12-1-1987

Nonparametric Estimation of the Bayes Error

Keinosuke Fukunaga
Purdue University

Donald M. Hummels
Purdue University

Nonparametric Estimation of the Bayes Error

Keinosuke Fukunaga
Donald M. Hummels

TR-EE 87-45
December 1987

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

This work was supported by the National Science Foundation under Grant ECS8513720.

NONPARAMETRIC ESTIMATION OF THE BAYES ERROR

Keinosuke Fukunaga and Donald M. Hummels

TR-EE 87-45

December 1987

This work was supported by the National Science Foundation under Grant ECS85 13720.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This thesis is concerned with the performance of nonparametric classifiers and their application to the estimation of the Bayes error. Although the behavior of these classifiers as the number of preclassified design samples becomes infinite is well understood, very little is known regarding their finite sample error performance. Here, we examine the performance of Parzen and k-nearest neighbor (k-NN) classifiers, relating the expected error rates to the size of the design set and the various design parameters (kernel size and shape, value of k, distance metric for nearest neighbor calculation, etc.). These results lead to several significant improvements in the design procedures for nonparametric classifiers, as well as improved estimates of the Bayes error rate.

Our results show that increasing the sample size is in many cases not an effective practical means of improving the classifier performance. Rather, careful attention must be paid to the decision threshold, selection of the kernel size and shape (for Parzen classifiers), and selection of k and the distance metric (for k-NN classifiers). Guidelines are developed toward propper selection of each of these parameters.

The use of nonparametric error rates for Bayes error estimation is also considered, and techniques are given which reduce or compensate for the biases of the nonparametric error rates. A bootstrap technique is also developed which allows the designer to estimate the standard deviation of a nonparametric estimate of the Bayes error.

CHAPTER 1
INTRODUCTION

1.1 Motivation and Problem Statement

The practical design of a statistical pattern recognition system may generally be broken down into three distinct phases. In the first stage, the designer must obtain and normalize a set of preclassified feature vectors which in some way characterize the various classes to be distinguished. Since the entire design is dependent on this limited set of design samples, it is important that this set adequately describes the statistical properties of the feature vectors for each class being considered. The second phase of design is generally termed data structure analysis. This stage may involve a wide variety of statistical tests. Various parametric models for the data may be examined. Data clustering and feature extraction are two important procedures which fall into this stage of design. The last phase is the actual classifier design and evaluation. In this phase, the designer uses his knowledge of the structural properties of the data to develop effective decision rules to determine the class of an unknown feature vector.

The estimation of the Bayes error rate (the minimum probability of error which may be obtained by any decision rule) should play a fundamental role in each of the three design phases. In the earliest stages of design, Bayes error estimates give the designer an indication of whether the feature vectors he obtains are adequate to design a system with a desired level of performance. If the design vectors have a Bayes error rate which is higher than the allowed error rate for the system, then immediately the designer knows that a more complicated feature set will be required to meet the goals, and no time is lost in the later two stages of classifier design. During the structure analysis phase, Bayes error estimates play their primary role as a guide toward feature extraction. By comparing the Bayes error in the original space to that of the extracted features, one may determine the amount of classification information which has been lost in the feature extraction process. Also, Bayes error estimates provide a means with which to compare different feature extraction techniques in terms of their potential error performance. Finally, Bayes error estimates provide a benchmark by which the designer may gauge the success or failure of any particular classifier structure. For example, if a linear classifier provides an error rate of 16% when the Bayes error for the data has been determined to be 15%, then the designer knows that more complicated classifier structures will not result in any dramatic improvements in the error performance. On the other hand, if the Bayes error for the data is 3%, then more complicated structures must be examined.

Since assumptions regarding the mathematical forms of the distributions of the data are not appropriate for Bayes error estimation, nonparametric procedures which converge for any distribution should be used. There are two nonparametric approaches to Bayes error estimation: those based on k-nearest neighbor (k-NN) procedures and those based on Parzen estimates. In both cases, the classification decision rule is formed by using nonparametric estimates of the density functions in place of the (unknown) actual density functions in the Bayes decision rule. The error is then estimated by counting the number of misclassified samples using this rule, and dividing by the total number of samples tested.

When an infinite number of samples is available, the behavior of the estimates is fairly well understood. Asymptotically (as the number of design samples becomes infinite), the k-NN errors are known to place bounds on the Bayes error: upper bounds for odd k's, and lower bounds for even k's. The bounds improve, and converge to the Bayes error, as k increases. Similar statements may be made regarding the error performance of classifiers based on Parzen density estimates. In practice, however, the number of available design samples is always very limited. Under these conditions, the k-NN errors do not necessarily bound the Bayes error, and the relationship of the error rates obtained using Parzen density estimates to the true Bayes error has not been determined. It is the goal of this thesis to clarify the behavior of these estimates when the number of design samples is limited, and to examine the effects of sample size, dimensionality, and distributions on the estimates. These results will be used to improve the estimation procedure, resulting in significantly more reliable estimates.

1.2 Thesis Organization

Chapter 2 of this thesis deals with the bias of the 1-NN and 2-NN risk estimates from their asymptotic values. Expressions are given which separate the effects of dimensionality and sample size from those of the distributions. A possible means of using this knowledge to compensate for the bias and obtain an improved estimate is also given.

In Chapter 3 we introduce the Parzen and k-NN density estimates, and their application to Bayes error estimation. The selection of each of the design parameters (Parzen kernel shape, k-NN metric, value of k used in k-NN classification, etc.) is discussed. The proper selection of the decision threshold is shown to play an important role for both k-NN and Parzen error estimates. By casting the k-NN classifier into a density estimation framework, we are able to develop the k-NN and Parzen procedures in a parallel manner, clarifying the fundamental relationships between the two. The discoveries of this chapter also result in a significant improvement for both the k-NN and Parzen error estimates. The results of this chapter are only valid if the Parzen kernel shape (or k-NN metric) is chosen independently from the data which is used to evaluate the classifiers. Chapter 4 presents an efficient, practical, computational algorithm to satisfy this requirement.

Chapter 5 uses the density estimation framework developed in Chapter 3 to obtain expressions relating the finite sample nonparametric error rates to the true Bayes error. This is in contrast to the results of chapter 2 which relate the finite sample 1-NN and 2-NN error rates to the asymptotic 1-NN and 2-NN errors. Through a curve fitting technique, we are able to use a set of observed (finite design set) error rates to obtain a true estimate of the Bayes error. The role of the decision threshold introduced in Chapter 3 is made more clear in this chapter. Also, we obtain results concerning the selection of the optimal Parzen kernel shape which nicely complement k-NN optimal metric results already in existence.

In Chapter 6, we examine a bootstrap procedure which may be used to estimate the variance of our Bayes error estimates. Thus we provide a means by which the reliability of our estimate may be judged.

Chapter 7 gives a summary of the major contributions of this thesis.

CHAPTER 2
BIAS OF NEAREST NEIGHBOR RISK ESTIMATES

2.1 Introduction

Classification error estimation using k-nearest neighbor (NN) rules has been a popular topic of research ever since Cover and Hart demonstrated the properties of the NN classifier [1]. Their results show that the expected error of the NN rule converges asymptotically (as the sample size becomes infinite) to a value which is between the Bayes error and twice the Bayes error. Thus k-NN errors provide a means of nonparametrically placing bounds on the Bayes error, provided the sample size is sufficiently large.

When a finite number of samples is available, however, the k-NN errors do not necessarily bound the Bayes error, and the NN classifier may yield an error which is much more than twice the Bayes error. Cover [2] investigated the finite sample performance of the NN classifier for the 1-dimensional case and found that, assuming almost surely continuous a posterior probability functions, the bias of the NN error from its asymptotic value is bounded by the function which is $0(1/N^2)$, where N is the sample size. Wagner [3] pointed out that, in terms of examining the convergence of the NN rule, a more meaningful criterion is $\Pr\{|L_N - \epsilon^*| > \mu\}$, where L_N is the probability of error given N preclassified samples, ϵ^* is its asymptotic value, and μ is a constant. He provided an exponential bound on this criterion under several additional assumptions including the continuity of the density function. Fritz [4] significantly reduced the conditions on the distributions and improved the bound provided by Wagner. Stone [5] demonstrated that if k is allowed to vary with N, the k-NN errors will converge to the Bayes error (in probability) for any distribution. Devroye [6] has shown that, with an additional assumptions of how k increases with N, the convergence holds with probability one. Further, Devroye [7] showed that, for a fixed k, the asymptotic results of the k-NN rule are distribution free. Other significant contributions concerning the convergence of NN rules have been made by a number of authors [8,9,10,11].

While these results give an indication of how the NN classifier performance improves as the sample size becomes infinite, they do little toward examining the behavior of the classifier when a finite number of preclassified samples is available. If the NN rules are to be used to place bounds on the Bayes error, then an important question is whether the finite sample NN error is significantly biased from its asymptotic value, and whether that bias may be effectively reduced by increasing the sample size. In this report, rather than finding bounds on a convergence criterion, we concern ourselves with the estimation of the bias between the finite sample and asymptotic errors. We have found that, depending on the distributions of the data, the k-NN errors in practice may exhibit a significant bias from their theoretical asymptotic value, so that they no longer place bounds on the Bayes error. Until now, most of the work in reducing the size of the bias has concentrated on the selection of an optimal metric [12,13]. In this chapter, we approach the problem in a different (and more general) point of view.

In Section 2.2, we begin by deriving the form of the bias of the NN classifier in terms of the sample size, metric, dimensionality, and distributions. We will show that the bias may be expressed as a product of two terms, the first of which is independent of the distributions, and the second of which is independent of the sample size. Thus we have isolated the effect of sample size from that of the distribution, giving an indication of the number of samples required to reduce the bias, and also of the relationship between that number and the dimensionality of the data. It can be shown that, for the one-dimensional case, our result agrees with Cover's result [2]. Thus, we have provided a generalization of his result to the n-dimensional case, while including the effects of other factors, such as the metric.

In Section 2.3 we separately consider the effects of dimensionality, sample size, metric, and distributions on the finite-sample NN error. In order to verify the theoretical results, some experimental results using two-class Gaussian mixture densities are given in Section 2.4. We then present in Section 2.5 a possible means of using these results to compensate for the bias and obtain an improved estimate of the asymptotic error. This procedure involves measuring the NN errors for several different samples sizes, and using our derived relationships to extrapolate an estimate of the asymptotic error. The results for the NN classifier are extended to the 2-NN case in Section 2.6. Finally, we consider the direct application of these results to the multiclass problem in Section 2.7.

