - \phi(t - A)]
\]

And whose entry \((k,k)\) is:

\[
\frac{p_2}{2} \Phi(t - \Delta)
\]

The lower left sub-matrix:

Since the top right sub-matrix is diagonal, the bottom left sub-matrix is equal to the top right sub-matrix.

The lower right sub-matrix:

This sub-matrix can be written in the following form:

\[
\int_{\Omega_1} \frac{P_2^2 P_2^2(x)}{2P_1 f_1(x)} (x - \mu_2)(x - \mu_2)^T dx
\]

\[
= \int_{\Omega_1} \frac{P_2^2}{2P_1} \frac{1}{(2\pi)^d} \exp\left[-\frac{1}{2} (x - \mu_2)(x - \mu_2)^T\right] dx
\]

\[
= \int_{\Omega_1} \frac{P_2^2}{2P_1} \frac{1}{\sqrt{(2\pi)^d}} \exp\left[-\frac{1}{2} \frac{1}{2} [x_1^2 + x_2^2 + \cdots + x_d^2] \right] dx
\]
The \((k,l)\) entry of the above equation is \((k\neq l)\):

\[
\frac{P_2^2}{2P_1} \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} x_1^2} dx_1 \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} x_k^2} x_k dx_k \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} x_1^2} x_1 dx_1 = 0
\]

The \((k,k)\) entry of equation (A.1.3) (where \(k\neq l\)) can be written as:

\[
\frac{P_2^2}{2P_1} \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} x_1^2} dx_1 \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} x_k^2} x_k dx_k = \frac{P_2^2}{2P_1} \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} x_1^2 + 2 \Delta \xi} d\xi
\]

\[
= \frac{P_2^2}{2P_1} e^{\Delta^2} \Phi(t - 2\Delta)
\]

Finally, the \((1,1)\) entry of equation (A.1.3) can be written as follows:

\[
\frac{P_2^2}{2P_1} \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2} x_1^2 + 2 \Delta \xi} - (x_1 - \Lambda)^2 dx_1
\]

\[
= \frac{P_2^2}{2P_1} e^{\Delta^2} \left[(\Delta^2)\Phi(t - 2\Delta) - \tau\Phi(2\Delta - t)\right]
\]

Now, consider the second term in the right hand side of equation (3.12).

The upper left sub-matrix:
The entry \((k,l)\) of the above equation (where \(k \neq l\)) is:

\[
\frac{P_1^2}{2P_2} \int_{\Omega_2} \frac{1}{\sqrt{(2\pi)^d}} e^{-x_i^2 + \frac{1}{2}(x_1 - \Delta)^2} e^{-\frac{1}{2}(\Delta^2 + L x_i^2)} \text{xx}^T dx_1
\]  

(A.4)

The entry \((k,k)\) of equation (A.1.4) (where \(k \neq l\)) can be written in the following form:

\[
= \frac{P_1^2}{2P_2} \int_{\Omega_2} \frac{1}{\sqrt{2\pi}} e^{-x_i^2 + \frac{1}{2}(x_1 - \Delta)^2} dx_1 \int_{\Omega_2} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_k^2} x_k dx_k \int_{\Omega_2} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_i^2} x_i dx_1 = 0
\]

The entry \((I_1, I_1)\) of equation (A.1.4) (where \(k \neq l\)) can be written in the following form:

\[
= \frac{P_1^2}{2P_2} \int_{\Omega_2} \frac{1}{\sqrt{2\pi}} e^{-x_i^2 + \frac{1}{2}(x_1 - \Delta)^2} dx_1 \int_{\Omega_2} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_k^2} x_k dx_k
\]

\[
= \frac{P_1^2}{2P_2} \int_{\Omega_2} e^{-\frac{1}{2}x_i^2 - \Delta x_i + \frac{1}{2} \Delta^2} dx_1
\]

\[
= \frac{P_1^2}{2P_2} e^{\Delta^2} \Phi(-\Delta - t)
\]

And finally, the entry \((1,1)\) of equation (A.1.4) can be written in the following form:

\[
= \frac{P_1^2}{2P_2} \int_{\Omega_2} \frac{1}{\sqrt{2\pi}} x_i^2 e^{-\frac{1}{2}x_i^2 - \Delta x_i + \frac{1}{2} \Delta^2} dx_1
\]

\[
= \frac{P_1^2}{2P_2} e^{\Delta^2} [(1 + \Delta^2) \Phi(-\Delta) + (t - \Delta) \Phi(t + \Delta)]
\]

