5-1-1988

Statistical Classifier Design and Evaluation

Keinosuke Fukunaga
Purdue University

Raymond Reynolds Hayes
Purdue University

FILE

Statistical Classifier Design and Evaluation

Keinosuke Fukunaga
Raymond Reynolds Hayes

TR-EE 88-19
May 1988

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

This work was partly supported by the National Science Foundation under Grant ESC-85137020 and by IBM under the Resident Study Program.

STATISTICAL CLASSIFIER

DESIGN AND EVALUATION

by

Keinosuke Fukunaga

and

Raymond R. Hayes

TR-EE 88-19

May 1988

School of Electrical Engineering

Purdue University

West Lafayette, Indiana 47907

This work was partly supported by the National Science Foundation under Grant ESC-8513720 and by IBM under the Resident Study Program.

TABLE OF CONTENTS

Page

APPENDICES

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

ABSTRACT

Hayes, Raymond Reynolds. Ph.D., Purdue University. May 1988. Statistical Classifier Design and Evaluation. Major Professor: Keinosuke Fukunaga.

This thesis is concerned with the design and evaluation of statistical classifiers. This problem has an optimal solution with a priori knowledge of the underlying probability distributions. Here, we examine the expected performance of parametric classifiers designed from a finite set of training samples and tested under various conditions. By investigating the statistical properties of the performance bias when tested on the true distributions, we have isolated the effects of the individual design components (i.e., the number of training samples, the dimensionality, and the parameters of the underlying distributions). These results have allowed us to establish a firm theoretical foundation for new design guidelines and to develop an empirical approach for estimating the asymptotic performance.

Investigation of the statistical properties of the performance bias when tested on finite sample sets has allowed us to pinpoint the effects of individual design samples, the relationship between the sizes of the design and test sets, and the effects of a dependency between these sets. This, in turn, leads to a better understanding of how a single training set can be used most efficiently. In addition, we have developed a theoretical framework for the analysis and comparison of various performance evaluation procedures.

Nonparametric and one-class classifiers are also considered. The reduced Parzen classifier, a nonparametric classifier which combines the error estimation capabilities of the Parzen density estimate with the computational feasibility of parametric classifiers, is presented. Also, the effect of the distance-space mapping in a one-class classifier is discussed through the approximation of the performance of a distance-ranking procedure.

CHAPTER 1
INTRODUCTION

1.1 Problem Statement

In the formulation of the statistical pattern recognition problem, multidimensional observations of a random event are assumed to have been generated from a set of underlying probability densities, each of which represents an event class. If one could accurately identify the underlying densities and determine from which an unknown observation came, one could classify the observation. A priori knowledge of the densities makes the problem relatively easy. However, the designer is usually just presented with a limited set of preclassified observations (training samples) from which the underlying structure of the problem must somehow be inferred.

The design of a pattern recognition system involves a number of steps: feature extraction, error estimation, classifier design, and classifier evaluation. Even though the measurement process determines the dimensionality of the observations, classification can take place in any space. Feature extraction involves heuristically or mathematically obtaining a reduced set of features which reflect the characteristics of the original measurements. Each observation can be mapped into a feature vector which is then used for classification. Classifier performance is bounded by the overlap of the underlying densities (the Bayes error). Once a set of features is chosen, the Bayes error in that space measures the maximum separability of the classes and provides a guideline for the evaluation of the classifier performance. Classifier design deals with the identification of the densities and the development of a discrimination rule. A common practice is to assume that the densities are uni-modal Gaussian, estimate the appropriate parameters from the training samples, use Bayes' rule to find the a posteriori class probabilities, and then take the natural log, generating a quadratic expression which is compared to a threshold (the quadratic classifier). Once the classifier has been designed, its expected performance in the field must be estimated and compared to the theoretical bounds (classifier evaluation).

When the assumptions made in the design stage are correct and a very large number of training samples are available, the resulting classifier will probably be capable of near-optimal performance. However, if a large number of training samples are not available, the density parameter estimates will not be accurate and classifier performance will be degraded. Furthermore, the same limited set of training samples used to design the classifier must also be used in its evaluation. This usually leads to an optimistic estimate of the classifier performance.

When the assumptions made in the design stage are not correct, the resulting parametric classifier will not provide a satisfactory level of performance. Currently, non-parametric techniques, such as k-nearest neighbor (k-NN) and the Parzen density estimate, are being used successfully for Bayes error estimation. These techniques can be re-cast as classifiers, but their dependence on a large number of training samples makes their implementation computationally impractical.

Finally, at times, because of the dynamic nature of the classification environment, training samples from all of the classes are not available. For example, suppose one wanted to be able to recognize the radar return of a tank located in a field. The characteristics of the tank will remain constant, but the field might contain grass, trees, rocks, or snow, all of which fall into the non-tank class. This one-class scenario is more of a detection problem than a classification problem and a slightly different approach must be taken.

This thesis focuses on the classifier design and evaluation concerns mentioned above. Even though they are theoretical in nature, the techniques developed and results presented should be useful in the solution of a number of practical problems.

1.2 Thesis Organization

Chapter 2 of this thesis deals with the effect of finite training sample size on parameter estimates and their subsequent use in functions. General and parameter-specific expressions for the expected bias and variance of the functions are derived. These expressions are then applied to the Bhattacharyya distance and to a new expression which characterizes the performance for the linear and quadratic classifiers, providing valuable insight into the relationship between the number of features and the number of training samples. Also, the functional form of these expressions allowed the development of an empirical approach which will enable asymptotic performance to be accurately estimated using a very small number of samples.

In Chapter 3, the expression for expected classifier performance derived in Chapter 2 is applied to a series of test procedures designed to compensate for the fact that only one set of training samples is available. For the holdout method, the roles of the independent design and test sets are identified. For the resubstitution and leave-one-out methods, the relationship between dependent design and test sets is investigated. Also, the statistical properties of the bootstrap re-sampling technique are analyzed.

Chapter 4 investigates the use of a new non-parametric classifier based on the error estimation capabilities of the Parzen density estimate. We develop an algorithm to select a given number of representative samples whose Parzen density estimate closely matches that of the entire sample set. Using these representatives, a piecewise quadratic classifier which provides nearly optimal performance is designed.

In Chapter 5, an approximation for the acquisition probability for a minimum distance one-class classifier is derived. In the original development of the classifier, it was shown that the acquisition probability is dependent upon the operating characteristics in the distance space, the number of targets detected, and the number of other objects detected. An approximate expression replaces the operating characteristics curve by a single point.

Chapter 6 gives a summary of the major contributions of this thesis and provides recommendations for further research.

CHAPTER 2
EFFECTS OF SAMPLE SIZE IN CLASSIFIER DESIGN

2.1 Introduction

In practical pattern recognition problems, the parameters of the underlying distributions are unknown and the number of training samples available frequently is small. The size of this set of samples, relative to the number of features used, determines the accuracy of the parameter estimates and the similarity between the sample set and the true distribution.

In this chapter, we will look at the effect of sample size on functions of the distributions' parameters. By viewing the estimated parameters as random variables, the expected value of a criterion can be computed by taking the expectation of the parameters over all possible N-size sets of training samples. This leads to a general expression for the expected bias and variance of the function, isolating the effects of functional form from the underlying distribution.

Pattern recognition research has considered various questions concerning the relationship between the limited size of the training set, the number of features, and the estimation of some performance criterion. A number of authors, including El-Sheikh and Wacker [1], have investigated the optimal number of features for a given finite design sample size in order to combat the "peaking phenomenon," the apparent loss of classifiability which accompaines an increase in the number of featues without an increase in the number of training samples. An excellent review of work done in this area is presented in Jain and Chandrasekaran [2]. Another group of authors has looked into the effect of the ratio of training sample size to feature set dimensionality on the expected performance of an empirically-designed classifier used on the true test distribution. In [3], Raudys and Pikelis catalog the development of a number of approximate expressions for the expected performance of the linear classifier and an exact expression for the quadratic classifier. Asymptotic expansions for the quadratic classifier have also been developed by Han [4] and McLachlan [5]. Unfortunately, these expressions are too complex to provide valuable insight and their accuracy has not been experimentally verified. Thus, the relationship between sample size and dimensionality has been inferred through simulation (eg., [3] and [6]), the investigation of related criteria (e.g., [7] for Bhattacharyya distance and [8] for divergence), and a look at the performance of these classifiers tested on the design set [9].

By applying our general expression to the Bhattacharyya distance and the classifier error equation, we have developed a useful framework for the analysis of classifier performance, design, and testing procedures. This provides valuable insight into the relationship between dimensionality and sample size and the importance of mean and covariance shifts in measuring separability. Also, we have developed an empirical approach which will allow the designer to estimate the asymptotic performance of a particular type of classifier. This can be used to evaluate trade-offs in classifier complexity and performance, and to determine the ratio of design samples to dimensionality needed for a particular performance level.

2.2 Bias and Variance Expressions

2.2.1 General Formulation

Let us consider the problem of estimating $f(y_1,,y_L)$ by $f(\hat{y}_1,,\hat{y}_L)$ where f is a given function, y_i's are the true parameter values and $\hat{y}_i$'s are their estimates. In this section, we will derive expressions for the expected value and variance of $f(\hat{y}_1,,\hat{y}_L)$, and propose a new method to estimate $f(y_1,,y_L)$.

Assuming that the deviation of $\hat{y}_i$ from y_i is small, $f(\hat{Y})$ can be expanded by a Taylor series up to the second-order terms as

$$\hat{f} \triangleq f(\hat{Y}) \cong f(Y) + \frac{\partial f^T}{\partial Y} \Delta Y + \frac{1}{2} \operatorname{tr} \left(\frac{\partial^2 f}{\partial Y^2} \Delta Y \Delta Y^T \right) \tag{2.1}$$

where $Y = [y_1 y_L]^T$ and $\hat{Y} = [\hat{y}_1 \hat{y}_L]^T$ are the column vectors of the true parameters and their estimates, respectively. $\Delta Y = \hat{Y} - Y$, V^T indicates the transpose of the vector V, and trA is the trace of a matrix A.

If the estimates are unbiased,

$$E\{\Delta Y\} = 0 \tag{2.2}$$

and subsequently

$$E\{\hat{f}\} \cong f + \frac{1}{2} \operatorname{tr} \left(\frac{\partial^2 f}{\partial Y^2} E\{\Delta Y \Delta Y^T\} \right) \tag{2.3}$$

Similarly, the variance of $\hat{f}$ can be derived as

$$\begin{aligned} \operatorname{Var}\{\hat{f}\} &\cong E\left\{ \left[\frac{\partial f^T}{\partial Y} \Delta Y + \frac{1}{2} \operatorname{tr} \left(\frac{\partial^2 f}{\partial Y^2} \Delta Y \Delta Y^T \right) - \frac{1}{2} \operatorname{tr} \left(\frac{\partial^2 f}{\partial Y^2} E\{\Delta Y \Delta Y^T\} \right) \right]^2 \right\} \\ &\cong E\left\{ \left(\frac{\partial f^T}{\partial Y} \Delta Y \right)^2 \right\} \\ &= \frac{\partial f^T}{\partial Y} E\{\Delta Y \Delta Y^T\} \frac{\partial f}{\partial Y} \end{aligned} \tag{2.4}$$

where the approximation from the first line to the second line was made by discarding terms higher than second-order.

Eq. (2.3) shows that $\hat{f}$ is a biased estimate in general and that the bias depends on $\partial^2 f/\partial Y^2$ and $E\{\Delta Y \Delta Y^T\}$, where $\partial^2 f/\partial Y^2$ is determined by the functional form of f and $E\{\Delta Y \Delta Y^T\}$ is determined by $p(\hat{Y})$, the density function of $\hat{Y}$, and N, the number of samples used to compute $\hat{Y}$. Likewise, the variance depends on $\partial f/\partial Y$ and $E\{\Delta Y \Delta Y^T\}$.

For many estimators, the effects of $p(\hat{Y})$ and N on $E\{\Delta Y \Delta Y^T\}$ can be separated as

$$E\{\Delta Y \Delta Y^T\} = g(N)\, K(p(\hat{Y})) \tag{2.5}$$

where the scalar g and the matrix K are functions determined by how $\hat{Y}$ is computed. Substituting (2.5) into (2.3),

$$E\{\hat{f}\} \cong f + c\, g(N) \tag{2.6}$$

where $c = \frac{1}{2} \operatorname{tr} (\partial^2 f/\partial Y^2\, K(p(\hat{Y})))$ is independent of N and treated as a constant determined by a given underlying problem. This leads to the following procedure to estimate f.

1) Change the sample size N as $N_1, N_2,, N_\ell$. For each N_i, compute $\hat{Y}$ and subsequently $\hat{f}$ empirically. Repeat the experiment M times, and approximate $E\{\hat{f}\}$ with the sample mean of the M experimental results.

2) Plot these empirical points $E\{\hat{f}\}$ vs. $g(N)$. Then, find the line best fitted to these points. The slope of this line is c and the y-intercept is the improved estimate of f. There are many possible ways of selecting a line. The standard procedure would be the minimum mean-square error approach.

2.2.2 Parametric Formulation

In pattern recognition, most of the expressions we would like to estimate are functions of the expected vectors and covariance matrices. In this section, we will show how the general discussion of the previous section can be applied to this particular family of parameters.

Assume that N samples are drawn from each of two n-dimensional Gaussian distributions with their expected vectors and covariance matrices given by

$$\begin{aligned} M_1 &= 0 \quad , \quad \textstyle\sum_1 = I \\ M_2 &= M \quad , \quad \textstyle\sum_2 = \Lambda \end{aligned} \tag{2.7}$$

Without loss of generality, any two covariance matrices can be simultaneously diagonalized to I and Λ, and a coordinate shift can bring the expected vector of one class to zero.

M_i and $\sum_i$ can be estimated without bias by the sample mean and sample covariance

$$\hat{M}_i = \frac{1}{N} \sum_{j=1}^{N} X_j^{(i)}$$

$$\textstyle\sum_i = \frac{1}{N-1} \sum_{j=1}^{N} \left(X_j^{(i)} - \hat{M}_i\right) \left(X_j^{(i)} - \hat{M}_i\right)^T \tag{2.8}$$

where $X_j^{(i)}$ is the jth sample vector from class i. Thus, the parameter vector $\hat{Y}$ of (2.1) consists of $2(n+n^2)$ components

$$\hat{Y} = \left[\hat{m}_1^{(1)} \hat{m}_n^{(1)} \; \hat{m}_1^{(2)} \hat{m}_n^{(2)} \; \hat{\alpha}_{11}^{(1)} \hat{\alpha}_{nn}^{(1)} \; \hat{\alpha}_{11}^{(2)} \hat{\alpha}_{nn}^{(2)}\right]^T \tag{2.9}$$

where $\hat{m}_i^{(r)}$ is the ith component of $\hat{M}_r$, and $\hat{\alpha}_{ij}^{(r)}$ is the ith row and jth column component of $\sum_r$.

The random variables of (2.9) satisfy the following statistical properties, where $\Delta m_i^{(r)} = \hat{m}_i^{(r)} - m_i^{(r)}$ and $\Delta\alpha_{ij}^{(r)} = \hat{\alpha}_{ij}^{(r)} - \alpha_{ij}^{(r)}$:

1) The sample mean and covariance are unbiased:

$$E\{\Delta m_i^{(r)}\} = 0 \quad , \quad E\{\Delta \alpha_{ij}^{(r)}\} = 0 \tag{2.10}$$

2) Samples from different classes are independent:

$$E\{\Delta m_i^{(1)} \Delta m_j^{(2)}\} = E\{\Delta m_i^{(1)}\}\, E\{\Delta m_j^{(2)}\} = 0$$

$$E\{\Delta \alpha_{ij}^{(1)} \Delta \alpha_{k\ell}^{(2)}\} = E\{\Delta \alpha_{ij}^{(1)}\}\, E\{\Delta \alpha_{k\ell}^{(2)}\} = 0$$

$$E\{\Delta m_i^{(r)} \Delta \alpha_{k\ell}^{(s)}\} = E\{\Delta m_i^{(r)}\}\, E\{\Delta \alpha_{k\ell}^{(s)}\} = 0 \quad \text{for } r \neq s \tag{2.11}$$

3) Diagonal $\sum_1$ and $\sum_2$ cause the mean estimate covariances to be diagonal:

$$E\{\Delta m_i^{(r)} \Delta m_j^{(r)}\} = 0 \quad \text{for } i \neq j$$

$$E\{\Delta m_i^{(1)2}\} = \frac{1}{N}$$

$$E\{\Delta m_i^{(2)2}\} = \frac{\lambda_i}{N} \tag{2.12}$$

where λ_i is the ith diagonal component of Λ.

4) The third-order central moments of a Gaussian distribution are zero:

$$E\{\Delta m_i^{(r)} \Delta \alpha_{k\ell}^{(r)}\} = 0 \tag{2.13}$$

5) The fourth-order central moments of a Gaussian distribution are known:

$$E\left\{\Delta \alpha_{ij}^{(1)}\ \Delta \alpha_{k\ell}^{(1)}\right\} = \begin{cases} \dfrac{1}{N} & \text{for } (i{\neq}j, i{=}k, j{=}\ell\,) \text{ or } (i{\neq}j, i{=}\ell\,, j{=}k) \\ \dfrac{2}{N-1} \cong \dfrac{2}{N} & \text{for } i{=}j{=}k{=}\ell \\ 0 & \text{otherwise} \end{cases}$$

$$E\left\{\Delta\alpha_{ij}^{(2)}\Delta\alpha_{k\ell}^{(2)}\right\} = \begin{cases} \dfrac{\lambda_i\lambda_j}{N} & \text{for } (i\neq j, i=k, j=\ell) \text{ or } (i\neq j, i=\ell, j=k) \\ \dfrac{2\lambda_i^2}{N-1} \cong \dfrac{2\lambda_i^2}{N} & \text{for } i=j=k=\ell \\ 0 & \text{otherwise} \end{cases} \tag{2.14}$$

Note that in the equal index cases of (2.14) N-1 is replaced by N for simplicity.

Substituting (2.9) through (2.14) into (2.3), the bias term of the estimate, $E\{\Delta f\} = E\{\hat{f}\} - f$, becomes

$$E\{\Delta f\} \cong \frac{1}{2}\,\mathrm{tr}\left(\frac{\partial^2 f}{\partial Y^2}\,E\{\Delta Y \Delta Y^T\}\right) = \frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L}\frac{\partial^2 f}{\partial y_i \partial y_j}\,E\{\Delta y_i\, \Delta y_j\}$$

$$= \frac{1}{2}\sum_{r=1}^{2}\left[\sum_{i=1}^{n}\frac{\partial^2 f}{\partial m_i^{(r)2}}\,E\{\Delta m_i^{(r)2}\} + \sum_{\substack{i=1\\ i\neq j}}^{n}\sum_{j=1}^{n}\frac{\partial^2 f}{\partial\alpha_{ij}^{(r)}\partial\alpha_{ij}^{(r)}}\,E\{\Delta\alpha_{ij}^{(r)}\Delta\alpha_{ij}^{(r)}\}\right.$$

$$\left. + \sum_{\substack{i=1\\ i\neq j}}^{n}\sum_{j=1}^{n}\frac{\partial^2 f}{\partial\alpha_{ij}^{(r)}\partial\alpha_{ji}^{(r)}}\,E\{\Delta\alpha_{ij}^{(r)}\Delta\alpha_{ji}^{(r)}\} + \sum_{i=1}^{n}\frac{\partial^2 f}{\partial\alpha_{ii}^{(r)2}}\,E\{\Delta\alpha_{ii}^{(r)2}\}\right]$$

$$\cong \frac{1}{2N}\left[\sum_{i=1}^{n}\frac{\partial^2 f}{\partial m_i^{(1)2}} + \sum_{i=1}^{n}\frac{\partial^2 f}{\partial m_i^{(2)2}}\,\lambda_i\right.$$

$$+ \sum_{\substack{i=1\\ i\neq j}}^{n}\sum_{j=1}^{n}\left(\frac{\partial^2 f}{\partial\alpha_{ij}^{(1)}\partial\alpha_{ij}^{(1)}} + \frac{\partial^2 f}{\partial\alpha_{ij}^{(1)}\partial\alpha_{ji}^{(1)}}\right) + \sum_{i=1}^{n}\frac{\partial^2 f}{\partial\alpha_{ii}^{(1)2}}\,2$$

$$\left. + \sum_{\substack{i=1\\ i\neq j}}^{n}\sum_{j=1}^{n}\left(\frac{\partial^2 f}{\partial\alpha_{ij}^{(2)}\partial\alpha_{ij}^{(2)}} + \frac{\partial^2 f}{\partial\alpha_{ij}^{(2)}\partial\alpha_{ji}^{(2)}}\right)\lambda_i\lambda_j + \sum_{i=1}^{n}\frac{\partial^2 f}{\partial\alpha_{ii}^{(2)2}}\,2\lambda_i^2\right] \tag{2.15}$$

Note that the effect of N is successfully separated, and that g(N) of (2.5) becomes 1/N. This is true for any functional form of f, provided f is a function of the expected vectors and covariance matrices of two Gaussian distributions. This conclusion can be extended to non-Gaussian cases in which (2.13) is satisfied and $E\{\Delta\alpha_{ij}^{(r)}\Delta\alpha_{k\ell}^{(r)}\}$ of (2.14) is proportional to 1/N.

Similarly, the variance can be computed from (2.4), resulting in

$$\mathrm{Var}\{\hat{f}\} \cong \frac{1}{N}\left[\sum_{i=1}^{n}\left(\frac{\partial f}{\partial m_i^{(1)}}\right)^2 + \sum_{i=1}^{n}\left(\frac{\partial f}{\partial m_i^{(2)}}\right)^2 \lambda_i\right.$$

$$+ \sum_{i=1}^{n}\sum_{\substack{j=1\\ i\neq j}}^{n}\left\{\left(\frac{\partial f}{\partial \alpha_{ij}^{(1)}}\right)^2 + \frac{\partial f}{\partial \alpha_{ij}^{(1)}}\frac{\partial f}{\partial \alpha_{ji}^{(1)}}\right\} + \sum_{i=1}^{n}\left(\frac{\partial f}{\partial \alpha_{ii}^{(1)}}\right)^2 2$$

$$\left. + \sum_{i=1}^{n}\sum_{\substack{j=1\\ i\neq j}}^{n}\left\{\left(\frac{\partial f}{\partial \alpha_{ij}^{(2)}}\right)^2 + \frac{\partial f}{\partial \alpha_{ij}^{(2)}}\frac{\partial f}{\partial \alpha_{ji}^{(2)}}\right\}\lambda_i\lambda_j + \sum_{i=1}^{n}\left(\frac{\partial f}{\partial \alpha_{ii}^{(2)}}\right)^2 2\lambda_i^2\right] \qquad (2.16)$$

Note that, in order to calculate the bias and variance, we only need to compute $\partial f/\partial m_i^{(r)}$, $\partial f/\partial \alpha_{ij}^{(r)}$, $\partial^2 f/\partial m_i^{(r)2}$, $\partial^2 f/\partial \alpha_{ij}^{(r)}\partial \alpha_{ij}^{(r)}$ and $\partial^2 f/\partial \alpha_{ij}^{(r)}\partial \alpha_{ji}^{(r)}$ for r=1,2.

2.3 Bhattacharyya Distance Between Two Distributions

A popular measure of similarity between two distributions is the Bhattacharyya distance [10]

$$B = \frac{1}{8}(M_2 - M_1)^T\left[\frac{\Sigma_1+\Sigma_2}{2}\right]^{-1}(M_2 - M_1) + \frac{1}{2}\ln\frac{\left|\frac{\Sigma_1+\Sigma_2}{2}\right|}{\sqrt{|\Sigma_1|}\sqrt{|\Sigma_2|}} \qquad (2.17)$$

Since B is a function of M_1, M_2, Σ_1 and Σ_2, it is a member of the family of functions discussed previously.

If two distributions are Gaussian, the Bhattacharyya distance gives an upper bound of the Bayes error, ϵ^*,

$$\epsilon^* \leqq \epsilon_u = \sqrt{P_1P_2}e^{-B} \qquad (2.18)$$

where P_i is the a priori probability of class i. The first and second terms of (2.17), B_1 and B_2, measure the difference between the two distributions due to the mean and covariance shifts respectively.

When $\hat{M}_i$ and $\hat{\Sigma}_i$ of (2.8) are used to compute B, the resulting $\hat{B}$ differs from its true value. The bias and variance of $\hat{B}$ can be obtained using (2.15) and (2.16).

2.3.1 First Term B_1

Since $B_1 = \frac{1}{8}\sum_{i=1}^{n}\frac{2}{1+\lambda_i}(m_i^{(2)} - m_i^{(1)})^2$, $\partial B_1/\partial m_i^{(r)}$ and $\partial^2 B_1/\partial m_i^{(r)2}$ can be easily obtained as

$$\frac{\partial B_1}{\partial m_i^{(1)}} = -\frac{m_i^{(2)}-m_i^{(1)}}{2(1+\lambda_i)} \quad , \quad \frac{\partial B_1}{\partial m_i^{(2)}} = \frac{m_i^{(2)}-m_i^{(1)}}{2(1+\lambda_i)} \tag{2.19}$$

$$\frac{\partial^2 B_1}{\partial m_i^{(1)2}} = \frac{\partial^2 B_1}{\partial m_i^{(2)2}} = \frac{1}{2(1+\lambda_i)} \tag{2.20}$$

The computation of $\partial B_1/\partial\alpha_{ij}^{(r)}$, $\partial^2 B_1/\partial\alpha_{ij}^{(r)}\partial\alpha_{ij}^{(r)}$ and $\partial^2 B_1/\partial\alpha_{ij}^{(r)}\partial\alpha_{ji}^{(r)}$ are more complex and presented in Appendix A. The results are

$$\frac{\partial B_1}{\partial\alpha_{ij}^{(1)}} = \frac{\partial B_1}{\partial\alpha_{ij}^{(2)}} = -\frac{m_i m_j}{4(1+\lambda_i)(1+\lambda_j)} \tag{2.21}$$

$$\frac{\partial^2 B_1}{\partial\alpha_{ij}^{(1)}\alpha_{ij}^{(1)}} = \frac{\partial^2 B_1}{\partial\alpha_{ij}^{(2)}\partial\alpha_{ij}^{(2)}} = 0 \quad \text{for } i\neq j \tag{2.22}$$

$$\frac{\partial^2 B_1}{\partial\alpha_{ij}^{(1)}\alpha_{ji}^{(1)}} = \frac{\partial^2 B_1}{\partial\alpha_{ij}^{(2)}\partial\alpha_{ji}^{(2)}} = \frac{1}{4(1+\lambda_i)(1+\lambda_j)}\left[\frac{m_i^2}{1+\lambda_i} + \frac{m_j^2}{1+\lambda_j}\right] \tag{2.23}$$

where $m_i = m_i^{(2)} - m_i^{(1)}$.

Substituting (2.19) through (2.23) into (2.15) and (2.16),

$$E\{\Delta B_1\} \cong \frac{1}{4N}\left[n + \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{m_j^2(1+\lambda_i\lambda_j)}{(1+\lambda_j)^2(1+\lambda_i)} + \sum_{i=1}^{n}\frac{m_i^2(1+\lambda_i^2)}{(1+\lambda_i)^3}\right] \tag{2.24}$$

$$\mathrm{Var}\{\hat{B}_1\} \cong \frac{1}{4N}\left[\sum_{i=1}^{n}\frac{m_i^2}{1+\lambda_i} + \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{m_i^2 m_j^2(1+\lambda_i\lambda_j)}{2(1+\lambda_i)^2(1+\lambda_j)^2}\right] \tag{2.25}$$

2.3.2 Second Term B_2

Similarly, the partial derivatives for B_2 are derived in Appendix B. They are listed as follows:

$$\frac{\partial B_2}{\partial m_i^{(r)}} = 0 \quad \text{and} \quad \frac{\partial^2 B_2}{\partial m_i^{(r)2}} = 0 \quad \text{for } r=1,2 \tag{2.26}$$

$$\frac{\partial B_2}{\partial \alpha_{ij}^{(1)}} = \frac{\delta_{ij}}{2(1+\lambda_i)} - \frac{\delta_{ij}}{4} \tag{2.27}$$

$$\frac{\partial B_2}{\partial \alpha_{ij}^{(2)}} = \frac{\delta_{ij}}{2(1+\lambda_i)} - \frac{\delta_{ij}}{4\lambda_i} \tag{2.28}$$

$$\frac{\partial^2 B_2}{\partial \alpha_{ij}^{(1)} \partial \alpha_{ij}^{(1)}} = \frac{1}{4} - \frac{1}{2(1+\lambda_i)(1+\lambda_j)} \tag{2.29}$$

$$\frac{\partial^2 B_2}{\partial \alpha_{ij}^{(2)} \partial \alpha_{ij}^{(2)}} = \frac{1}{4\lambda_i\lambda_j} - \frac{1}{2(1+\lambda_i)(1+\lambda_j)} \tag{2.30}$$

$$\frac{\partial^2 B_2}{\partial \alpha_{ij}^{(1)} \partial \alpha_{ji}^{(1)}} = \frac{\partial^2 B_2}{\partial \alpha_{ij}^{(2)} \partial \alpha_{ji}^{(2)}} = 0 \quad \text{for } i \neq j \tag{2.31}$$

Substituting (2.26) through (2.31) into (2.15) and (2.16),

$$E\{\Delta B_2\} \cong \frac{1}{4N}\left[n(n+1) - \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{1+\lambda_i\lambda_j}{(1+\lambda_i)(1+\lambda_j)} - \sum_{i=1}^{n} \frac{1+\lambda_i^2}{(1+\lambda_i)^2} \right] \tag{2.32}$$

$$\mathrm{Var}\{\hat{B}_2\} \cong \frac{1}{2N} \sum_{i=1}^{n} \left[\left(\frac{1}{1+\lambda_i} - \frac{1}{2} \right)^2 + \left(\frac{1}{1+\lambda_i} - \frac{1}{2\lambda_i} \right)^2 \lambda_i^2 \right] \tag{2.33}$$

2.3.3 Discussions and Experimental Verification

Table 2.1 shows the dependence of $E\{\Delta B_1\}$ and $E\{\Delta B_2\}$ on n and k (=N/n) for three different cases. In Case 1, samples from both classes are drawn from the same source N(O,I), a Gaussian distribution with zero mean and identity covariance matrix. In Case 2, the two distributions share a covariance matrix but differ in the means. In Case 3, the means are the same, but the covariances are different. As Table 2.1 indicates, for all three cases, $E\{\Delta B_1\}$ is proportional to 1/k while $E\{\Delta B_2\}$ is proportional to (n+1)/k. Also, note that $E\{\Delta B_1\}$ is the same for Cases 1 and 3 because the sources have the same mean. Similarly, $E\{\Delta B_2\}$ is the same for Cases 1 and

Table 2.1 Sample bias expressions for the Bhattacharyya distance.

	Case 1 N(0,I) N(0,I)	Case 2 N(0,I) N(M,I)	Case 3 N(0,I) N(0,4I)
m_i	$m_i = 0$	$m_1 = 2.56$ $m_i = 0\ (i \neq 1)$	$m_i = 0$
λ_i	$\lambda_i = 1$	$\lambda_i = 1$	$\lambda_i = 4$
B_1	0	0.82	0
B_2	0	0	0.11 n
ϵ^*	50%	10%	Depends on n
$E\{\Delta B_1\}$	$\frac{0.25}{k}$	$\frac{0.35}{k}$	$\frac{0.25}{k}$
$E\{\Delta B_2\}$	$0.125\frac{n+1}{k}$	$0.125\frac{n+1}{k}$	$0.08\frac{n+1}{k}$

2 because the sources share a covariance matrix.