2.2 The Bias of the NN Error

Let **X** be an n-dimensional random vector with density p(X), where the boldface type indicates randomness. In NN classification, **X** is to be assigned to one of two classes, ω_1 and ω_2, according to the class of its NN, $\mathbf{X}_{NN}$. In order to keep our initial discussion as simple as possible, only the two class problem is discussed here. Extensions to the multiclass problem are given in Section 2.7. The risk associated with the NN classifier, then, is given by

$$r_1(\mathbf{X},\mathbf{X}_{NN}) = q_1(\mathbf{X})q_2(\mathbf{X}_{NN}) + q_2(\mathbf{X})q_1(\mathbf{X}_{NN}) \tag{1}$$

where $q_i(X)$ is the a posteriori probability of class ω_i given X. Under an asymptotic analysis, it is assumed that the sample size is made arbitrarily large and $\mathbf{X}_{NN}$ converges to **X** with probability one. Thus, the asymptotic NN risk is obtained by setting $q_i(\mathbf{X}_{NN}) = q_i(\mathbf{X})$ in (1) giving

$$r_1^*(\mathbf{X}) = 2q_1(\mathbf{X})q_2(\mathbf{X}). \tag{2}$$

In the finite sample case, however, there is a finite difference between $q_i(\mathbf{X})$ and $q_i(\mathbf{X}_{NN})$. Thus we define ϵ so that

$$q_1(\mathbf{X}_{NN}) = q_1(\mathbf{X}) + \epsilon \quad \text{and} \quad q_2(\mathbf{X}_{NN}) = q_2(\mathbf{X}) - \epsilon \tag{3}$$

Eq. (3) holds since $q_1(\mathbf{X}) + q_2(\mathbf{X}) = 1$ and $q_1(\mathbf{X}_{NN}) + q_2(\mathbf{X}_{NN}) = 1$. Substituting (3) into (1) we obtain

$$r_1(\mathbf{X},\mathbf{X}_{NN}) = r_1^*(\mathbf{X}) + b(\mathbf{X},\mathbf{X}_{NN}) \tag{4}$$

where

$$b(\mathbf{X},\mathbf{X}_{NN}) = (q_2(\mathbf{X}) - q_1(\mathbf{X}))\,\epsilon \tag{5}$$

Thus, the bias between the finite sample and asymptotic NN errors may be found by taking $E\{b(\mathbf{X},\mathbf{X}_{NN})\}$ with respect to both **X** and $\mathbf{X}_{NN}$. In order to accomplish this, we first approximate ϵ using a second order Taylor series around **X**:

$$\epsilon \cong \tilde{\epsilon} = \nabla^T q_1(\mathbf{X})\delta_1 + \frac{1}{2}\delta_1^T \nabla^2 q_1(\mathbf{X})\delta_1 \tag{6}$$

where

$$\delta_1 = \mathbf{X}_{NN} - \mathbf{X} \tag{7}$$

Before we can take the expectation, we must also specify the metric which is to be used. In order to simplify our derivation and ensure that our results are applicable to practical situations, we will assume that a quadratic metric

$$d_A(X,Y) = [(X-Y)^T A(X-Y)]^{1/2} \tag{8}$$

is used to measure distances between samples X and Y. In the case that A is held fixed, this is a global metric. However, in the more general case, A may be allowed to vary with X, forming a local metric. Thus, in our derivation we will assume that A could be a function of X.

In order to evaluate the expectation of (5), we follow a procedure very similar to that used by Fukunaga and Flick [13]. We begin by breaking the expectation into three stages as $\underset{\mathbf{X}}{E}\ \underset{\rho}{E}\ \underset{\mathbf{X}_{NN}}{E}\{(q_2(\mathbf{X})-q_1(\mathbf{X}))\bar{\varepsilon}\,|\rho,\mathbf{X}\}$ where ρ is the distance between $\mathbf{X}$ and its NN, $\rho = d_A(\mathbf{X},\mathbf{X}_{NN})$.

2.2.1 Expectation with respect to $\mathbf{X}_{NN}$

The first step in evaluation of the bias is to calculate the expectation with respect to $\mathbf{X}_{NN}$, given $\mathbf{X}=X$ and $\rho=\rho$. That is, the bias is to be averaged over all $\mathbf{X}_{NN}$ on a hyperellipsoidal surface with a constant ρ, $S(\rho) = \{Y:\rho = d_A(Y,X)\}$.

$$\underset{\mathbf{X}_{NN}}{E}\{(q_2(X)-q_1(X))\bar{\varepsilon}\,|\rho,X\} = \frac{(q_2(X)-q_1(X))\int_{S(\rho)} \bar{\varepsilon}p(X_{NN})dX_{NN}}{\int_{S(\rho)} p(X_{NN})dX_{NN}}$$

$$\cong \frac{\mathrm{tr}\left\{(q_2(X)-q_1(X))\,(\nabla p(X)\nabla^T q_1(X) + 1/2p(X)\nabla^2 q_1(X)) \int_{S(\rho)} \delta_1\delta_1^T dX_{NN}\right\}}{p(X)\int_{S(\rho)} dX_{NN}}$$

$$= \mathrm{tr}\left\{A^{-1}[1/n(q_2(X)-q_1(X))\,(\nabla p(X)\nabla^T q_1(X)/p(X) + 1/2\nabla^2 q_1(X))]\right\}\rho^2 \tag{9}$$

The second line of (9) is obtained by approximating $p(X_{NN})$ by a Taylor series as

$$p(X_{NN}) \cong p(X) + \delta_1^T \nabla p(X) \tag{10}$$

Note that all odd order terms of δ_1 disappear after taking the integration, since $S(\rho)$ is symmetric around X.

2.2.2 Expectation with respect to ρ

Let **u** represent the probability enclosed by the surface $S(\rho)$. Assuming (10),

$$u \cong \alpha \rho^n p(X) |A|^{-1/2} \tag{11}$$

where $\alpha\rho^n |A|^{-1/2}$ is the volume of the region surrounded by $S(\rho)$ and

$$\alpha = \frac{\pi^{n/2}}{\Gamma(n/2+1)} \quad . \tag{12}$$

Here Γ represents the gamma function and n is the dimensionality. The random variable **u** is known to have the beta distribution [14]

$$p(u) = N(1-u)^{N-1} \tag{13}$$

where N is the number of samples. Thus the expectation of (9) with respect to ρ may be accomplished by substituting **u** for ρ using (11) and taking the expectation with respect to u. This yields

$$\underset{\rho}{E}\{\rho^2 | X\} = \underset{u}{E}\{\alpha^{-2/n} p^{-2/n}(X) |A|^{1/n} u^{2/n} | X\} \tag{14}$$

$$= \alpha^{-2/n} p^{-2/n}(X) |A|^{1/n} \frac{\Gamma(2/n+1)\Gamma(N+1)}{\Gamma(N+1+2/n)} \tag{15}$$

2.2.3 Expectation with respect to X

Substituting (15) into (9) and taking the expectation we obtain

$$E\{b(X,X_{NN})\} \cong \beta_1 \underset{X}{E}\{ |A|^{1/n} \operatorname{tr}(A^{-1}B_1(X))\} \tag{16}$$

where

$$B_1(X) = p^{-2/n}(X)(q_2(X) - q_1(X))(\nabla p(X) \nabla^T q_1(X)/p(X) + 1/2 \nabla^2 q_1(X)) \tag{17}$$

$$\beta_1 = \frac{\Gamma^{2/n}(n/2+1)}{n\pi} \frac{\Gamma(2/n+1)\Gamma(N+1)}{\Gamma(N+1+2/n)} \quad . \tag{18}$$

Eq. (16) is the bias of the NN error estimator due to a finite sample size.

For large value of N, β_1 become proportional to $N^{-2/n}$. Thus our results are in agreement with those obtained by Cover [2] which showed that in the 1-dimensional case the bias drops as N^{-2}.

2.3 Effect of Parameters, n, N, A and p(X)

Several observations may be made at this point. First, note that the value of β_1 is completely independent of the mixture density. It depends only on the dimensionality of the data (n) and the sample size (N) and does not depend on the particular distributions involved. The term inside the expectation in (16) on the other hand, does not depend on the sample size. For any given set of distributions this term remains fixed regardless of the number of samples. This term does, however, depend heavily on the selection of the metric, A.

Our expression, therefore, yields much information about how the bias is effected by each of the parameters of interest (n, N, A, and p(X)). Each of these parameters will be discussed separately as follows.

2.3.1 Sample Size (N)

Eq. (18) gives an explicit expression showing how the sample size effects the size of the bias of the NN error. β_1 vs. N is plotted in Figure 2.1 for various n. Thus, we have obtained valuable insight into the number of samples which are required to obtain a reliable NN error. As shown in Figure 2.1, the bias tends to drop off rather slowly as the sample size increases, particularly when the dimensionality of the data is high. This is not an encouraging result, since it tends to indicate that, when the dimensionality of the data is high, increasing the sample size N is not an effective means of reducing the bias. For example, with a dimensionality of 64, increasing the number of samples from 1,000 to 10,000 results in only a 6.9% reduction in the bias (β_1 from .0504 to .0469). Further reduction by 6.9% would require increasing the number of samples to over 100,000. Thus it does not appear that the asymptotic NN error may be estimated simply by "choosing a large enough N" as generally believed, especially when the dimension of the data is high. The required value of N would be prohibitively large. This has been a repeated observation in our experience, and has motivated us to initiate this investigation.