The upper right sub-matrix:
Consider the first term in the right hand side of the above equation. The entry \((k,l)\) of this term (where \(k \neq l\)) can be written in the following form:

\[
\frac{P_1}{2} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_1^2} dx_1 \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_k^2} x_k dx_k \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_i^2} x_i dx_i = 0
\]

The entry \((k,k)\) of the first term in the right hand side of (A.1.5) can be written in the following form:

\[
\frac{P_1}{2} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_i^2} dx_i \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_k^2} x_k dx_k = \frac{P_1}{2} \Phi(-t)
\]

The entry \((1,1)\) of the first term in the right hand side of (A.1.5) can be written in the following form:

\[
\frac{P_1}{2} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_i^2} x_i^2 dx_i = \frac{P_1}{2} [\Phi(-t) + t\phi(t)]
\]

Where the second order statistic of the left truncated normal density is used to get the above result.

Now, let's study the second term in (A.1.5). Consider the following integral:

\[
\frac{P_1}{2} \int_{\Omega_2} f_1(x) dx
\]

The first entry of the above equation can be written as follows:

\[
\frac{P_1}{2} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_1^2} x_1 dx_1 = \frac{P_1}{2} \phi(t)
\]
Any other entry $k$, can be written as:

$$\frac{P_l}{2} \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_t^2} dx_t \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_k^2} x_k dx_k = 0$$

Therefore, the second term of (A.1.5) can be written in the following form:

$$\frac{P_l}{2} \Delta\phi(t) \begin{bmatrix} 1 & [0] \\ [0] & [0] \end{bmatrix}$$

Putting all these together, the upper right sub-matrix of the second term in equation (3.12) can be written as a diagonal matrix whose $(1,1)$ entry is:

$$\frac{P_l}{2} |\Phi(-t) + (t - \Delta)\phi(t)|$$

and whose $(k,k)$ entry is:

$$\frac{P_l}{2} \Phi(-t)$$

The lower left sub-matrix:

Since the top right sub-matrix is diagonal, the bottom left sub-matrix is equal to the top right one.

The lower right sub-matrix:

$$\int_{\Omega_2} \frac{P_2 f_2(x)}{2} (x - \mu_2)(x - \mu_2)^T dx$$

The $(k,l)$ entry (where $k \neq l$) of the above equation is:
The $(k,k)$ entry (where $k \neq 1$) can be written as follows:

\[ \frac{P_2}{2} \int_1^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x_1-\Delta)^2} \, dx_1 \int_1^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_k} \, dx_k \int_1^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_1} \, dx_1 = 0 \]

And the entry $(1,1)$ can be written as:

\[ \frac{P_2}{2} \int_1^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(x_1-\Delta)^2} \, dx_1 = \frac{P_2}{2} \Phi(\Delta - t) \]

\[ (x_1 - \Delta)^2 = \frac{P_2}{2} \left| \Phi(\Delta - t) + (t - \Delta)\phi(t - \Delta) \right| \]
Appendix B

In this appendix it is shown that \( \text{var}(e^{\hat{\theta}}) \) is order \( O(1/n^2) \).

Consider \( E(e^{\hat{\theta}}^2) \):

\[
E(e^{\hat{\theta}}^2) = E\left[ \text{err}^* + \frac{1}{2} \text{tr}\left( \frac{\partial^2 \text{err}}{\partial \theta^2} \right)_{\theta = \theta^*} (\hat{\theta} - \theta^*)(\hat{\theta} - \theta^*)^T \right]^2
\]

Expanding the square one gets:

\[
= \left( \text{err}^* \right)^2 + \text{err}^* \text{tr}\left( \frac{\partial^2 \text{err}}{\partial \theta^2} \right)_{\theta = \theta^*} \text{cov}(\hat{\theta}) + \frac{1}{4} E\left[ \text{tr}\left( \frac{\partial^2 \text{err}}{\partial \theta^2} \right)_{\theta = \theta^*} (\hat{\theta} - \theta^*)(\hat{\theta} - \theta^*)^T \right]^2
\]

Using equation (3.28) in the above equation, one gets:

\[
E(e^{\hat{\theta}}^2) = \left( \text{err}^* \right)^2 + 2\text{err}^* \left[ E(e^{\hat{\theta}}) - \text{err}^* \right] + \frac{1}{4} E\left[ \text{tr}\left( \frac{\partial^2 \text{err}}{\partial \theta^2} \right)_{\theta = \theta^*} (\hat{\theta} - \theta^*)(\hat{\theta} - \theta^*)^T \right]^2
\]