Since the trend is the same for all three cases, let us study Case 1 closely. Table 2.1 demonstrates that in high dimensional space ($n >> 1$) the distortion due to the covariance estimate ($E\{\Delta B_2\} = 0.125\ (n+1)/k$) dominates that caused by the mean estimate ($E\{\Delta B_1\} = 0.25/k$). Also, since $E\{\Delta B_2\} = 0.125\ (n+1)/k$, an increasingly large value of k is required to maintain a constant value of $E\{\hat{B}\}(= E\{\hat{B}_1\} + E\{\hat{B}_2\})$ as the dimensionality increases. For example, Table 2.2 shows the value of k required to keep the value of $E\{\hat{B}\}$ less than 0.223. The true Bayes error for this case is 50%, and $E\{\hat{B}\} = 0.223$ gives an upper bound of 40% using (2.18). Only 16 samples (3.9 times the dimensionality) are needed to achieve $E\{\hat{B}\} = 0.223$ in a 4-dimensional space, while 9396 samples (73.4 times the dimensionality) are needed in a 128 dimensional space. This result is sharply contrasted with the common belief that a fixed multiple of the dimensionality such as 5 or 10 could be used to determine the sample size.

Since the theoretical results of (2.24) and (2.32) for bias and (2.25) and (2.33) for variance are approximations, we have conducted three sets of experiments to verify these results. The first two cases are Cases 2 and 3 of Table 2.1, while the third, which will be called Case 4, uses both mean and covariance differences. Case 4 uses an 8-dimensional Gaussian data set taken from [6] with a Bayes error of 1.9%, and λ_i's and m_i's listed in Table 2.3.

For Cases 2 and 3, the dimensionality ranged from 4 to 64 in powers of 2, and k was selected as 3, 5, 10, 15, 20, 30, 40, and 50. $N(= nk)$ samples were generated from each class according to the given mean and covariance, and $\hat{B}_1$ and $\hat{B}_2$ were computed. This procedure was repeated 10 times independently, and the mean and standard deviation were computed. Tables 2.4, 2.5, and 2.6 present a comparison of the theoretical predictions (first lines) and the means of the 10 trials (second lines) for Cases 2, 3, and 4 respectively. These tables show that the theoretical predictions of the biases match the experimental results very closely.

The third lines of Tables 2.4 and 2.5 present the standard deviations of the 10 trials. Table 2.7 shows the theoretical predictions computed from (2.25) and (2.33) for $\hat{B}_1$ of Case 2 and $\hat{B}_2$ of Case 3. Again the theoretical predictions match the experimental results closely. It should be noted that the variances for $\hat{B}_2$ of Case 2 and $\hat{B}_1$ of Case 1 are zero theoretically. This suggests that the variances for these cases come from the Taylor expansion

Table 2.2 Values of k and N required to maintain $E\{\hat{B}\} \leqq 0.223$.

n	4	8	16	32	64	128
k	3.9	6.2	10.7	19.6	39.6	73.4
N=nk	16	50	172	628	2407	9396

Table 2.3 Statistics for Case 4.

i	1	2	3	4	5	6	7	8
λ_i	8.41	12.06	0.12	0.22	1.49	1.77	0.35	2.73
m_i	3.86	3.10	0.84	0.84	1.64	1.08	0.26	0.01

Table 2.4 Biases of $\hat{B}$ for Case 2.

(a) $\hat{B}_1$ $(B_1=0.82)$

			n		
k	4	8	16	32	64
3	1.1101	1.0758	1.0587	1.0502	1.0459
	1.0730	0.9933	1.0502	1.0754	1.0825
	0.4688	0.3791	0.2221	0.1551	0.0955
5	0.9946	0.9740	0.9638	0.9586	0.9561
	1.0941	1.0702	1.0396	0.9659	0.9764
	0.3867	0.2745	0.1542	0.1091	0.0733
10	0.9080	0.8977	0.8926	0.8900	0.8887
	0.9593	0.9277	0.8421	0.9128	0.8911
	0.2240	0.1424	0.1045	0.0720	0.0709
15	0.8791	0.8723	0.8688	0.8671	0.8663
	0.8802	0.8705	0.8909	0.8634	0.8730
	0.1634	0.1493	0.1053	0.0794	0.0493
20	0.8647	0.8595	0.8570	0.8557	0.8551
	0.8778	0.8891	0.8261	0.8685	0.8361
	0.1356	0.1060	0.0929	0.0455	0.0387
30	0.8502	0.8468	0.8451	0.8443	0.8438
	0.7901	0.8477	0.8583	0.8436	0.8373
	0.0702	0.0992	0.0712	0.0361	0.0366
40	0.8430	0.8405	0.8392	0.8385	0.8382
	0.7917	0.8251	0.8578	0.8343	0.8444
	0.0786	0.1118	0.0522	0.0283	0.0271
50	0.8387	0.8366	0.8356	0.8351	0.8348
	0.8524	0.8383	0.8364	0.8301	0.8290
	0.1060	0.0404	0.0515	0.0475	0.0287

(b) $\hat{B}_2$ $(B_2=0)$

			n		
k	4	8	16	32	64
3	0.2083	0.3750	0.7083	1.3750	2.7083
	0.2546	0.4106	0.8930	1.7150	3.2875
	0.0787	0.0653	0.0588	0.0776	0.1083
5	0.1250	0.2250	0.4250	0.8250	1.6250
	0.1133	0.2791	0.5244	0.9252	1.8035
	0.0266	0.0785	0.0581	0.0302	0.0775
10	0.0625	0.1125	0.2125	0.4125	0.8125
	0.0803	0.1179	0.2280	0.4365	0.8578
	0.0339	0.0191	0.0218	0.0279	0.0234
15	0.0417	0.0750	0.1417	0.2750	0.5417
	0.0437	0.0742	0.1416	0.2894	0.5566
	0.0243	0.0146	0.0143	0.0257	0.0170
20	0.0313	0.0563	0.1063	0.2063	0.4063
	0.0389	0.0566	0.1079	0.2099	0.4129
	0.0101	0.0140	0.0132	0.0154	0.0058
30	0.0208	0.0375	0.0708	0.1375	0.2708
	0.0190	0.0344	0.0707	0.1416	0.2777
	0.0063	0.0082	0.0097	0.0098	0.0062
40	0.0156	0.0281	0.0531	0.1031	0.2031
	0.0170	0.0282	0.0561	0.1034	0.2061
	0.0072	0.0084	0.0086	0.0046	0.0063
50	0.0125	0.0225	0.0425	0.0825	0.1625
	0.0102	0.0219	0.0417	0.0831	0.1650
	0.0037	0.0062	0.0041	0.0060	0.0057

(1st line: Theoretical prediction,
2nd line: The mean of 10 trials,
3rd line: The standard deviation of 10 trials)

Table 2.5 Biases of $\hat{B}$ for Case 3.

(a) $\hat{B}_1$ (B_1=0)

	n				
k	4	8	16	32	64
3	0.0833	0.0833	0.0833	0.0833	0.0833
	0.1435	0.1212	0.1051	0.1118	0.1061
	0.0971	0.0633	0.0415	0.0385	0.0160
5	0.0500	0.0500	0.0500	0.0500	0.0500
	0.0489	0.0709	0.0579	0.0545	0.0605
	0.0284	0.0314	0.0141	0.0209	0.0071
10	0.0250	0.0250	0.0250	0.0250	0.0250
	0.0192	0.0267	0.0266	0.0276	0.0262
	0.0151	0.0124	0.0066	0.0079	0.0035
15	0.0167	0.0167	0.0167	0.0167	0.0167
	0.0159	0.0155	0.0207	0.0166	0.0181
	0.0078	0.0049	0.0106	0.0046	0.0036
20	0.0125	0.0125	0.0125	0.0125	0.0125
	0.0135	0.0156	0.0139	0.0120	0.0141
	0.0055	0.0071	0.0036	0.0038	0.0025
30	0.0083	0.0083	0.0083	0.0083	0.0083
	0.0050	0.0097	0.0085	0.0087	0.0085
	0.0037	0.0050	0.0030	0.0014	0.0013
40	0.0063	0.0063	0.0063	0.0063	0.0063
	0.0066	0.0082	0.0056	0.0062	0.0065
	0.0045	0.0050	0.0021	0.0014	0.0010
50	0.0050	0.0050	0.0050	0.0050	0.0050
	0.0042	0.0040	0.0054	0.0049	0.0052
	0.0037	0.0017	0.0015	0.0008	0.0009

(b) $\hat{B}_2$ (B_2=0.11 n)

	n				
k	4	8	16	32	64
3	0.5796	1.1326	2.2385	4.4503	8.8739
	0.7129	1.0732	2.4527	4.7841	9.3263
	0.1447	0.1653	0.2332	0.1893	0.1642
5	0.5263	1.0366	2.0572	4.0983	8.1806
	0.5081	1.0063	2.1341	4.1041	8.4000
	0.1119	0.1546	0.1129	0.0868	0.1209
10	0.4863	0.9646	1.9212	3.8343	7.6606
	0.4901	0.9463	1.9345	3.8014	7.6630
	0.1016	0.0722	0.0759	0.0702	0.1206
15	0.4730	0.9406	1.8758	3.7463	7.4873
	0.5085	0.9675	1.9030	3.7952	7.5133
	0.0686	0.0350	0.0567	0.0306	0.0658
20	0.4663	0.9286	1.8532	3.7023	7.4006
	0.4708	0.9331	1.8277	3.7019	7.4049
	0.0658	0.0686	0.0966	0.0394	0.0672
30	0.4596	0.9166	1.8305	3.6583	7.3139
	0.4478	0.9033	1.8656	3.7053	7.3493
	0.0328	0.0646	0.0411	0.0884	0.0531
40	0.4473	0.9106	1.7886	3.5769	7.1536
	0.4713	0.8937	1.8058	3.6374	7.2596
	0.0444	0.0328	0.0353	0.0563	0.0392
50	0.4543	0.9070	1.8124	3.6231	7.2446
	0.4456	0.8872	1.8116	3.6279	7.2212
	0.0562	0.0506	0.0362	0.0449	0.0610

(1st line: Theoretical prediction,
2nd line: The mean of 10 trials,
3rd line: The standard deviation of 10 trials)

Table 2.6 Biases of $\hat{B}$ for Case 4.

(a) $\hat{B}_1$

	Theoretical	Experimental	
k		Mean	St. Dev.
3	1.6453	1.5056	0.4995
5	1.4951	1.5104	0.1650
10	1.3824	1.3864	0.1997
15	1.3448	1.3365	0.1886
20	1.3261	1.3266	0.1712
30	1.3073	1.2884	0.1136
40	1.2979	1.3104	0.0658
50	1.2923	1.2997	0.0769

(b) $\hat{B}_2$

	Theoretical	Experimental	
k		Mean	St. Dev.
3	1.4431	1.5695	0.2081
5	1.3002	1.2287	0.1446
10	1.1929	1.1638	0.0766
15	1.1572	1.1497	0.0523
20	1.1393	1.1255	0.0539
30	1.1214	1.1005	0.0337
40	1.1125	1.1093	0.0405
50	1.1071	1.1063	0.0276

Table 2.7 Predicted standard deviations.

	$\hat{B}_1$ for Case 2					$\hat{B}_2$ for Case 3
k n	4	8	16	32	64	for all n
3	0.3531	0.2497	0.1765	0.1248	0.0883	0.1732
5	0.2735	0.1934	0.1368	0.0967	0.0684	0.1342
10	0.1934	0.1368	0.0967	0.0684	0.0483	0.0949
15	0.1579	0.1117	0.0790	0.0558	0.0395	0.0775
20	0.1368	0.0967	0.0684	0.0483	0.0342	0.0671
30	0.1117	0.0790	0.0558	0.0395	0.0279	0.0548
40	0.0967	0.0684	0.0483	0.0342	0.0242	0.0474
50	0.0865	0.0612	0.0432	0.0306	0.0216	0.0424

terms higher than second-order and therefore are expected to be smaller than the variances for the other cases. This is confirmed by comparing the variances between $\hat{B}_1$ and $\hat{B}_2$ of Cases 2 and 3. Also, note that the variances of $\hat{B}_2$ for Case 3 are independent of n.

In addition to the experimental verification, when n=1, our theoretical predictions agree with those presented for univariate Gaussian densities in Jain [7]. Note that, because of the presence of cross-terms (e.g. $\lambda_i \lambda_j$), Jain's univariate expression cannot be applied to the multivariate case by summing the contributions of each feature even when these features are mutually independent.

2.3.4 Verification of the Proposed Estimation Procedure

The proposed estimation procedure following (2.6) was tested on a set of 66-dimensional, millimeter-wave radar data. The samples were collected by rotating a Camaro and a Dodge Van on a turntable and taking approximately 8800 readings. 66 range bins were selected and the resulting 66 dimensional vectors were normalized by energy. The vectors were then selected at each half-degree to form 720-sample sets. The Bhattacharyya distance estimated from 720 samples, $\hat{B}_{720}$, was 2.29 which corresponds to an upper bound of the Bayes error of 5.1% ($\epsilon_u = 5.1\%$). These 720 samples per class were then divided into two sets of 360 samples. Since two sets were available from each class, there were 4 possible combinations of selecting one set from each class and forming a two-class problem. $\hat{B}$ was computed for each combination and the average of the 4 cases was taken. The resulting $\hat{B}_{360}$ was 3.27 ($\epsilon_u = 1.9\%$). Since g(N) of (2.6) is 1/N for this case, two equations, $\hat{B}_{720} = 2.29 = B + c/720$ and $\hat{B}_{360} = 3.27 = B + c/360$, were set up and solved for B. Note that we replaced $E\{\hat{B}_{720}\}$ by $\hat{B}_{720}$ because $Var\{\hat{B}_{720}\}$ was expected to be small from the experimental results for Cases 2 and 3. The resulting B was 1.31 ($\epsilon_u = 13.5\%$). On the other hand, when all available 8800 samples per class were used, $\hat{B}_{8800}$ was 1.51 ($\epsilon_u = 11.0\%$).

Although the radar data is not guaranteed to be Gaussian, the above results indicate that the prediction of the true B from a relatively small number of samples (720 per class for the 66 dimensional space) seems possible. Also, note that $\hat{B}_{360}$, $\hat{B}_{720}$ and $\hat{B}_{8800}$ are significantly different. Without the proposed compensation, $\hat{B}_{360}$ and $\hat{B}_{720}$ could not provide a useful upper bound of the Bayes error.

2.4 Classifier Degradation

An even more important measurement in pattern recognition is the expected performance of a given classifier. The discriminant functions for some of the popular classifiers, including the linear and quadratic classifiers, are functions of M_1, M_2, $\sum_1$ and $\sum_2$. Thus, they are the members of the family of functions presented in Section 2.2. However, unlike the Bhattacharyya distance, the degradation of the expected classifier performance due to a finite sample size comes from two sources: the finite sample set used for design and the finite number of test samples. Thus, we need to study their effects separately.

2.4.1 Effect of Test Sample Size

When the design and test samples are independent, the effect of test sample size is well-understood. Let us assume that a classifier is given and ϵ_i ($i=1,2$) is the true probability of error from class i (ω_i). In order to estimate ϵ_i, N_i samples from ω_i are drawn and tested by the given classifier and the number of misclassified samples, $\hat{\tau}_i$, is counted. The random variables $\hat{\tau}_1$ and $\hat{\tau}_2$ are independent and each is binomially distributed as [10]:

$$\Pr\{\hat{\tau}_1 = \tau_1, \hat{\tau}_2 = \tau_2\} = \underset{i=1}{\overset{2}{\pi}} \Pr\{\hat{\tau}_1 = \tau_i\}$$

$$= \underset{i=1}{\overset{2}{\pi}} \begin{pmatrix} N_i \\ \tau_i \end{pmatrix} \epsilon_i^{\tau_i}(1-\epsilon_i)^{N_i-\tau_i} \tag{2.34}$$

ϵ_i is estimated by $\hat{\tau}_i/N_i$ and subsequently, the total probability of error is estimated by

$$\hat{\epsilon} = \sum_{i=1}^{2} P_i \frac{\hat{\tau}_i}{N_i} \tag{2.35}$$

where P_i is the a priori probability of ω_i. The expected value and variance are known:

$$E\{\hat{\epsilon}\} = \epsilon \tag{2.36}$$

$$\mathrm{Var}\{\hat{\epsilon}\} = \sum_{i=1}^{2} P_i^2 \frac{\epsilon_i(1-\epsilon_i)}{N_i} \tag{2.37}$$

2.4.2 Expression of ϵ

The effect of design sample size is much harder to analyze. In order to discuss this subject, we need to express the probability of error, ϵ, in terms of the classifier. Let us assume that the classifier is defined as

$$h(X) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} 0 \tag{2.38}$$

The characteristic function of h for ω_i is

$$\phi_i(\omega) = E\{e^{j\omega h(X)} | \omega_i\} = \int_S e^{j\omega h(X)} p_i(X) dX \tag{2.39}$$

where S indicates the entire n-dimensional space and $p_i(X)$ is the density function of **X** for ω_i. Since the characteristic function of h is the Fourier transform of the density function of h (except for the sign of $j\omega$), the density function of h for ω_i, $q_i(X)$, can be obtained by the inverse Fourier transform as

$$q_i(h) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \phi_i(\omega) e^{-j\omega h} d\omega \tag{2.40}$$

The probabilities of error for ω_1 and ω_2 are

$$\epsilon_1 = \int_0^{\infty} q_1(h)dh = 1 - \int_{-\infty}^{0} q_1(h)dh \tag{2.41}$$

$$\epsilon_2 = \int_{-\infty}^{0} q_2(h)dh \tag{2.42}$$

According to Fourier transform theory, the integration in the h-space can be converted to multiplication by $1/j\omega$ in the ω-space. That is,

$$g_i(t) = \int_{-\infty}^{t} q_i(h)dh = \frac{\phi_i(0)}{2} - \frac{1}{2\pi} \int_{-\infty}^{+\infty} \frac{\phi_i(\omega)}{j\omega} e^{-j\omega t} d\omega \tag{2.43}$$

Inserting $g_i(0)$ into (2.41) and (2.42), and realizing that (2.39) guarantees $\phi_i(0) = 1$,

$$\epsilon = P_1\epsilon_1 + P_2\epsilon_2$$

$$= \frac{1}{2} + P_1 \frac{1}{2\pi} \int_{-\infty}^{+\infty} \frac{\phi_1(\omega)}{j\omega} d\omega - P_2 \frac{1}{2\pi} \int_{-\infty}^{+\infty} \frac{\phi_2(\omega)}{j\omega} d\omega$$

$$= \frac{1}{2} + \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{S} \frac{e^{j\omega h(X)}}{j\omega} [P_1 p_1(X) - P_2 p_2(X)] dX \, d\omega \qquad (2.44)$$

When the design sample size is finite, the parameters Y of the distributions are estimated and the discriminant function is based on these estimated parameters $\hat{Y}$. That is, $\hat{h}(X) = h(X,\hat{Y})$ is a random variable shifted from h(X,Y). Taking the expectation with respect to $\hat{Y}$,

$$E\{\hat{\epsilon}\} = \frac{1}{2} + \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{S} \frac{E\{e^{j\omega \hat{h}(X)}\}}{j\omega} [P_1 p_1(X) - P_2 p_2(X)] dX \, d\omega (2.45)$$

Treating $e^{j\omega \hat{h}(X)}$ as $\hat{f}$ in (2.3)

$$E\{e^{j\omega \hat{h}(X)}\} \cong e^{j\omega h(X)} + \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} \frac{\partial^2 e^{j\omega h(X)}}{\partial y_i \partial y_j} E\{\Delta y_i \Delta y_j\}$$

$$= e^{j\omega h(X)} + \frac{j\omega}{2} e^{j\omega h(X)} \sum_{i=1}^{L} \sum_{j=1}^{L} \left[\frac{\partial^2 h(X)}{\partial y_i \partial y_j} + j\omega \frac{\partial h(X)}{\partial y_i} \frac{\partial h(X)}{\partial y_j} \right] E\{\Delta y_i \Delta y_j\} \quad (2.46)$$

Substituting (2.46) into (2.45) and realizing $E\{\Delta\epsilon\} = E\{\hat{\epsilon}\} - \epsilon$,

$$E\{\Delta\epsilon\} \cong \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{S} \frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} \left[\frac{\partial^2 h(X)}{\partial y_i \partial y_j} + j\omega \frac{\partial h(X)}{\partial y_i} \frac{\partial h(X)}{\partial y_j} \right] E\{\Delta y_i \Delta y_j\}$$

$$\times e^{j\omega h(X)} [P_1 p_1(X) - P_2 p_2(X)] dX \, d\omega \qquad (2.47)$$

Eq. (2.47) is a very general expression for $E\{\Delta\epsilon\}$ which is valid regardless of the selection of h(X), P_i and $p_i(X)$. The term $E\{\Delta y_i \Delta y_j\}$ gives the effect of the sample size, N. Therefore, if (2.5) is satisfied, $E\{\Delta\epsilon\}$ can be expressed by cg(N) where c is determined by h(X), P_i and $p_i(X)$, and the proposed estimation procedure following (2.6) can be applied. Furthermore, if h(X) is a function of M_i and $\sum_i$, g(N) becomes 1/N.

2.4.3 The Quadratic Classifier for Gaussian Distributions

When a quadratic classifier is designed from N training samples, drawn from two simultaneously diagonalized distributions, N(O,I) and N(M,Λ), with a priori probabilities $P_1 = P_2 = 0.5$, the discriminant function can be found as

$$\hat{h}(X) = \frac{1}{2}(X-\hat{M}_1)^T\Sigma_1^{-1}(X-\hat{M}_1) - \frac{1}{2}(X-\hat{M}_2)^T\Sigma_2^{-1}(X-\hat{M}_2)$$

$$+\frac{1}{2}\ln|\Sigma_1| - \frac{1}{2}\ln|\Sigma_2| \tag{2.48}$$

where $\hat{M}_i$ and Σ_i are estimated by (2.8). Forming $\hat{Y}$ as in (2.9), we only need to compute $\partial h/\partial m_i^{(r)}$, $\partial h/\partial \alpha_{ij}^{(r)}$, $\partial^2 h/\partial m_i^{(r)2}$, $\partial^2 h/\partial \alpha_{ij}^{(r)} \partial \alpha_{ij}^{(r)}$ and $\partial^2 h/\partial \alpha_{ij}^{(r)} \partial \alpha_{ji}^{(r)}$, since $E\{\Delta y_i \Delta y_j\} = 0$ for other combinations.

With M_i and Σ_i given in (2.7), these partial derivatives can be easily computed and are listed in Appendix C. Substituting these results into (2.47),

$$\frac{1}{2}\sum_{i=1}^{L}\sum_{j=1}^{L}[\,\cdot\,]\,E\{\Delta y_i \Delta y_j\}$$

$$\cong \frac{1}{2N}\left[(n+1)\sum_{i=1}^{n}\left\{x_i^2 - \frac{(x_i-m_i)^2}{\lambda_i}\right\}\right.$$

$$\left. + j\omega\left[n + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left\{x_i^2 x_j^2 + \frac{(x_i-m_i)^2(x_j-m_j)^2}{\lambda_i \lambda_j}\right\}\right]\right]$$

$$\triangleq \frac{1}{N} f_q(X,\omega) \tag{2.49}$$

Thus, (2.47) may be rewritten as

$$E\{\hat{\epsilon}\} \cong \epsilon + \frac{c_q}{N} \tag{2.50}$$

where

$$c_q = \frac{1}{2\pi}\int_{-\infty}^{+\infty}\int_S f_q(X,\omega)e^{j\omega h(X)}[P_1 p_1(X) - P_2 p_2(X)]dX\, d\omega \tag{2.51}$$

That is, c_q is determined by the underlying distributions, and stays constant for experiments with various sample sizes. Thus, as was proposed in Section 2.2, we may choose various values of N as $N_1,,N_\ell$, and measure $\hat{\epsilon}$. Computing $E\{\hat{\epsilon}\}$ from several independent trials, we may solve (2.50) for ϵ and c_q by a line fit technique.

The above technique was applied to the radar data. The entire 8800-sample set was divided into two groups, each consisting of 4400 samples. When one group was used to design a quadratic classifier and the other was used for testing, the error, $\hat{\epsilon}_{4400}$, was 17.2%. Then, 720 samples were selected from the design group and used to design a quadratic classifier. The entire 4400 samples of the test group were tested, resulting in $\hat{\epsilon}_{720} =$ 21.4%. Such a large number of test samples was used to eliminate the variation of $\hat{\epsilon}$ due to test sample size. The same experiment was performed for 360 samples. Since there were two groups of 360 samples from 720 samples for each class, four error estimates were obtained; they were averaged, resulting in $\hat{\epsilon}_{360} = 25.4\%$. $\hat{\epsilon}_{720}$ and the averaged $\hat{\epsilon}_{360}$ were used to obtain ϵ by solving (2.50), resulting in $\epsilon = 17.4\%$. This result is very close to $\hat{\epsilon}_{4400} = 17.2\%$, and confirms that we can predict the potential performance of the quadratic classifier even if the available sample size is relatively small for a high dimensional space.

Although we do not need to know the value of c_q to conduct the above experiment to estimate ϵ, c_q can be computed by carrying through the integration of (2.51). Let us consider the simplest case, Case 2 of Table 2.1, in which $p_1(X)$ and $p_2(X)$ are Gaussian N(0,I) and N(M,I) respectively. Then, $e^{j\omega h}(X)p_i(X)$ may be rewritten as

$$e^{j\omega h(X)}p_1(X) = \frac{\sqrt{2\pi}}{\sqrt{\beta}}\, e^{-\beta/8}N_\omega(-\frac{j}{2},\frac{1}{\beta})N_X(j\omega M,I) \tag{2.52}$$

$$e^{j\omega h(X)}p_2(X) = \frac{\sqrt{2\pi}}{\sqrt{\beta}}\, e^{-\beta/8}N_\omega(\frac{j}{2},\frac{1}{\beta})N_X((1+j\omega)M,I) \tag{2.53}$$

where $\beta = M^TM$. $N_\omega(a,b)$ and $N_X(D,K)$ are Gaussian density functions of ω and X with the expected value of a and variance b for N_ω, and the expected vector D and covariance matrix K for N_X.

Since $f_q(\omega,X)$ is a linear combination of $x_i^a x_j^b (a,b \leqq 4)$ as is seen in (2.49), $\int f_q(X,\omega)N_X(\cdot,\cdot)dX$ is the linear combination of the moments of $N_X(\cdot,\cdot)$. The result of the integration becomes a polynomial in ω

$$\gamma_i(\omega) = \frac{\beta^2}{2}(j\omega)^5 \mp \beta^2(j\omega)^4 + \frac{\beta}{2}(n+5+3\beta)(j\omega)^3 \mp \frac{\beta}{2}(n+5+2\beta)(j\omega)^2$$

$$+\frac{1}{4}\left[n(n+7)+(5n+9)\beta+\beta^2\right](j\omega) \mp \frac{(n+1)\beta}{2} \tag{2.54}$$

where $-$ and $+$ of $\mp$ are for i=1 and 2 respectively. Again, the $\int\gamma_i(\omega)N_\omega(\cdot,\cdot)d\omega$ is a linear combination of the moments of $N_\omega(\cdot,\cdot)$. Thus, c_q for P_1=P_2=0.5 is

$$c_q = \frac{1}{4\sqrt{2\pi\beta}}e^{-\beta/8}\left[n^2+(1+\frac{\beta}{2})n+(\frac{\beta^2}{16}-\frac{\beta}{2}-1)\right] \tag{2.55}$$

$E\{\hat{\epsilon}\}$ can be predicted theoretically by $\epsilon + c_q/N$. Table 2.8 lists the theoretical predictions for various N and k (=N/n) for the distribution parameters given in Case 2 of Table 2.1. These parameters yield $\beta = 2.56^2$ and $\epsilon = 0.1$ (10%). Also shown in Table 2.8 are experimental results verifying these predictions. For each combination of n and k, N samples were generated from each class and used to design a quadratic classifier which was then tested on true distributions. Novak developed an algorithm which numerically computes the error of any discriminant function with a quadratic form tested on two Gaussian distributions [11]. This procedure was repeated 10 times. The second and third lines in Table 2.8 show the means and standard deviations of the experimental results. The theoretical prediction accurately reflects the experimental trends. Also, the standard deviations are small. Notice that as n gets larger, k must increase to maintain the same performance, since c_q is proportional to n^2 for $n >> 1$. This conclusion agrees with Pipberger's experimental results [6] and the numerical tables in Raudys and Pikelis [3]. Together, these present design guidelines significantly different from the traditional rules of thumb which suggest a particular fixed value of k for all values of n.

2.4.4 The Linear Classifier for Gaussian Distributions

The analysis of the linear classifier proceeds in a similar fashion. Fisher's linear discriminant function is

$$h(X) = (M_2-M_1)^T\bar{\Sigma}^{-1}X+\frac{1}{2}(M_1^T\bar{\Sigma}^{-1}M_1-M_2^T\bar{\Sigma}^{-1}M_2) \tag{2.56}$$

where $\bar{\Sigma} = (\Sigma_1+\Sigma_2)/2$. Again, we assume, without lost of generality, that $M_1 = 0$, $M_2 = M$, $\Sigma_1 = I$ and $\Sigma_2 = \Lambda$.

The partial derivatives of h can be computed as is shown in Appendix D. Thus, (2.47) becomes

Table 2.8 Quadratic classifier degradation for Case 2.

				n		
		4	8	16	32	64
k	3	0.1450	0.1689	0.2115	0.3067	0.4894
		0.1668	0.2041	0.2204	0.2673	0.3131
		0.0351	0.0235	0.0289	0.0195	0.0133
	5	0.1270	0.1414	0.1691	0.2240	0.3336
		0.1403	0.1640	0.1734	0.2081	0.2554
		0.0211	0.0186	0.0091	0.0057	0.0074
	10	0.1135	0.1207	0.1345	0.1620	0.2168
		0.1152	0.1240	0.1366	0.1573	0.1934
		0.0081	0.0061	0.0070	0.0054	0.0085
	15	0.1090	0.1138	0.1230	0.1413	0.1779
		0.1086	0.1184	0.1232	0.1415	0.1658
		0.0044	0.0061	0.0042	0.0053	0.0042
	20	0.1067	0.1103	0.1173	0.1310	0.1584
		0.1077	0.1105	0.1190	0.1393	0.1513
		0.0021	0.0023	0.0051	0.0022	0.0032
	30	0.1045	0.1069	0.1115	0.1207	0.1389
		0.1054	0.1071	0.1114	0.1307	0.1365
		0.0019	0.0021	0.0020	0.0019	0.0022
	40	0.1034	0.1052	0.1086	0.1155	0.1292
		0.1037	0.1057	0.1087	0.1150	0.1275
		0.0024	0.0013	0.0013	0.0013	0.0018
	50	0.1027	0.1041	0.1069	0.1124	0.1234
		0.1025	0.1044	0.1068	0.1125	0.1221
		0.0013	0.0010	0.0013	0.0009	0.0007

(1st line: Theoretical prediction, 2nd line: The mean of 10 trials,
3rd line: The standard deviation of 10 trials)

$$E\{\hat{\epsilon}\} \cong \epsilon + \frac{c_\ell}{N} \tag{2.57}$$

$$c_\ell = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_S f_\ell(X,\omega) e^{j\omega h(X)} [P_1 p_1(X) - P_2 p_2(X)] dX \, d\omega \tag{2.58}$$

$$\begin{aligned} f_\ell(X,\omega) = & \sum_{i=1}^{n} \left[\frac{1-\lambda_i}{1+\lambda_i} + (2x_i - m_i) \left\{ \frac{(1+\lambda_i^2)m_i}{(1+\lambda_i)^3} + \frac{m_i}{(1+\lambda_i)^2} \sum_{j=1}^{n} \frac{1+\lambda_i\lambda_j}{1+\lambda_j} \right\} \right] \\ & + \frac{j\omega}{2} \left[4 \sum_{i=1}^{n} \left\{ \frac{x_i^2}{(1+\lambda_i)^2} + \frac{(x_i-m_i)^2\lambda_i}{(1+\lambda_i)^2} \right\} \right. \\ & \left. + \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{m_i(2x_j-m_j)(1+\lambda_i\lambda_j)}{(1+\lambda_i)^2(1+\lambda_j)^2} \left\{ m_i(2x_j-m_j) + m_j(2x_i-m_i) \right\} \right] \end{aligned} \tag{2.59}$$

Again, c_ℓ is determined by the underlying distributions, and ϵ can be estimated from experiments with various N. Also, since $f_\ell(X,\omega)$ is a linear combination of $x_i^a (a \leq 2)$, c_ℓ can be theoretically computed for Case 2 of Table 2.1, resulting in

$$c_\ell = \frac{1}{2\sqrt{2\pi\beta}} e^{-\beta/8} [(1 + \frac{\beta}{4}) n - 1] \tag{2.60}$$

Eq. (2.60) was experimentally verified in the same manner as (2.55). The results are shown in Table 2.9.