2.3.2 Effects of Dimensionality (n)

The dimensionality of the data appears to play an important role in determining the relationship between the size of the bias and the sample size. As is shown in Figure 2.1, for small values of n (say, $n \leq 4$), changing the sample size is an effective means of reducing the bias. For larger values

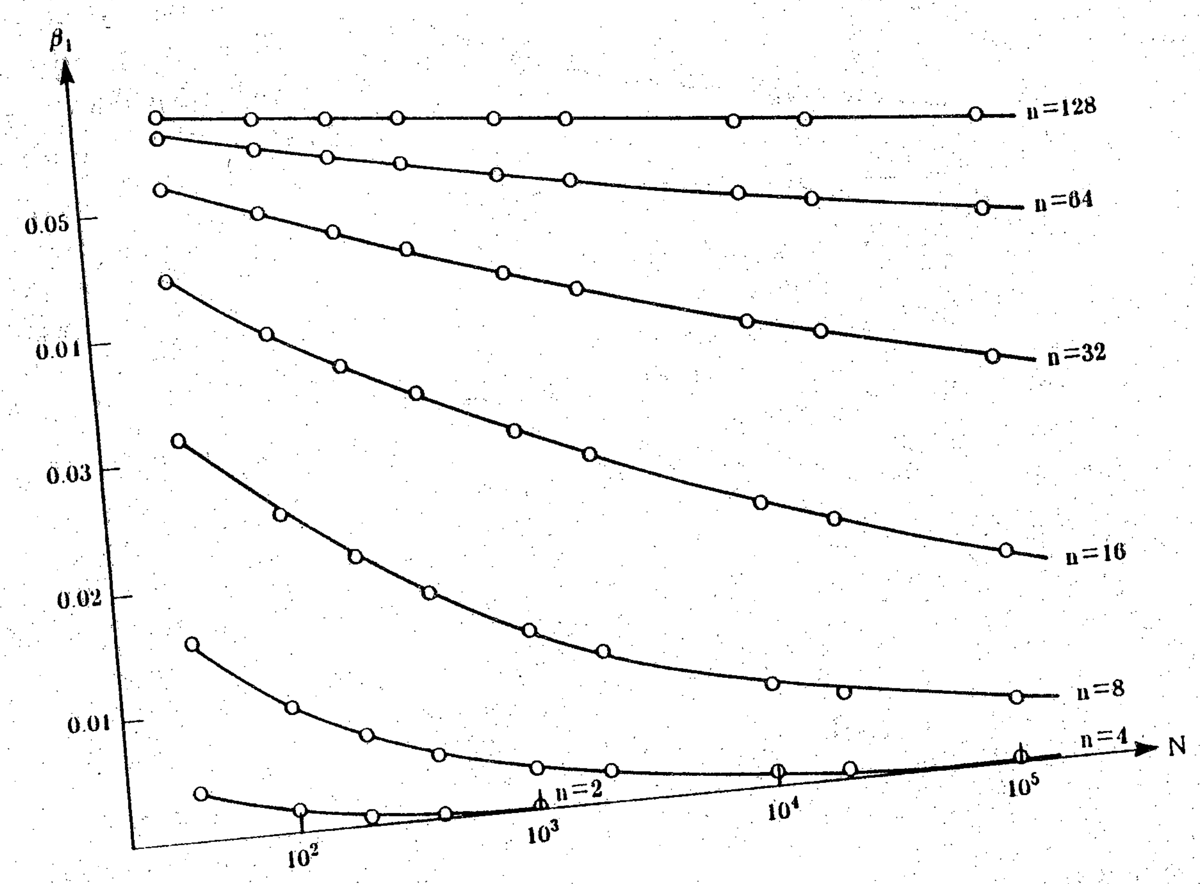


Figure 2.1 β_1 vs. N

of n, however, increasing the number of samples becomes a more and more futile means of improving the estimate. It is in these higher dimensional cases that improved techniques of accurately estimating the Bayes error are needed.

It should be pointed out that, in our expression for the bias of the NN error, n represents the "local" or "intrinsic" dimensionality of the data*. NN statistics are determined by local sample distribution, and not related to global one. In many applications, intrinsic dimensionality is much smaller than the dimensionality of the observation space. Therefore, it is necessary that the intrinsic dimensionality be estimated from the data in order to calculate β_1. A convenient means of estimating the intrinsic dimensionality which works well in conjunction with NN techniques is given by [15], [16].

$$n = \frac{1}{\dfrac{\overline{d_{2NN}}}{\overline{d_{NN}}} - 1} \tag{19}$$

where $\overline{d_{NN}}$ and $\overline{d_{2NN}}$ are the mean distances to the first and second NN respectively.

2.3.3 Effects of Densities

The expectation term of (16) gives the effect of densities on the size of the bias. In general, it is very hard to determine the effect of this term because of its complexity. In order to investigate the general trends, however, we have computed the term numerically for a Gaussian case. When two density functions $p_i(X)$ $(i = 1,2)$ are Gaussian with the expected vector M_i and covariance matrix Σ_i, B_1 of (17) becomes

$$\begin{aligned} B_1(X) = p^{-2/n}(X)\,(q_2(X) - q_1(X))q_1(X)q_2(X)\,\cdot \\ [V_1(X)V_1^T(X) - V_2(X)V_2^T(X) + \Sigma_2^{-1} - \Sigma_1^{-1}] \end{aligned} \tag{20}$$

where

$$V_i(X) = \Sigma_i^{-1}(X - M_i)\ . \tag{21}$$

The derivation of (20) is given in the Appendix.

*This was pointed out by Arnold C. Williams of Martin Marietta Aerospace, Orlando, FL, and was confirmed empirically by him on radar signature data, for which the intrinsic dimensionality was about a quarter of the measurement dimensionality.

Gaussian data was generated with $\Sigma_1 = \Sigma_2 = I$. The class means, M_1 and M_2, were separated to obtain the desired Bayes error. The Euclidean metric ($A = I$) was used. The B_1 of (20) was evaluated at each generated sample point where the mathematical formulas based on the Gaussian assumption were used to compute $p(X)$ and $q_i(X)$. The expectation of (16) was replaced by the sample mean taken over 1600n samples. The result is shown in Table 2.1 for dimensionality ranging from 2 to 16 and the Bayes error ranging from 2% to 30%.

Table 2.1 reveals many properties of the expectation term. But, special attention must be paid to the fact that, once n becomes large ($n > 4$), its value has little effect on the size of the expectation. This implies that β_1 of (18) dominates the effect of n on the bias. That is, the bias is much larger for high dimensions. This coincides with our frequent observations in practice that the NN error comes down, contrary to theoretical expectation, by selecting a smaller number of features. That is, the bias is reduced more than the increase of the Bayes error. In order to compare two sets of features in different dimensions, this dependency of the bias on n must be carefully examined.

Also, note in Table 2.1 that the second order term due to $\nabla^2 q_1(X)$ is comparable to or even larger than the first order term due to $\nabla q_1(X)$. It is for this reason that we have included the second order term in the Taylor series of (6).

2.3.4 Effect of metric (A)

The expectation term of (16) also indicates how the matrix, A, affects the bias. Certainly, proper selection of a metric may reduce the bias significantly. Unfortunately, B_1 is a very complex function of X and very hard to estimate for any given set of data. An exception is the Gaussian case in which (20), with known mathematical formulas for $q_1(X)$ and $p(X)$, provides a means to compute B_1.

As for optimization of A, Fukunaga and Hostetler [17] showed that an expression of the $|A|^{1/n}\operatorname{tr}(A^{-1}B_1)$ is minimized by setting $A = B_1$, provided B_1 is a positive definite matrix. However, (20) reveals that B_1 might not be positive definite.

Thus, it is not immediately clear how to choose A to minimize the bias. Nevertheless, selection of an appropriate metric remains an important topic in NN error estimation.

Table 2.1 Estimates of the Expectation Term in Eq. (16) for Data 1

Bayes Error		n = 2	n = 4	n = 8	n = 16
30%	1st order term	3.4	1.2	1.2	1.1
	2nd order term	2.2	1.2	0.4	0.3
	Sum	5.6	2.4	1.6	1.4
20%	1st order term	2.2	1.3	0.9	0.8
	2nd order term	1.8	1.2	1.1	1.0
	Sum	4.0	2.5	2.0	1.8
10%	1st order term	-1.3	-.2	-0.2	-0.2
	2nd order term	4.7	2.1	1.7	1.6
	Sum	3.4	1.9	1.5	1.4
5%	1st order term	-1.9	-1.0	-0.8	-0.6
	2nd order term	3.8	2.3	1.8	1.5
	Sum	1.9	1.3	1.0	0.9
2%	1st order term	-2.0	-1.5	-0.8	-0.7
	2nd order term	3.5	2.3	1.3	1.1
	Sum	1.5	0.8	0.5	0.4

2.4 Experimental Verification

2.4.1 Experiment for Data 1

In order to verify our results, an experiment was run for the same Gaussian data which were used to produce Table 2.1. The NN errors were measured empirically by generating N samples, performing the NN classification, counting the errors, and averaging the resulting error over 20 trials. Sample sizes of $N = 20n$, $40n$, $80n$, and $160n$ were used.

According to (16), the bias of the NN rule (and hence the actual NN error) varies linearly with β_1 for any given set of distributions. Therefore, taking the expectation of (4) with respect to X and X_{NN}, the finite sample NN error may be written as

$$\hat{\epsilon}_{NN} = \overset{*}{\epsilon}_{NN} + \beta_1 c_1 \tag{22}$$

where $\hat{\epsilon}_{NN} = E\{r_1(X,X_{NN})\}$, $\overset{*}{\epsilon}_{NN}$ is its asymptotic value, and c_1 is the expectation in (16). For the distributions used in these tests, the values of c_1 were previously estimated in Section 2.3, and are shown in Table 2.1. The theoretical asymptotic error, $\overset{*}{\epsilon}_{NN}$, was also estimated by generating a large number (1600n) of samples, calculating the risk at each sample point from (2) using the known mathematical forms of $q_i(X)$ for the Gaussian case, and averaging the result. These estimates are shown in Table 2.2.

Thus, using (22) in conjunction with the estimates of $\overset{*}{\epsilon}_{NN}$ and c_1, we can predict the finite sample NN error as a linear function of β_1. The dotted lines of Figure 2.2 (for $n = 8$) show these predicted NN errors for various values of the Bayes error. The $\hat{\epsilon}_{NN}$'s, which were obtained empirically by averaging the NN error over of 20 trials, are also plotted in Figure 2.2. Its standard deviation is also shown by the vertical bars ($\pm$ one standard deviation). Note that these measured $\hat{\epsilon}_{NN}$'s are reasonably close to the predicted values.

2.4.2 Experiment for Data 2

The experiment was repeated, again using 8-dimensional Gaussian random vectors, but this time choosing Σ_1 and Σ_2 to be significantly different (after simultaneous diagonalization of $\Sigma_1 \rightarrow I$ and $\Sigma_2 \rightarrow \Lambda$, the diagonal elements of Λ vary from 0.12 to 12.06 [18].) The empirically measured NN errors, as well as the predicted NN errors are plotted in Figure 2.3. In this case, the empirical NN errors tend to lie above values predicted by our estimates of $\overset{*}{\epsilon}_{NN}$ and c_1, although still preserving the linear relationship predicted by (22).