Since \( \text{var}(e^{\hat{\theta}}) = E(e^{\hat{\theta}}^2) - (E(e^{\hat{\theta}}))^2 \) one gets the following expression for \( \text{var}(e^{\hat{\theta}}) \):

\[
\text{var}(e^{\hat{\theta}}) = -\left(E(e^{\hat{\theta}}) - \text{err}^* \right)^2 + \frac{1}{4} E\left[ \text{tr}\left( \frac{\partial^2 \text{err}}{\partial \theta^2} \right)_{\theta = \theta^*} (\hat{\theta} - \theta^*)(\hat{\theta} - \theta^*)^T \right]^2
\]

The first term in the right hand side is \( O(1/n^2) \) from (3.28) and the fact that \( \text{cov}(\hat{\theta}) \) is the inverse of the Fisher information matrix and hence is \( O(1/n) \). The second term contains fourth order moments such as \( E((\hat{\theta}_i - \theta^*_i)(\hat{\theta}_j - \theta^*_j)(\hat{\theta}_k - \theta^*_k)(\hat{\theta}_l - \theta^*_l)) \). But, since the ML estimate \( \hat{\theta} \) is asymptotically normal with mean equal to \( \theta^* \) and covariance matrix equal to the inverse of the Fisher information \[10\], one can write \[32\]:

\[
E\left\{ (\hat{\theta}_i - \theta^*_i)(\hat{\theta}_j - \theta^*_j)(\hat{\theta}_k - \theta^*_k)(\hat{\theta}_l - \theta^*_l) \right\} = \sigma_{ij} \sigma_{kl} + \sigma_{ik} \sigma_{jl} + \sigma_{il} \sigma_{jk}
\]

- 171 -
Appendix B

Where $\sigma_{ij}$ is the $ij^{th}$ element of $\text{cov}(\hat{\theta})$ and is therefore $O(1/n)$. Therefore the second term in the expression of $\text{var}(\epsilon \hat{r})$ is also $O(1/n^2)$. 
Appendix C

This appendix contains the derivation of equation (5.13) from equation (5.12). Consider equation (5.12):

\[
Q^j_c(\theta | \theta^c) = \sum_{i,j_1 \in S_j} \sum_{j_2 \in S_j} \cdots \sum_{i_N \in S_j} \sum_{k=1}^{N_j} \log \left( \frac{\alpha_{i,j_k}}{\sum_{t \in S_j} \alpha_t} f_{i,j_k}(z_{i,j_k} | \phi_{i,j_k}) \right) \prod_{k=1}^{N_j} P^c_j(i_{j_k} | z_{j_k}) 
\]

\[
= \sum_{i,j_1 \in S_j} \sum_{j_2 \in S_j} \cdots \sum_{i_N \in S_j} \left\{ \log \left( \frac{\alpha_{i,j_1}}{\sum_{t \in S_j} \alpha_t} f_{i,j_1}(z_{i,j_1} | \phi_{i,j_1}) \right) + \log \left( \frac{\alpha_{i,N} \cdot \prod_{k \neq 1}^{N_j} P^c_j(i_{j_k} | z_{j_k})}{\sum_{t \in S_j} \alpha_t} f_{i,N}(z_{i,N} | \phi_{i,N}) \right) \right\} 
\]

\[
= \sum_{i,j_1 \in S_j} \sum_{j_2 \in S_j} \cdots \sum_{i_N \in S_j} \left\{ \log \left( \frac{\alpha_{i,j_1}}{\sum_{t \in S_j} \alpha_t} f_{i,j_1}(z_{i,j_1} | \phi_{i,j_1}) \right) \cdot P^c_j(i_{j_1} | z_{j_1}) \cdots P^c_j(i_{N} | z_{N}) \right\} 
\]

\[
+ \log \left( \frac{\alpha_{i,N} \cdot \prod_{k \neq 1}^{N_j} P^c_j(i_{j_k} | z_{j_k})}{\sum_{t \in S_j} \alpha_t} f_{i,N}(z_{i,N} | \phi_{i,N}) \right) \cdot P^c_j(i_{j_1} | z_{j_1}) \cdots P^c_j(i_{N} | z_{N}) \right\} 
\]
But, since $P^c_j$'s are posterior probabilities, the above can be simplified to:

\[
= \left( \sum_{i_{j1} \in S_j} \log \left( \frac{\alpha_{ij1}}{\sum_{i \in S_j} \alpha_{i}} f_{ij1}(z_{j1} | \phi_{ij1}) \right) \right) P^c_j(i_{j1} | z_{j1})
\]

+ \sum_{i_{j2} \in S_j} \log \left( \frac{\alpha_{ij2}}{\sum_{i \in S_j} \alpha_{i}} f_{ij2}(z_{j2} | \phi_{ij2}) \right) P^c_j(i_{j2} | z_{j2})

+ \sum_{i_{jNj} \in S_j} \log \left( \frac{\alpha_{ijNj}}{\sum_{i \in S_j} \alpha_{i}} f_{ijNj}(z_{jNj} | \phi_{ijNj}) \right) P^c_j(i_{jNj} | z_{jNj})
\[ \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} \log \left( \frac{\alpha_i}{\sum_{i=1}^{N_i} \alpha_i} f_i(z_{jk} | \phi_i) \right) p_j^c(z_{jk}) \]
Appendix D

This appendix contains the details of the maximization step of chapter 5, section 5.3.