Comparison of (2.55) and (2.60) reveals an important distinction between quadratic and linear classifiers. For Case 2, the two covariances are the same. Thus, if the true underlying parameters are used, the quadratic classifier of (2.48) becomes identical to the linear classifier of (2.56). However, when the estimated covariances are used, $\Sigma_1 \neq \Sigma_2$ even though $\Sigma_1 = \Sigma_2$. Thus, the classifier of (2.48) differs from that of (2.56). As a result, $E\{\Delta\epsilon\}$ for quadratic is proportional to n^2/N $(=n/k)$ while $E\{\Delta\epsilon\}$ for linear is proportional to n/N $(=1/k)$ as in (2.55) and (2.60) when $n >> 1$. Although it depends on the values of n and β, we may generally conclude that c_q is larger than c_ℓ for $n >> 1$. This implies that many more samples are needed to properly design a quadratic classifier than a linear classifier. Novak reported in [11] that the linear classifier is more robust (less sensitive to parameter estimation errors) than the quadratic classifier, particularly in

Table 2.9 Linear classifier degradation for Case 2.

			n		
k	4	8	16	32	64
3	0.1273	0.1287	0.1294	0.1298	0.1300
	0.1437	0.1436	0.1336	0.1302	0.1319
	0.0365	0.0174	0.0135	0.0081	0.0040
5	0.1164	0.1172	0.1177	0.1179	0.1180
	0.1165	0.1223	0.1207	0.1199	0.1207
	0.0128	0.0153	0.0071	0.0048	0.0041
10	0.1082	0.1086	0.1088	0.1089	0.1090
	0.1050	0.1089	0.1093	0.1086	0.1092
	0.0030	0.0041	0.0024	0.0021	0.0019
15	0.1055	0.1057	0.1059	0.1060	0.1060
	0.1048	0.1080	0.1064	0.1058	0.1064
	0.0030	0.0032	0.0027	0.0013	0.0012
20	0.1041	0.1043	0.1044	0.1045	0.1045
	0.1039	0.1039	0.1058	0.1040	0.1045
	0.0021	0.0018	0.0026	0.0011	0.0008
30	0.1027	0.1029	0.1029	0.1030	0.1030
	0.1036	0.1033	0.1027	0.1033	0.1028
	0.0023	0.0021	0.0009	0.0006	0.0006
40	0.1020	0.1022	0.1022	0.1022	0.1022
	0.1022	0.1027	0.1021	0.1023	0.1022
	0.0021	0.0014	0.0009	0.0005	0.0004
50	0.1016	0.1017	0.1018	0.1018	0.1018
	0.1016	0.1021	0.1018	0.1018	0.1017
	0.0011	0.0007	0.0005	0.0004	0.0003

(1st line: Theoretical prediction, 2nd line: The mean of 10 trials, 3rd line: The standard deviation of 10 trials) high dimensional spaces. Our results support his claim both theoretically and experimentally.

Also note that for large n, c_ℓ/N is proportional to $1/k$. This indicates that, as far as the design of a linear classifier is concerned, a fixed multiple could be used to determine the sample size from the dimensionality. This coincides with the conclusions of many reports in the past. However, (2.60) indicates that the value of the multiple depends on β, which measures the separability between two distributions with a common covariance matrix.

2.5 Conclusions

The main purpose of this chapter was to investigate the effect of finite sample size parameter estimates on the evaluation of a family of functions. To this end, we have presented general expressions for the expected bias and variance in terms of the statistical properties of the parameter estimates.

Applying these expressions to the Bhattacharyya distance has provided insight into the relationship between the dimensionality and the number of training samples and their effect on measuring separability due to mean and covariance shifts. Applying them to classifier evaluation equations, we have derived explicit expressions for the degradation of the quadratic and linear classifiers. This provides a new guideline for the selection of the number of samples or features necessary for a certain level of classifier performance. We have provided theoretical evidence that, as the dimensionality increases, covariance-based similarity measures and the quadratic classifier require an increasing multiple of samples. We have also presented support for the claim that the linear classifier is more robust.

Finally, the form of the bias expression allows the dependence on the sample size to be separated from the distribution-specific terms. Since the distribution and dimension are fixed for a given sample set, an empirical approach was employed to use estimates of expected performance for different sized samples to find an estimate of the asymptotic performance. This allows small sample sets to provide accurate, unbiased estimates.

CHAPTER 3
ESTIMATION OF CLASSIFIER PERFORMANCE

3.1 Introduction

Evaluating the performance of a classifier is an important, yet difficult problem in pattern recognition. In practice, the true distributions are never known and only a finite number of training samples are available. The designer must decide whether this sample size is adequate or not, and also decide how many samples should be used to design the classifier and how many should be used to test it. The effect of the test sample size is well-known. However, the effect of the design sample size is hardly understood in spite of substantial effort in the past.

The leave-one-out method [13] is designed to alleviate one of the above difficulties. That is, it avoids dividing the available sample set into design and test, while maintaing an independence between them. Thus, the procedure utilizes all available samples more efficiently, and produces a pessimistic error estimate. On the other hand, the resubstitution method, in which the available samples are used for both design and test without any modification, produces an optimistic error. Thus, using both methods simultaneously, we can obtain an upper and lower bounds of the true performance of the classifier.

More recently, Efron [14] proposed a re-sampling procedure, called the bootstrap method, in which artificial samples are generated from the existing samples, and the optimistic bias of the resubstitution error is estimated from them.

All these procedures work well experimentally. However, it was still very difficult to analize them theoretically and to find the effects of sample size and other parameters on the errors. Raudys and Pikelis [3] gave an excellent review of work done in approximating the expected performance in the parametric case. The difficulty came from the fact that the explicit expression for the classification error was not available or too complex for further theoretical development, except in the case of linear classifiers.

In Chapter 2, we investigated the effect of sample size on a family of functions, and found a manageable expression for the errors of classifiers, including quadratic and Fisher linear classifiers. Using the expression, we computed the biases of these classifier errors due to a finite design set.

The objective of this chapter is to apply the error expression of Chapter 2 to the various methods of error estimation mentioned above, and to offer a unified and complehensive approach to the analysis of classifier performance. In Section 3.2, after the error expression is introduced, it is applied to three cases: (1) a given classifier and a finite test set, (2) given test distributions and a finite design set, and (3) finite and independent design and test sets. For all cases, the expected values and variances of the classifier errors are presented. Although the study of Case 1 does not produce any new results, it is important to confirm that the proposed approach produces the known results, and also to show how these results are modified when the design set becomes finite, as in Cases 2 and 3. In Section 3.3, the error expression of Chapter 2 is used to compute the bias between the leave-one-out and resubstitution errors for quadratic classifiers. Note that in this case the design and test sample sets are no longer independent. Again, the expected value and variance of the bias are presented. Also, because of its similarity to the analysis of the leave-one-out method, the effect of outliers in design samples on the classification error is discussed. Finally, in Section 3.4, the theoretical analysis of the bootstrap method is presented for quadratic classifiers. The explicit error expression can be obtained for the optimistic bias of the bootstrap resubstitution error. The expected value of the bias with respect to the bootstrap procedure is shown to be very close to the bias between the conventional leave-one-out and resubstitution errors. The variance of the bootstrap bias also can be computed in a closed form.

Throughout all sections, the theoretical conclusions are experimentally verified. The results of these analyses allow us to delve into the theoretical differences between the methods and account for a series of frequently-observed experimental trends.

3.2 Classification Errors For Finite Samples

In this section, we will discuss the effects of finite test and design samples on classification performance.

3.2.1 Error Expression

For the two-class problem, a classifier can be expressed by

$$h(X) \overset{\omega_1}{\underset{\omega_2}{\lessgtr}} 0 \tag{3.1}$$

where h(X) is the descriminant function of an n-dimensional vector X, and ω_i indicates the class i (i=1,2). The probabilities of errors for this classifier from ω_1 and ω_2 are

$$\epsilon_1 = \int_{h(X)>0} p_1(X)dX = \int_S u(h)p_1(X)dX$$

$$= \frac{1}{2\pi}\int_S \int_{-\infty}^{+\infty} [\frac{1}{j\omega} + \pi\delta(\omega)]e^{j\omega h(X)}p_1(X)d\omega dX$$

$$= \frac{1}{2} + \frac{1}{2\pi}\int_S \int_{-\infty}^{+\infty} \frac{e^{j\omega h(X)}}{j\omega}p_1(X)d\omega dX \tag{3.2}$$

and

$$\epsilon_2 = \int_{h(X)<0} p_2(X)dX = \frac{1}{2} - \frac{1}{2\pi}\int_S \int_{-\infty}^{+\infty} \frac{e^{j\omega h(X)}}{j\omega}p_2(X)d\omega dX \tag{3.3}$$

where $p_i(X)$ is the density function of class i tested by the classifier, and S indicates the entire n-dimensional space. The second line of (3.2) is obtained using the fact that the Fourier transform of a step function, u(h), is $[1/j\omega + \pi\delta(\omega)]$.

The total probability of error is

$$\epsilon = P_1\epsilon_1 + P_2\epsilon_2$$

$$= \frac{1}{2} + \frac{1}{2\pi}\int_S \int_{-\infty}^{+\infty} \frac{e^{j\omega h(X)}}{j\omega}\tilde{p}(X)d\omega dX \tag{3.4}$$

where P_i is the a priori prrobability of ω_i and

$$\tilde{p}(X) = P_1p_1(X) - P_2p_2(X) \tag{3.5}$$

3.2.2 Effect of Test Samples

When a finite number of samples are tested by a given classifier, $p_i(X)$ of (3.5) may be replaced by

$$\hat{p}_i(X) = \frac{1}{N_i}\sum_{j=1}^{N_i}\delta(X - \mathbf{X}_j^{(i)}) \tag{3.6}$$

where $\mathbf{X}_1^{(i)},,\mathbf{X}_{N_i}^{(i)}$ are N_i test samples drawn from $p_i(X)$. Throughout the chapter, boldface indicates randomness.

Thus, the estimate of the error probability is

$$\hat{\epsilon} = \frac{1}{2} + \frac{1}{2\pi}\int_S\int_{-\infty}^{+\infty}\frac{e^{j\omega h(X)}}{j\omega}\left[\frac{P_1}{N_1}\sum_{j=1}^{N_1}\delta(X - \mathbf{X}_j^{(1)}) - \frac{P_2}{N_2}\sum_{j=1}^{N_2}\delta(X - \mathbf{X}_j^{(2)})\right]d\omega dX$$

$$= \frac{1}{2} + \frac{P_1}{N_1}\sum_{j=1}^{N_1}\alpha_j^{(1)} - \frac{P_2}{N_2}\sum_{j=1}^{N_2}\alpha_j^{(2)} \tag{3.7}$$

where

$$\alpha_j^{(i)} = \frac{1}{2\pi}\int_{-\infty}^{+\infty}\frac{e^{j\omega h(\mathbf{X}_j^{(i)})}}{j\omega}d\omega \tag{3.8}$$

The expected value of $\alpha_j^{(i)}$ with respect to $\mathbf{X}_j^{(i)}$ (w.r.t. the test samples) is

$$\bar{\alpha}_i = E_t\{\alpha_j^{(i)}\} = \frac{1}{2\pi}\int_S\int_{-\infty}^{+\infty}\frac{e^{j\omega h(X)}}{j\omega}p_i(X)d\omega dX$$

$$= \begin{cases} \epsilon_1 - \dfrac{1}{2} & \text{for } i=1 \\ \dfrac{1}{2} - \epsilon_2 & \text{for } i=2 \end{cases} \tag{3.9}$$

The second line of (3.9) can be obtained from (3.2) and (3.3) respectively. The second-order moments are also computed as

$$E_t\{\alpha_j^{(i)2}\} = E_t\{[\frac{1}{2\pi}\int_{-\infty}^{+\infty}\frac{e^{j\omega h(X)}}{j\omega}d\omega]^2\} = E_t\{[\frac{1}{2}\text{sgn}(h(\mathbf{X}))]^2\}$$

$$= \frac{1}{4} \tag{3.10}$$

$$E_t\{\alpha_j^{(i)}\alpha_k^{(\ell)}\} = \bar{\alpha}_i\bar{\alpha}_\ell \qquad \text{otherwise} \tag{3.11}$$

where sgn(h) equals +1 for $h>0$ and -1 for $h<0$. Eq. (3.11) is obtained because $\alpha_j^{(i)}$ and $\alpha_k^{(\ell)}$ are independent due to the independence between $X_j^{(i)}$ and $X_k^{(\ell)}$.

From (3.7) and (3.9)-(3.11),

$$E_t\{\hat{\epsilon}\} = \frac{1}{2} + P_1\bar{\alpha}_1 - P_2\bar{\alpha}_2$$

$$= \frac{1}{2} + P_1(\epsilon_1 - \frac{1}{2}) - P_2(\frac{1}{2} - \epsilon_2) = \epsilon \tag{3.12}$$

$$Var_t\{\hat{\epsilon}\} = \frac{P_1^2}{N_1}Var_t\{\alpha_j^{(1)}\} + \frac{P_2^2}{N_2}Var_t\{\alpha_j^{(2)}\}$$

$$= \frac{P_1^2}{N_1}[\frac{1}{4} - (\epsilon_1 - \frac{1}{2})^2] + \frac{P_2^2}{N_2}[\frac{1}{4} - (\frac{1}{2} - \epsilon_2)^2]$$

$$= P_1^2\frac{\epsilon_1(1-\epsilon_1)}{N_1} + P_2^2\frac{\epsilon_2(1-\epsilon_2)}{N_2} \tag{3.13}$$

That is, $\hat{\epsilon}$ is an unbiased estimate, and its variance has the well-known form derived from the binomial distribution [10].

3.2.3 Effect of Design Samples

It is more difficult to discuss the effect of using a finite number of design samples. Although we would like to keep the formula as general as possible, in this section a specific family of discriminant functions is investigated to help determine which approximations should be used.

Assume that the discriminant function is a function of two expected vectors, M_1 and M_2, and covariance matrices, Σ_1 and Σ_2. Typical examples are the quadratic classifier and Fisher's linear classifier:

$$h(X) = \frac{1}{2}(X-M_1)^T\Sigma_1^{-1}(X-M_1) - \frac{1}{2}(X-M_2)^T\Sigma_2^{-1}(X-M_2) + \frac{1}{2}\ln\frac{|\Sigma_1|}{|\Sigma_2|} \tag{3.14}$$

$$h(X) = (M_2-M_1)^T\Sigma^{-1}X + \frac{1}{2}(M_1^T\Sigma^{-1}M_1-M_2^T\Sigma^{-1}M_2) \tag{3.15}$$

where $\Sigma = [\Sigma_1+\Sigma_2]/2$. When only a finite number of design samples are available and M_i and Σ_i are estimated from them,

$$\Delta h(X) = \hat{h}(X) - h(X) = \sum_{k=1}^{\infty} \mathbf{0}^{(k)} \tag{3.16}$$

where $\hat{h}(X)=h(X,\hat{M}_1,\hat{M}_2,\Sigma_1,\Sigma_2)$,$h(X)=h(X,M_1,M_2,\Sigma_1,\Sigma_2)$ and $\mathbf{0}^{(k)}$ is the k-th order term of the Taylor series expansion in terms of the variations of $\hat{M}_i$ and Σ_i. If the design samples are drawn from Gaussian distributions, and $\hat{M}_i$ and Σ_i are unbiased estimates (e.g., the sample mean and sample covariance), it was shown in Chapter 2 that

$$E_d\{\mathbf{0}^{(1)}\} = 0,\ E_d\{\mathbf{0}^{(2)}\}\sim 1/\mathcal{N},\ E_d\{\mathbf{0}^{(3)}\} = 0,\ E_d\{\mathbf{0}^{(4)}\}\sim 1/\mathcal{N}^2 \tag{3.17}$$

where E_d indicates the expectation with respect to the design samples, and $\mathcal{N}$ is the number of design samples (while N indicates the number of test samples). Therefore, from (3.16) and (3.17),

$$E_d\{\Delta h(X)\}\sim 1/\mathcal{N},\ E_d\{\Delta h^2(X)\}\sim 1/\mathcal{N},\ E_d\{\Delta h^3(X)\}\sim 1/\mathcal{N}^2$$

$$E_d\{\Delta h^4(X)\}\sim 1/\mathcal{N}^2 \ldots \tag{3.18}$$

Assuming that $\mathcal{N}$ is reasonably large, we can eliminate $E\{\Delta h^m(X)\}$ for m larger than 2.

Thus, the error of a random classifier for given test distributions is expressed by (3.4)

$$\hat{\epsilon} = \frac{1}{2} + \frac{1}{2\pi}\int_S\int_{-\infty}^{+\infty} \frac{e^{j\omega\hat{h}(X)}}{j\omega}\tilde{p}(X)d\omega dX \tag{3.19}$$

The expected value $\bar{\epsilon}$ with respect to the design samples is

$$\bar{\epsilon} = E_d\{\hat{\epsilon}\} = \frac{1}{2} + \frac{1}{2\pi}\int_S\int_{-\infty}^{+\infty} \frac{E_d\{e^{j\omega\hat{h}(X)}\}}{j\omega}\tilde{p}(X)d\omega dX \tag{3.20}$$

$$= \epsilon + \overline{\Delta\epsilon}$$

where

$$\overline{\Delta\epsilon} \cong \frac{1}{2\pi}\int_{S}\int_{-\infty}^{+\infty} E_d\{\Delta h(X) + \frac{j\omega}{2}\Delta h^2(X)\}e^{j\omega h(X)}\tilde{p}(X)d\omega dX \qquad (3.21)$$

The approximation from (3.20) to (3.21) was made by using

$$e^{j\omega\hat{h}(X)} = e^{j\omega h(X)}e^{j\omega\Delta h(X)} \cong e^{j\omega h(X)}[1+j\omega\Delta h(X)+\frac{(j\omega)^2}{2}\Delta h^2(X)].$$

When h(X) is the Bayes classifier, ϵ must be a minimum. Appendix E gives the proof that $\hat{\epsilon}$ of (3.19) is indeed larger than ϵ of (3.4).

When two Gaussian distributions are classified by the quadratic or linear classifier whose parameters are estimated from a finite sample set, $\overline{\Delta\epsilon}$ of (3.21) can be computed. Explicit solutions for the case with $M_1=0$, $M_2=M$ and $\Sigma_1=\Sigma_2=I$ are given in Chapter 2.

The variance of $\hat{\epsilon}$ may be computed from (3.19) and (3.20) as

$$\mathrm{Var}_d\{\hat{\epsilon}\} = \frac{1}{4\pi^2}\int_{S_x}\int_{S_y}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\frac{E_d\{e^{j\omega_1\hat{h}(X)}e^{j\omega_2\hat{h}(Y)}\}}{j\omega_1 j\omega_2}\tilde{p}(X)\tilde{p}(Y)d\omega_1 d\omega_2 dXdY$$

$$-(\bar{\epsilon}-\frac{1}{2})^2$$

$$\cong \frac{1}{4\pi^2}\int_{S_x}\int_{S_y}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}E_d\{\Delta h(X)\Delta h(Y)\}e^{j\omega_1 h(X)}e^{j\omega_2 h(Y)}\tilde{p}(X)\tilde{p}(Y)d\omega_1 d\omega_2 dXdY$$

$$= \int_{S_x}\int_{S_y}E_d\{\Delta h(X)\Delta h(Y)\}\delta(h(X))\delta(h(Y))\tilde{p}(X)\tilde{p}(Y)dXdY$$

$$= \int_{h(X)=0}\int_{h(Y)=0} E_d\{\Delta h(X)\Delta h(Y)\}\tilde{p}(X)\tilde{p}(Y)dXdY \qquad (3.22)$$

where the derivation from the first line to the second line is given in Appendix F. Eq. (3.22) indicates that the integration is carried out along the classification boundary where h(X) = 0. When h(X) is the Bayes classifier, $\tilde{p}(X)$ of (3.5) must be zero at the boundary. Thus, (3.22) becomes 0. Since we neglected the higher order terms of $\Delta h(X)$, $\mathrm{Var}_d\{\hat{\epsilon}\}$ is not zero, but proportional to $1/\mathcal{N}^2$. When h(X) is not the Bayes classifier, $\tilde{p}(X) \neq 0$ at h(X) = 0. Thus, we may observe a variance dominated by a term proportional to $1/\mathcal{N}$ due to the fact that $E_d\{\Delta h(X)\Delta h(Y)\} \sim 1/\mathcal{N}$.

In order to confirm the above theoretical conclusion, an experiment has been run for the quadiatic classifier between two Gaussian distributions which share the same covariance matrix I and differ in the means to give a Bayes error of 10%. The dimensionality n was varied from 4 to 64 in powers of 2 and the ratio of the sample size and the dimensionality $k(=\mathcal{N}/n)$ was varied from 3 to 50. $\mathcal{N}$ $(=nk)$ samples were generated from each class according to the given mean and covariance, and $\hat{M}_i$ and Σ_i were estimated from the generated data using the sample mean and sample covariance. The quadratic classifier was designed by (3.14). Testing was done by Novak's program which numerically computes the error of any discriminant function with a quadratic form tested on separately specified Gaussian distributions [11]. This procedure was repeated 10 times. The second and third lines of Table 3.1 show the average and standard deviation of these experiments. The first line shows the theoretically computed errors from (3.20) and (3.21). Also, Fig. 3.1 shows the relationship between $1/k(=n/\mathcal{N})$ and the standard deviation. From these results, we may confirm that the standard deviation is very small and roughly proportional to $1/\mathcal{N}$. Thus, the variance is proportional to $1/\mathcal{N}^2$.

An intuitive reason why the standard deviation due to a finite number of design samples is proportional to $1/\mathcal{N}$ may be observed as follows. When the Bayes classifier is implemented, $\Delta\epsilon$ is always positive and thus generates a positive bias. As (3.21) suggests, the bias is proportional to $1/\mathcal{N}$. Since $\Delta\epsilon$ varies between 0 and some positive value with an expected value $a/\mathcal{N}$ (where a is a positive number), we can expect that the standard deviation is also proportional to $1/\mathcal{N}$.

In addition, it should be noted that design samples affect the variance of the error in a different way from test samples. When a classifier is fixed, the variations of the two test distributions are independent. Thus, $\mathrm{Var}_t\{\hat{\epsilon}\} = P_1^2\mathrm{Var}\{\hat{\epsilon}_1\} + P_2^2\mathrm{Var}\{\hat{\epsilon}_2\}$ as is seen is (3.13). On the other hand, when the test distributions are fixed and the classifier varies, $\hat{\epsilon}_1$ and $\hat{\epsilon}_2$ are strongly correlated with a correlation coefficient close to -1. That is, when $\hat{\epsilon}_1$ increases, $\hat{\epsilon}_2$ decreases and vice versa. Thus, when $P_1=P_2, \mathrm{Var}_d\{\hat{\epsilon}\}=(0.5)^2 E_d\{\Delta\epsilon_1^2\}+(0.5)^2E_d\{\Delta\epsilon_2^2\}+2(0.5)^2E_d\{\Delta\epsilon_1\Delta\epsilon_2\} \cong (0.5)^2[E_d\{\Delta\epsilon_1^2\}+E_d\{(-\Delta\epsilon_1)^2\} +2E_d\{\Delta\epsilon_1(-\Delta\epsilon_1)\}]=0$. The covariance of $\hat{\epsilon}_1$ and $\hat{\epsilon}_2$ cancels the individual variances of $\hat{\epsilon}_1$ and $\hat{\epsilon}_2$.

Table 3.1 Quadratic classifier degradation for I-I (%).

		n				
		4	8	16	32	64
		14.50	16.89	21.15	30.67	48.94
	3	16.68	20.41	22.04	26.73	31.31
		3.51	2.35	2.89	1.95	1.33
		12.70	14.14	16.91	22.40	33.36
	5	14.03	16.40	17.34	20.81	25.54
		2.11	1.86	0.91	0.57	0.74
		11.35	12.07	13.45	16.20	21.68
	10	11.52	12.40	13.66	15.73	19.34
		0.81	0.61	0.70	0.54	0.85
		10.90	11.38	12.30	14.13	17.79
	15	10.86	11.84	12.32	14.15	16.58
k		0.44	0.61	0.42	0.53	0.42
		10.67	11.03	11.73	13.10	15.84
	20	10.77	11.05	11.90	13.93	15.13
		0.21	0.23	0.51	0.22	0.32
		10.45	10.69	11.15	12.07	13.89
	30	10.54	10.71	11.14	13.07	13.65
		0.19	0.21	0.20	0.19	0.22
		10.34	10.52	10.86	11.55	12.92
	40	10.37	10.57	10.87	11.50	12.75
		0.24	0.13	0.13	0.13	0.18
		10.27	10.41	10.69	11.24	12.34
	50	10.25	10.44	10.68	11.25	12.21
		0.13	0.10	0.13	0.09	0.07

(1st line: Theoretical prediction,
2nd line: The mean of 10 trials,
3rd line: The standard deviation of 10 trials)

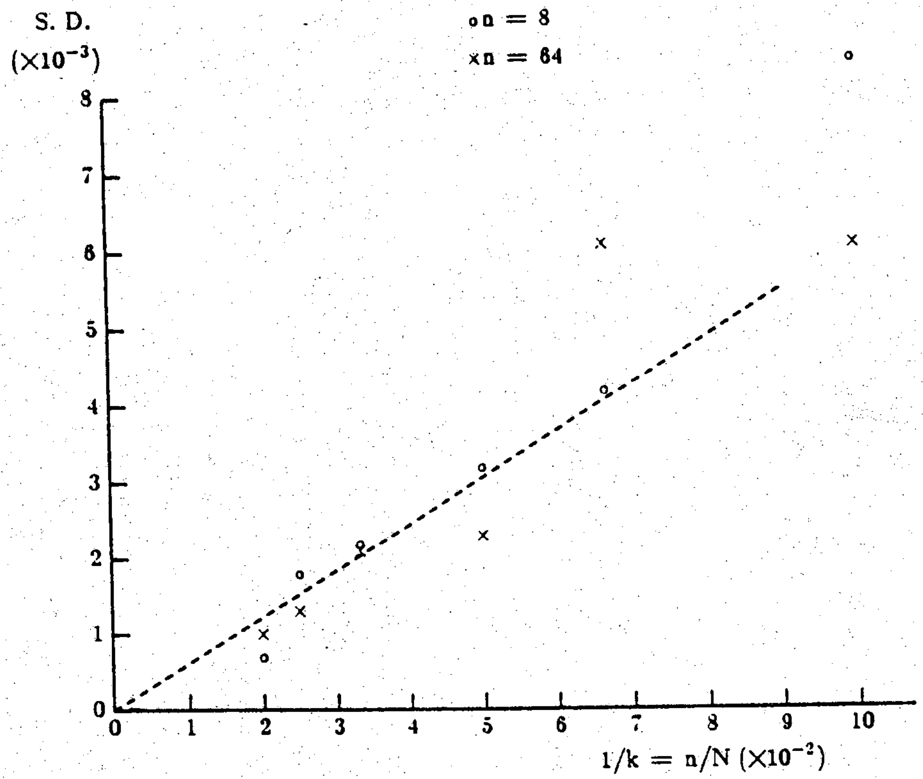


Figure 3.1 Quadratic Classifier Degradation for I-I.
(Standard deviation vs. n/N)

3.2.4 Effect of Independent Design and Test Samples

When both design and test sample sizes are finite, the error is expressed as

$$\hat{\epsilon} = \frac{1}{2} + \frac{P_1}{N_1}\sum_{j=1}^{N_1}\hat{\alpha}_j^{(1)} - \frac{P_2}{N_2}\sum_{j=1}^{N_2}\hat{\alpha}_j^{(2)} \tag{3.23}$$

where

$$\hat{\alpha}_j^{(i)} = \frac{1}{2\pi}\int_{-\infty}^{+\infty}\frac{e^{j\omega\hat{h}(X_j^{(i)})}}{j\omega}d\omega \tag{3.24}$$

That is, the randomness comes from $\hat{h}$ due to the finite design samples as well as from the test samples $X_j^{(i)}$.

The expected value and variance of $\hat{\epsilon}$ can be computed as follows:

$$\bar{\epsilon} = E\{\hat{\epsilon}\} = E_t E_d\{\hat{\epsilon}\} = \frac{1}{2} + P_1\bar{\alpha}_1 - P_2\bar{\alpha}_2 \tag{3.25}$$

where

$$\bar{\alpha}_i = \frac{1}{2\pi}\int_S\int_{-\infty}^{+\infty}\frac{E_d\{e^{j\omega\hat{h}(X)}\}}{j\omega}p_i(X)d\omega dX$$

$$= \begin{cases} \bar{\epsilon}_1 - \dfrac{1}{2} & \text{for } i=1 \\[2ex] \dfrac{1}{2} - \bar{\epsilon}_2 & \text{for } i=2 \end{cases} \tag{3.26}$$

Substituting (3.26) into (3.25),

$$\bar{\epsilon} = P_1\bar{\epsilon}_1 + P_2\bar{\epsilon}_2 \tag{3.27}$$

This average error is the same as the error of (3.20). That is, the bias of the error due to finite design and test samples is identical to the bias due to finite design samples alone. Finite test samples do not contribute to the bias.