Table 2.2 Asymptotic NN Errors (%) for Data 1

Bayes Error	n = 2	n = 4	n = 8	n = 16
30%	38.9	38.9	38.9	38.8
20%	27.6	27.8	27.8	27.7
10%	14.5	14.6	14.6	14.6
5%	7.6	7.5	7.4	7.6
2%	3.2	3.0	3.0	3.2

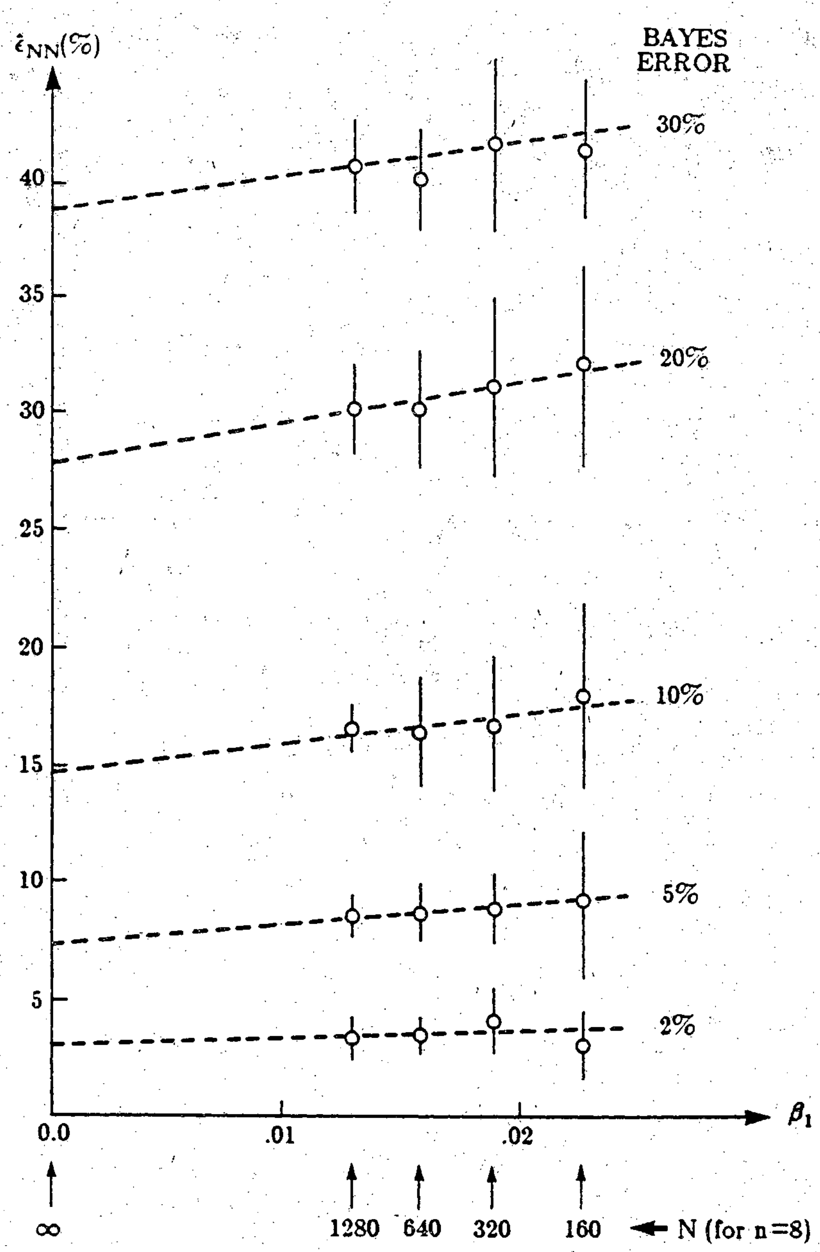


Figure 2.2 NN Errors for Data 1

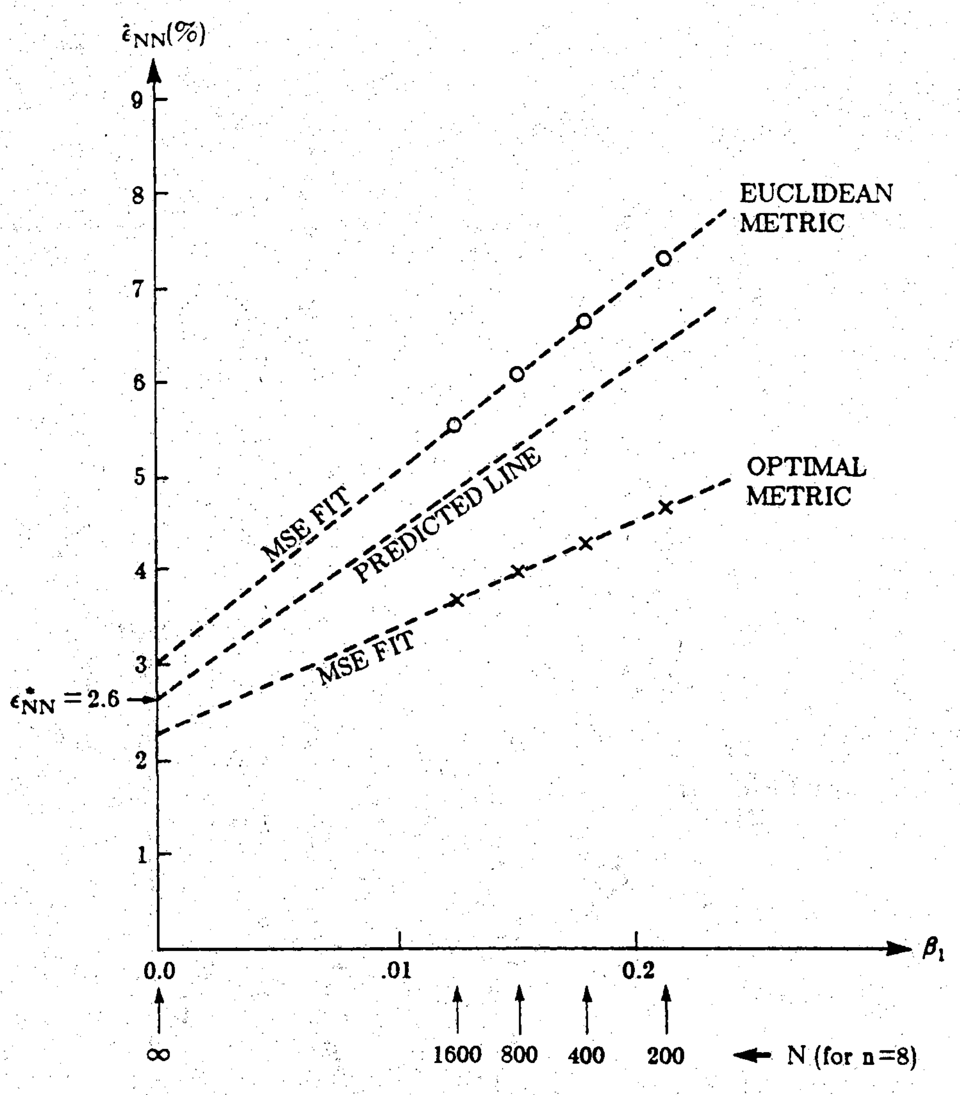


Figure 2.3 NN Errors for Data 2

2.4.3 Effect of Metric for Data 2

In order to demonstrate the effect of properly selecting the metric, the empirical NN errors were also measured using an optimized metric. In this case a global metric was assumed, so that the matrix A is not a function of X. The expected value of $B_1(X)$ in (16) was estimated by evaluating $B_1(X)$ from (20) at each generated sample point using the known mathematical forms for $p(X)$ and $q_i(X)$ under the Gaussian assumption, and taking the sample mean. A total of 100,000 samples were used to estimate this mean. In this case, the expectation of $B_1(X)$ happened to be a positive definite matrix, and hence the optimal metric is found by setting $A = E\{B_1(X)\}$.

The resulting NN errors, obtained using the optimal metric, $A_0 = E\{B_1(X)\}$, are plotted vs. β_1 in Figure 2.3 for various values of N. The results shown were averaged over 20 trials. Figure 2.3 confirms our expectation that the selection of an appropriate metric may significantly reduce the slope of the bias curve, resulting in more reliable estimates.

2.5 Estimation of the Asymptotic NN Error

While it may not be practical to obtain the asymptotic NN errors simply by increasing the sample size, it may be possible to use our information concerning how the bias changes with sample size to our advantage. $\hat{\epsilon}_{NN}$ could be measured empirically for several sample sizes, and β_1 obtained using either (18) or Figure 2.1. These values could be used in conjunction with (22) to obtain an estimate of the asymptotic NN error, ϵ^*_{NN}. Thus, our proposed procedure to estimate ϵ^*_{NN} is as follows:

1. Change the sample size N as $N_1, N_2, \ldots, N_m$. For each N_i, calculate β_1 and measure $\hat{\epsilon}_{NN}$ empirically.

2. Plot these m empirical points $\hat{\epsilon}_{NN}$ vs. β_1. Then, find a line best fitted to these m points. The slope of this line is c_1 and the y-intercept is ϵ^*_{NN} which we would like to estimate.

There are many possible ways of selecting a line. The standard procedure would be the minimum mean-square error approach. An alternative could be to weight the square error for each value of β_1 by the corresponding sample size, β_1, or the variance of the NN error.

This procedure was tested using Data 1 of Section 2.4. The standard minimum mean square approach was adopted to find lines. Table 2.3 shows the estimated values of ϵ^*_{NN} for each of the test cases. As may be seen by comparing Table 2.3 and Table 2.2, the predicted NN errors using the above procedure come very close to our estimates of the asymptotic error obtained in Table 2.2.

In order to test a more difficult case, the experiment was again repeated for Data 2 of Section 2.4. These distributions yield a theoretical Bayes error of 1.9% and an experimentally determined (see Section 2.4) asymptotic NN error of 2.6%. The minimum mean-square error line fitting the empirically measured NN errors has been plotted in Figure 2.3, and is seen to yield a much improved estimate of the asymptotic error. While increasing the sample size to 1600 (800 samples per class) only gave a predicted NN error of 5.5% in the conventional NN procedure (Euclidean metric), the proposed procedure has given a predicted error of 3.0% - much closer to the actual 2.6%. The same procedure was applied by using the optimal metric. Figure 2.3 shows that the line hits the y-axis at 2.2%.