In equation (5.14), the effects of $\alpha_i$ and $\phi_i$ are separated into different terms, therefore the maximization can be done separately. We first consider the maximization with respect to $\alpha_i$. Using the Lagrangian multiplier $\omega$, $\alpha_i^+$ is obtained by solving the following equations:

$$\frac{\partial}{\partial \alpha_i} \left[ Q(\theta|\theta^c) - \omega \left( \sum_{j=1}^{m} \alpha_j - 1 \right) \right] = 0$$

and

$$\sum_{j=1}^{m} \alpha_j = 1$$

Using equation (5.14), we can write the following:

$$\frac{\partial}{\partial \alpha_i} \left[ \sum_{k=1}^{N} P^c(i|x_k) \log(\alpha_i) + \sum_{k=1}^{N} P^f(i|z_{jk}) \log(\alpha_i) - \sum_{q \in S_j} \left[ \sum_{k=1}^{N_q} P_f^r(q|z_{jk}) \right] \log \left( \sum_{\theta \in S_j} \alpha_i \right) \right. \right.$$

$$- \omega \left( \sum_{j=1}^{m} \alpha_j - 1 \right) = 0$$

Where $i \in S_j$.

Notice that in the third term of the left hand side, the index $q$ is used instead of $i$, in order to emphasize its distinction from the index of $\alpha_i$. Taking the derivative and substituting $\alpha_i^+$ for $\alpha_i$ we get:
Multiplying both sides of the above equation by \( \alpha_i^+ \) and summing all \( m \) such equations over \( i \), we get:

\[
N + \sum_{j=1}^{J} N_j - \sum_{j=1}^{J} N_j \cdot \omega = 0
\]

Therefore, \( \omega = N \). Now, substituting for \( \omega \) in (A.6), multiplying by \( a_j \) and rearranging, the following is obtained:

\[
\sum_{k=1}^{N} P^c(ill_k) + \sum_{k=1}^{N_j} P^c_j(illz_{jk}) = \left( \sum_{q \in S_j} \left[ \sum_{k=1}^{N_j} P^c_j(q|lz_{jk}) \right] \cdot \frac{1}{\sum_{i \in S_j} \alpha_i^+} + N \right) \cdot \alpha_i^+
\]

But:

\[
\sum_{q \in S_j} \sum_{k=1}^{N_j} P^c_j(q|lz_{jk}) = N_j
\]

Therefore, we have:

\[
\sum_{k=1}^{N} P^c(ill_k) + \sum_{k=1}^{N_j} P^c_j(illz_{jk}) = \left( N_j \cdot \frac{1}{\sum_{i \in S_j} \alpha_i^+} + N \right) \cdot \alpha_i^+
\]  \hspace{1cm} (A.7)

Adding all such equations for all \( i \) which belong to the class \( S_j \), we get:
\[
\sum_{r \in S_j} \sum_{k=1}^{N} P_r(r|x_k) + N_j = N_j + N \sum_{r \in S_j} \alpha_r^+
\]

Therefore:

\[
\sum_{r \in S_j} \alpha_r^+ = \frac{\sum_{r \in S_j} \sum_{k=1}^{N} P_r(r|x_k)}{N}
\]

Substituting the above into equation (A.7) and rearranging we get the iterative equations for obtaining the prior probabilities:

\[
\alpha_r^+ = \frac{\sum_{k=1}^{N_i} P_j(z_{jk}) + \sum_{k=1}^{N} P_r(r|x_k)}{N(1 + \frac{N_j}{\sum_{r \in S_j} \sum_{k=1}^{N} P_r(r|x_k)})}
\]

Notice that in doing the above optimization, we did not use the constraint \( \alpha_i > 0 \). However, starting from any admissible initial point satisfying \( \alpha_i > 0 \), the positive property remains intact by using the above equation.