The variance of $\hat{\epsilon}$ can be obtained from (3.23) as

$$Var\{\hat{\epsilon}\} = P_1^2[\frac{1}{N_1}Var\{\hat{\alpha}_j^{(1)}\} + (1-\frac{1}{N_1})Cov\{\hat{\alpha}_j^{(1)}\hat{\alpha}_k^{(1)}\}]$$

$$+ P_2^2[\frac{1}{N_2}\mathrm{Var}\{\hat{\alpha}_j^{(2)}\} + (1-\frac{1}{N_2})\mathrm{Cov}\{\hat{\alpha}_j^{(2)}\hat{\alpha}_k^{(2)}\}]$$

$$- 2P_1P_2\mathrm{Cov}\{\hat{\alpha}_j^{(1)}\hat{\alpha}_k^{(2)}\} \qquad (3.28)$$

where

$$\mathrm{Var}\{\hat{\alpha}_j^{(i)}\} = E\{[\frac{1}{2\pi}\int_{-\infty}^{+\infty}\frac{e^{j\omega\hat{h}(X)}}{j\omega}d\omega]^2\} - (\overline{\epsilon}_i - \frac{1}{2})^2$$

$$= \frac{1}{4} - (\overline{\epsilon}_i - \frac{1}{2})^2$$

$$= \overline{\epsilon}_i(1 - \overline{\epsilon}_i) \qquad (3.29)$$

$$\mathrm{Cov}\{\hat{\alpha}_j^{(i)}\hat{\alpha}_k^{(\ell)}\} = \frac{1}{4\pi^2}\int\!\!\int_{S_xS_y}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\frac{E_d\{e^{j\omega_1\hat{h}(X)}e^{j\omega_2\hat{h}(Y)}\}}{j\omega_1 j\omega_2}p_i(X)p_\ell(Y)d\omega_1 d\omega_2 dXdY$$

$$-\overline{\alpha}_i\overline{\alpha}_\ell \qquad (3.30)$$

The second line of (3.29) can be derived from the first line as is seen in (3.10). From (3.30), a portion of (3.28) can be expressed as

$$P_1^2\mathrm{Cov}\{\hat{\alpha}_j^{(1)}\hat{\alpha}_k^{(1)}\} + P_2^2\mathrm{Cov}\{\hat{\alpha}_j^{(2)}\hat{\alpha}_k^{(2)}\} - 2P_1P_2\mathrm{Cov}\{\hat{\alpha}_j^{(1)}\hat{\alpha}_k^{(2)}\}$$

$$= \frac{1}{4\pi^2}\int\!\!\int_{S_xS_y}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\frac{E_d\{e^{j\omega_1\hat{h}(X)}e^{j\omega_2\hat{h}(Y)}\}}{j\omega_1 j\omega_2}\tilde{p}(X)\tilde{p}(Y)d\omega_1 d\omega_2 dXdY - (\overline{\epsilon}-\frac{1}{2})^2$$

$$= \mathrm{Var}_d\{\hat{\epsilon}\} \qquad (3.31)$$

where $\mathrm{Var}_d\{\hat{\epsilon}\}$ is the same one as (3.22). On the other hand, (3.30) can be approximated as

$$\mathrm{Cov}\{\hat{\alpha}_j^{(i)}\hat{\alpha}_k^{(\ell)}\} \cong \frac{1}{4\pi^2}\int\!\!\int_{S_xS_y}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}E_d\{\Delta h(X)\Delta h(Y)\}$$

$$\times e^{j\omega_1 h(X)} e^{j\omega_1 h(Y)} p_i(X) p_\ell(Y) d\omega_1 d\omega_2 dX dY$$

$$= \int_{h(X)=0} \int_{h(Y)=0} E_d\{\Delta h(X) \Delta h(Y)\} p_i(X) p_\ell(X) dX dY$$

$$\sim \frac{1}{\mathcal{N}} \tag{3.32}$$

Eq. (3.32) is proportional to $1/\mathcal{N}$ because $E_d\{\Delta h(X) \Delta h(Y)\}$ is proportional to $1/\mathcal{N}$.

Substituting (3.29)-(3.32) into (3.28), and ignoring the terms proportional to $1/N_i\mathcal{N}$,

$$\mathrm{Var}\{\hat{\epsilon}\} \cong P_1^2 \frac{\overline{\epsilon}_1(1-\overline{\epsilon}_1)}{N_1} + P_2^2 \frac{\overline{\epsilon}_2(1-\overline{\epsilon}_2)}{N_2} + \mathrm{Var}_d\{\hat{\epsilon}\} \tag{3.33}$$

As we discussed in Section 3.2.3, $\mathrm{Var}_d\{\hat{\epsilon}\}$ is proportional to $1/\mathcal{N}^{\,2}$ when the Bayes classifier is used for Gaussian distributions. Therefore, $\mathrm{Var}\{\hat{\epsilon}\}$ of (3.33) is dominated by the first two terms which are due to the finite test set. A comparison of (3.33) and (3.13) shows that the effect of the finite design set appears in $\overline{\epsilon}_1$ and $\overline{\epsilon}_2$ of (3.33) instead of ϵ_1 and ϵ_2 of (3.13). That is, the bias due to the finite design set increases the variance proportionally. However, since $\overline{\epsilon}_i - \epsilon_1 \sim 1/\mathcal{N}$, this effect can be ignored. It should be noted that $\mathrm{Var}_d\{\hat{\epsilon}\}$ could be proportional to $1/\mathcal{N}$ if the classifier is not the Bayes.

Thus, we can draw the following conclusions from (3.27) and (3.33). When both design and test sets are finite,

1. the bias of the classification error comes entirely from the finite design set, and
2. the variance comes predominantly from the finite test set.

3.3 Dependent Design and Test Sets

In the previous section, we assumed that the design and test sets were finite and mutually independent. When only one set of samples is available, independence can be achieved by using either the holdout method or the leave-one-out method. In the holdout method, the available sample set is divided into two groups; one group is used for designing the classifier and the other for testing the classifier. The ratio of design sample size to test sample size must be determined by the desired bias and variance of the estimated error, as derived in Section 3.2.4. On the other hand, in the leave-one-out method, each sample is tested by the classifier which was designed using the remaining samples [13]. With N available samples, the test sample size is N and the design sample size is N$-$1($\cong$ N). Experimental results have confirmed that the holdout method with equal sample sizes for design and test gives the same bias and variance as the leave-one-out method.

It has been shown that the above procedures tend to give a larger error than the true one. The true error is the error of the classifier designed using the true distributions, tested with the true distributions. On the other hand, an error smaller than the true one can be obtained by the resubstitution method, in which all available samples are used to design the classifier and the same sample set is used to test the classifier. Since the resubstitution and leave-one-out methods can be carried out simultaneously without additional computation time [10], it is a common practice to compute both estimates to obtain upper and lower bounds of the true error.

When the resubstitution method in used, the design and test sample sets are no longer independent. In this section, we would like to address the dependency of the design and test sample sets. The bias and variance of the resubstitution error and the statistical properties of the bias between the resubatition and leave-one-out errors depend on the classifiers to be used. Therefore, in this section, we limit our discussions to parametric classifiers such as the quadratic and linear classifiers. Extending this discussion to other types of classifiers could be handled in a similar way.

3.3.1 Modifications of M and Σ

Let us assume that the expected vector, M, and covariance matrix, Σ, of a distribution are estimated from the available sample set. $X_1,,X_{N-1}$ by the sample mean and sample covariance as

$$\hat{M} = \frac{1}{N-1}\sum_{i=1}^{N-1} X_i \qquad (3.34)$$

$$\Sigma = \frac{1}{N-2}\sum_{i=1}^{N-1}(X_i-\hat{M})(X_i-\hat{M})^T \qquad (3.35)$$

When an additional sample Y is used, the above estimates are modified as

$$\hat{M}_R = \frac{1}{N}[(N-1)\hat{M}+Y] = \hat{M} + \frac{1}{N}(Y-\hat{M}) \tag{3.36}$$

or

$$Y - \hat{M}_R = \frac{N-1}{N}(Y-\hat{M}) \tag{3.37}$$

and

$$\Sigma_R = \frac{1}{N-1}[\sum_{i=1}^{N-1}(X_i-\hat{M}_R)(X_i-\hat{M}_R)^T + (Y-\hat{M}_R)(Y-\hat{M}_R)^T]$$

$$= \Sigma - \frac{1}{N-1}\Sigma + \frac{1}{N}(Y-\hat{M})(Y-\hat{M})^T \tag{3.38}$$

The deviations of these estimates from the true parameters, M and Σ, are

$$\Delta M_R = \Delta M + \frac{1}{N}(Y-M-\Delta M) \cong \Delta M + \frac{1}{N}(Y-M) \tag{3.39}$$

$$\Delta\Sigma_R = \Delta\Sigma - \frac{1}{N-1}(\Sigma+\Delta\Sigma) + \frac{1}{N}(Y-M-\Delta M)(Y-M-\Delta M)^T$$

$$\cong \Delta\Sigma - \frac{1}{N}\Sigma + \frac{1}{N}(Y-M)(Y-M)^T \tag{3.40}$$

ΔM and $\Delta\Sigma$ assumed to be proportional to $1/N$ and approximations were made by ignoring $1/N^2$ and higher-order terms.

With this approximation, a function of $\hat{M}_R$ and Σ_R, $f(\hat{M}_R, \Sigma_R)$, can be expanded around $f(M,\Sigma)$ as

$$f(\hat{M}_R,\Sigma_R) \cong f(M,\Sigma) + \frac{\partial f^T}{\partial M}\Delta M_R + \mathrm{tr}\frac{\partial f}{\partial \Sigma}\Delta\Sigma_R \tag{3.41}$$

In the general Taylor series expansion, components of the second-order terms are also proportional to $1/N$. Using (3.39) and (3.40),

$$\Delta M_R \Delta M_R^T \cong \Delta M\ \Delta M^T + \frac{2}{N}(Y-M)\Delta M^T \tag{3.42}$$

$$\Delta\Sigma_R \Delta\Sigma_R^T \cong \Delta\Sigma\ \Delta\Sigma^T - \frac{2}{N}[\Sigma-(Y-M)(Y-M)^T]\Delta\Sigma^T \tag{3.43}$$

$$\Delta M_R \Delta\Sigma_R^T \cong \Delta M\ \Delta\Sigma^T - \frac{1}{N}\Delta M\Sigma^T + \frac{1}{N}\Delta M(Y-M)(Y-M)^T$$

$$+ \frac{1}{N}(Y-M)\Delta\Sigma^T \tag{3.44}$$

In the above expressions, each 1/N term contains a random variable which is assumed to be proportional to 1/N, making the entire term proportional to $1/N^2$. Thus, (3.41) is consistent with the approximations made by ignoring $1/N^2$ and higher-order terms.

Substituting (3.39) and (3.40) into (3.41),

$$\begin{aligned} f(\hat{M}_R,\Sigma_R) \cong\ & [f(M,\Sigma) + \frac{\partial f^T}{\partial M}\Delta M + \mathrm{tr}\frac{\partial f}{\partial \Sigma}\Delta\Sigma] \\ & + \frac{1}{N}[\frac{\partial f^T}{\partial M}(Y-M) + \mathrm{tr}\frac{\partial f}{\partial \Sigma}\{(Y-M)(Y-M)^T-\Sigma\}] \\ & = f(\hat{M},\Sigma) + \frac{1}{N}[\frac{\partial f^T}{\partial M}(Y-M) + \mathrm{tr}\frac{\partial f}{\partial \Sigma}\{(Y-M)(Y-M)^T-\Sigma\}] \end{aligned} \tag{3.45}$$

Note that the difference between the two random variables $f(\hat{M}_R,\Sigma_R)$ and $f(\hat{M},\Sigma)$ is not random, as long as Y is fixed and the first-order approximation is valid.

Example: Let us examine the case where f is given by

$$f(M,\Sigma) = \frac{1}{2}(Y-M)^T\Sigma^{-1}(Y-M) + \frac{1}{2}\ln|\Sigma| \tag{3.46}$$

Then,

$$\frac{\partial f}{\partial M} = -\Sigma^{-1}(Y-M) \tag{3.47}$$

$$\frac{\partial f}{\partial \Sigma} = -\frac{1}{2}\Sigma^{-1}(Y-M)(Y-M)^T\Sigma^{-1} + \frac{1}{2}\Sigma^{-1} \tag{3.48}$$

Therefore,

$$f(\hat{M}_R,\Sigma_R) - f(\hat{M},\Sigma) \cong -\frac{1}{2N}[d^4(Y)+n] \tag{3.49}$$

where

$$d^2(Y) = (Y-M)^T\Sigma^{-1}(Y-M) \tag{3.50}$$

3.3.2 Quadratic Classifiers

In this section, the quadratic classifier of (3.14) is discussed. Using (3.46), (3.14) can be rewritten as

$$h(X) = f(M_1,\Sigma_1) - f(M_2,\Sigma_2) \tag{3.51}$$

When a sample X from ω_1 is tested in the resubstitution method,

$$\begin{aligned}\hat{h}_R(X) &= f(\hat{M}_{1R},\Sigma_{1R}) - f(\hat{M}_2,\Sigma_2)\\ &\cong f(\hat{M}_1,\Sigma_1) - \frac{1}{2N_1}[d_1^4(X)+n] - f(\hat{M}_2,\Sigma_2)\\ &= \hat{h}_L(X) - \frac{1}{2N_1}[d_1^4(X)+n] \quad \text{for } X\epsilon\omega_1\end{aligned} \tag{3.52}$$

Likewise, when X comes from ω_2,

$$\hat{h}_R(X) \cong \hat{h}_L(X) + \frac{1}{2N_2}[d_2^4(X)+n] \quad \text{for } X\epsilon\omega_2 \tag{3.53}$$

where $\hat{h}_R(X)$ and $\hat{h}_L(X)$ are the discriminant functions for the resubstitution and leave-one-out methods, N_i is the sample size for ω_i and $d_i^2(X) = (X-M_i)^T\Sigma_i^{-1}(X-M_i)$.

Now, the resubstitution error can be computed by (3.23) and (3.24) with $\hat{h}$ of (3.24) replaced by $\hat{h}_R$ of either (3.52) or (3.53) depending on i=1 or 2. The result is

$$\begin{aligned}\hat{\epsilon}_R &= \frac{1}{2} + \frac{P_1}{N_1}\sum_{j=1}^{N_1}\frac{1}{2\pi}\int_{-\infty}^{+\infty}\frac{e^{j\omega\hat{h}_R(X_j^{(1)})}}{j\omega}d\omega - \frac{P_2}{N_2}\sum_{j=1}^{N_2}\frac{1}{2\pi}\int_{-\infty}^{+\infty}\frac{e^{j\omega\hat{h}_R(X_j^{(2)})}}{j\omega}d\omega\\ &\cong \hat{\epsilon}_L - [\frac{P_1}{N_1^2}\sum_{j=1}^{N_1}\beta_j^{(1)} + \frac{P_2}{N_2^2}\sum_{j=1}^{N_2}\beta_j^{(2)}]\end{aligned} \tag{3.54}$$

where

$$\beta_j^{(i)} = \frac{1}{2\pi}\int_{-\infty}^{+\infty}\frac{d_i^4(X_j^{(i)})+n}{2}e^{j\omega\hat{h}_L(X_j^{(i)})}d\omega \tag{3.55}$$

In order to obtain the second line of (3.54), an approximation of $e^{j\omega a/N} \cong 1+j\omega a/N$ is used. Also, note that, in the leave-one-out method, the design and test samples are independent and, therefore, the discussion of Section 3.2.4 can be applied without modification. However, in the leave-one-out method, the number of design samples ($\mathcal{N}_i$) is always the same as the number of test samples (N_i).

Now, the statistical properties of the bias, $\hat{\epsilon}_b = \hat{\epsilon}_L - \hat{\epsilon}_R$, can be studied. The expected value of $\hat{\epsilon}_b$ is

$$E\{\hat{\epsilon}_b\} \cong \frac{P_1}{N_1}\beta_1 + \frac{P_2}{N_2}\beta_2 \tag{3.56}$$

where

$$\beta_i = \frac{1}{2\pi}\int_S \int_{-\infty}^{+\infty} \frac{d_i^4(X)+n}{2} E_d\{e^{j\omega \hat{h}_L(X)}\} p_i(X) d\omega dX \tag{3.57}$$

And, the variance of $\hat{\epsilon}_b$ is

$$\begin{aligned} Var\{\hat{\epsilon}_b\} = & \frac{P_1^2}{N_1^2}[\frac{1}{N_1}Var\{\beta_j^{(1)}\} + (1-\frac{1}{N_1})Cov\{\beta_j^{(1)}\beta_k^{(1)}\}] \\ & + \frac{P_2^2}{N_2^2}[\frac{1}{N_2}Var\{\beta_j^{(2)}\} + (1-\frac{1}{N_2})Cov\{\beta_j^{(2)}\beta_k^{(2)}\}] \\ & + \frac{2P_1P_2}{N_1N_2}Cov\{\beta_j^{(1)}\beta_k^{(2)}\} \end{aligned} \tag{3.58}$$

The explicit expression for β_i of (3.57) can be obtained by using the same technique used to compute $\bar{\epsilon}$ in Chapter 2, if two distributions are Gaussian with $M_1 = 0$, $M_2 = M$ and $\Sigma_1 = \Sigma_2 = I$ and the quadratic classifier of (14) is used. For $N_1 = N_2 = N$

$$E_d\{e^{j\omega \hat{h}_i(X)}\} \cong e^{j\omega h(X)}[1 + \frac{1}{N}a] \cong e^{j\omega h(X)} \tag{3.59}$$

$$e^{j\omega h(X)}p_1(X) = \frac{\sqrt{2\pi}}{\sqrt{M^TM}} e^{\frac{-M^TM}{8}} N_\omega(-\frac{j}{2}, \frac{1}{M^TM}) N_x(j\omega M,I) \tag{3.60}$$

$$e^{j\omega h(X)}p_2(X) = \frac{\sqrt{2\pi}}{\sqrt{M^TM}} e^{\frac{-M^TM}{8}} N_\omega(\frac{j}{2}, \frac{1}{M^TM}) N_x((1+j\omega)M,I) \tag{3.61}$$

where a is a constant given in Chapter 2. $N_\omega(d,k)$ and $N_x(D,K)$ are Gaussian density functions of ω and X with the expected value d and variance k for N_ω, and the expected vector D and covariance matrix K for N_X. Thus, the integration of (3.57) merely involves computing the moments of the Gaussian distributions of (3.60) and (3.61), resulting in

$$\beta_i \cong \frac{1}{2\sqrt{2\pi M^T M}} e^{\frac{-M^T M}{8}} \left[n^2+(1+M^T M/2)n+[(M^T M)^2/16-M^T M/2-1] \right] \quad (3.62)$$

The first lines of Table 3.2 show the values of $E\{\hat{\varepsilon}_b\}$ computed from (3.56) and (3.62) with $M^T M = 2.56^2$ and $P_1 = P_2 = 0.5$ for various k (= N/n) and n. The theoretical values are compared with the experimental ones in the second lines. The experiments were conducted by generating N samples, estimating M_i and Σ_i, designing the quadratic classifier of (3.14), estimating the resubstitution and leave-one-out errors and computing the bias between them. The experiment was repeated 10 times and the average and standard deviation of the estimated biases are listed in the second and third lines. As Table 3.2 shows, the first and second lines are close, confirming the validity of our discussion.

An important fact is that, from (3.56) and (3.62), $E\{\hat{\varepsilon}_b\}$ is roughly proportional to n^2/N for large n. A simpler explanation for this fact can be obtained by observing (3.57) more closely. Assuming (3.59) and carrying through the integration of (3.57) with respect to ω,

$$\beta_i \cong \int_S \frac{d_i^4(X) + n}{2} \delta(h(X))p_i(X)dX$$

$$= \int_{h(X)=0} \frac{d_i^4(X) + n}{2} p_i(X)dX \quad (3.63)$$

It is well known that $d_i^2(X)$ is χ^2-distributed with an expected value of n and standard deviation of $\sqrt{2n}$, if X is Gaussianly distributed. Particularly when n is large, $d_i^2(X)$ on the classification boundary should be n times some number not far from 1. That is, $d_i^4(X)$ is close to n^2. Thus, β_i should be proportional to n^2.

The analysis of the variance (3.58) is more complex. Though the order of magnitude may not be immediately clear from (3.58), our experimental results, presented in Fig. 3.2 and the third line of Table 3.2, show that the standard deviation is roughly proportional to 1/N. The intuitive explanation should be the same as that presented in Section 3.2.3.

Table 3.2 Bias between leave-one-out and resubstitution errors for I-I (%).

		n				
		4	8	16	32	64
	3	9.00	13.79	23.03	41.34	77.87
		13.33	15.42	19.69	22.86	30.29
		7.03	5.22	4.12	4.26	3.40
	5	5.40	8.27	13.82	24.80	46.72
		7.50	9.25	10.75	17.75	24.47
		4.56	3.24	2.28	2.69	1.53
	10	2.70	4.14	6.91	12.40	23.36
		2.25	4.63	6.34	9.58	16.01
		1.84	2.02	1.59	1.61	1.24
	15	1.80	2.76	4.61	8.27	15.57
		1.33	3.13	4.42	7.44	11.92
k		0.90	1.29	0.87	0.47	1.18
	20	1.35	2.07	3.45	6.20	11.68
		1.38	2.09	3.14	5.05	9.56
		1.05	1.00	0.64	0.53	0.45
	30	0.90	1.38	2.30	4.13	7.79
		0.63	1.58	2.39	3.94	6.41
		0.45	0.52	0.41	0.35	0.33
	40	0.67	1.03	1.73	3.10	5.84
		0.44	1.08	1.55	2.96	5.21
		0.30	0.39	0.30	0.30	0.36
	50	0.54	0.83	1.38	2.48	4.67
		0.30	0.75	1.38	2.29	4.27
		0.23	0.23	0.37	0.25	0.25

(1st line: Theoretical prediction,
2nd line: The mean of 10 trials,
3rd line: The standard deviation of 10 trials)

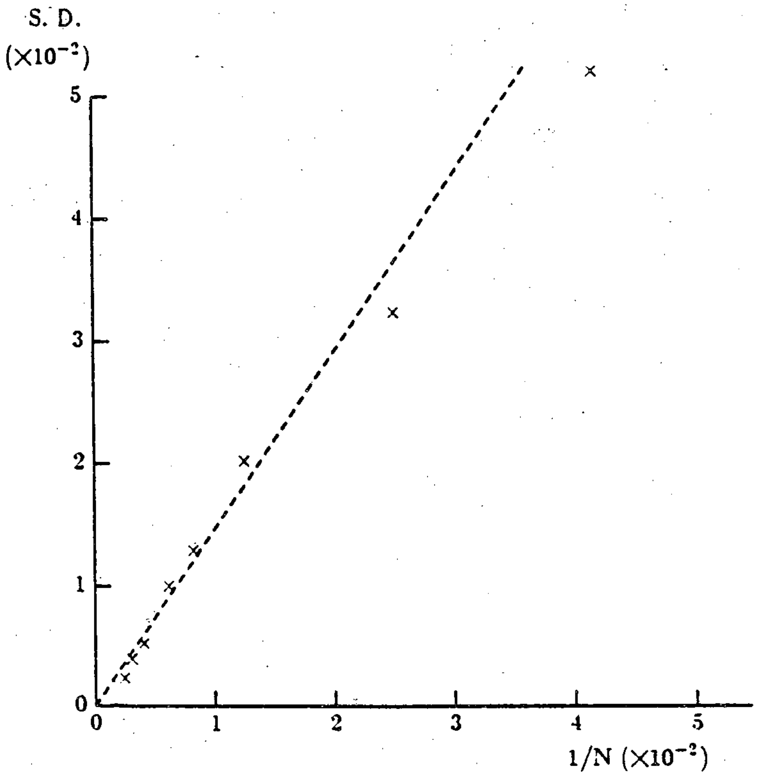


Figure 3.2 Bias between Leave-one-out and Resubstitution Errors for I-I
(Standard deviation vs. 1/N for n=8)

3.3.3 Effect of Outliers

It is widely believed in the pattern recognition field that classifier performance can be improved by removing outliers, points far from a class's inferred mean which seem to distort the distribution. The approach used in Section 3.3.1 to analyze the difference between the resubstitution and leave-one-out parameters can be extended to handle the effect of a single point of the design set on classifier performance.

As in (3.34)-(3.38), assume that N-1 samples have been used to estimate a distribution's parameters $(\hat{M},\Sigma)$ and that these estimates will now be modified by including a new point, Y. These new estimates $(\hat{M}_y,\Sigma_y)$ are defined by (3.36) and (3.38). The approximations in (39)-(44) are still valid, so (3.45) can also be used. For the quadratic classifier, (3.47) and (3.48) can be substituted into (3.45) to yield

$$f(\hat{M}_y,\Sigma_y) - f(\hat{M},\Sigma) = \frac{1}{2N}[-2(Y-M)^T\Sigma^{-1}(X-M)+(Y-M)^T\Sigma^{-1}(Y-M)-n$$

$$-\{(Y-M)^T\Sigma^{-1}(X-M)\}^2 + (X-M)^T\Sigma^{-1}(X-M)]$$

$$= \frac{1}{N}g(X,Y) \tag{3.64}$$

The corresponding change in the discriminant function for $Y\epsilon\omega_1$ can be found by inserting (3.64) into (3.51)

$$\hat{h}_y(X) = f(\hat{M}_{1y},\Sigma_{1y}) - f(\hat{M}_2,\Sigma_2)$$

$$\cong f(\hat{M}_1,\Sigma_1) + \frac{1}{N}g_1(X,Y) - f(\hat{M}_2,\Sigma_2)$$

$$= \hat{h}(X) + \frac{1}{N}g_1(X,Y) \quad \text{for } Y\epsilon\omega_1 \tag{3.65}$$

Likewise, when Y comes from ω_2,

$$\hat{h}_y(X) = \hat{h}(X) - \frac{1}{N}g_2(X,Y) \quad \text{for } Y\epsilon\omega_2 \tag{3.66}$$

where g_i indicates that M_i and Σ_i are used instead of M and Σ in (3.64).

When this modified classifier is used on an independent set of test samples, the result is, using (3.19),

$$\begin{aligned}\hat{\varepsilon}_y &= \frac{1}{2} + \frac{1}{2\pi}\int_S\int_{-\infty}^{+\infty} \frac{e^{j\omega\hat{h}_y(X)}}{j\omega}\tilde{p}(X)d\omega dX \\ &\cong \frac{1}{2} + \frac{1}{2\pi}\int_S\int_{-\infty}^{+\infty} \frac{e^{j\omega\hat{h}(X)}}{j\omega}[1 \pm \frac{j\omega}{N}g_i(X,Y)]\tilde{p}(X)d\omega dX \\ &= \hat{\varepsilon} \pm \frac{1}{2\pi}\int_S\int_{-\infty}^{+\infty} e^{j\omega\hat{h}(X)}\frac{1}{N}g_i(X,Y)\tilde{p}(X)d\omega dX \\ &\cong \hat{\varepsilon} \pm \frac{1}{2\pi}\int_S\int_{-\infty}^{+\infty} e^{j\omega h(X)}\frac{1}{N}g_i(X,Y)\tilde{p}(X)d\omega dX \qquad (3.67)\end{aligned}$$

where + and $i=1$ are used for $Y\epsilon\omega_1$ and − and $i=2$ are for $Y\epsilon\omega_2$. The approximation in the last line involves expressing $e^{j\omega\hat{h}(X)}$ in terms of $e^{j\omega h(X)}$ and ignoring terms smaller than 1/N. Unlike the case of the resubstitution error, (3.67) keeps $\tilde{p}(X)$ in its integrand. This makes the integral in (3.67) particularly easy to handle. If the quadratic classifier is the Bayes classifier, the integration with respect to ω results in

$$\Delta\hat{\varepsilon}_y = \pm \int_S \delta(h(X))\frac{1}{N}g_i(X,Y)\tilde{p}(X)dX = 0 \qquad (3.68)$$

That is, as long as $\tilde{p}(X) = 0$ at $h(X) = 0$, the effect of an individual sample is negligible. Even if the quadratic classifier is not optimal, $\Delta\hat{\varepsilon}_y$ is dominated by a 1/N term. Thus, as one would expect, as the number of samples becomes larger, the effect of an individual sample diminishes.

These results were confirmed in three sets of experiments. The first was the mean difference case used earlier. In the second experiment, the two classes share a mean, but have different covariances (I for ω_1, 4I for ω_2). The third experiment used Standard Data from [10] where the classes differ widely in both the mean and the covariance. Eight-dimensional data was used in each case.

The experiments were run in the following manner. N samples were generated for each class. Then, an additional sample, Y, was generated from class 1 and scaled to a specific normalized distance from the mean. Classifiers were designed with and without Y and were tested on the true distributions using Novak's program computing the performance of a classifier with a given test distribution [11]. This procedure was repeated 10 times for each particular value of N. The entire process was run a number of times with varying distances. Experimental results are presented in Tables 3.3, 3.4, and 3.5. Notice that even when the squared distance is much larger than its expected value, n, the outlier's effect is still negligible.

3.4 Bootstrap Methods

As an alternative to the holdout and leave-one-out error estimates, Efron [14] has suggested using a bootstrap technique to estimate the optimistic bias of the resubstitution error and, in turn, to estimate the expected error rate for a given decision rule. In the bootstrap procedure, one assumes that the existing sample set represents the true distributions. That is, these density functions consist of impulses located at the existing sample points

$$p_i^*(X) = \frac{1}{N_i} \sum_{j=1}^{N_i} \delta(X-X_j^{(i)}) \quad i = 1,2 \tag{3.69}$$

where * indicates something related to the bootstrap operation. Note that in this section, $X_j^{(i)}$ is considered a given fixed vector and is not random as it was in the previous sections.

When samples are drawn from $p_i^*(X)$ randomly, we select only the existing sample points with random frequencies. Thus, the N_i samples drawn from $p_i^*(X)$ form a density function

$$\hat{p}_i^*(X) = \sum_{j=1}^{N_i} \theta_j^{(i)} \delta(X-X_j^{(i)}) \quad i = 1,2 \tag{3.70}$$

Within each class, the $\theta_j^{(i)}$'s are identically distributed under the condition $\sum_{j=1}^{N_i} \theta_j^{(i)} = 1$. Their statistical properties are known [14]:

$$E\{\theta_j^{(i)}\} = \frac{1}{N_i} \tag{3.71}$$

$$E\{\theta_j^{(i)}\theta_k^{(i)}\} = \frac{1}{N_i^2}\delta_{jk} - \frac{1}{N_i^3} \tag{3.72}$$

$$E\{\theta_j^{(i)}\theta_k^{(\ell)}\} = 0 \quad \text{for } i \neq \ell \tag{3.73}$$

The holdout error in the bootstrap procedure, $\hat{\epsilon}_H^*$, is obtained by generating samples, designing a classifier based on $\hat{p}_i^*(X)$ and testing $p_i^*(X)$. On the other hand, the resubstitution error, $\hat{\epsilon}_R^*$, is computed by testing

Table 3.3 Bias between error without outlier and error including outlier for various outlier distances from the class 1 mean for I-I ($\epsilon^* = 10\%$).

N	ERROR WITHOUT OUTLIER (%)	BIAS BETWEEN ERROR WITHOUT OUTLIER AND ERROR INCLUDING OUTLIER (%)			
		$d^2 = n/2$	$d^2 = n$	$d^2 = 2n$	$d^2 = 3n$
24	20.18	0.519	0.689	0.769	0.762
40	15.61	0.124	0.211	0.279	0.274
80	12.04	0.029	0.035	0.027	0.018
120	11.71	0.008	0.012	0.011	0.003
160	11.04	0.006	0.010	0.014	0.013
240	10.74	0.004	0.006	0.010	0.001
320	10.53	0.004	0.006	0.009	0.011
400	10.34	-.001	-.001	-.003	-.001

Table 3.4 Bias between error without outlier and error including outlier for various outlier distances from the class 1 mean for I-4I (ϵ^* = 9%).

N	ERROR WITHOUT OUTLIER (%)	BIAS BETWEEN ERROR WITHOUT OUTLIER AND ERROR INCLUDING OUTLIER (%)			
		$d^2 = n/2$	$d^2 = n$	$d^2 = 2n$	$d^2 = 3n$
24	23.53	0.792	1.213	1.451	1.356
40	16.19	0.222	0.423	0.619	0.658
80	11.79	0.025	0.060	0.091	0.083
120	10.83	0.015	0.032	0.047	0.045
160	10.32	-.003	0.005	0.014	0.013
240	9.92	0.003	0.012	0.025	0.034
320	9.52	0.003	0.006	0.012	0.015
400	9.41	0.000	0.000	0.001	-.001

Table 3.5 Bias between error without outlier and error including outlier for various outlier distances from the class 1 mean for standard data ($\epsilon^* = 1.9\%$).