In order for (22) to be valid with a constant c_1, $\hat{\epsilon}_{NN}$ must be the expected value of $r_1(\mathbf{X},\mathbf{X}_{NN})$ with respect to both $\mathbf{X}$ and $\mathbf{X}_{NN}$. For artificially generated data, the expectation may be approximated by generating many sets of the data and averaging the resulting NN errors. However, in practice, only one set of data is available. In this case it becomes necessary to partition the available data to obtain the various desired sample sizes. For example, given 1000 samples, only one trial could be used to estimate $\hat{\epsilon}_{NN}$ for $N = 1000$, two trials to estimate $\hat{\epsilon}_{NN}$ for $N = 500$, 4 trials for $N = 250$ and so on. The actual value used for $\hat{\epsilon}_{NN}$ in the proposed procedure would be the average of all trials at a particular sample size. Although the number of trials used decreases as the sample size increases, the variance of the NN error at the large sample size is smaller, so that hopefully the damage is minimized.

2.6 The Bias of the 2-NN Error

2.6.1 Derivation

An analysis of the 2-NN bias may be carried out in much the same manner as the NN bias. In 2-NN classification, a sample is counted as misclassification only if both of its two NN's are of the opposite class. The tie vote is treated as correct-classification. Because of this, the 2-NN error

Table 2.3 Estimates of Asymptotic NN Errors for Data 1

Bayes Error	n = 2	n = 4	n = 8	n = 16
30%	39.4	38.1	39.3	40.1
20%	28.1	27.6	27.3	28.0
10%	14.2	13.7	14.2	13.1
5%	8.1	7.5	7.6	5.9
2%	3.1	3.3	3.4	4.1

gives the lower bound of the Bayes error. Thus, the risk is given by

$$r_2(X,X_{NN},X_{2NN}) = q_1(X)q_2(X_{NN})q_2(X_{2NN}) + q_2(X)q_1(X_{NN})q_1(X_{2NN}) \quad (23)$$

where X_{2NN} is the second NN to X. Under the asymptotic assumptions, both X_{NN} and X_{2NN} converge to X, giving the asymptotic 2-NN risk

$$r_2^*(X) = q_1(X)q_2(X) \quad (24)$$

Thus, the 2-NN risk is (asymptotically) one half the NN risk. We now define ϵ_i such that $q_1(X_{iNN}) = q_1(X) + \epsilon_i$ and $q_2(X_{iNN}) = q_2(X) - \epsilon_i$ $(i=1,2)$. Substituting these into (23), and subtracting (24) from this gives the bias

$$b(X,X_{NN},X_{2NN}) = \epsilon_1\epsilon_2 \quad (25)$$

In order to find the expected value of (25), we approximate ϵ_1 and ϵ_2 using a second order Taylor series, and first take the expectation with respect to X_{NN} and X_{2NN}, holding their distances from X (given by ρ_1 and ρ_2) constant. This gives

$$E\{\tilde{\epsilon}_1\tilde{\epsilon}_2|X,\rho_1,\rho_2\} = [\mathrm{tr}\,\{A^{-1}(1/n(\nabla p(X)\nabla^T q_1(X)/p(X) + 1/2\nabla^2 q_1(X)))\}]^2\rho_1^2\rho_2^2 \quad (26)$$

We now let u_i be the probability enclosed by $S(\rho_i)$

$$u_i = \alpha\rho_i^n p(X)|A|^{-1/2} \quad (27)$$

where the joint density of u_1 and u_2 is known to be [14]

$$p(u_1,u_2) = \frac{N!}{(N-2)!}(1-u_2)^{N-2}, \quad 1 \geq u_2 \geq u_1 \geq 0 \quad (28)$$

Substituting u_i for ρ_i and taking the expectation, we obtain

$$E\{b(X,X_{NN},X_{2NN})\} = \beta_2 \underset{X}{E}\{[|A|^{1/n}\,\mathrm{tr}\,\{A^{-1}B_2(X)\}]^2\} \quad (29)$$

$$B_2(X) = p^{-2/n}(X)\,(\nabla p(X)\nabla^T q_1(X)/p(X) + 1/2\nabla^2 q_1(X)) \quad (30)$$

$$\beta_2 = \left(\frac{\Gamma^{2/n}(n/2+1)}{n\pi}\right)^2 \frac{\Gamma(1+4/n)\Gamma(N+1)}{\Gamma(N+1+4/n)} \frac{1+4/n}{1+2/n} \quad (31)$$

Thus we have obtained expressions very similar to those obtained for the NN error bias. The effect of sample size has successfully been isolated from that of the distribution.

Eq. (31) gives an explicit expression showing how the 2-NN error is affected by sample size. β_2 is plotted vs. N for various values of n in Figure 2.4. By comparing (31) to (18), we see that the 2-NN error converges to its

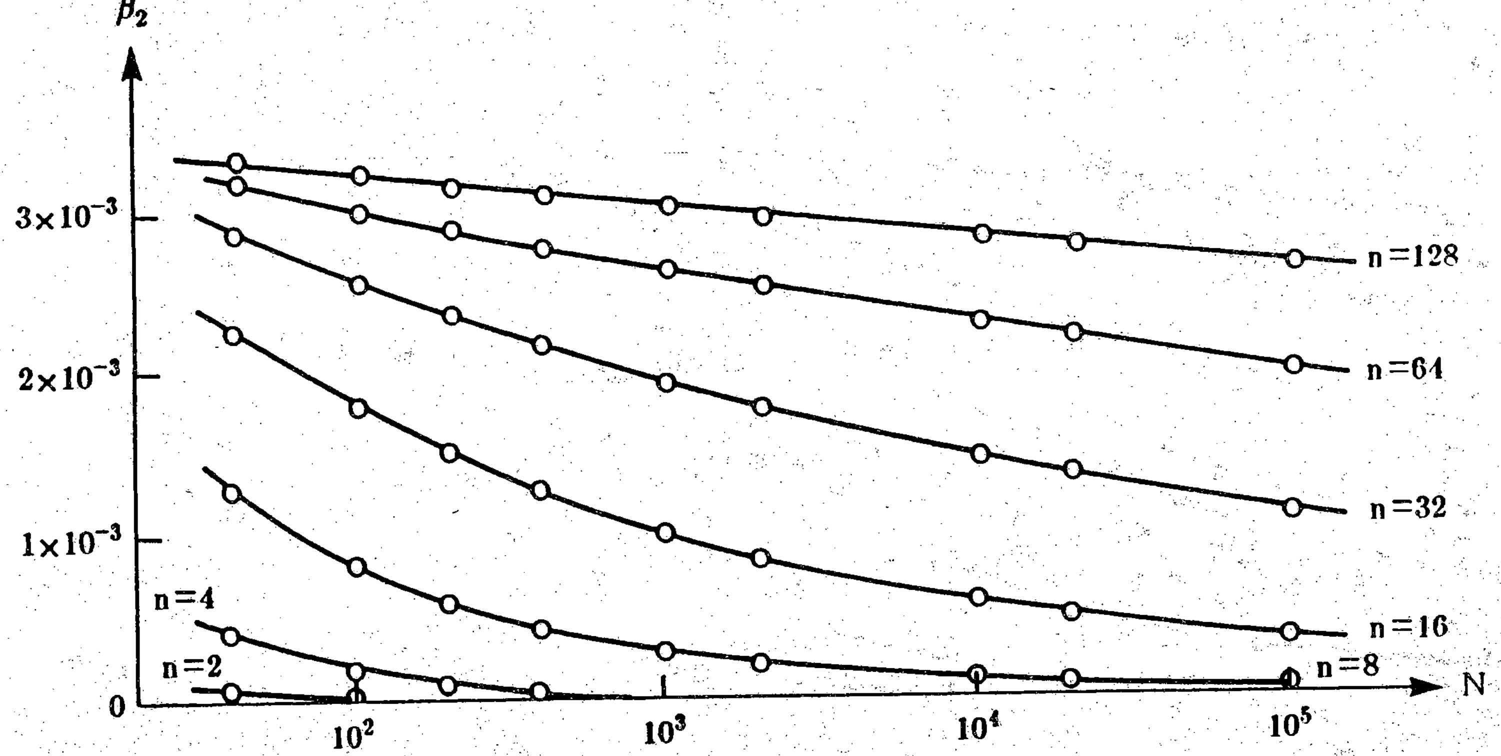


Figure 2.4 β_2 vs. N

asymptotic value more quickly than the NN error - as if the dimensionality, n, were half as large. Also note that β_2 is significantly smaller in magnitude than β_1. To some extent, this may be offset by an increased value for the expectation term of (29).

Also noteworthy is the fact that, under the approximations made here, the 2-NN bias is always positive, regardless of the distributions and metric used. Once again we see that the only apparent means of reducing the size of the expectation term in (29) is to adjust the metric. Just as in the NN case, when the expectation of (29) is taken we obtain a linear relationship between the 2-NN error and value of β_2

$$\hat{\epsilon}_{2NN} = \epsilon^*_{2NN} + \beta_2 c_2 \tag{32}$$

where $\hat{\epsilon}_{2NN} = E\{r_2(X, X_{NN}, X_{2NN})\}$, ϵ^*_{2NN} is its asymptotic value and c_2 is the expectation term of (29). Thus most of the arguments presented for NN error estimation (including the procedure for estimating the asymptotic error) are directly applicable to the 2-NN case, provided the new constant β_2 is used.

2.6.2 Experimental Verification

These results were tested using the same Gaussian test cases as were used in Section 2.4. Table 2.4 shows the estimated values for c_2. The empirically measured 2-NN error is plotted, along with our theoretically predicted results, in Figure 2.5 for $n = 8$. As may be seen in the figure, there is a close correspondence between the empirically measured 2-NN errors and the values predicted using (32). The experiment was also repeated using Data 2. The results are plotted in Figure 2.6. Also shown in Figure 6 is the minimum mean-square error line best fitting the empirical data. This line yields a predicted asymptotic 2-NN error rate of 1.85%, higher than the actual 1.3%, but still closer than the 2.33% value predicted by accepting the $N = 1600$ estimate.