The iterative equations for \( \mu_i \) and \( \Sigma_i \) are obtained by solving the following equations:

\[
\frac{\partial}{\partial \mu_i} Q(\theta|\theta^\circ) = 0 \quad (A.8)
\]

and

\[
\frac{\partial}{\partial \Sigma_i} Q(\theta|\theta^\circ) = 0 \quad (A.9)
\]
Appendix D

Where using (5.14) in (A.8), one gets:

\[
\frac{\partial}{\partial x_k} \sum_{i=1}^{N} P^c(i|x_k) \left[ -\frac{1}{2} \log |\Sigma| - \frac{1}{2} (x_k - \mu_i)^T \Sigma^{-1} (x_k - \mu_i) - \frac{d}{2} \log(2\pi) \right] + \frac{\partial}{\partial \mu_i} \left\{ \sum_{k=1}^{N} P^c(i|x_k) \left[ -\frac{1}{2} \log |\Sigma| - \frac{1}{2} (z_k - \mu_i)^T \Sigma_i^{-1} (z_k - \mu_i) - \frac{d}{2} \log(2\pi) \right] \right\} = 0
\]

Taking the derivatives and substituting \( \alpha_i^+ \), \( \mu_i^+ \) and \( \Sigma_i^+ \) for \( \alpha_i \), \( \mu_i \) and \( \Sigma_i \) respectively, we get:

\[
\sum_{k=1}^{N} P^c(i|x_k) \Sigma_i^{-1}(x_k - \mu_i^+) + \sum_{k=1}^{N} P^c(i|x_k) \Sigma_i^{-1}(z_k - \mu_i^+) = 0
\]

Multiplying both sides from left by \( \Sigma_i^+ \) and rearranging the following iterative equation is obtained:

\[
\mu_i^+ = \frac{\sum_{k=1}^{N} P^c(i|x_k) \cdot z_k + \sum_{k=1}^{N} P^c(i|x_k) \cdot x_k}{\sum_{k=1}^{N} P^c(i|x_k) + \sum_{k=1}^{N} P^c(i|x_k)}
\]

Finally, in order to obtain the iterative equations for covariance matrices, we substitute (5.14) in (A.9) and use the matrix derivative formulas in [9] to get the following:
\[
\sum_{k=1}^{N} P^c(\text{i}x_k) \cdot \left[ -\Sigma_i^{+1} + \frac{1}{2} D(\Sigma_i^{+1}) + \Sigma_i^{+1}(x_k - \mu_i^+)(x_k - \mu_i^+)^T \Sigma_i^{+1} - \frac{1}{2} D(\Sigma_i^{+1})(x_k - \mu_i^+)(x_k - \mu_i^+)^T \Sigma_i^{+1} \right] + \\
\sum_{j=1}^{N_i} P_j^c(\text{i}lz_{jk}) \cdot \left[ -\Sigma_i^{+1} + \frac{1}{2} D(\Sigma_i^{+1}) + \Sigma_i^{+1}(z_{jk} - \mu_i^+)(z_{jk} - \mu_i^+)^T \Sigma_i^{+1} - \frac{1}{2} D(\Sigma_i^{+1})(z_{jk} - \mu_i^+)(z_{jk} - \mu_i^+)^T \Sigma_i^{+1} \right] = [0]
\]

Where \( D(A) \) is the diagonal matrix which is obtained by replacing all the non-diagonal entries of matrix \( A \) with zero. The above equation can be reduced to the following:

\[
\sum_{k=1}^{N} P^c(\text{i}x_k) \cdot \left[ -\Sigma_i^{+1} + \Sigma_i^{+1}(x_k - \mu_i^+)(x_k - \mu_i^+)^T \Sigma_i^{+1} \right] + \\
\sum_{j=1}^{N_i} P_j^c(\text{i}lz_{jk}) \cdot \left[ -\Sigma_i^{+1} + \Sigma_i^{+1}(z_{jk} - \mu_i^+)(z_{jk} - \mu_i^+)^T \Sigma_i^{+1} \right] = [0]
\]

By multiplying the above from left and right by \( \Sigma_i^+ \) and rearranging the following iterative equation is obtained:

\[
\Sigma_i^+ = \frac{\sum_{j=1}^{N_i} P_j^c(\text{i}lz_{jk})(z_{jk} - \mu_i^+)(z_{jk} - \mu_i^+)^T + \sum_{k=1}^{N} P^c(\text{i}x_k)(x_k - \mu_i^+)(x_k - \mu_i^+)^T}{\sum_{j=1}^{N_i} P_j^c(\text{i}lz_{jk}) + \sum_{k=1}^{N} P^c(\text{i}x_k)}
\]
Appendix E

The source codes of the programs used in this thesis are available upon request from B.M. Shahshahani, or Dr. David Landgrebe, School of Electrical Engineering, Purdue University, West Lafayette, IN 47907-1285.