N	ERROR WITHOUT OUTLIER (%)	BIAS BETWEEN ERROR WITHOUT OUTLIER AND ERROR INCLUDING OUTLIER (%)			
		$d^2 = n/2$	$d^2 = n$	$d^2 = 2n$	$d^2 = 3n$
24	5.58	0.374	0.555	0.664	0.673
40	3.70	0.054	0.088	0.110	0.103
80	2.54	0.005	0.007	0.008	0.003
120	2.35	0.005	0.007	0.007	0.005
160	2.25	-.001	0.000	0.001	0.001
240	2.14	0.001	0.002	0.003	0.004
320	2.08	0.000	0.000	0.001	0.001
400	2.05	0.000	0.000	0.000	0.000

$\hat{p}_i^*(X)$. The bias between them can be expressed by

$$\hat{\epsilon}_b^* = \hat{\epsilon}_H^* - \hat{\epsilon}_R^*$$

$$= \frac{1}{2\pi} \int_S \int_{-\infty}^{+\infty} \frac{e^{j\omega\hat{h}^*(X)}}{j\omega} \left[P_1 \sum_{j=1}^{N_1} \left(\frac{1}{N_1} - \theta_j^{(1)} \right) \delta(X - X_j^{(1)}) \right.$$

$$\left. - P_2 \sum_{j=1}^{N_2} \left(\frac{1}{N_2} - \theta_j^{(2)} \right) \delta(X - X_j^{(2)}) \right] d\omega dX$$

$$= P_1 \sum_{j=1}^{N_1} \gamma_j^{(1)} - P_2 \sum_{j=1}^{N_2} \gamma_j^{(2)} \qquad (3.74)$$

where

$$\gamma_j^{(i)} = -\frac{\Delta\theta_j^{(i)}}{2\pi} \int_{-\infty}^{+\infty} \frac{e^{j\omega\hat{h}^*(X_j^{(i)})}}{j\omega} d\omega \qquad (3.75)$$

and $\Delta\theta_j^{(i)} = \theta_j^{(i)} - \frac{1}{N_i}$.

When a quadratic classifier is used, $\hat{h}^*(X)$ in (75) becomes

$$\hat{h}^*(X) = f(\hat{M}_1^*, \Sigma_1^*) - f(\hat{M}_2^*, \Sigma_2^*) \qquad (3.76)$$

where $f(\cdot,\cdot)$ is defined in (3.46). The bootstrap parameters, $\hat{\mathbf{M}}_i^*$ and Σ_i^* are

$$\hat{\mathbf{M}}_i^* = \sum_{j=1}^{N_i} \theta_j^{(i)} X_j^{(i)} \qquad (3.77)$$

$$\Sigma_i^* = \sum_{j=1}^{N_i} \theta_j^{(i)} (X_j^{(i)} - \hat{M}_i)(X_j^{(i)} - \hat{M}_i)^T \qquad (3.78)$$

Note that $\hat{M}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} X_j^{(i)}$ is used to compute Σ_i^*. $\hat{M}_i$ is available in the bootstrap operation and the use of $\hat{M}_i$ instead of $\hat{\mathbf{M}}_i^*$ simplifies the discussion significantly. Their expectations are

$$E_*\{\hat{\mathbf{M}}_i^*\} = \sum_{j=1}^{N_i} E\{\theta_j^{(i)}\} X_j^{(i)} = \frac{1}{N_i} \sum_{j=1}^{N_i} X_j^{(i)} = \hat{M}_i \qquad (3.79)$$

$$E_*\{\Sigma_i^*\} = \sum_{j=1}^{N_i} E\{\theta_j^{(i)}\}(X_j^{(i)}-\hat{M}_i)(X_j^{(i)}-\hat{M}_i)^T$$

$$= \frac{N_i-1}{N_i}\Sigma_i \cong \Sigma_i \tag{3.80}$$

where E_* indicates the expectation with respect to the θ's.

$f(\hat{M}_i^*,\Sigma_i^*)$ can be expanded around $f(\hat{M}_i,\Sigma_i)$ by the Taylor series as

$$f(\hat{M}_i^*,\Sigma_i^*) \cong f(\hat{M}_i,\Sigma_i) + \frac{\partial f^T}{\partial \hat{M}_i}\Delta\hat{M}_i + \mathrm{tr}\,\frac{\partial f}{\partial \Sigma_i}\Delta\Sigma_i \tag{3.81}$$

where $\Delta\hat{M}_i = \hat{M}_i^* - \hat{M}_i$ and $\Delta\Sigma_i = \Sigma_i^* - \Sigma_i$. Since $\hat{h}(X) = f(\hat{M}_1,\Sigma_1) - f(\hat{M}_2,\Sigma_2)$,

$$\Delta\hat{h}(X) = \hat{h}^*(X) - \hat{h}(X)$$

$$\cong \frac{\partial f^T}{\partial \hat{M}_1}\Delta\hat{M}_1 - \frac{\partial f^T}{\partial \hat{M}_2}\Delta\hat{M}_2 + \mathrm{tr}\left(\frac{\partial f}{\partial \Sigma_1}\Delta\Sigma_1 - \frac{\partial f}{\partial \Sigma_2}\Delta\Sigma_2\right) \tag{3.82}$$

The partial derivatives of (3.82) can be obtained by (3.47) and (3.48).

3.4.1 Bootstrap Expectation

Using the approximation of (3.21), (3.75) can be approximated as

$$\gamma_j^{(i)} \cong -\frac{\Delta\theta_j^{(i)}}{2\pi}\int_{-\infty}^{+\infty}\frac{e^{j\omega\hat{h}(X_j^{(i)})}}{j\omega}[1+j\omega\Delta\hat{h}(X_j^{(i)}) + \frac{(j\omega)^2}{2}\Delta\hat{h}^2(X_j^{(i)})]d\omega \tag{3.83}$$

The third term contains third-order moments with the combination of $\Delta\theta_j^{(i)}$ and $\Delta\hat{h}^2$ and can be ignored. Thus, our analysis will focus on the first and second terms. With this in mind, substituting (3.77), (3.78) and (3.82) into (3.83) produces

$$\gamma_j^{(i)} \cong -\frac{\Delta\theta_j^{(i)}}{2\pi}\int_{-\infty}^{+\infty}\frac{e^{j\omega\hat{h}(X_j^{(i)})}}{j\omega}d\omega$$

$$-\frac{1}{2\pi}\int_{-\infty}^{+\infty}e^{j\omega\hat{h}(X_j^{(i)})}[\frac{\partial f^T}{\partial \hat{M}_1}\sum_{k=1}^{N_1}\Delta\theta_j^{(i)}\Delta\theta_k^{(1)}X_k^{(1)} - \frac{\partial f}{\partial \hat{M}_2}\sum_{k=1}^{N_2}\Delta\theta_j^{(i)}\Delta\theta_k^{(2)}X_k^{(2)}$$

$$+\sum_{k=1}^{N_1} \Delta\theta_j^{(i)}\,\Delta\theta_k^{(1)}(X_k^{(1)}-\hat{M}_1)^T \frac{\partial f}{\partial\Sigma_1}(X_k^{(1)}-\hat{M}_1)$$

$$-\sum_{k=1}^{N_2} \Delta\theta_j^{(i)}\,\Delta\theta_k^{(2)}(X_k^{(2)}-\hat{M}_2)^T \frac{\partial f}{\partial\Sigma_2}(X_k^{(2)}-\hat{M}_2)]d\omega \tag{3.84}$$

Using the partial derivatives of (3.47) and (3.48) and the expectations in (3.71)-(3.73), $E_*\{\gamma_j^{(i)}\}$ becomes

$$E_*\{\gamma_j^{(i)}\} \cong \frac{(-1)^{i+1}}{N_i^2}\frac{1}{2\pi}\int_{-\infty}^{+\infty}\frac{\hat{d}_i^4(X_j^{(i)})+n}{2}e^{j\omega\hat{h}(X_j^{(i)})}d\omega \tag{3.85}$$

where

$$\hat{d}_i^2(X) = (X-\hat{M}_i)^T\Sigma_i^{-1}(X-\hat{M}_i) \tag{3.86}$$

In the derivation of (3.85), we utilized the relationship that

$$\frac{1}{N_i}\sum_{k=1}^{N_i}(X_k^{(i)}-\hat{M}_i)^T\Sigma_i^{-1}(X_k^{(i)}-\hat{M}_i) = \text{tr } \Sigma_i^{-1}\frac{1}{N_i}\sum_{k=1}^{N_i}(X_k^{(i)}-\hat{M}_i)(X_k^{(i)}-\hat{M}_i)^T$$

$$= \frac{N_i-1}{N_i}\text{ tr } \Sigma_i^{-1}\Sigma_i$$

$$= \frac{N_i-1}{N_i}n \cong n\,.$$

Thus, the expectation of the bootstrap bias for a quadratic classifier given a sample set $S=\{X_1^{(1)},,X_{N_1}^{(1)},X_1^{(2)},,X_{N_2}^{(2)}\}$ becomes

$$E_*\{\hat{\varepsilon}_b^*|S\} = \frac{P_1}{N_1^2}\sum_{j=1}^{N_1}\beta_j^{*(1)} + \frac{P_2}{N_2^2}\sum_{j=1}^{N_2}\beta_j^{*(2)} \tag{3.87}$$

where

$$\beta_j^{*(i)} = \frac{1}{2\pi}\int_{-\infty}^{+\infty}\frac{\hat{d}_i^4(X_j^{(i)})+n}{2}e^{j\omega\hat{h}(X_j^{(i)})}d\omega \tag{3.88}$$

Note that (3.55) and (3.88) are very similar. The differences are $\hat{d}_i^2$ of (3.86) vs. d_i^2 of (3.50) and $\hat{h}$ vs. $\hat{h}_L$. $\hat{h}$ is the discriminent function designed with $\hat{M}_i$ and Σ_i, the sample mean and sample covariance of the sample set S. The test samples $X_j^{(i)}$ are the members of the same set, S. Therefore, $\hat{h}$ is the same as the resubstitution discriminant function $\hat{h}_R$ of the previous sections, while $\hat{h}_L$ is the leave-one-out discriminant function. As is shown in (3.52) and (3.53), the difference between $\hat{h}_L$ and $\hat{h}_R$ is proportional to 1/N. Thus, the difference between $e^{j\omega\hat{h}_L}$ and $e^{j\omega\hat{h}_R}$ is proportional to 1/N. Also, as (3.50) suggests, it can be shown that the difference between $\hat{d}_i^2$ and d_i^2 is proportional to 1/N. Thus, ignoring terms with 1/N, $\hat{\epsilon}_b$ of (3.54) and $E_*\{\hat{\epsilon}_b^* | S\}$ of (3.87) (note that **S** is now a random set) become equal and have the same statistical properties. Practically, this means that estimating the expected error rate using the leave-one-out and bootstrap methods should yield the same results.

These conclusions have been confirmed experimentally. For several values of N_i, 8-dimensional sample vectors were generated from the Gaussian distributions used in Section 3.3. The generated samples were bootstrapped and used to design a quadratic classifier. This classifier was then tested on the original sample set ($\hat{\epsilon}_H^*$) and the bootstrap sample set ($\hat{\epsilon}_R^*$). Each sample set (S) was bootstrapped 100 times and the results were averaged to simulate the bootstrap expectation ($E_*\{\hat{\epsilon}_b^* | S\}$.) The whole procedure was repeated 10 times to estimate the expectation with respect to the training sample set ($E_S E_*\{\hat{\epsilon}_b^* | S\}$.) Results are presented in Tables 3.6, 3.7, and 3.8. In columns 3-7, the first line of each entry is the mean of 10 trials and the second line is the standard deviation. In column 2, the first line is still the mean, but the variance is presented in the second line.

When N_i is particularly small, our approximations might not be valid and the leave-one-out and bootstrap methods may produce different results. Though the bootstrap bias estimate does seem to have a slightly smaller standard deviation (column 4 vs. column 6 of Tables 3.6-3.8), both our results and those presented in Jain, Dubes, and Chen [15] show that the leave-one-out and bootstrap methods are equivalent (column 3 vs. column 5 of Tables 3.6-3.8).

3.4.2 Bootstrap Variance

The variance with respect to the bootstrap can be evaluated in a fashion similar to (3.58)

$$\mathrm{Var}_*\{\hat{\epsilon}_b^* | S\} = P_1^2 [\sum_{j=1}^{N_1} \mathrm{Var}_*\{\gamma_j^{(1)}\} + \sum_{j=1}^{N_1} \sum_{\substack{k=1 \\ j \neq k}}^{N_1} \mathrm{Cov}_*\{\gamma_j^{(1)} \gamma_k^{(1)}\}]$$

Table 3.6 Bootstrap results for I-I ($\epsilon^* = 10\%$).

	CONVENTIONAL LEAVE-ONE-OUT & RESUBSTITUTION			BOOTSTRAP		
N	$E\{\epsilon_R\}$	$E\{\epsilon_L\}$	$E\{\epsilon_L-\epsilon_R\}$	$E_S\{\epsilon_R+E_*\{\epsilon_b^* \mid S\}\}$	$E_S E_*\{\epsilon_b^* \mid S\}$	$E_S Var_*\{\epsilon_b^* \mid S\}$
24	3.54 0.11	17.08 4.89	13.54 3.14	12.77 4.17	9.23 1.38	0.18 0.04
40	5.75 0.07	13.38 6.04	7.63 3.88	11.68 4.44	5.92 1.90	0.08 0.02
80	7.13 0.04	11.19 2.47	4.06 1.29	10.67 2.50	3.55 0.56	0.04 0.01
120	9.04 0.06	11.79 2.97	2.75 1.01	11.45 2.79	2.41 0.43	0.03 0.01
160	9.13 0.03	11.28 2.35	2.16 1.09	11.17 1.94	2.05 0.44	0.02 0.00
240	8.27 0.02	9.35 1.61	1.08 0.51	9.46 1.61	1.19 0.15	0.01 0.00
320	9.78 0.01	10.67 0.80	0.89 0.37	10.78 0.91	1.00 0.11	0.01 0.00
400	9.18 0.01	9.78 0.91	0.60 0.26	9.96 0.84	0.77 0.10	0.01 0.00

(All numbers are percentages.)

Table 3.7 Bootstrap results for I-4I ($\epsilon^* = 9\%$).

	CONVENTIONAL LEAVE-ONE-OUT & RESUBSTITUTION			BOOTSTRAP		
N	$E\{\epsilon_R\}$	$E\{\epsilon_L\}$	$E\{\epsilon_L - \epsilon_R\}$	$E_S\{\epsilon_R + E_*\{\epsilon_b^* \mid S\}\}$	$E_S E_*\{\epsilon_b^* \mid S\}$	$E_S Var_*\{\epsilon_b^* \mid S\}$
24	3.54	18.33	14.79	15.08	11.54	0.21
	0.12	4.79	3.86	4.35	1.26	0.03
40	4.88	13.75	8.88	12.10	7.22	0.12
	0.06	3.23	2.97	2.27	0.92	0.03
80	7.19	11.19	4.00	10.82	3.63	0.04
	0.08	2.72	1.56	3.12	0.54	0.01
120	8.25	10.75	2.50	10.86	2.61	0.03
	0.03	2.14	1.23	2.04	0.37	0.01
160	7.59	9.88	2.28	9.56	1.96	0.02
	0.01	1.58	0.68	1.23	0.33	0.00
240	8.38	9.75	1.38	9.80	1.42	0.02
	0.03	1.94	0.49	1.83	0.22	0.00
320	9.11	10.09	0.98	10.14	1.03	0.01
	0.71	0.83	0.40	0.77	0.15	0.00
400	9.09	9.99	0.90	9.99	0.90	0.01
	0.01	0.95	0.24	0.89	0.13	0.00

(All numbers are percentages.)

Table 3.8 Bootstrap results for standard data ($\epsilon^* = 1.9\%$).

	CONVENTIONAL LEAVE-ONE-OUT & RESUBSTITUTION			BOOTSTRAP		
N	$E\{\epsilon_R\}$	$E\{\epsilon_L\}$	$E\{\epsilon_L - \epsilon_R\}$	$E_S\{\epsilon_R + E_*\{\epsilon_b^* \mid S\}\}$	$E_S E_*\{\epsilon_b^* \mid S\}$	$E_S Var_*\{\epsilon_b^* \mid S\}$
24	0.63	5.00	4.38	4.14	3.52	0.10
	0.01	3.43	3.02	1.69	0.84	0.02
40	1.88	3.63	1.75	3.74	1.86	0.03
	0.02	1.99	1.21	1.95	0.67	0.02
80	1.44	2.31	0.88	2.26	0.82	0.01
	0.01	1.10	0.94	1.08	0.24	0.00
120	1.75	2.71	0.96	2.31	0.56	0.01
	0.01	1.04	0.48	1.05	0.19	0.00
160	1.94	2.34	0.41	2.35	0.42	0.01
	0.00	0.90	0.36	0.80	0.17	0.00
240	2.21	2.50	0.29	2.50	0.29	0.00
	0.00	0.71	0.26	0.60	0.13	0.00
320	2.00	2.17	0.17	2.18	0.18	0.00
	0.00	0.48	0.14	0.53	0.07	0.00
400	2.01	2.24	0.23	2.21	0.19	0.00
	0.00	0.45	0.16	0.38	0.07	0.00

(All numbers are percentages.)

$$+ P_2^2[\sum_{j=1}^{N_2} \mathrm{Var}_*\{\gamma_j^{(2)}\} + \sum_{j=1}^{N_1}\sum_{\substack{k=1\\ j\neq k}}^{N_2} \mathrm{Cov}_*\{\gamma_j^{(2)}\gamma_k^{(2)}\}]$$

$$- 2P_1P_2\sum_{j=1}^{N_1}\sum_{k=1}^{N_2} \mathrm{Cov}_*\{\gamma_j^{(1)}\gamma_k^{(2)}\} \tag{3.89}$$

Because the samples from each class were bootstrapped independently, $\mathrm{Cov}_*\{\gamma_j^{(1)}\gamma_k^{(2)}\} = 0$.

Using a property of the inverse Fourier transform,

$$\gamma_j^{(i)} = -\frac{\Delta\theta_j^{(i)}}{2\pi}\int_{-\infty}^{+\infty}\frac{e^{j\omega\hat{h}^*(X_j^{(i)})}}{j\omega}d\omega = -\frac{1}{2}\mathrm{sgn}(\hat{h}^*(X_j^{(i)}))\Delta\theta_j^{(i)} \tag{3.90}$$

Thus, the variance of $\gamma_j^{(i)}$ is

$$\begin{aligned}\mathrm{Var}_*\{\gamma_j^{(i)}\} &= E_*\{\gamma_j^{(i)2}\} - E_*^2\{\gamma_j^{(i)}\}\\ &= \frac{1}{4}E\{\Delta\theta_j^{(i)2}\} - E_*^2\{\gamma_j^{(i)}\}\\ &\cong \frac{1}{4}(\frac{1}{N_i^2} - \frac{1}{N_i^3})\end{aligned} \tag{3.91}$$

where $E_*^2\{\gamma_j^{(i)}\}$ is proportional to $1/N_i^4$ from (3.85) and therefore can be ignored. $\mathrm{Cov}_*\{\gamma_j^{(i)}\gamma_k^{(i)}\}$ may be approximated by using the first term only of (3.84). Again, using (3.90),

$$\begin{aligned}\mathrm{Cov}_*\{\gamma_j^{(i)}\gamma_k^{(i)}\} &= E_*\{\gamma_j^{(i)}\gamma_k^{(i)}\} - E_*\{\gamma_j^{(i)}\}E_*\{\gamma_k^{(i)}\}\\ &\cong \frac{1}{4}\mathrm{sgn}(\hat{h}(X_j^{(i)}))\mathrm{sgn}(\hat{h}(X_k^{(i)}))E\{\Delta\theta_j^{(i)}\Delta\theta_k^{(i)}\} - E_*\{\gamma_j^{(i)}\}E\{\gamma_k^{(i)}\}\\ &\cong -\frac{1}{4N_i^3}\mathrm{sgn}(\hat{h}(X_j^{(i)}))\mathrm{sgn}(\hat{h}(X_k^{(i)}))\end{aligned} \tag{3.92}$$

where $E\{\Delta\theta_j^{(i)}\Delta\theta_k^{(i)}\} = -1/N_i^3$ for $j\neq k$ by (3.72), and $E_*\{\gamma_j^{(i)}\}E_*\{\gamma_k^{(i)}\}$ is proportional to $1/N_i^4$ by (3.85) and therefore can be ignored.

Thus, substituting (3.91) and (3.92) into (3.89) and using $\mathrm{Cov}_*\{\gamma_j^{(1)}\gamma_k^{(2)}\} = 0$,

$$\mathrm{Var}_*\{\hat{\varepsilon}_b^*|S\} \cong \frac{1}{4}\sum_{i=1}^{2}\frac{P_i}{N_i}\left[1-\sum_{j=1}^{N_i}\frac{\mathrm{sgn}(\hat{h}(X_j^{(i)}))}{N_i}\sum_{k=1}^{N_i}\frac{\mathrm{sgn}(\hat{h}(X_k^{(i)}))}{N_i}\right]$$

$$= \frac{1}{4}\sum_{i=1}^{2}\frac{P_i}{N_i}[1-(1-2\hat{\varepsilon}_{Ri})(1-2\hat{\varepsilon}_{Ri})]$$

$$= \sum_{i=1}^{2}P_i\frac{\hat{\varepsilon}_{Ri}(1-\hat{\varepsilon}_{Ri})}{N_i} \tag{3.93}$$

Note that $\Sigma\mathrm{sgn}(\hat{h}(X_j^{(i)}))/N_i = (-1)^i$ [(# of correctly classified ω_i samples by $\hat{h}\overset{\omega_1}{\underset{\omega_2}{\lessgtr}} 0)/N_i$ - (# of misclassified ω_i samples by $\hat{h}\overset{\omega_1}{\underset{\omega_2}{\gtrless}} 0)/N_i]$ = $(-1)^i[(1-\hat{\varepsilon}_{Ri})-\hat{\varepsilon}_{Ri}]$]= $(-1)^i(1-2\hat{\varepsilon}_{Ri})$. Since $\hat{h}$ is the resubstitution discriminant function for the original sample set, the resulting error is the resubstitution error.

Note that (3.93) is the variance expression of the resubstitution error estimate. This is seen in Tables 3.6-3.8 (second line of column 2 vs. first line of column 7) and theoretically substantiates a claim of Efron [14]. Also, note that, since (3.93) only involves bootstrap operations, this value can be estimated using just one set of samples. When S becomes a random set, $\mathrm{Var}_*\{\hat{\varepsilon}_b^*|S\}$ varies with $\hat{\varepsilon}_{Ri}(1-\hat{\varepsilon}_{Ri}) \cong \hat{\varepsilon}_{Ri}$.

3.5 Conclusions

The objective of this chapter was to apply the error expression derived in Chapter 2 to various classifier test procedures in order to theoretically analyze their estimates of the expected classifier performance. It was shown that the design samples alone account for a classifier's bias, while the test samples dominate the variance of the error estimate. These results had been demonstrated empirically. But, this chapter offers a new theoretical approach to understanding how design and test sample sizes affect the performance of classifiers. A general expression showing the relationship between the resubstitution and leave-one-out estimates of functions of Gaussian parameters was derived. As an example, the statistical properties of the difference between the resubstitution and leave-one-out error estimates for the quadratic classifier were investigated. The difference was found to be inversely proportional to the number of design samples and roughly proportional to n^2. In a related discussion, the effect of outlier design samples was found to be negligible, other than their effective reduction of the number of design samples in the training set. Finally, Efron's bootstrap estimate of the optimistic bias of the resubstitution error was analyzed. The resulting error estimate was shown to be statistically equivalent to the leave-one-out error estimate under reasonable design conditions.

Though not exhaustive, this study should provide a better understanding of the role of dependent and independent design and test samples in classifier design and evaluation. Hopefully, the tools and methodology can be applied to other statistical testing procedures and may help propose new ones.

CHAPTER 4
THE REDUCED PARZEN CLASSIFIER

4.1 Introduction

In pattern recognition, the quadratic classifier is very popular. However, in practice, with non-Gaussian distributions, it has been frequently observed that the error of a quadratic classifier is much larger than the Bayes error estimated by nonparametric techniques. On the other hand, nonparametric classifiers are too complex and time-consuming for on-line operation. Thus, there is a need to fill the gap between these two kinds of classifiers.

One possible solution is to find clusters and to design quadratic classifiers around cluster centers. Unfortunately, conventional clustering techniques give very poor estimates of the expected vectors and covariance matrices of the clusters. For example, let us consider a distribution which consists of two Gaussian distributions with some overlap. If we divide the mixture distribution into two clusters by setting a boundary, each cluster includes one true Gaussian distribution with a tail cut off, plus the tail of the other Gaussian distribution. Thus, the estimates of the expected vector and covariance matrix of the cluster based on samples in that cluster region could be significantly different from the true parameters of the Gaussian distribution.

In this chapter, we have taken a different approach. Our solution is to find a small number of representatives, maintaining that the Parzen density estimate with these representatives is as close as possible to the Parzen density estimate with all available samples [16], [17]. The resulting Parzen density estimate represents the distribution of each class. Combining these estimates from different classes, the Bayes classifier is designed. With Gaussian kernel functions, this closely resembles a piecewise quadratic classifier.

The idea of using reduced sample sets as representatives of larger sample sets has been around for a long time for various purposes, particularly for the k-nearest neighbor (NN) approach. For examples, the condensed NN for reducing storage and computation time, the edited NN for better performance and so on [18], [19], [20]. For the Parzen approach, smaller sample sets were sought in a pure density estimation setting [21]. Also, various parametric techniques have been developed for the decomposition of Gaussian mixtures [22], [23].

4.2 The Data Reduction Algorithm

In [24], Fukunaga and Mantock presented an algorithm for finding a reduced sample set which had virtually the same nearest neighbor (NN) density estimate as the original sample set. In this section, we will show how this algorithm has been adapted to use the Parzen density estimate.

Given N samples drawn from a density function, p(X), of a random vector, X, we wish to select r samples ($r < N$) such that we make the Parzen density estimates for the N sample set and the r sample set as close as possible.

Assuming that a Gaussian kernel is used, the Parzen density estimate at X given N samples is

$$\hat{p}_N(X) = \frac{1}{N} \sum_{i=1}^{N} k(X-X_i) \tag{4.1}$$

where

$$k(X-X_i) = \frac{1}{(2\pi)^{n/2} h^n \sqrt{|\Sigma|}} \exp[-\frac{1}{2h^2} (X-X_i)^T \Sigma^{-1} (X-X_i)] \tag{4.2}$$

n is the dimensionality, Σ is the kernel covariance matrix, and h is the kernel size control parameter. Similarly, when r representatives, $Y_1,...,Y_r$, are selected, the density function at X is estimated by

$$\hat{p}_r(X) = \frac{1}{r} \sum_{i=1}^{r} k(X - Y_i) \tag{4.3}$$

In order to measure the similarity between $\hat{p}_r(X)$ and $\hat{p}_N(X)$, the entropy $\int \ln[\hat{p}_r(X)/\hat{p}_N(X)]\,\hat{p}_N(X)dX$ is used in this chapter. The entropy expression satisfies

$$\int \ln \left[\frac{\hat{p}_r(X)}{\hat{p}_N(X)} \right] \hat{p}_N(X)dX \leq 0 \tag{4.4}$$

where equality holds when $\hat{p}_N(X) = \hat{p}_r(X)$. A larger entropy means that $\hat{p}_r(X)$ is closer to $\hat{p}_N(X)$. The inequality of (4.4) can be proved easily by using a property of the logarithmic function, $\ln a \leq a-1$. That is, $\int \ln [\hat{p}_r(X)/\hat{p}_N(X)]\, \hat{p}_N(X)dX \leq \int [\hat{p}_r(X)/\hat{p}_N(X)-1]\, \hat{p}_N(X)dX = \int \hat{p}_r(X)dX - \int \hat{p}_N(X)dX = 1-1=0$.

The entropy expression may be rewritten as $E\{\ln[\hat{p}_r(X)/\hat{p}_N(X)]\}$ where the expectation is taken with respect to $\hat{p}_N(X)$. Thus, if the expectation is replaced by the sample mean over the existing samples $X_1,\ldots,X_N$, (4.4) is approximated by

$$J = \frac{1}{N} \sum_{i=1}^{N} [\ln \hat{p}_r(X_i) - \ln \hat{p}_N(X_i)]. \tag{4.5}$$

Since the expectation is replaced by the sample mean, (4.5) $\leqq$ 0 is no longer guaranteed. However, this approximation significantly simplifies the criterion and subsequently the selection algorithm. The experimental results are good, as reported in the next section, and justifies the use of (4.5) as a criterion. Substituting (4.1) and (4.3) into (4.5),

$$J = \frac{1}{N} \sum_{i=1}^{N} [\ln \frac{1}{r} \sum_{j=1}^{r} k(X_i - Y_j) - \ln \frac{1}{N} \sum_{j=1}^{N} k(X_i - X_j)] \tag{4.6}$$

In order to find the best r representatives from the existing samples $X_1,\ldots,X_N$, we would like to maximize J over all possible r element subsets of the original N element set. Unfortunately, an exhaustive search of all $\binom{N}{r}$ subsets is not computationally feasible. Instead, we will settle for the maximum J for subsets formed by replacing one element of the representative set by the best candidate not yet selected.

The proposed procedure is as follows:

1) Select an initial assignment of r samples from the N sample data set. Call the r sample set STORE and the remaining N-r samples TEST.

2) For each element, X_t, in TEST, compute the change in J that results if the sample is transferred to STORE.

$$\Delta J_1(X_t) = J_{r+1}(X_t) - J_r$$

$$= \frac{1}{N} \sum_{i=1}^{N} [\ln \frac{1}{r+1} \{\sum_{j=1}^{r} k(X_i - Y_j) + k(X_i - X_t)\}$$

$$- \ln \frac{1}{r} \sum_{j=1}^{r} k(X_i - Y_j)] \qquad (4.7)$$

3) Pick the element, X_t, corresponding to the largest ΔJ_1 (and call it X_t^*).

4) For each element, X_s, in STORE, compute the change in J that results if the sample is transferred to TEST.

$$\Delta J_2(X_s) = J_r(X_s) - J_{r+1}$$

$$= \frac{1}{N} \sum_{i=1}^{N} [\ln \frac{1}{r} \{\sum_{j=1}^{r} k(X_i - Y_j) + k(X_i - X_t^*) - k(X_i - X_s)\}$$

$$- \ln \frac{1}{r+1} \{\sum_{j=1}^{r} k(X_i - Y_j) + k(X_i - X_t^*)\}] \qquad (4.8)$$

5) Find the element, X_s, corresponding to the largest ΔJ_2 (and call it X_s^*).

6) The change of J due to these two operations is $\Delta J = \Delta J_1(X_t^*) + \Delta J_2(X_s^*)$. In order to maximize J, we would like to have $\Delta J > 0$. If X_s^* exists to satisfy $\Delta J > 0$, transfer X_s^* to TEST, transfer X_t^* to STORE, and go to step 2.

7) Otherwise, find the element, X_t, corresponding to the next largest ΔJ_1 (and call it X_t^*).

8) If X_t^* exists, go to step 4.

9) Otherwise, stop.

Generally, this kind of iterative process produces a result which depends on the initial selection of r representatives in STORE. However, Steps 7 and 8 allow us to search more possible combinations of X_t and X_s and thus insure that the final representative set is independent of the initial assignment.

This procedure should be applied to the sample set of each class separately. The resulting Parzen density estimate with r representatives from each class is used to design the Bayes classifier.

4.3 Experimental Results

Two types of experiments were run to demonstrate the feasibility of the proposed procedure: one is for various Gaussian cases and the other is for non-Gaussian cases.

4.3.1 Gaussian Cases

The experiments for Gaussians used the test distributions mentioned in Chapters 2 and 3. Case a characterizes mean-separable problems. Case b characterizes covariance-separable problems. Case c is a complex case in which both the means and covariances are different. For all cases, the experiments were run by the following procedure:

1. **Classifier design:**

   a) Generate 100 samples per class, the design set, Gaussianly with given M_i and $\sum_i$.

   b) Generate the Parzen density estimate of (4.1) for each class by using the known covariance matrix as the kernel covariance in (4.2). Each class has a different kernel covariance.

   c) The kernel size control parameter, h, will be selected experimentally as will be presented in Step (b) of classifier test.

   d) For a fixed h, classify the existing 200 samples (100 per class) by

$$\hat{p}_N(X_i/\omega_1) \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \hat{p}_N(X_i/\omega_2) + t \qquad (i=1,...,200) \tag{4.9}$$

   where ω_i indicates class i and t is the threshold. t is selected to minimize the classification error. When the true $p(X/\omega_1)$ and $p(X/\omega_2)$ are used, the Bayes classifier requires a value of zero for t. However, when these densities are unknown, their estimates are biased. Adjusting t has been shown to be an excellent way to minimize the effect of the biases on classification [25].