2.7 Extensions to Multiclass Classifiers

In the proceeding, we have assumed that X belongs to one of two classes, ω_1 and ω_2. We now generalize these results to the M class problem. The NN risk associated with the M class classifier is given by

Table 2.4 Estimates of the Expectation Term in Eq. (29) for Data 1

Bayes Error	2-NN n = 2	n = 4	n = 8	n = 16
30%	11,000	144.5	18.2	11.1
20%	9,000	45.8	16.6	10.3
10%	14,000	35.0	11.5	6.4
5%	460	24.5	6.5	3.9
2%	2,800	23.4	3.2	1.9

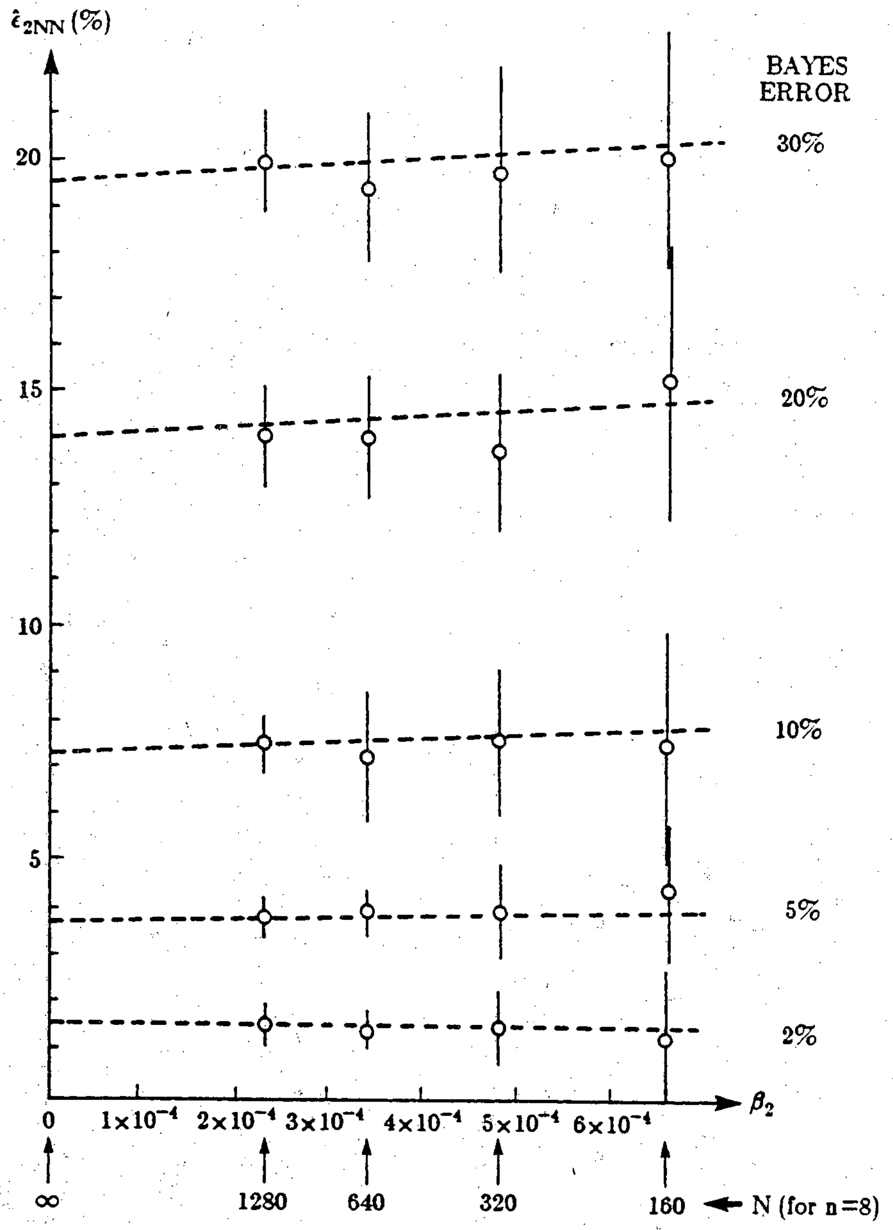


Figure 2.5 2-NN Errors for Data 1

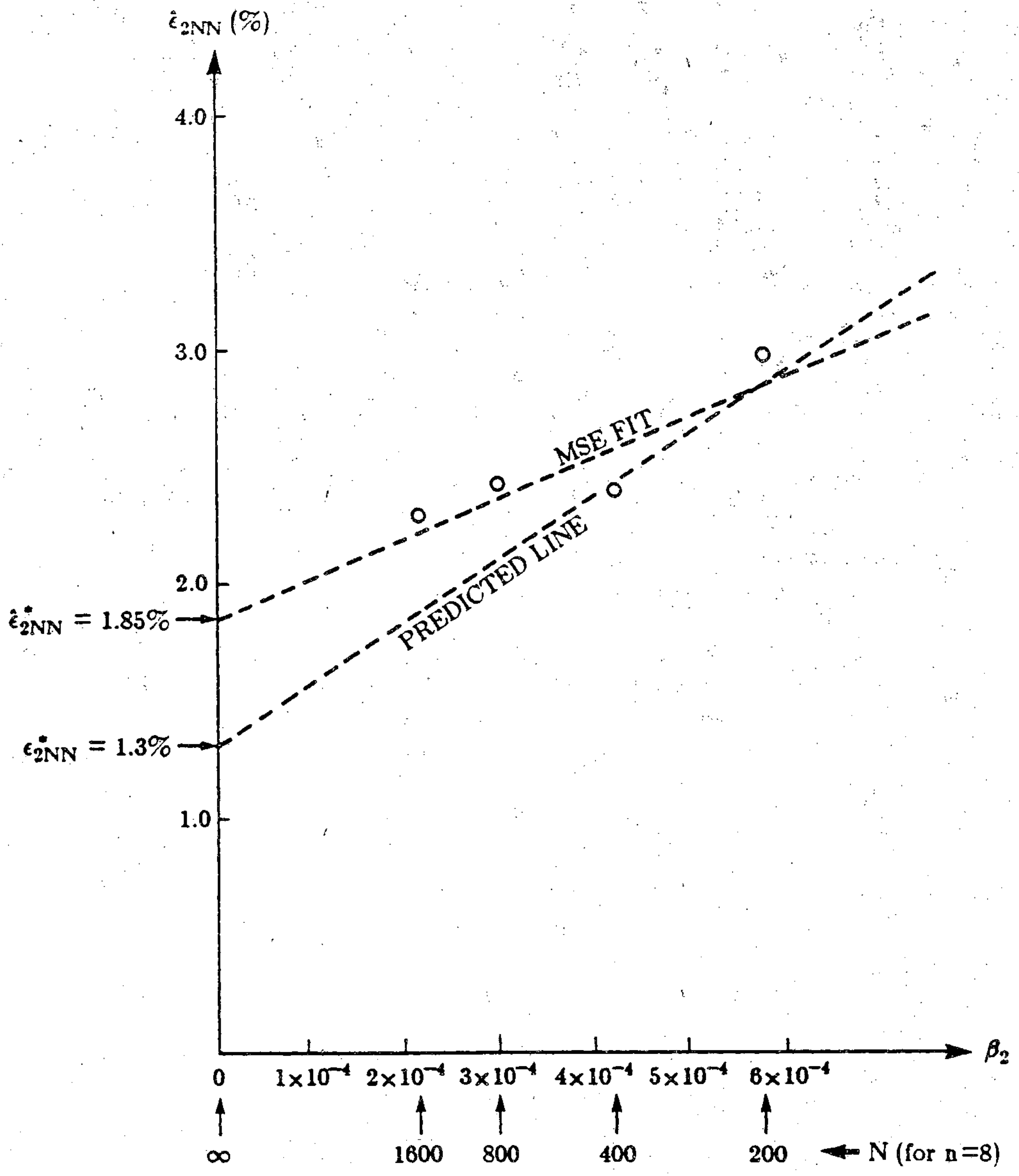


Figure 2.6 2-NN Errors for Data 2

$$r_1(X,X_{NN}) = \sum_{i=1}^{M} q_i(X)\,(1 - q_i(X_{NN})) = 1 - \sum_{i=1}^{M} q_i(X) q_i(X_{NN}) \tag{33}$$

The asymptotic NN risk is found by letting $X_{NN} \rightarrow X$

$$r_1^*(X) = 1 - \sum_{i=1}^{M} q_i^2(X) \tag{34}$$

The bias is then

$$b(X,X_{NN}) = r_1(X,X_{NN}) - r_1^*(X) = \sum_{i=1}^{M} q_i(X)\,(q_i(X) - q_i(X_{NN})) \tag{35}$$

Expanding the error term into a Taylor series, and taking the expectation in exactly the same procedure as used in Section 2.2 gives the final form of the bias

$$E\{b(X,X_{NN})\} = \beta_1 \underset{X}{E}\{\,|A|^{1/n}\,\mathrm{tr}\,[A^{-1}B_M(X)]\} \tag{36}$$

where

$$B_M(X) = \sum_{i=1}^{M} p^{-2/n}(X) q_i(X)\,(-\nabla p(X)\nabla^T q_i(X)/p(X) + 1/2\nabla^2 q_i(X)) \tag{37}$$

and β_1 is as given in (18). Once again we have isolated the effects of sample size from that of the distributions. Therefore, all discussion in this chapter concerning the 2-class NN classifier may be applied directly to the multiclass classifier. The only exception is that the value of the expectation term, c_M, is different from c_1 of (22).

2.8 Summary

An analysis has been given of the biases of the finite sample NN and 2-NN errors away from their asymptotic values. The effect of sample size on the bias was completely isolated from the effects of the distribution, giving insight into many questions concerning k-NN statistics. It was shown that in many cases, increasing the sample size is not an effective means of estimating asymptotic NN errors.

A possible procedure for compensating for the bias has been presented. Under this procedure, the NN errors are first measured for a variety of sample sizes. Since we now have explicit expressions showing how the bias changes with sample size, these empirically measured values may be used to form an estimate of the asymptotic NN error. Experimental results show that the procedure yields significantly improved estimates of the asymptotic

NN errors, and hence more reliable bounds on the Bayes error.

Finally, the analysis was extended to the multiclass problem. It was shown that the bias associated with the multi-class NN classifier behaves in virtually the same manner as the 2-class classifier.