2. **Classifier test:**

   a) Generate independently another set of 100 samples per class, the test set, and classify them using the classifier of (4.9) with the t fixed by 1-d. This was repeated 10 times and the average error is denoted as ϵ.

   b) Repeat 1-d and 2-a for various h. The optimal h is selected which minimizes ϵ [25]. When ϵ decreases monotonically with increased h, as has been observed for Gaussian distributions, h is selected at the point where ϵ starts to flatten out. Figs. 4.1(a), (b), and (c) show the plots of ϵ vs. h for cases a, b and c. Since the design and test sets are independent, these curves are supposed to give an upper bound of the Bayes error. The dotted lines were obtained by testing the original design samples. Because of the dependency between the design set and the test set, this procedure is supposed to give a lower bound of the Bayes error. From these figures, 2.0, 2.0, and 3.0 were selected as the optimal h for cases a, b, and c, respectively.

3. **Representative selection:** Select the r representatives by the proposed procedure of Section 4.2.

4. **Design of the reduced Parzen classifier:** Classify the original 100 design samples using

$$\hat{p}_r(X_i/\omega_1) \underset{\omega_2}{\overset{\omega_1}{\gtrless}} \hat{p}_r(X_i/\omega_2) + t \qquad (i=1,...,200) \tag{4.10}$$

   Select t which minimizes the error. Note that t must be readjusted each time a different value of r is selected.

5. **Test of the reduced Parzen classifier:** Generate another set of 100 test samples per class and classify them by (4.10). The resulting errors are plotted in Figs. 4.2(a), (b), and (c), (corresponding to cases a, b and c respectively), for various values of r. The curves of these figures are the averages of 10 trials and their standard deviations are shown by vertical bars.

In the cases presented above, the estimated errors bound the theoretical Bayes error closely. The reduced Parzen classifier provided excellent results until a very small ($\cong$ 3) number of representatives was selected. For

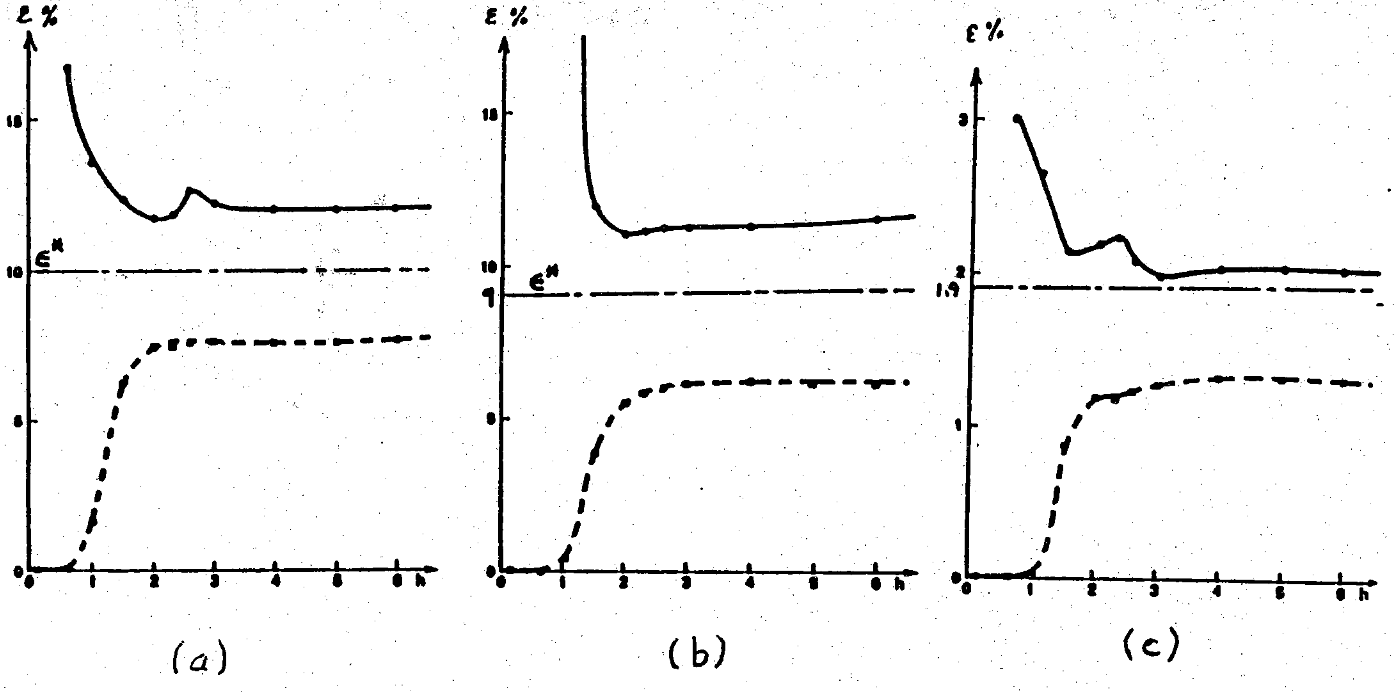


Figure 4.1 Parzen Density Error Estimates for Gaussians.

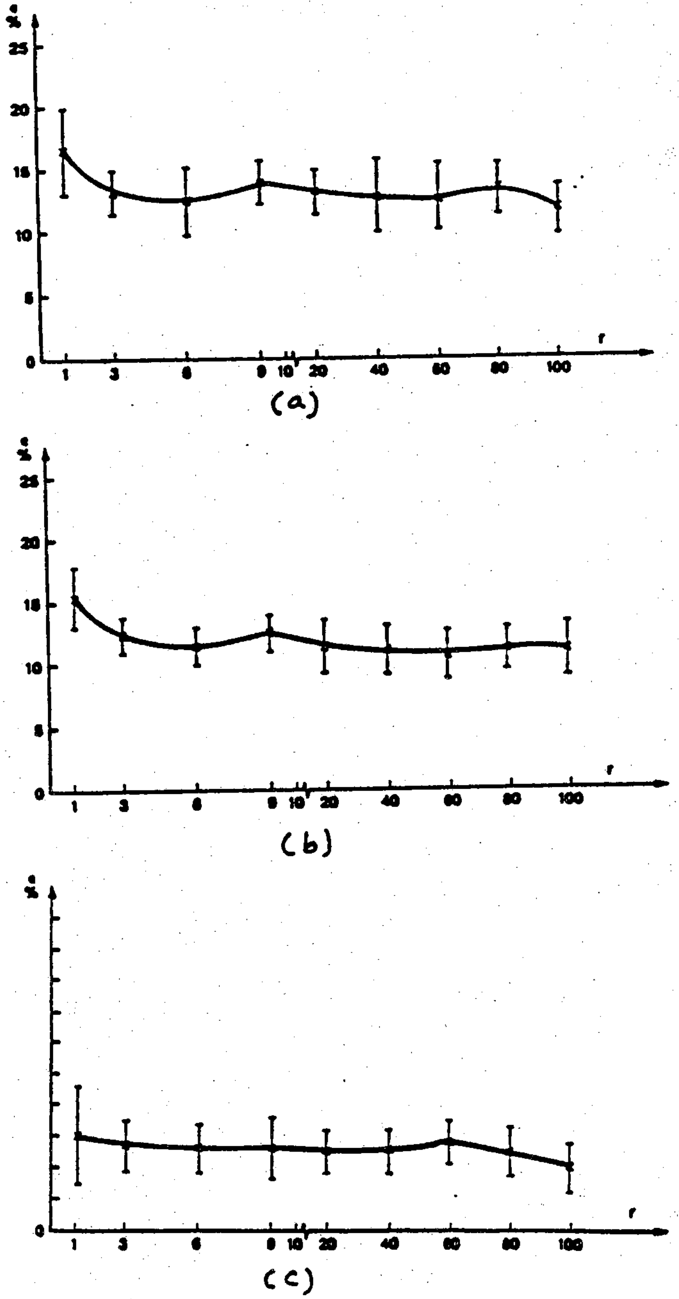


Figure 4.2 Reduced Parzen Classifier Results for Gaussians.

Gaussian cases, selecting the expected vector as the one representative from each class and the covariance matrix as the kernel covariance, the reduced Parzen classifier becomes the Bayes classifier. So, the error curves of Fig. 4.1 could be flat down to r=1. However, the proposed data reduction algorithm selects a vector from the existing design samples, which may or may not be close to the expected vector.

4.3.2 Non-Gaussian Cases

Two cases were studied as follows:

a) n=8, 4 Gaussian clusters with

$$M_1 = [-3.28,0,\cdots,0]^T \text{ and } \textstyle\sum_1 = I$$

$$M_2 = [0,0,\cdots,0]^T \text{ and } \textstyle\sum_2 = I$$

$$M_3 = [+3.28,0,\cdots,0]^T \text{ and } \textstyle\sum_3 = I$$

$$M_4 = [+6.56,0,\cdots,0]^T \text{ and } \textstyle\sum_4 = I$$

Clusters 1 and 3 form class 1 and clusters 2 and 4 form class 2.

$\epsilon^* = 7.5\%,\quad \epsilon_a = 29.5\%$

b) n=8, 6 Gaussian clusters with

$$M_1 = [-3.28,0,\cdots,0]^T \text{ and } \textstyle\sum_1 = I$$

$$M_2 = [0,0,\cdots,0]^T \text{ and } \textstyle\sum_2 = I$$

$$M_3 = [0,3.28,0,\cdots,0]^T \text{ and } \textstyle\sum_3 = I$$

$$M_4 = [3.28,3.28,0,\cdots,0]^T \text{ and } \textstyle\sum_4 = I$$

$$M_5 = [0,-3.28,0,\cdots,0]^T \text{ and } \textstyle\sum_5 = I$$

$$M_6 = [3.28,-3.28,0,\cdots,0]^T \text{ and } \textstyle\sum_6 = I$$

Clusters 1, 3 and 5 form class 1 and clusters 2, 4 and 6 form class 2.

$\epsilon^* = 8.3\%,\quad \epsilon_a = 19\%$

If we blindly assume that each class has a Gaussian distribution and design a two-class quadratic classifier using overall means and covariance matrices, the resulting error is much larger than the Bayes error. This error is called the apparent error and is listed above as ϵ_a. When the size of the Gaussian kernel function, h, is large, the Parzen density estimate becomes close to a Gaussian distribution, and the resulting classification error is expected to be close to ϵ_a.

All experiments were run as in the Gaussian case, with 75 samples per cluster for case a and 50 samples per cluster for case b. The Gaussian kernel function with $\sum=I$ was used for both cases.

Figs. 4.3(a) and (b) show ϵ vs. h in Step 2-b for cases a and b. The optimal h's were selected as 2.0 for case a and 1.3 for case b, respectively.

With these h's, the reduced Parzen classifiers were designed and the resulting error vs. r curves are plotted in Figs. 4.4(a) and (b).

As in the Gaussian cases, the reduced Parzen classifier provided excellent results until a very small number of representatives was selected. In case a, this number was 6. The data reduction algorithm picked 3 representatives from each cluster; the Gaussian results show that fewer non-optimal representatives cannot accurately represent the distribution. For case b, degradation occurred after 9 representatives. Again, the data reduction algorithm picked 3 representatives from each cluster.

4.3.3 Radar Data

To test its performance in a real, high-dimensional case, the reduced Parzen classifier was used on a set of 66-dimensional milimeter-wave radar data. The samples were collected by rotating a target (a Camaro and a Dodge Van) on a turntable and taking approximately 8800 readings. 66 range bins were selected and the resulting vectors were normalized by energy. For each class, the samples were alternately picked to form independent design and test sets, 4400 in each. Every sixth point in the design sets was chosen to form two sets of 720 reference representatives. Using a kernel covariance estimated from the 4400-sample design sets, these 720-sample sets were used to compute the Bayes error estimate. Then, the reduced Parzen classifier was designed for different numbers of representatives. Each classifier was tested on the 4400-sample test sets. Results are presented in Fig. 4.5.

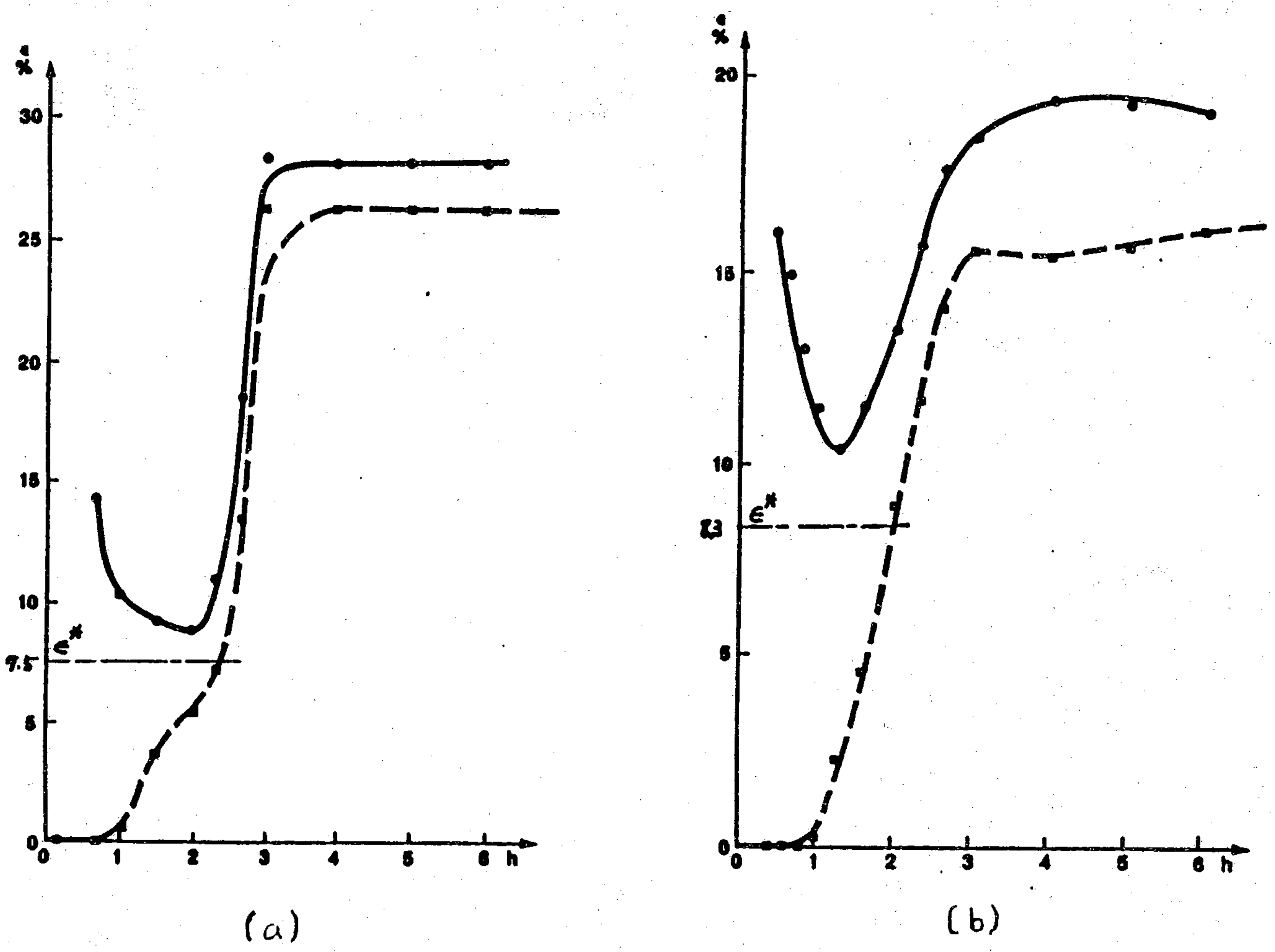
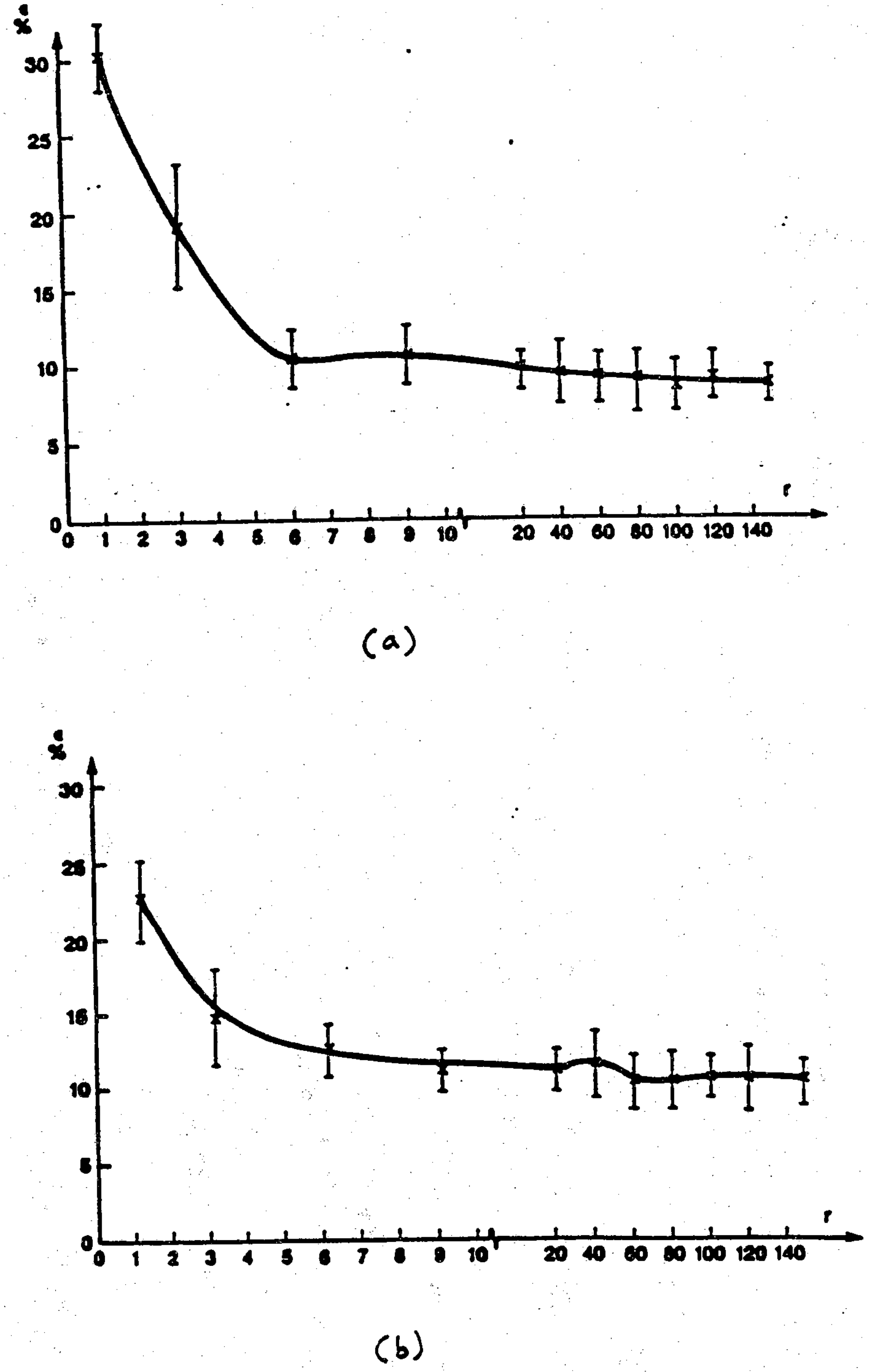


Figure 4.3 Parzen Density Error Estimates for Non-Gaussians.



(a)

(b)

Figure 4.4 Reduced Parzen Classifier Results for Non-Gaussians.

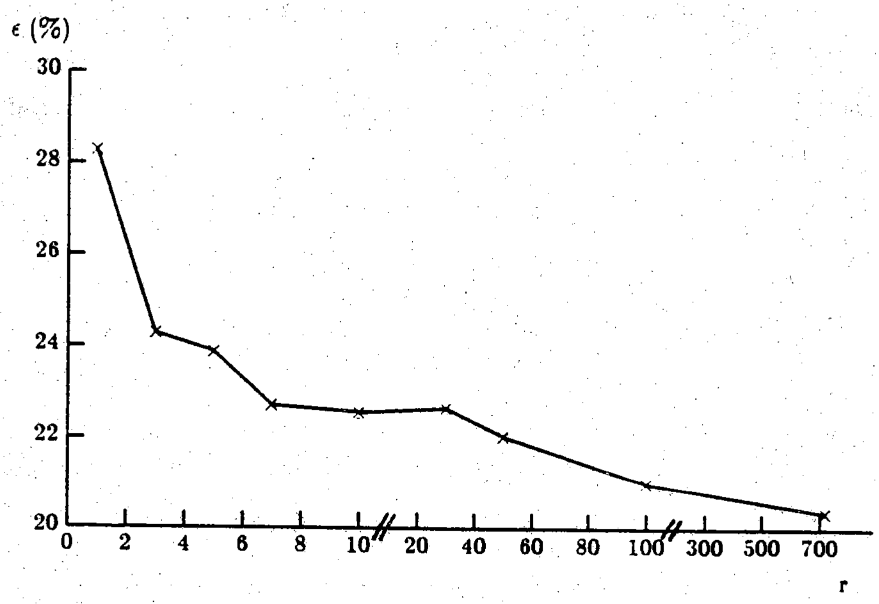


Figure 4.5 Reduced Parzen Classifier Results for Radar Data.

A flat performance is observed until the number of representatives is reduced to 3, suggesting that the underlying distributions are dominated by a Gaussian-like mode. The introduction of a few more representatives improves performance slightly, as if the distributions also contained small clusters of outliers. Nearly-optimal results can be achieved with a very samll ($\cong$ 7) number of representatives. This illustrates the reduced Parzen classifier's applicability in complicated, high-dimensional, real-life situations.

The proposed procedure selects the optimal kernel size h and the threshold, t, of the classifier. However, the selection of the kernel function and the kernel covariance matrix is not clearly understood. For Gaussian distributions, the Gaussian kernel with the sample covariance matrix works very well, provided that a large number of samples is used to estimate the covariance. Unfortunately, in non-Gaussian cases, the sample covariance matrix does not reflect the local structure of the distribution, producing poor experimental results. If we could estimate the local covariance accurately, the Parzen density estimate would provide a good estimate of the Bayes error, and we could design the reduced Parzen classifier with a small number of representatives.

The last paragraph suggests an interesting byproduct. In the past, it has been believed that a nonparametric procedure needs a large number of samples, N, for high-dimensional data, in order to reliably estimate the Bayes error or reasonable upper and lower bounds. Any nonparametric operation with a large N requires a large amount of computer time, and the turn-around time normally becomes days or even weeks. The results of this chapter contradict these common beliefs, and suggest that we may need only a relatively small sample size after all.

For example, Fig. 4.5 reveals that r could be reduced to 100 from 720 with an increase in the upper bound of the Bayes error from 22% to 25%. These 100 representatives are selected to optimize the criterion of (4.6). However, if r is a reasonably large number, such as 100, we may select them in a non-optimal way without significantly hurting the performance. For example, we may pick a sample every 3.6 degrees of the viewing angle. Previously, it was reported in [25] that the upper and lower bounds of the Bayes error for 60-dimensional data were reliably estimated using N = 115 and 230 from each class. The results were surprising at that time. But, that result is very consistent with our observations in this chapter. However, it should be pointed out that in [25] 5000 samples per class were used to estimate the covariance matrix which was used as the kernel covariance of the Parzen density estimate.

4.4 Summary

An algorithm was proposed to select a subset of representative samples from a given data set which preserves the Parzen density estimate. If an approximate Bayes classifier is designed using these representatives, nearly optimal discrimination is achieved, even for a significantly reduced number of representatives. Experimental results were presented, covering a wide range of Gaussian and non-Gaussian test cases.

CHAPTER 5
THE ACQUISITION PROBABILITY FOR A MINIMUM DISTANCE ONE-CLASS CLASSIFIER

5.1 Introduction

In many targeting scenarios, objects from different classes are detected and classified. As long as all of the classes are well-defined, standard Bayesian classification techniques work very well. However, in some cases, one class can be well-defined, while the other is not. For example, when we want to distinguish tanks (targets) from all other possible objects (non-targets), the non-targets may include trucks, automobiles and all kinds of other vehicles as well as trees and clutter discretes which are detected erroneously. Because of the wide variety, it is almost impossible to study the distributions of all possible non-targets before a classifier is designed.

One-class classification schemes have been proposed to solve this problem. Typically, they involve measuring the object's distance from the target mean and applying a threshold to determine if it is or isn't a target [26]. This technique, however, greatly increases the classification error. The mapping from the original n-dimensional feature space to a one-dimensional distance space destroys valuable classification information which existed in the original feature space.

However, this large increase in error can be reduced if one uses ranking instead of thresholding. If many objects are detected in a field and the goal is to acquire that one object which is most target-like, rank the objects according to their distances from the target mean and select the closest one. The acquisition probability of this procedure was derived and studied by Parenti and Tung [27] and Novak [28]. In this chapter, we will point out that this probability is determined by the operating characteristics in the distance space as well as the numbers of targets and non-targets detected in the field. Also, we will show that, if an exact measure is not required, the probability of acquisition can be approximated from just one point of the operating characteristics.

5.2 Computation of Acquisition Probability

Let X be an n-dimensional vector, representing an object in the feature space, and let us assume that k_1 targets $(X_1,\ldots,X_{k_1})$ and k_2 non-targets $(X_{k_1+1},\ldots,X_{k_1+k_2})$ are detected in a field. The acquisition procedure which will be studied in this chapter is:

(1) Compute the squared distance of X_i from the target's expected vector (M_1), normalized by the target covariance matrix $(\sum_1)$:

$$z_i = \frac{1}{n}(X_i - M_1)^T \textstyle\sum_1^{-1}(X_i - M_1) \qquad (i=1,2,\ldots,k_1+k_2) \tag{5.1}$$

where T indicates the transpose of the vector. M_1 and $\sum_1$ are assumed to be known.

(2) Rank the X_i's according to their z_i values. The X_i with the smallest z is selected as the target to be acquired.

The probability of acquiring any one of the k_1 targets in the field by this procedure (the probability of correct classification) can be expressed as [27]

$$P_a = \int_0^1 k_1(1-u_1)^{k_1-1}(1-u_2)^{k_2}\,du_1 \tag{5.2}$$

where

$$u_i(t) = \int_0^t p_i(z)dz \qquad (i=1,2) \tag{5.3}$$

and $p_i(z)$ is the density function of z for class i. Classes 1 and 2 are assigned to the targets and the non-targets respectively. As is seen in (5.3), $u_i(t)$ is the probability of a sample from class i falling in $0 \leqq z < t$. $u_1(t)$ and $u_2(t)$ are known as the detection and false alarm probabilities in the z-space when the threshold is chosen at $z=t$. In (5.2), du_1, $(1-u_1)^{k_1-1}$ and $(1-u_2)^{k_2}$ represent the probability of one of the k_1 targets falling in $t \leqq z < t+\Delta t$, k_1-1 of the targets falling in $t+\Delta t \leqq z < \infty$ and all k_2 non-targets falling in $t+\Delta t \leqq z < \infty$. The product of these three gives the probability of the combined event. Since the acquisition of any one of the k_1 targets is a correct classification, the probability is multiplied by k_1. The integration is taken with respect to t from 0 to ∞, that is, with respect to u_1 from 0 to 1. The derivation of (5.2) is given in Appendix H.

Rewriting $(1-u_1)$ as v and $(1-u_2)$ as f(v), the acquisition probability becomes

$$P_a = \int_0^1 k_1 v^{k_1-1} f^{k_2}(v)dv \tag{5.4}$$

Eq. (5.4) indicates that P_a is a function of k_1, k_2 and f(v). f(v) is a function relating $1-u_2$ to $1-u_1$. Since u_1 and u_2 are the detection and false-alarm probabilities in the z-space, f(v) represents the operating characteristics when each sample is classified in the z-space without ranking.

Fig. 5.1 shows typical operating characteristics from a series of experiments which will be described in the next section. Also shown are plots of v^{k_1-1} for $k_1=5$ and 20, which were used in these experiments. $f(v)=v$ represents the worst case in which the summation of the class 1 error and the class 2 error is always 100%, regardless of the operating point or the threshold value. That is, the distributions of class 1 and class 2 are identical. Therefore, if the distributions are classifiable through this ranking procedure, $f(v)>v$. Thus, v^{k_1-1} is reasonably assumed to drop to zero more quickly than $f^{k_2}(v)$, for realistic values of k_1 and k_2. This means that only the rightmost part of the operating characteristics, where v is close to 1, contributes to P_a. The other part of the operating characteristics will not affect P_a.

Although (5.2) is the exact expression for P_a, it is desirable to have an approximation formula through which P_a can be estimated faster and which shows the effects of k_1, k_2 and f(v) more explicitly. Since only a small portion of f(v) affects P_a and f(v) is very flat in that portion, we tried to approximate f(v) in this region with a constant, a line and other simple constructs. We have found that a constant gives us the simplest and most robust approximation of P_a, although it is rather crude. Thus,

$$f(v) \cong 1-\gamma \quad \text{for} \quad v^{k_1-1} \not\cong 0 \tag{5.5}$$

$$P_a \cong \int_0^1 k_1 v^{k_1-1} (1-\gamma)^{k_2} dv = (1-\gamma)^{k_2}$$

$$\cong 1-k_2\gamma \tag{5.6}$$

We have found empirically that γ may be selected in the following manner:

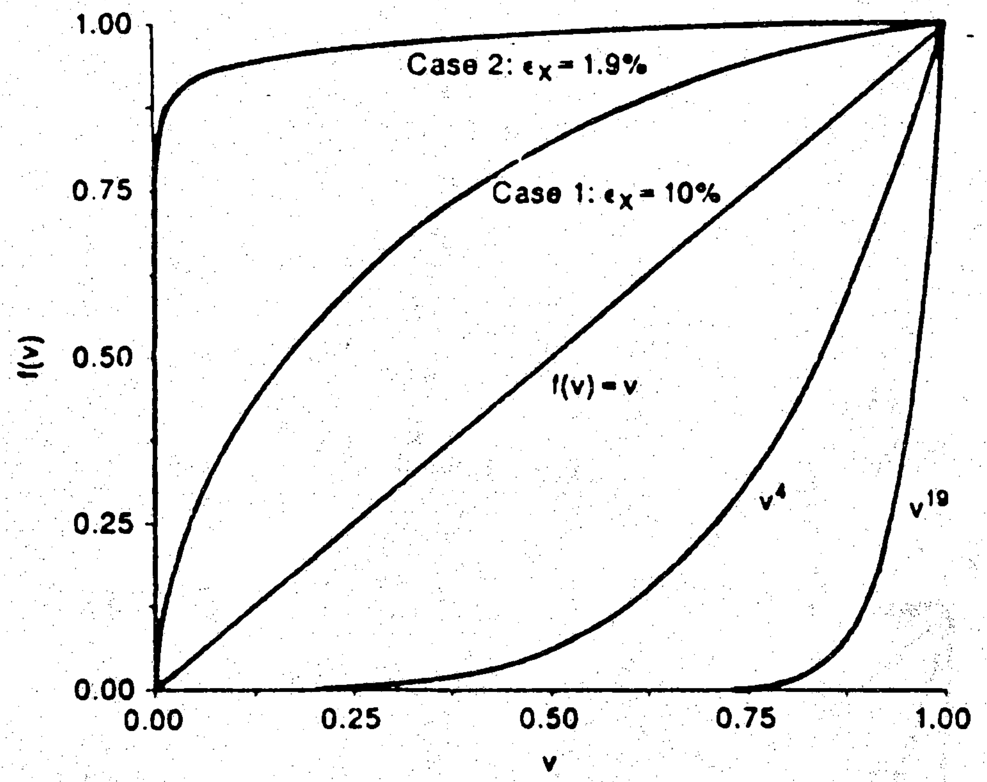


Figure 5.1 Operating Characteristics and $v^{k_1 - 1}$.