CHAPTER 3
BAYES ERROR ESTIMATION USING PARZEN AND k-NN DENSITY ESTIMATES

3.1 Introduction

Nonparametric density estimation, and the closely related area of nonparametric error estimation, play fundamental roles in statistical pattern recognition. The k-nearest neighbor (k-NN) estimate of the density function was introduced by Fix and Hodges [19,20]. Cover and Hart [1] strengthened the case for using k-NN procedures in error estimation by showing that the error for the 1-NN classifier is bounded by twice the Bayes error. The Parzen kernel type estimation of the density function was introduced by Rosenblatt [21] and later studied by Parzen [22]. Cacoullos [23] extended the estimation to the multidimensional case.

There are many theoretical results dealing with the asymptotic properties of the above estimators. Although the properties of the k-NN and Parzen estimates are well understood as the number of preclassified design samples becomes large, very little is known about the use of the estimates when the design set is limited. Our investigations in Chapter 2 have shown that the k-NN estimates may be severely biased even for large sample sizes if the dimensionality of the data is large. This unreliability of the estimators in finite sample conditions is the major obstacle toward implementation of these techniques in practice.

The goals of this chapter are twofold. First, we wish to develop reliable procedures for estimating the Bayes error using Parzen and k-NN techniques under limited design set conditions. The procedures developed here will be shown to be much less sensitive to the sample size and underlying distributions than are the Parzen and k-NN procedures currently in use. Second, by developing the k-NN and Parzen estimates in a parallel manner, we hope to expose and clarify the relationship between the two.

Because of the extreme difficulty of analysis of the k-NN and Parzen estimates under finite sample conditions, this chapter tends to be experimental, rather than theoretical, in nature. Most of the arguments presented here are intuitive, and lack theoretical rigor. However, the experimental results do back up these arguments, and the resulting procedures are seen to be significant improvements over the commonly used k-NN and Parzen Bayes error estimates.

Section 3.2 of this chapter introduces the k-NN and Parzen density estimates, and their application to Bayes error estimation. The next three sections deal with the selection of the design parameters used in the estimators, including the Parzen kernel size and shape, the value of k used for k-NN classification, and the value of the decision threshold. The proper selection of the decision threshold is shown to be an extremely important procedure for both the Parzen and k-NN error estimates, and it is this discovery which is the major contribution of this chapter. Section 3.6 gives an experimental result showing the behavior of the error estimates as the sample size is varied, and Section 3.7 presents an experiment in which the design set is not Gaussian, and cannot effectively be classified using a quadratic classifier. Finally, Section 3.8 gives a comparison of the procedures developed in this chapter with those currently in use.

3.2 Error Estimation Procedures

In this section, we introduce the Bayes error estimates to be used for both the k-NN and Parzen error estimates. In both cases, the error estimates are obtained by first forming the corresponding k-NN or Parzen estimates of the densities, designing the Bayes classifier based on these estimates, classifying the available samples, and counting the number of classification errors. Resubstitution and leave-one-out estimates are used to form upper and lower bounds on the Bayes error. In the following, for class ω_i, we assume we are given N_i samples $X_1^{(i)}, \ldots, X_{N_i}^{(i)}$ taking values in R^n which are independent and have a common density $p_i(X)$.

3.2.1 Parzen Density Estimates

The Parzen, or kernel, estimate of the density at X is given by [21-23]

$$\hat{p}_i(X) = (1/N_i) \sum_{j=1}^{N_i} (1/h^n) k_i((X-X_j^{(i)})/h) \tag{1}$$

where $k_i(\cdot)$ is a nonnegative Borel measurable kernel function satisfying $\int k_i(X)dX = 1$, and h is a linear scale factor determining the "spread" of the kernel function. Convergence results for this estimate may be found in [24-

27]. A extensive discussion of the statistical properties of $\hat{p}_i(X)$ may be found in [28].

In error estimation, we commonly wish to estimate the ω_i density at one of the sample points $X_k^{(i)}$. In this case, use of (1) with $X_k^{(i)}$ included in the design set will produce biased results. In order to reduce this bias, leave-one-out estimates of the density may be employed. The resulting estimate is

$$\hat{p}_i(X_k^{(i)}) = \frac{1}{N_i-1}\left[\sum_{j=1}^{N_i}(1/h^n)k_i((X_k^{(i)}-X_j^{(i)})/h) - (1/h^n)k_i(0)\right] . \tag{2}$$

Note that the leave-one-out density estimate may be calculated by a simple modification of (1), so that in practice no additional computational burden is assumed by calculating both estimates.

The form of the kernel function and the appropriate value of h are key questions when the sample size N_i is limited. In this chapter, we will always choose the functional form of the kernel so that the covariance of the kernel density $k_i(X)$ is equal to the covariance of the design samples, Σ_i. Hence the covariance of the scaled kernel, $(1/h^n)k(X/h)$ is given by $h^2\Sigma_i$. In selecting the best value for h, several factors must be considered. Taking the expectation of (1) with respect to X gives

$$\underset{X}{E}\{\hat{p}_i(X)\} = p_i(X)*(1/h^n)k(X/h) \tag{3}$$

where * represents convolution in $\mathbf{R}^n$. From this we see that for any nonzero value of h, the estimated density is a smoothed version of the true density, and that as h increases, the smoothing becomes more severe. For small values of h, the scaled kernel $(1/h^n)k(X/h)$ approaches an impulse function and the expected value of the estimate given in (3) approaches the true density. However, while the bias of the estimate decreases for smaller values of h, the variance of the estimate becomes very large. In practice, when a finite number of design samples is available, the value of h must be carefully chosen. This problem is further explored in Section III.

3.2.2 k-NN Density Estimates

The k-NN density estimate is given by [19,20]

$$\hat{p}_i(X) = \frac{k-1}{N_i v_k^{(i)}(X)} \tag{4}$$

where $v_k^{(i)}(X)$ is the volume of $\{Y\in\mathbf{R}^n;\ d(X,Y)\leq d(X,X_{kNN}^{(i)})\}$, d(X,Y) is any metric measuring the distance from X to Y and $X_{kNN}^{(i)}$ is the k^th NN to X in the design set from class ω_i. Loftsgaarden [29] showed that the k-NN density estimate is asymptotically unbiased and consistent. These results were strengthened by Moore and Yackel [30], and by Devroye and Wagner [31]. As in the case for the Parzen estimates, in many cases we will be evaluating the k-NN density estimate at one of the sample points $X_j^{(i)}$. In this case if we leave $X_j^{(i)}$ in the design set (and hence count it as its own first NN) the density estimate given in (4) will be biased high. Effectively, we have set $v_k^{(i)}$ by counting only k-1 neighbors, other than $X_j^{(i)}$. We will call this biased estimate the resubstitution density estimate. This bias may be removed by using a "leave-one-out" procedure in which $X_j^{(i)}$ is not counted as its own neighbor.

In theory, any metric d(X,Y) may be used in (4). In this chapter, we are interested in forming a comparison between the Parzen and k-NN error estimates. For the Parzen approach, we have restricted ourselves to kernel densities where covariance is equal to the covariance for class ω_i, Σ_i. In analogy with the Parzen estimate, and in order to form a valid comparison of the two procedures, we will restrict the distance metric to the quadratic metric based on the covariance Σ_i:

$$d_i(X,Y^{(i)}) = [(X-Y^{(i)})^T \Sigma_i^{-1} (X-Y^{(i)})]^{1/2} \tag{5}$$

Note that we have chosen to use a different metric to form the density estimates for each class.

3.2.3 The Error Estimates

An estimate of the Bayes error may now be formed by using the estimated densities (k-NN or Parzen) to calculate a log likelihood ratio, classifying each of the design samples using this likelihood ratio estimate, counting the number of classification errors, and dividing by the total number of samples classified. Our decision rule, then, is given by

$$\ell(X) = -\ln \frac{\hat{p}_1(X)}{\hat{p}_2(X)} \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} t \tag{6}$$

where $\hat{p}_i(X)$ is given in (1) for the Parzen estimate and (4) for the k-NN error estimate, and t is the decision threshold. Many authors have considered the asymptotic properties of the error performance of classifiers based on Parzen type density estimates (see, for example [32-36]). With restrictions on the kernel $k(\cdot)$, if $h \rightarrow 0$ and $Nh^n/\log N \rightarrow \infty$ as $N \rightarrow \infty$, then the classification error (for $t=0$) approaches the Bayes error for any distribution for ω_1 and ω_2 [6,8,37]. For k-NN, if $k/N \rightarrow 0$ and $k/\log(\log N) \rightarrow \infty$ as $N \rightarrow \infty$, then for any distributions the error performance also converges to the Bayes error [5,38]. Further, for fixed k, the error performance forms upper and lower bounds on the Bayes error regardless of the distributions [7]. Results concerning rates of convergence may be found in [10,11].

In practice both the resubstitution and leave-one-out density estimates are used in (6) to form lower and upper bounds on the Bayes error. The parameters to be chosen by the designer for Parzen error estimation are the kernel size (h), decision threshold (t) and kernel shape $k_i(\cdot)$. Correspondingly, k-NN error estimation the designer is free to select the number of neighbors (k), the decision threshold (t), and the metric used, $d_i(X,Y)$.

A few remarks are in order concerning the use of (6) to perform k-NN error estimation. Under the more conventional k-NN error estimation procedure, the k-NN's from the mixture density are located, and the classification decision is based on the majority class among these k-neighbors [1]. When k is even, and equal numbers of neighbors are found from both classes, the sample is rejected, and no error is counted. Using this procedure gives (asymptotically) upper bounds on the Bayes error for odd values of k and lower bounds on the Bayes error for even values of k. These bounds improve as k becomes large, provided the sample sizes is large enough. Under the procedure used in this chapter, for each sample we find the volume to the k^{th} NN from each class, and use this value to estimate the density at the sample point. Equation (6) is then used to classify the sample. While this may seem a very different procedure, a little thought verifies that, if the metrics used are kept the same for both classes, and if the decision threshold is set to 0, then the leave-one-out k-NN error estimate considered here is identical to the more conventional (2k-1)-NN sample counting rule. Similarly, the resubstitution k-NN error estimate used here is the same as the conventional (2k-2)-NN sample counting procedure. Thus, under these conditions the k-NN resubstitution and leave-one-out procedure used here do in fact asymptotically give lower and upper bounds on the Bayes error. By using the decision rule as it appears in (6), we hope to make the relationships between the k-NN and Parzen estimates more clear. Also, we gain the ability to use different metrics for each class of data. Finally, the representation given in (6) allows us to adjust the decision threshold.