(1) For a given k_1, find v_o which satisfies $v_o^{k_1-1} = 0.5$.

(2) Read the operating characteristics $f(v)$ at v_o. Then, $f(v_o) = 1 - \gamma$.

The experimental results of this approximation will be reported in the next section.

It might seem that (5.6) is too sensitive to changes in the value of γ. However, a small change in γ corresponds to a significant change in the operating characteristics. So, in practice, the variation of γ stays very small and the approximation of (5.6) works well as reported in the next section.

5.3 Experimental Results

In order to test the validity of the proposed approximation and to find a way to select the value of γ, a series of experiments were run.

For $p_i(z)$ $(i=1,2)$ of (5.1), Gamma densities were chosen as

$$p_i(z) = \frac{c_i^{b_i+1}}{\Gamma(b_i+1)} z^{b_i} e^{-c_i z} \tag{5.7}$$

whose expected value and variance are

$$m_i = \frac{b_i+1}{c_i} \quad \text{and} \quad \sigma_i^2 = \frac{b_i+1}{c_i^2} \tag{5.8}$$

The reasons for this selection are as follows:

1. For class 1, if X is distributed Gaussianly with the expected vector M_1 and covariance matrix $\sum_1$, z of (5.1) has the Gamma distribution of (5.7) with $m_1 = 1$ and $\sigma_1^2 = \frac{2}{n}$.

2. For class 2, even if X is distributed Gaussianly, z does not have an exact Gamma density, since the expected vector, M_2, differs from M_1. However, our experiments show that the empirical distributions of z are very close visually to the Gamma distributions for a wide variety of M_2 and $\sum_2$ values. The empirical distribution of z was obtained from samples generated Gaussianly with given M_2 and $\sum_2$ in the X-space and converted to z by (5.1). The corresponding Gamma density function was specified by the expected value and variance computed by the following equations:

$$m_2 = \frac{1}{n}\left(\sum_{i=1}^{n} \lambda_i + \sum_{i=1}^{n} \mu_i^2\right) \tag{5.9}$$

$$\sigma_2^2 = \frac{1}{n^2}\left(2\sum_{i=1}^{n} \lambda_i^2 + 4\sum_{i=1}^{n} \lambda_i\mu_i^2\right) \tag{5.10}$$

where λ_i and μ_i are obtained as the results of simultaneous diagonalization. That is, a linear transformation A^TX is applied to X such that $A^T\sum_1 A = I$ and $A^T\sum_2 A = \Lambda$. λ_i and μ_i are the i-th components of the diagonal matrix Λ and the transformed vector $A^T(M_2 - M_1)$, respectively. The derivations of (5.9) and (5.10) are given in the Appendix G.

In order to cover various cases for the class 2 distribution, two types of Gaussian distributions were chosen for the experiments. Note that the selection of I for $\sum_1$ does not hurt generality, since we can always linearly transform $\sum_1$ to I without changing the subsequent results. Throughout the experiments, it was assumed that a priori probabilities of classes 1 and 2 are equal. ϵ_x and ϵ_z indicate the Bayes errors in the X- and z-spaces respectively. The Bayes error is the smallest error which can be obtained by the optimal classifier (the Bayes classifier) for given distributions [26].

1. Case 1: $\sum_1 = \sum_2 = I$, $M_2 - M_1 = M$, $n = 20$

   The Bayes classifier in the X-space is linear in this case and ϵ_x is determined by the length of the vector M, $\|M\|$. We selected $\|M\|$'s to get 1, 5, 10 and 20% for ϵ_x.

2. Case 2: $\sum_1 = I$, $\sum_2 = \Lambda$, $M_2 - M_1 = [\mu_1, \ldots, \mu_8]^T$, $n = 8$

   Λ and $M_2 - M_1$ were chosen from Standard Data of [10], and their components are $\lambda_1 = 8.41$, $\lambda_2 = 12.06$, $\lambda_3 = 0.12$, $\lambda_4 = 0.22$, $\lambda_5 = 1.49$, $\lambda_6 = 1.77$, $\lambda_7 = 0.35$, $\lambda_8 = 2.73$ and $\mu_1 = 3.86$, $\mu_2 = 3.10$, $\mu_3 = 0.84$, $\mu_4 = 0.84$, $\mu_5 = 1.64$, $\mu_6 = 1.08$, $\mu_7 = 0.26$, $\mu_8 = 0.01$. This data is suitable to test the case where $\sum_1$ and $\sum_2$ are significantly different, since the λ's vary from 0.12 to 12.06. The Bayes classifier is quadratic for this case and the resulting ϵ_x is 1.9% [10]. In order to obtain various ϵ_x's, we multiplied $M_2 - M_1$ by constants while keeping the covariances fixed.

   The experiments were carried out as follows:

1. Compute m_2 and σ_2^2 of (5.9) and (5.10) from given M_2 and $\sum_2$.
2. We assumed that the class 2 distribution in the z-space is Gamma with m_2 and σ_2^2 computed in step 1. The class 1 distribution is Gamma with $m_1=1$ and $\sigma_1^2=2/n$. These two Gamma density functions are plotted in Fig. 5.2, (a) for Case 1 and (b) for Case 2 respectively.
3. In Fig. 5.2, select the threshold t and compute $u_1(t)$ and $u_2(t)$ by (5.3). Changing t from 0 to ∞, plot the relationship between $1-u_2$ and $1-u_1$. The results are the operating characteristics as shown in Fig. 5.1.
4. Compute P_a of (5.4) and the approximated P_a of (5.6) for several values of γ. Table 5.1 shows the results when γ is selected as $1-f(v_0)$ where $v_0^{k_1-1}=0.5$. Although the approximations are somewhat crude, they predict the trend of P_a reasonably well.

A counter-intuitive result was observed in the Case 2 experiment. Intuitively, as k_1 and k_2 increase (i.e., as the numbers of targets and non-targets detected increase), the probability of misacquisition should decrease since there are now more targets, the acquisition of any one of which is considered correct. This is shown clearly in Case 1. However, in Case 2, the probability of misacquisition actually increases with an increase in k_1 and k_2. From (5.4), it should be apparent that an increase in k_1 makes the far rightmost position of the operating characteristics more dominant. Due to the construction of f(v), the rightmost position of the operating characteristic corresponds to the integration of the leftmost portion (the section closest to zero) of the probability duration of the distances. Ordinarily, one would expect $p_1(z)>p_2(z)$ for small values of z (i.e., one would expect the probability of a target being very close to the target mean to be greater than the probability of a non-target being very close to the target mean). However, Fig. 5.2 shows that, in spite of the fact that $m_2>m_1$, $p_2(z)>p_1(z)$ for small values of z! Thus, increasing k_1 compresses the range of significant distances from the target mean and amplifies the effect of the small region in which non-targets are more likely to be closer to the target mean than the targets themselves, increasing the probability of misacquisition. This result suggests that a careful examination is needed for the starting edges of the density functions in the z-space before deriving any conclusions by intuition.

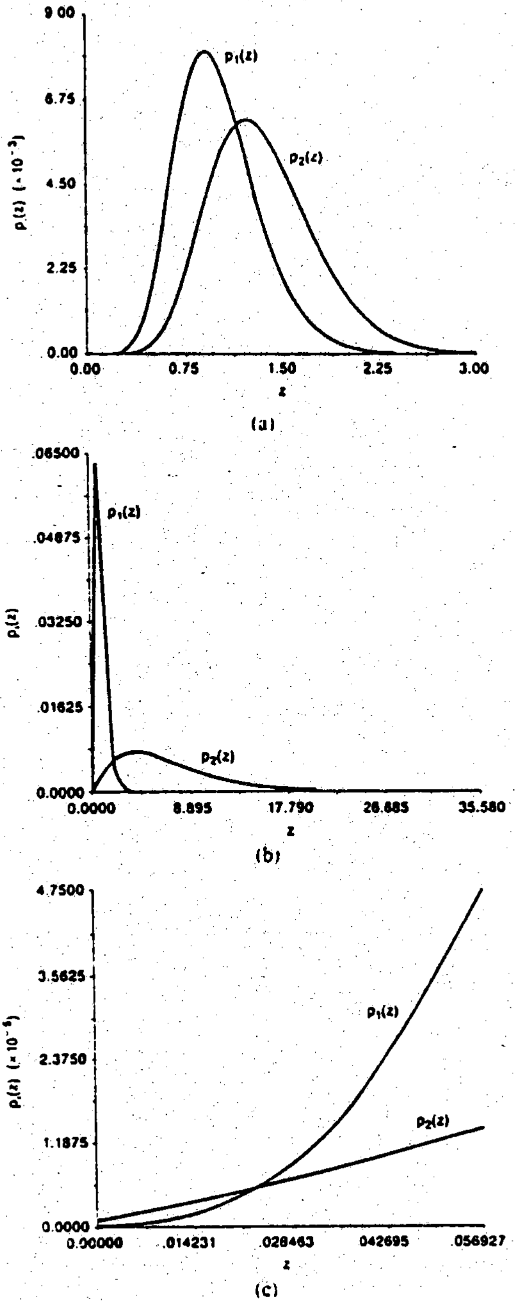


Figure 5.2 (a) Case 1 with $\epsilon_x = 10\%$. (b) Case 2 with $\epsilon_x = 1.9\%$. (c) Blow-up of the left-most part of (b).

Table 5.1 Results of P_a Approximation for Cases 1 and 2.
(All numbers are percentages.)

			$k_1=k_2=5$			$k_1=k_2=20$		
	ϵ_x	ϵ_z	$1-P_a$	$1-(1-\gamma)^{k_2}$	$k_2\gamma$	$1-P_a$	$1-(1-\gamma)^{k_2}$	$k_2\gamma$
CASE 1:	1.0	10.0	0.9	0.3	0.3	0.6	0.1	0.1
$\sum_1=\sum_2$	5.0	24.0	8.9	6.2	6.4	4.4	2.9	3.0
$=I$	10.0	32.0	17.6	14.8	15.8	14.9	8.4	8.7
	20.0	42.0	34.2	35.5	42.0	32.0	26.9	31.1
CASE 2:	1.9	12.9	4.4	3.1	3.1	6.9	4.0	4.1
$\sum_1\neq\sum_2$	*	29.7	17.6	17.3	18.7	30.1	27.5	31.9
	*	35.8	23.1	23.4	25.9	37.0	36.3	44.6

* - unknown error rates

At this point, we would like to point out the purpose of the ranking procedure. As Table 5.1 shows, the transformation of (5.1) from n-dimensional X to one dimensional z increases the classification error from ϵ_x to ϵ_z, if a simple threshold in the z-space is applied. Although the ranking procedure reduces ϵ_z to $1-P_a$, this reduction barely compensates the loss from ϵ_x to ϵ_z. Therefore, there is no need to use the proposed procedure, if the class 2 distribution is unimodal Gaussian as in the experiments. The conventional Bayes classifier in the X-space gives the classification error ϵ_x. However, if the class 2 distribution consists of many Gaussians surrounding the class 1 distribution as shown in Fig. 5.3, we must use a one-class classifier such as $z \gtrless t$, accepting ϵ_z as the resulting error. In this case, ϵ_x merely serves as a measure of how far the neighboring Gaussians are apart from the class 1 center. As was discussed in the introduction, in many target classification scenarios, class 2 includes various objects such as trucks, automobiles and all kinds of other vehicles as well as trees and clutter discretes, thus creating a distribution like the one in Fig. 5.3. Therefore, in this chapter, we point out how much the error can be reduced (from ϵ_z to $1-P_a$) by the ranking procedure, and discuss the effects of k_1, k_2, and the relative locations of the class 2 distributions.

5.4 Supplementary Discussions

5.4.1 Combinatorial Results

The expression and approximation derived for P_a are only good for fixed k_1 and k_2. More realistically, one is given the total number of objects detected in a field, k, and the a priori probability that a sample is a target, P_1. In this case, P_a can be computed by

$$P_a = \sum_{i=0}^{k} \binom{k}{i} P_1^i (1-P_1)^{k-i} \, P_a(i, k-i) \tag{5.11}$$

where $P_a(i, k-i)$ is the acquisition probability for $k_1 = i$ and $k_2 = k-i$.

5.4.2 Effect of Distance-Space Mapping

Even though the ranking procedure outperforms conventional one-class classification techniques, it is still hampered by the error introduced by the mapping from the original n-dimensional feature space to the one-dimensional distance space. In order to see how much the error is increased,

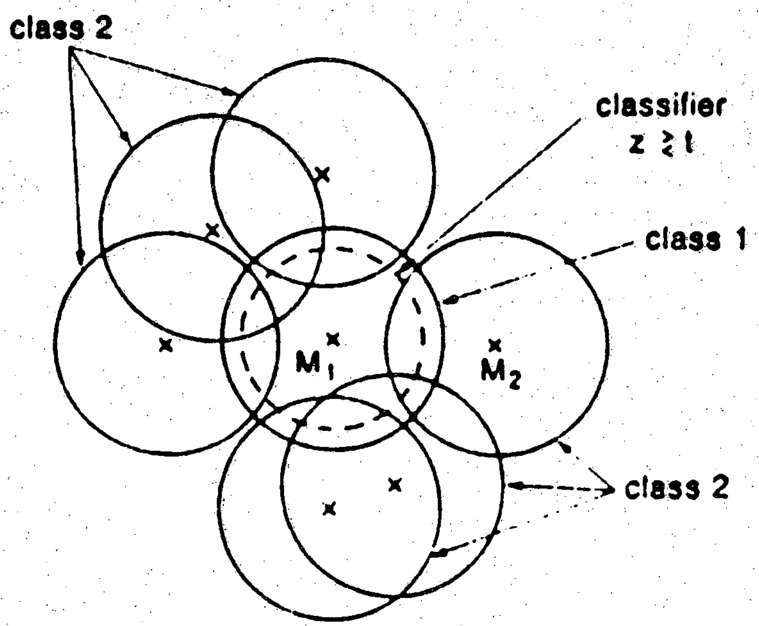


Figure 5.3 A Possible Class 2 Distribution with Multi-modal Gaussian.

the relationship between the Bayes errors in the X- and z-spaces, ϵ_x and ϵ_z, was examined for Case 1 of the previous section. Results are presented in Fig. 5.4. Figure 5.4 was obtained as follows:

1. Fix n (10, 50, 10, 150, 200).
2. Change $\|M\|$ in Case 1 experiment and obtain the corresponding ϵ_x in the X-space.
3. Compute $u_1(t_o)$ and $u_2(t_o)$ of (3) by numerical integration. $p_1(z)$ and $p_2(z)$ are assumed to be Gamma densities with $m_1=1$ and $\sigma_1^2=2/n$ for $p_1(z)$ and m_2 and σ_2^2 computed by (5.9) and (5.10) for $p_2(z)$. t_o is the value of z where $p_1(z)$ and $p_2(z)$ cross. When $p_1(z)$ and $p_2(z)$ cross at two values of z, as is the case for Case 2 experiment, choose the larger z.
4. $\epsilon_z = \frac{1}{2}(1-u_1(t_o)) + \frac{1}{2}u_2(t_0)$, since the a priori probabilities for classes 1 and 2 are both assumed to equal 1/2.

As one would expect, ϵ_z becomes very large as n increases.

5.4.3 Trade-off Between Number of Features and Original Error

Increasing the number of features, n, has both advantages and disadvantages in our targeting scenario. It reduces ϵ_x in general, but increases the information lost by mapping to the z-space. Thus, there should be some sort of trade-off between the number of features which would provide a reasonable error in the X-space and limit the amount of information lost in the distance mapping. Fig. 5.5 shows the same experimental results as Fig. 5.4, but this time ϵ_x vs. n is plotted for a fixed ϵ_z. Fig. 5.5 indicates that, in order to achieve $\epsilon_z=27\%$ for example, we may have many choices such as $\epsilon_x=10\%$ with $n=10$, $\epsilon_x=3\%$ with $n=60$ and so on. There is no reason to use an incredibly large number of features if the additional classifiability in the X-space cannot be transferred to the z-space. By keeping ϵ_z fixed, we were able to see just how much error could be introduced in the X-space while maintaining ϵ_z and reducing the number of features used.

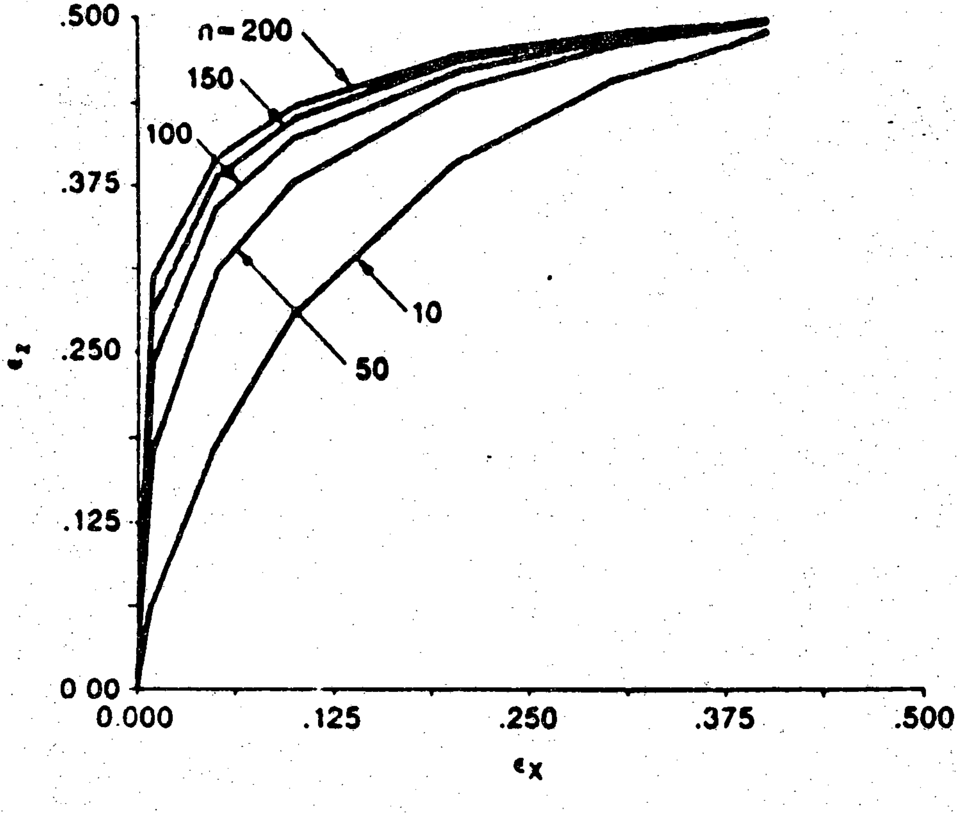


Figure 5.4 ϵ_z vs. ϵ_x for Constant n.

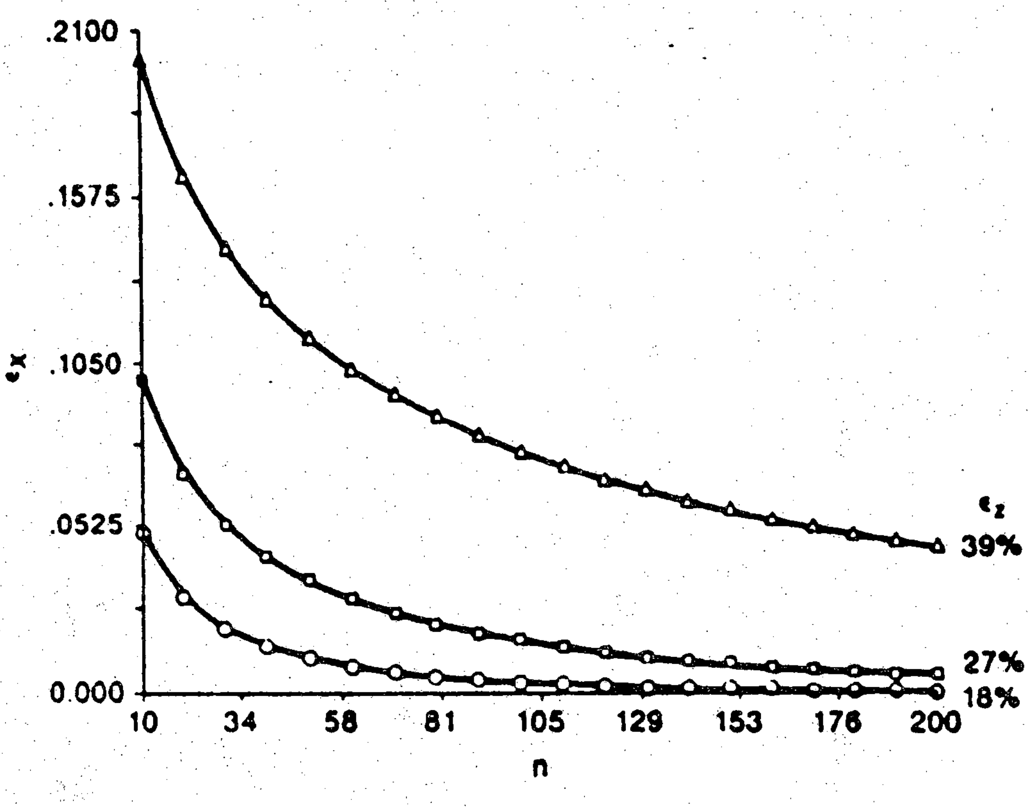


Figure 5.5 ϵ_x vs. n for Constant ϵ_z.

5.4.4 Extension to Multiple-Target, Multiple-Shot Case

So far, our scenarios have assumed that one and only one acquisition is attempted. However, in a different situation, where α acquisitions are attempted, we need to compute the probability that β targets are acquired in α attempts ($\alpha > \beta$). In this case, the probability of acquisition equals the probability that, if the α smallest distances are selected, β are targets and $\alpha - \beta$ are non-targets. The probability is presented here without derivation:

$$P_{\alpha\beta} = \int_0^1 \left[\binom{k_1}{\beta-1} u_1^{\beta-1} (1-u_1)^{k_1-\beta+1} \binom{k_2}{\alpha-\beta} u_2^{\alpha-\beta} \binom{k_1-\beta+1}{1} \frac{du_1}{1-u_1} \right.$$
$$\left. + \, g(k_1,\beta,u_1)\, g(k_2,\alpha-\beta-1,u_2) \binom{k_2-\alpha+\beta+1}{1} \frac{du_2}{1-u_2} \right] \tag{5.12}$$

where

$$g(k,\delta,u) = \binom{k}{\delta} u^{\delta} (1-u)^{k-\delta}$$

In [27], an approximation was developed which defines the multiple-target, multiple-shot acquisition probability as a function of P_a, k_1, α and β:

$$P_{\alpha\beta} = \begin{cases} \binom{k_1}{\beta} P_a^{\beta} (1-P_a)^{k_1-\beta} & : \quad \beta < \alpha, \ \beta \leqq k_1 \\ \sum_{i=\alpha}^{k_1} \binom{k_1}{i} P_a^{i} (1-P_a)^{k_1-i} & : \quad \beta = \alpha, \ \beta \leqq k_1 \end{cases} \tag{5.13}$$

If our proposed approximation for P_a is used together with this expression, $P_{\alpha\beta}$ can be estimated directly from the empirical operating characteristics.

5.5 Conclusions

Targeting scenarios, in which one class is known and well-defined and the other is unknown, point out the need for one-class classifiers. Conventional one-class classification techniques introduce a great deal of error by mapping the n-dimensional feature space into a one-dimensional distance space. An exact expression for the acquisition probability is dependent upon the empirical operating characteristics, the number of targets detected, and the number of other objects detected. An approximate expression is dependent on a single point of the operating characteristics, the number of targets detected, and the number of non-targets detected. Combinational techniques can be used when only the total number of objects detected is known. All of these results can be extended to include the multiple-target, multiple-shot case.

CHAPTER 6
SUMMARY AND RECOMMENDATIONS

6.1 Summary of Contributions

This thesis has examined several aspects of the classifier design and evaluation stages of the statistical pattern recognition system design process. Chapter 2 provides general and parametric expressions of the bias and variance of functions of estimated parameters. It was shown that when the dependence on the sample size can be separated from the distributions' effects, an empirical method for estimating the asymptotic value of the function can be applied. Also, an explicit expression for the error of a given classifier when used on a given test distribution was derived. The bias expressions were then applied to this error function to generate bias expressions for the linear and quadratic classifier, characterizing the degradation in their performance due to the design conditions.

In Chapter 3, the tools developed in Chapter 2 were applied to classifiers under finite design and test conditions. A number of testing procedures were investigated and compared. This chapter provides a unified framework for the analysis of classifier evaluation techniques and guidelines for the development of new ones. In addition, an explicit expression for the effect of outliers in the design set was presented.

The reduced Parzen classifier was developed in Chapter 4. This classifier combines the error estimation capabilities of the Parzen density estimate with the computational feasibility of parametric classifiers. It also shows that nonparametric techniques can be effective when implemented with a small number of carefully selected samples.

In Chapter 5, an approximation for the acquisition probability of a minimum-distance one-class classifier was presented. This provides insight into how the distance-space mapping of a one-class classifier degrades separability and how some of this can be recovered by applying a ranking procedure rather than a threshold.

6.2 Recommendations for Further Research

There are several topics related to this thesis which deserve further study.

1) The bias and variance expressions of Chapter 2 and Chapter 3 were only calculated for the I-I case. A symbolic or numerical tool must be developed in order to calculate these expressions for the general case.

2) Now that the degradation of classifier performance due to the design conditions has been characterized, perhaps the design process itself can be improved.

3) The Chapter 2 error bias expressions can be used to characterize the sensitivity of a classifier to changes in the measurement of a feature and to measure the separability provided by a feature.

4) The variance expression of (3.58) needs to be studied further in order to provide a more theoretical explanation of the trend in Fig. 3.2.

5) Chapters 2 and 3 suggest that a finite test set presents more difficulties than a finite design set. That is, it is impossible to characterize the degradation due to a finite test set without hypothesizing the true test distributions. We would like to develop an intelligent system which could manage these hypotheses, determine their likelihoods, evaluate classifiers within this hypothetical framework, and somehow find the optimal classifier for the most likely underlying distributions.

6) In Chapter 4, the reduced Parzen classifier design process employed the entropy as a sample selection criterion. Although this provided satisfactory results, other criteria, such as the mean-square error, need to be investigated.

7) Many important issues related to the Parzen density estimate have been investigated from density estimation and Bayes error estimation perspectives. While these have provided insight into the selection of the

kernel size and the threshold, the estimation of the kernel covariance remains a mystery. Perhaps this can be compensated for by allowing the locations of the representatives to move away from the sample points or by allowing the size of each individual kernel to vary.

8) The approximation presented in Chapter 5 was based on an intuitive understanding of the mechanics of the integral expression (5.4). While this has provided a great deal of insight, other approximation techniques, such as Gaussian quadrature, need to be investigated.

LIST OF REFERENCES

LIST OF REFERENCES

[1] T. S. El-Sheikh and A. G. Wacker, "Effect of dimensionality and estimation on the performance of Gaussian classifiers," Pattern Recognition, vol. 12, pp. 115-126, 1980.

[2] A. K. Jain and B. Chandrasekaran, "Dimensionality and sample size considerations in pattern recognition practice," Handbook of Statistics, Vol. 2, P. R. Krishnaiah and L. N. Kanal (eds.), North Holland, pp. 835-855, 1982.

[3] S. Raudys and V. Pikelis, "On dimensionality, sample size, classification error, and complexity of classification algorithm in pattern recognition," IEEE Trans. Pattern Anal. and Machine Intell., vol. PAMI-2, no. 3, pp. 242-252, May 1980.

[4] C. P. Han, "Distribution of discriminant function in circular models," Inst. Stat. Mathematics Annals, vol. 22, no. 1, pp. 117-125, 1970.

[5] G. J. McLachlan, "Some expected values for the error rates of the sample quadratic discriminant function," Australian Journal of Statistics, vol. 17(3), pp. 161-165, 1975.

[6] H. V. Pipberger, "Computer analysis of electrocardiogram," Clinical Electrocardiography and Computers, C. A. Caceres and L. S. Dreifus (eds.), New York: Academic, pp. 109-119, 1970.

[7] A. K. Jain, "On an estimate of the Bhattacharyya distance," IEEE Trans. Systems, Man, and Cybernetics, pp. 763-766, Nov. 1976.

[8] H. M. Kalayeh and D. A. Landgrebe, "Predicting the required number of training samples," IEEE Trans. Pattern Anal. and Machine Intell., vol. PAMI-5, no. 6, pp. 664-667, Nov. 1983.

[9] D. H. Foley, "Considerations of sample and feature size," IEEE Trans. Inform. Theory, vol. IT-18, no. 5, pp. 618-626, Sept. 1972.

[10] K. Fukunaga, Introduction to Statistical Pattern Recognition, New York: Academic, 1972.

[11] L. Novak, "On the sensitivity of Bayes and Fisher classifiers in radar target detection," in Proc. 18th Asilomar Conf. on Circuits, Systems, and Computers, Nov. 5-7, 1984.

[12] W. Beyer, CRC Standard Mathematical Tables, 26th ed., Boca Raton: CRC Press, pp. 44-45, 1981.

[13] P. A. Lachenbruch and R. M. Mickey, "Estimation of error rates in discriminant analysis," Technometrics, vol. 10, no. 1, pp. 1-11, 1968.

[14] B. Efron, "Bootstrap methods: Another look at the jackknife," Ann. Statist., vol. 7, pp. 1-26, 1979.

[15] A. K. Jain, R. C. Dubes, and C.-C. Chen, "Bootstrap techniques for error estimation," IEEE Trans. Pattern Anal. and Machine Intell., vol. PAMI-9, no. 5, pp. 628-633, Sept. 1987.

[16] E. Parzen, "An Estimation of a probability density function and mode," Ann. Math. Stat., 33, pp. 1065-1076, 1962.

[17] D. J. Hand, Kernel Discriminant Analysis, Chichester, U.R.: Research Studies Press, 1982.

[18] P. A. Devijver and J. Kittler, "On the edited nearest neighbor rule," in Proceedings of the Fifth International Conference on Pattern Recognition, pp. 72-80, 1980.

[19] G. W. Gates, "The reduced nearest neighbor rule," IEEE Trans. Inform. Theory, vol. IT-18, pp. 431-433, 1972.

[20] P. E. Hart, "The condensed nearest neighbor rule," IEEE Trans. Inform. Theory, vol. IT-14, pp. 515-516, 1968.

[21] S. Geman and C. R. Hwang, "Nonparametric maximum likelihood estimation by the method of sieves," Annals of Statistics 10, pp. 401-414, 1982.

[22] B. S. Everitt and D. J. Hand, Finite Mixture Distributions. London: Chapman and Hall, 1981.

[23] D. M. Titterington, A.F.M. Smith and U.E. Markov, Statistical Analysis of Finite Mixture Distributions, Wiley, 1985.

[24] K. Fukunaga and J. M. Mantock, "Nonparametric data reduction," IEEE Trans. Pattern Anal. and Machine Intell., vol. PAMI-6, pp. 115-118, 1984.

[25] K. Fukunaga and D. M. Hummels, "Bayes error estimation using Parzen and k-NN procedures," IEEE Trans. Pattern Anal. and Machine Intell., vol. PAMI-9, 634-643, 1987.

[26] K. Fukunaga, "Statistical Pattern Classification," Handbook of Pattern Recognition and Image Processing, edited by T. Young and K. S. Fu, Academic Press, New York, 1986.

[27] R. R. Parenti and E. W. Tung, "A Statistical Analysis of the Multiple-Target Multiple-Shot Target Acquisition Problem," Project Report TT-43, Lincoln Laboratory, M.I.T. (28 January 1981).

[28] L. Novak, "Optimal Target Designation Techniques," IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-17, Sept. 1981, pp. 676-684.

APPENDICES

Appendix A
Computation of the Derivatives of B_1

In order to compute the derivatives of B_1, we need the following formula for matrix differentiation [12].

$$\frac{\partial A^{-1}}{\partial a_{ij}} = -\,A^{-1}\frac{\partial A}{\partial a_{ij}}\,A^{-1} = -\,A^{-1}\,I(i,j)A^{-1} \tag{A1}$$

where a_{ij} is the i,j component of a matrix A, and I(i,j) is a matrix with an i,j component of 1 and all other components equal to 0. The s,t component of (A1) is

$$\left[\frac{\partial A^{-1}}{\partial a_{ij}}\right]_{st} = -\,[A^{-1}]_{si}[A^{-1}]_{jt} \tag{A2}$$

Applying (A1) repeatedly,

$$\frac{\partial^2 A^{-1}}{\partial a_{ij}\partial a_{k\ell}} = A^{-1}\frac{\partial A}{\partial a_{k\ell}}A^{-1}I(i,j)A^{-1}+A^{-1}I(i,j)A^{-1}\frac{\partial A}{\partial a_{k\ell}}A^{-1}$$

$$= A^{-1}I(k,\ell\,)A^{-1}I(i,j)A^{-1}+A^{-1}I(i,j)A^{-1}I(k,\ell\,)A^{-1} \tag{A3}$$

and

$$\left[\frac{\partial^2 A^{-1}}{\partial a_{ij}\partial a_{k\ell}}\right]_{st} = [A^{-1}]_{sk}[A^{-1}]_{\ell i}[A^{-1}]_{jt} + [A^{-1}]_{si}[A^{-1}]_{jk}[A^{-1}]_{\ell t} \tag{A4}$$

In the computation of the derivatives of B_1 with respect to $\alpha_{ij}^{(r)}$, let $A=\Sigma=(\Sigma_1+\Sigma_2)/2$ and $M=M_2-M_1$ from (2.7).

From (2.17), (A1) and (A2)

$$\frac{\partial B_1}{\partial \alpha_{ij}^{(r)}} = -\,\frac{1}{8}M^T\Sigma^{-1}\left(\frac{1}{2}\,\frac{\partial \Sigma_r}{\partial \alpha_{ij}^{(r)}}\right)\Sigma^{-1}M$$

$$= -\frac{1}{16}M^T\Sigma^{-1}I(i,j)\Sigma^{-1}M$$

$$= -\frac{1}{16}\sum_{s=1}^{n}\sum_{t=1}^{n}[\Sigma^{-1}]_{si}[\Sigma^{-1}]_{jt}m_sm_t$$

$$= -\frac{1}{16}\sum_{s=1}^{n}\sum_{t=1}^{n}\frac{2\delta_{si}}{1+\lambda_i}\ \frac{2\delta_{jt}}{1+\lambda_j}m_sm_t$$

$$= -\frac{m_im_j}{4(1+\lambda_i)(1+\lambda_j)} \tag{A5}$$

where $\delta_{ij} = 0$ or 1 depending on $i \neq j$ or $i = j$ and m_i is the ith component of M.

Also from (2.17), (A3) and (A4)

$$\frac{\partial^2 B_1}{\partial\alpha_{ij}^{(r)}\partial\alpha_{ij}^{(r)}} = \frac{1}{8}M^T\left[\Sigma^{-1}\frac{I(i,j)}{2}\Sigma^{-1}\frac{I(i,j)}{2}\Sigma^{-1}+\Sigma^{-1}\frac{I(i,j)}{2}\Sigma^{-1}\frac{I(i,j)}{2}\Sigma^{-1}\right]M$$

$$= \frac{1}{16}\sum_{s=1}^{n}\sum_{t=1}^{n}[\Sigma^{-1}]_{si}[\Sigma^{-1}]_{ji}[\Sigma^{-1}]_{jt}m_sm_t$$

$$= \frac{1}{16}\sum_{s=1}^{n}\sum_{t=1}^{n}\frac{2\delta_{si}}{1+\lambda_i}\ \frac{2\delta_{ji}}{1+\lambda_i}\ \frac{2\delta_{jt}}{1+\lambda_j}m_sm_t$$

$$= 0 \quad \text{for} \quad i \neq j \tag{A6}$$

Likewise,

$$\frac{\partial^2 B_1}{\partial\alpha_{ij}^{(r)}\partial\alpha_{ji}^{(r)}} = \frac{1}{8}M^T\left[\Sigma^{-1}\frac{I(i,j)}{2}\Sigma^{-1}\frac{I(j,i)}{2}\Sigma^{-1}+\Sigma^{-1}\frac{I(j,i)}{2}\Sigma^{-1}\frac{I(i,j)}{2}\Sigma^{-1}\right]M$$

$$= \frac{1}{32}\sum_{s=1}^{n}\sum_{t=1}^{n}\left[\frac{2\delta_{si}}{1+\lambda_i}\ \frac{2\delta_{jj}}{1+\lambda_j}\ \frac{2\delta_{it}}{1+\lambda_i}m_sm_t+\frac{2\delta_{sj}}{1+\lambda_j}\ \frac{2\delta_{ii}}{1+\lambda_i}\ \frac{2\delta_{jt}}{1+\lambda_j}m_sm_t\right]$$

$$= \frac{1}{4}\left[\frac{m_i^2}{(1+\lambda_i)^2(1+\lambda_j)} + \frac{m_j^2}{(1+\lambda_j)^2(1+\lambda_i)}\right] \tag{A7}$$

Eqs. (A5), (A6) and (A7) are shown in (2.21), (2.22) and (2.23) respectively.

Appendix B
Computation of the Derivatives of B_2

From [12], if a matrix A is symmetric

$$\frac{\partial \ln|A|}{\partial A} = A^{-1^T} = A^{-1} \tag{B1}$$

or,

$$\frac{\partial \ln|A|}{\partial a_{ij}} = [A^{-1}]_{ij} \tag{B2}$$

Using (A1),

$$\frac{\partial^2 \ln|A|}{\partial A \partial a_{k\ell}} = \frac{\partial A^{-1}}{\partial a_{k\ell}} = -\, A^{-1} I(k,\ell\,) A^{-1} \tag{B3}$$

or,

$$\frac{\partial^2 \ln|A|}{\partial a_{ij} \partial a_{k\ell}} = -\,[A^{-1} I(k,\ell\,) A^{-1}]_{ij} = -\,[A^{-1}]_{ik} [A^{-1}]_{\ell j} \tag{B4}$$

Since $B_2 = \frac{1}{2} \ln |\Sigma| - \frac{1}{4} \ln |\Sigma_1| - \frac{1}{4} |\Sigma_2|$ from (2.17) and $\Sigma = (\Sigma_1 + \Sigma_2)/2$,

$$\frac{\partial B_2}{\partial \alpha_{ij}^{(1)}} = \frac{1}{2}\,\frac{1}{2}[\Sigma^{-1}]_{ij} - \frac{1}{4}[\Sigma_1^{-1}]_{ij} = \frac{1}{4}\,\frac{2\delta_{ij}}{1+\lambda_i} - \frac{1}{4}\delta_{ij} \tag{B5}$$

$$\frac{\partial B_2}{\partial \alpha_{ij}^{(2)}} = \frac{1}{2}\,\frac{1}{2}[\Sigma^{-1}]_{ij} - \frac{1}{4}[\Sigma_2^{-1}]_{ij} = \frac{1}{4}\,\frac{2\delta_{ij}}{1+\lambda_i} - \frac{1}{4}\,\frac{\delta_{ij}}{\lambda_i} \tag{B6}$$

Eqs. (B5) and (B6) are shown in (2.27) and (2.28) respectively.

The second order derivatives of B_2 are obtained by using (B4):

$$\frac{\partial^2 B_2}{\partial \alpha_{ij}^{(r)} \partial \alpha_{ij}^{(r)}} = -\,\frac{1}{2}\,\frac{1}{4}[\Sigma^{-1}]_{ii}[\Sigma^{-1}]_{jj} + \frac{1}{4}[\Sigma_r^{-1}]_{ii}[\Sigma_r^{-1}]_{jj} \tag{B7}$$

$$\frac{\partial^2 B_2}{\partial \alpha_{ij}^{(r)} \partial \alpha_{ji}^{(r)}} = -\frac{1}{2}\ \frac{1}{4}[\overline{\sum}^{-1}]_{ij}[\overline{\sum}^{-1}]_{ji} + \frac{1}{4}[\sum\nolimits_r^{-1}]_{ij}[\sum\nolimits_r^{-1}]_{ji} \tag{B8}$$

Therefore,

$$\frac{\partial^2 B_2}{\partial \alpha_{ij}^{(1)} \partial \alpha_{ij}^{(1)}} = -\frac{1}{8}\frac{2}{1+\lambda_i}\ \frac{2}{1+\lambda_j} + \frac{1}{4} \tag{B9}$$

$$\frac{\partial^2 B_2}{\partial \alpha_{ij}^{(2)} \partial \alpha_{ij}^{(2)}} = -\frac{1}{8}\frac{2}{1+\lambda_i}\ \frac{2}{1+\lambda_j} + \frac{1}{4}\frac{1}{\lambda_i}\frac{1}{\lambda_j} \tag{B10}$$

$$\frac{\partial^2 B_2}{\partial \alpha_{ij}^{(r)} \partial \alpha_{ji}^{(r)}} = 0 \quad \text{for} \quad i \neq j \tag{B11}$$

Eqs. (B8) through (B11) are shown in (2.29) through (2.31).

Appendix C
The Derivatives of h for the Quadratic Classifier

The derivatives of h with respect to $m_i^{(r)}$ can be obtained easily from (2.48) as follows:

$$\frac{\partial h(X)}{\partial M_1} = -\,\Sigma_1^{-1}(X-M_1)\;, \qquad \frac{\partial h(X)}{\partial M_2} = \Sigma_2^{-1}(X-M_2) \tag{C1}$$

$$\frac{\partial^2 h(X)}{\partial M_1^2} = \Sigma_1^{-1}\;, \qquad \frac{\partial^2 h(X)}{\partial M_2^2} = -\,\Sigma_2^{-1} \tag{C2}$$

Using $M_1=0$, $M_2=M$, $\Sigma_1=I$ and $\Sigma_2=\Lambda$ of (2.7),

$$\frac{\partial h(X)}{\partial m_i^{(1)}} = -x_i\;, \qquad \frac{\partial h(X)}{\partial m_i^{(2)}} = \frac{x_i-m_i}{\lambda_i} \tag{C3}$$

$$\frac{\partial^2 h(X)}{\partial m_i^{(1)2}} = 1\;, \qquad \frac{\partial^2 (X)}{\partial m_i^{(2)2}} = -\frac{1}{\lambda_i} \tag{C4}$$

In order to derive the derivatives with respect to $\alpha_{ij}^{(r)}$, we need the derivatives for matrix inversion as in Appendix A and the derivatives of the log-determinant as in Appendix B. They can be computed as follows:

$$\frac{\partial h(X)}{\partial \alpha_{ij}^{(1)}} = -\frac{1}{2}(X-M_1)^T\Sigma_1^{-1}I(i,j)\Sigma_1^{-1}(X-M_1) + \frac{1}{2}[\Sigma_1^{-1}]_{ij}$$

$$= -\frac{1}{2}x_ix_j + \frac{1}{2}\delta_{ij} \tag{C5}$$

$$\frac{\partial h(X)}{\partial \alpha_{ij}^{(2)}} = \frac{1}{2}(X-M_2)^T\Sigma_2^{-1}I(i,j)\Sigma_2^{-1}(X-M_2) - \frac{1}{2}[\Sigma_2^{-1}]_{ij}$$

$$= \frac{1}{2}\frac{(x_i-m_i)(x_j-m_j)}{\lambda_i\lambda_j} - \frac{1}{2}\frac{\delta_{ij}}{\lambda_i} \tag{C6}$$

$$\frac{\partial^2 h(X)}{\partial \alpha_{ij}^{(1)}\partial \alpha_{ij}^{(1)}} = \frac{1}{2}(X-M_1)^T\Big[\Sigma_1^{-1}I(i,j)\Sigma_1^{-1}I(i,j)\Sigma_1^{-1}$$

$$+ \sum_1^{-1}I(i,j)\sum_1^{-1}I(i,j)\sum_1^{-1}\Big](X-M_1)$$

$$-\frac{1}{2}[\sum_1^{-1}]_{ii}[\sum_1^{-1}]_{jj}$$

$$=\sum_{s=1}^{n}\sum_{t=1}^{n}\delta_{si}\delta_{ji}\delta_{jt}x_s x_t-\frac{1}{2}\delta_{ii}\delta_{jj}$$

$$=-\frac{1}{2}\quad \text{for}\quad i\neq j \tag{C7}$$

$$\frac{\partial^2(X)}{\partial\alpha_{ij}^{(2)}\partial\alpha_{ij}^{(2)}}=-\frac{1}{2}(X-M_2)^T\Big[\sum_2^{-1}I(i,j)\sum_2^{-1}I(i,j)\sum_2^{-1}$$

$$+ \sum_2^{-1}I(i,j)\sum_2^{-1}I(i,j)\sum_2^{-1}\Big](X-M_2)$$

$$+\frac{1}{2}[\sum_2^{-1}]_{ii}[\sum_2^{-1}]_{jj}$$

$$=\frac{1}{2}\frac{1}{\lambda_i\lambda_j}\quad \text{for}\quad i\neq j \tag{C8}$$

$$\frac{\partial^2h(X)}{\partial\alpha_{ij}^{(1)}\partial\alpha_{ji}^{(1)}}=\frac{1}{2}(X-M_1)^T\Big[\sum_1^{-1}I(i,j)\sum_1^{-1}I(j,i)\sum_1^{-1}$$

$$+ \sum_1^{-1}I(j,i)\sum_1^{-1}I(i,j)\sum_1^{-1}\Big](X-M_1)$$

$$-\frac{1}{2}[\sum_1^{-1}]_{ij}[\sum_1^{-1}]_{ji}$$

$$=\frac{1}{2}\sum_{s=1}^{n}\sum_{t=1}^{n}(\delta_{si}\delta_{jj}\delta_{it}x_s x_t+\delta_{sj}\delta_{ii}\delta_{jt}x_s x_t)-\frac{1}{2}\delta_{ij}\delta_{ji}$$

$$=\frac{1}{2}(x_i^2+x_j^2)-\frac{1}{2}\delta_{ij} \tag{C9}$$

$$\frac{\partial^2h(X)}{\partial\alpha_{ij}^{(2)}\partial\alpha_{ji}^{(2)}}=-\frac{1}{2}(X-M_2)^T\Big[\sum_2^{-1}I(i,j)\sum_2^{-1}I(j,i)\sum_2^{-1}$$

$$+ \sum_2^{-1}I(j,i)\sum_2^{-1}I(i,j)\sum_2^{-1}\Big](X-M_2)$$

$$+\ \frac{1}{2}[\Sigma_2^{-1}]_{ij}[\Sigma_2^{-1}]_{ji}$$

$$= -\ \frac{1}{2}\sum_{s=1}^{n}\sum_{t=1}^{n}\left(\frac{\delta_{si}}{\lambda_i}\ \frac{\delta_{jj}}{\lambda_j}\ \frac{\delta_{it}}{\lambda_i}\ (x_s-m_s)(x_t-m_t)\right.$$

$$\left. +\ \frac{\delta_{sj}}{\lambda_j}\ \frac{\delta_{ii}}{\lambda_i}\ \frac{\delta_{jt}}{\lambda_j}\ (x_s-m_s)(x_t-m_t)\right) + \frac{1}{2}\frac{\delta_{ij}}{\lambda i}\ \frac{\delta_{ji}}{\lambda_j}$$

$$= -\ \frac{1}{2}\left(\frac{(x_i-m_i)^2}{\lambda_i^2\lambda_j} + \frac{(x_j-m_j)^2}{\lambda_j^2\lambda_i}\right) + \frac{1}{2}\frac{\delta_{ij}}{\lambda_i\lambda_j} \tag{C10}$$

Plugging all these equations into (2.47), we obtain (2.49).

Appendix D
The Derivatives of h for the Linear Classifier

The derivatives of h with respect to $m_i^{(r)}$ can be obtained from (2.56) as follows:

$$\frac{\partial h(X)}{\partial M_1} = -\sum^{-1}X + \sum^{-1}M_1 \qquad \text{(D1)}$$

$$\frac{\partial h(X)}{\partial M_2} = \sum^{-1}X - \sum^{-1}M_2 \qquad \text{(D2)}$$

$$\frac{\partial^2 h(X)}{\partial M_1^2} = \sum^{-1} \qquad \text{(D3)}$$

$$\frac{\partial^2 h(X)}{\partial M_2^2} = -\sum^{-1} \qquad \text{(D4)}$$

Using $M_1=0$, $M_2=M$, $\sum_1=I$, $\sum_2=\Lambda$ and $\sum=(I+\Lambda)/2$,

$$\frac{\partial h(X)}{\partial m_i^{(1)}} = -\frac{2x_i}{1+\lambda_i}\ , \quad \frac{\partial h(X)}{\partial m_i^{(2)}} = \frac{2(x_i-m_i)}{1+\lambda_i} \qquad \text{(D5)}$$

$$\frac{\partial^2 h(X)}{\partial m_i^{(1)2}} = \frac{2}{1+\lambda_i}\ , \quad \frac{\partial^2 h(X)}{\partial m_i^{(2)2}} = -\frac{2}{1+\lambda_i} \qquad \text{(D6)}$$

The derivatives with respect to $\alpha_{ij}^{(r)}$ are computed as follows:

$$\frac{\partial h}{\partial \alpha_{ij}^{(r)}} = -\frac{1}{4}M^T\sum^{-1}I(i,j)\sum^{-1}(2X-M)$$

$$= -\frac{1}{4}\sum_{s=1}^{n}\sum_{t=1}^{n}\frac{2\delta_{si}}{1+\lambda_i}\ \frac{2\delta_{jt}}{1+\lambda_j}\ m_s(2x_t-m_t)$$

$$= -\frac{m_i(2x_j-m_j)}{(1+\lambda_i)(1+\lambda_j)} \tag{D7}$$

$$\frac{\partial^2 h}{\partial\alpha_{ij}^{(r)}\partial\alpha_{ij}^{(r)}} = \frac{1}{8}M^T\left[\Sigma^{-1}I(i,j)\Sigma^{-1}I(i,j)\Sigma^{-1}\right.$$

$$\left.+\ \Sigma^{-1}I(i,j)\Sigma^{-1}I(i,j)\Sigma^{-1}\right](2X-M)$$

$$= \frac{1}{4}\sum_{s=1}^{n}\sum_{t=1}^{n}\frac{2\delta_{si}}{1+\lambda_i}\ \frac{2\delta_{ji}}{1+\lambda_j}\ \frac{2\delta_{jt}}{1+\lambda_j}\ m_s(2x_t-m_t)$$

$$= 0 \qquad \text{for} \quad i\neq j \tag{D8}$$

$$\frac{\partial^2 h}{\partial\alpha_{ij}^{(r)}\partial\alpha_{ji}^{(r)}} = \frac{1}{8}M^T\left[\Sigma^{-1}I(i,j)\Sigma^{-1}I(j,i)\Sigma^{-1}\right.$$

$$\left.+\ \Sigma^{-1}I(j,i)\Sigma^{-1}I(i,j)\Sigma^{-1}\right](2X-M)$$

$$= \frac{1}{8}\sum_{s=1}^{n}\sum_{t=1}^{n}\left\{\frac{2\delta_{si}}{1+\lambda_i}\ \frac{2\delta_{jj}}{1+\lambda_j}\ \frac{2\delta_{it}}{1+\lambda_i}\right.$$

$$\left.+\ \frac{2\delta_{sj}}{1+\lambda_j}\ \frac{2\delta_{ii}}{1+\lambda_i}\ \frac{2\delta_{jt}}{1+\lambda_j}\right\}m_s(2x_t-m_t)$$

$$= \frac{m_i(2x_i-m_i)}{(1+\lambda_i)^2(1+\lambda_j)} + \frac{m_j(2x_j-m_j)}{(1+\lambda_j)^2(1+\lambda_i)} \tag{D9}$$

Plugging all these results into (2.47), we obtain (2.59).

Appendix E
Proof of $\hat{\epsilon} \geq \epsilon$

The first step is to prove that the first-order variation of (3.19) is zero regardless of $\Delta h(X)$. From (3.21), the first-order variation of (3.19) is

$$\frac{1}{2\pi}\int_S \int_{-\infty}^{+\infty} \Delta h(X) e^{j\omega h(X)} \tilde{p}(X) d\omega dX = \int_S \Delta h(X) \delta(h(X)) \tilde{p}(X) dX$$

$$= \int_{h(X)=0} \Delta h(X) \tilde{p}(X) dX$$

$$= 0 \qquad \text{(E1)}$$

The last equality comes from the fact that $\tilde{p}(X) = 0$ at $h(X) = 0$ if $h(X)$ is the Bayes classifier of $P_1 p_1(X)$ and $P_2 p_2(X)$.

The second step involves showing that the second-order variation of (3.19) is positive regardless of $\Delta h(X)$. Again from (3.21)

$$\frac{1}{2\pi}\int_S \int_{-\infty}^{+\infty} \frac{j\omega}{2} \Delta h^2(X) e^{j\omega h(X)} \tilde{p}(X) d\omega dX = \frac{1}{2}\int_S \Delta h^2(X) \frac{d\delta(h)}{dh} \tilde{p}(X) dX \qquad \text{(E2)}$$

In the region very close to $h(X) = 0$, $d\delta(h)/dh > 0$ and $\tilde{p}(X) > 0$ for $h < 0$, while $d\delta(h)/dh < 0$ and $\tilde{p}(X) < 0$ for $h > 0$. Since $\Delta h^2(X) > 0$ regardless of $\Delta h(X)$, (E2) is always positive.

Appendix F
Derivation of Var $\{\hat{\epsilon}\}$

Keeping up to the second-order terms of $\Delta\mathbf{h}$,

$$e^{j\omega_1\hat{\mathbf{h}}(X)}e^{j\omega_2\hat{\mathbf{h}}(Y)} = e^{j\omega_1\mathbf{h}(X)}e^{j\omega_2\mathbf{h}(Y)}e^{j\omega_1\Delta\mathbf{h}(X)}e^{j\omega_2\Delta\mathbf{h}(Y)}$$

$$\cong e^{j\omega_1\mathbf{h}(X)}e^{j\omega_2\mathbf{h}(Y)}[1 + j\omega_1\Delta\varsigma_1(X) + j\omega_2\Delta\varsigma_2(Y) - \omega_1\omega_2\Delta\mathbf{h}(X)\Delta\mathbf{h}(Y)] \quad \text{(F1)}$$

where

$$\Delta\varsigma_i(X) = \Delta\mathbf{h}(X) + \frac{j\omega_i}{2}\Delta\mathbf{h}^2(X) \quad \text{(F2)}$$

Thus, the first line of (3.22) can be expanded to

$$\begin{aligned}
\mathrm{Var}_d\{\hat{\epsilon}\} \cong\ & \frac{1}{2\pi}\int_{S_x}\int_{-\infty}^{+\infty}\frac{e^{j\omega_1\mathbf{h}(X)}}{j\omega_1}\tilde{p}(X)d\omega_1 dX \cdot \frac{1}{2\pi}\int_{S_y}\int_{-\infty}^{+\infty}\frac{e^{j\omega_2\mathbf{h}(Y)}}{j\omega_2}\tilde{p}(Y)d\omega_2 dY \\
& + \frac{1}{2\pi}\int_{S_x}\int_{-\infty}^{+\infty}E_d\{\Delta\varsigma_1(X)\}e^{j\omega_1\mathbf{h}(X)}\tilde{p}(X)d\omega_1 dX \cdot \frac{1}{2\pi}\int_{S_y}\int_{-\infty}^{+\infty}\frac{e^{j\omega_2\mathbf{h}(Y)}}{j\omega_2}\tilde{p}(Y)d\omega_2 dY \\
& + \frac{1}{2\pi}\int_{S_x}\int_{-\infty}^{+\infty}\frac{e^{j\omega_1\mathbf{h}(X)}}{j\omega_1}\tilde{p}(X)d\omega_1 dX \cdot \frac{1}{2\pi}\int_{S_y}\int_{-\infty}^{+\infty}E_d\{\Delta\varsigma_2(Y)\}e^{j\omega_2\mathbf{h}(Y)}\tilde{p}(Y)d\omega_2 dY \\
& + \frac{1}{4\pi^2}\int_{S_y}\int_{S_x}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}E_d\{\Delta\mathbf{h}(X)\Delta\mathbf{h}(Y)\}e^{j\omega_1\mathbf{h}(X)}e^{j\omega_2\mathbf{h}(Y)}\tilde{p}(X)\tilde{p}(Y)d\omega_1 d\omega_2 dX dY \\
& - (\overline{\epsilon} - \frac{1}{2})^2 \qquad \text{(F3)}
\end{aligned}$$

The first line of (F3) is $(\epsilon - \frac{1}{2})^2$ from (4), and the second and third lines are each $(\epsilon - \frac{1}{2})\overline{\Delta\epsilon}$ from (3.21). Furthermore, the summation of the first, second, third and fifth lines is $(\epsilon - \frac{1}{2})^2 + 2(\epsilon - \frac{1}{2})\overline{\Delta\epsilon} - (\overline{\epsilon} - \frac{1}{2})^2 = -\overline{\Delta\epsilon}^2$ where

$\overline{\epsilon} = \epsilon + \overline{\Delta\epsilon}$. Since $\overline{\Delta\epsilon}$ is proportional to $\mathrm{E_d}\{\Delta\mathbf{h}(\mathrm{X}) + \frac{\mathrm{j}\omega}{2}\Delta\mathbf{h}^2(\mathrm{X})\}$ ($\sim 1/\mathcal{N}$) from (3.21), $\overline{\Delta\epsilon}^2$ is proportional to $1/\mathcal{N}^{\;2}$ and can be neglected. Thus, only the fourth line remains uncancelled, which is the second line of (3.22).

Appendix G
Derivation of Expressions for m_2 and σ_2^2

Let us assume that $\sum_1 = I$, $\sum_2 = \Lambda$, $M_1 = 0$, and $M_2 = M = [\mu_1, \ldots, \mu_n]^T$. These assumptions do not hurt any generality. First, z of (5.1) can be modified as

$$z = \frac{1}{n} X^T X = \frac{1}{n}(X-M+M)^T(X-M+M)$$
$$= \frac{1}{n}(X-M)^T(X-M) + \frac{2}{n} M^T(X-M) + \frac{M^T M}{n} \qquad (G1)$$

The expected value of z for class 2 (ω_2) is

$$m_2 = E\{z \mid \omega_2\} = \frac{1}{n} E\{(X-M)^T(X-M) \mid \omega_2\} + \frac{2}{n} M^T E\{(X-M) \mid \omega_2\} + \frac{M^T M}{n}$$
$$= \frac{1}{n} \operatorname{tr} E\{(X-M)(X-M)^T \mid \omega_2\} + \frac{M^T M}{n} = \frac{1}{n} \operatorname{tr} \Lambda + \frac{M^T M}{n}$$
$$= \frac{1}{n}\left(\sum_i \lambda_i + \sum_i \mu_i^2\right) \qquad (G2)$$

Likewise, the second order moment of z for ω_2 is

$$E\{z^2 \mid \omega_2\} = \frac{1}{n^2} E\{(X-M)^T(X-M)(X-M)^T(X-M) \mid \omega_2\}$$
$$+ \frac{4}{n^2} M^T E\{(X-M)(X-M)^T \mid \omega_2\} M$$
$$+ \frac{1}{n^2}(M^T M)^2 + \frac{2}{n^2} E\{(X-M)^T(X-M) \mid \omega_2\} M^T M$$

$$= \frac{1}{n}\left[3\sum_i \lambda_i^2 + 2\sum_{i>j}\sum \lambda_i\lambda_j\right] + \frac{4}{n^2}\sum_i \lambda_i \mu_i^2 + \frac{1}{n^2}(\sum_i \mu_i^2)^2$$

$$+ \frac{2}{n^2}(\sum_i \lambda_i)(\sum_i \mu_i^2) \qquad \text{(G3)}$$

where X is assumed to be Gaussian. Thus, the variance of z for ω_2 is

$$\sigma_2^2 = E\{z^2 | \omega_2\} - m_2^2 = \frac{1}{n^2}\left[2\sum_i \lambda_i^2 + 4\sum_i \lambda_i \mu_i^2\right] \qquad \text{(G4)}$$

Appendix H
Derivation of the Acquisition Probability of (5.2)

The acquisition probability of (5.2) is derived as follows:

$$P_a = \Pr\{\text{the smallest z is from class 1}\}$$

$$= \sum_{i=1}^{\infty} \Pr\{A_i \text{ and } B_i \text{ and } C_i\}$$

$$= \sum_{i=1}^{\infty} \Pr\{A_i\} \Pr\{B_i \mid A_i\} \Pr\{C_i \mid A_i,B_i\} \tag{H1}$$

where $A_i = \{\text{no sample in } 0 \leqq z < i\Delta t\}$, $B_i = \{\text{one class 1 sample is in } i\Delta t \leqq z < (i+1)\Delta t\}$ and $C_i = \{k_1-1 \text{ class 1 samples and } k_2 \text{ class 2 samples in } (i+1)\Delta t \leqq z < \infty\}$. $\Pr\{A_i\}$, $\Pr\{B_i \mid A_i\}$ and $\Pr\{C_i \mid A_i,B_i\}$ may be computed as follows:

$$\Pr\{A_i\} = \binom{k_1}{0} u_1^0(i\Delta t)(1-u_1(i\Delta t))^{k_1} \binom{k_2}{0} u_2^0(i\Delta t)(1-u_2(i\Delta t))^{k_2}$$

$$= (1-u_1(i\Delta t))^{k_1} (1-u_2(i\Delta t))^{k_2} \tag{H2}$$

$$\Pr\{B_i \mid A_i\} = \binom{k_1}{1} \left[\frac{\Delta u_1(i\Delta t)}{1-u_1(i\Delta t)}\right]^1 \left[1-\frac{\Delta u_1(i\Delta t)}{1-u_1(i\Delta t)}\right]^{k_1-1}$$

$$\times \binom{k_2}{0} \left[\frac{\Delta u_2(i\Delta t)}{1-u_2(i\Delta t)}\right]^0 \left[1-\frac{\Delta u_2(i\Delta t)}{1-u_2(i\Delta t)}\right]^{k_2}$$

$$\simeq k_1 \frac{\Delta u_1(i\Delta t)}{1-u_1(i\Delta t)} \tag{H3}$$

$$\Pr\{C_i \mid A_i,B_i\} = 1 \tag{H4}$$

where $\Delta u_j(i\Delta t)$ is the probability of a class j sample filling in

$i\Delta t \leqq z < (i+1)\Delta t$. The approximation of (H3) is obtained by making $\Delta u_j \rightarrow 0$. Substituting (H2), (H3) and (H4) into (H1) and letting $\Delta u_j \rightarrow 0$, we can obtain

$$P_a = \int_0^1 k_1(1-u_1)^{k-1}(1-u_2)^{k_2}\,du_1 \qquad \text{(H5)}$$

The summation of (H1) is taken by changing t from 0 to ∞. Since $u_i(0)=0$ and $u_i(\infty)=1$ and $u_i(t)$'s are the monotonic functions of t, the integration is taken with respect to u_1 from 0 to 1.

VITA

VITA

Raymond R. Hayes was born in Ann Arbor, Michigan on July 10, 1962. He received his B.S. in computer and electrical engineering at Purdue University, West Lafayette, Indiana, in May of 1984 under the Bell Labs Engineering Scholarship Program. Since then he has continued at Purdue in pursuit of the Ph.D. degree in electrical engineering under the IBM Resident Study Program. Mr. Hayes is a member of Phi Kappa Phi, Golden Key, Tau Beta Pi, Eta Kappa Nu, and the IEEE computer society.