Our experimental results will show that the decision threshold plays a crucial role in both k-NN and Parzen type error estimates, and should not be neglected.

The following sections will present arguments on how to select each of the parameters of interest. Since analysis of the Parzen and k-NN classifiers in the finite design sample case is extremely difficult, many of the arguments presented here are rather heuristic in nature. However, the experimental results presented will verify that the procedures proposed here will result in a significant improvement of the performance of the k-NN and Parzen classifiers as they are currently employed. The experiments run here are based on three test cases, whose statistics are summarized in Table 3.1. Gaussian data was used in all cases, and throughout this chapter, all results are averaged over 10 independent trials. Case 1 is a simple, equicovariance case with means separated to give a Bayes error of 10%. For case 2, the means were set equal and the covariances are proportional to each other. This case was specifically chosen so that the covariance determinants would be very different. Case 3 is a complex case in which Σ_1 and Σ_2 are very different. The parameters for case 3 were chosen from Fukunaga's "standard data" after simultaneous diagonalization [18]. The class 2 covariance Λ is a diagonal matrix with diagonal terms as given in Table 3.2. Table 3.2 also shows the corresponding components of M_2—M_1. The three test cases were chosen as to present a wide range of practical problems. For example, while the Bayes classifier for case 1 is a simple linear classifier, the Bayes classifier for case 2 contains only 2nd order terms, and the Bayes classifier for case 3 is a complex quadratic decision rule. Also, while the Bayes error rates for cases 1 and 2 are moderate (10% and 9% respectively), the Bayes error of case 3 is fairly small (1.9%).

3.3 Selection of Kernel Size/Optimal Value of k

As mentioned earlier, the choice of the Parzen scale parameter h is largely a tradeoff between reducing the bias of the estimate, and increasing the variance. Many authors have considered the problem of determining the optimal value of h (or correspondingly, the optimal value of k in k-NN classification). Fukunaga and Hostetler found expressions for the optimal k in k-NN density estimation and used this expression to determine the optimal h for Parzen estimation assuming a uniform kernel function [17]. Increasingly popular are estimates based on cross-validation procedures which maximize some criterion - usually based on the maximum likelihood

Table 3.1 Statistics for three experiment test cases.

Case	$\|\|M_1-M_2\|\|$	Σ_1	Σ_2	n	$\|\Sigma_1\|$	$\|\Sigma_2\|$	Bayes Error
1	2.563	I	I	8	1	1	10%
2	0	I	4I	8	1	65,536	9%
3	5.463	I	Λ	8	1	6.747	1.9%

Table 3.2 Statistics for Case 3.

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
$M_2(i)-M_1(i)$:	3.86	3.10	0.84	0.84	1.64	1.08	0.26	0.01

principle. Such estimates were first suggested by Duin [39] and Habbema et. al. [40] and later investigated by a number of authors [41-43]. Another approach involves relating the chosen value for h to the distances to the kth nearest neighbor [44,45]. Most of these results find the optimal h based on some density estimation criterion. In many cases, such estimation criteria place the emphasis on obtaining accurate estimates in regions in which the densities are large. In error estimation, however, many times we are primarily concerned with obtaining accurate estimates in the tails of the densities, particularly when the Bayes error is small. Hence, while the optimal kernel size/optimal k results do give us a general idea of the magnitude of h or k, blind use of these values may result in disastrous estimates of the Bayes error.

In our experiments, use of the theoretically determined values of h or k gave very discouraging and inconsistent results. After many unsuccessful attempts to determine the optimal value of h theoretically, we found that the only reliable means of selecting the best h for a particular data set is experimentally. That is, by using several values of h or k on a given data set and plotting the results to obtain the best bounds on the Bayes error. In the case of Parzen error estimation, this requires that the estimation procedure be completely repeated for each value of h. In the k-NN case, however, the computational load is only slightly increased since the error may be computed for all values of k simultaneously.

Figure 3.1 shows a graph of the Parzen errors for the data from cases 1,2 and 3. Gaussian kernel function were used for the Parzen classifier. In this experiment 100 samples per class were used in each trial. The decision threshold t was set to zero.

The behavior of the Parzen classifier may be explained as follows. For small values of h, the Parzen classifier gives basically a 1-NN type of performance, since as h decreases, only the nearest neighbor to X_k has a significant contribution to the sum given in (2). As h increases, the variance of the density estimate decreases and the density estimate approaches its expected value given in (3) As h becomes very large, the kernel function begins to dominate the convolution in (3), and the density estimate approaches the scaled kernel function, shifted to the mean of the true density, $1/h^n k_i((X-M_i)/h)$. Thus the decision rule as h becomes large becomes almost completely determined by the functional form of the kernel function, and not by the true density. When the kernel function is

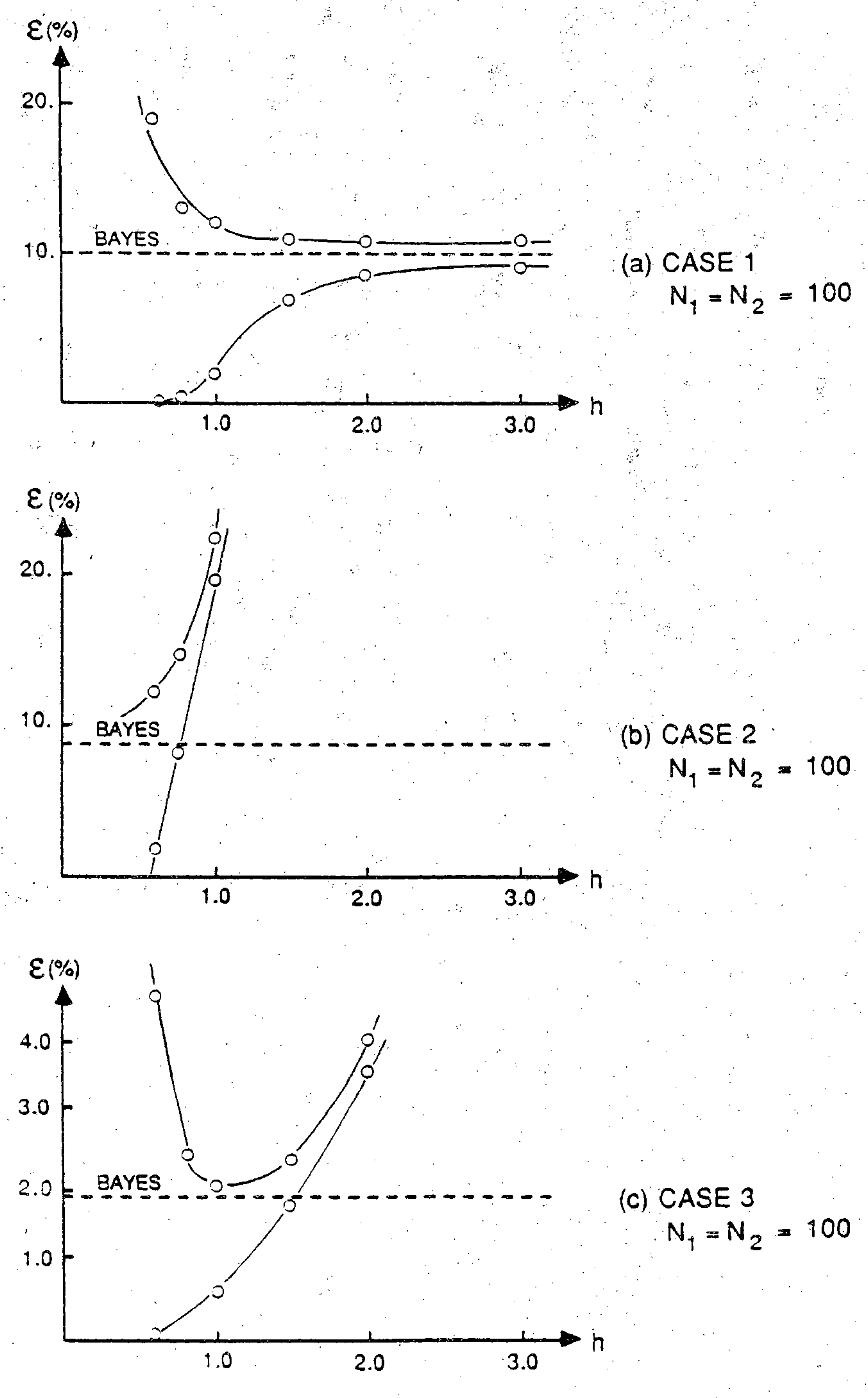


Figure 3.1 Parzen error for various values of h. Upper and lower curves represent leave-one-out and resubstitution results respectively.

Gaussian, this results in a quadratic classifier. For the data in Figure 3.1(a), as h increases the Parzen classifier approaches the quadratic classifier which happens to be the Bayes classifier. Thus, in this case, the performance of the Parzen classifier does not degrade as h increases. This is not the case in general as demonstrated in Figures 3.1(b) and 3.1(c). In these cases, although the Parzen classifier does approach a quadratic classifier, it does not approach the particular Bayes classifier for cases 2 and 3. Thus as h increases the performance of the Parzen classifier degrades sharply. Similar performance may be expected when the data is not Gaussian, and the Bayes classifier is not quadratic. In general the leave-one-out error dips down to some minimum point and, as h increases, performance degrades to the performance of a quadratic classifier which is determined by the shape of the kernel functions. The best bounds on the Bayes error are obtained by using the minimum of the leave-one-out performance curve, and the corresponding resubstitution error. It is interesting to note the size of the optimal value of h in Figure 3.1. In these experiments ($n=8$, $N_1=N_2=100$) the best value of h appears to be close to $h=1$, meaning that the covariance of the kernel function is nearly equal to the covariance of the original data. While on first glance this seems to be an extremely large value for h, further examination shows that such values of h are required if the Parzen density estimate at a sample is to receive reasonable contributions from its nearest neighbors, particularly as the dimensionality of the data becomes large. A discussion of the relationship between dimensionality, sample size, and nearest neighbor distances may be found in [16]. For Gaussian data with covariance I, the expected distance to the kth nearest neighbor is given by [16]

$$E\{d(X,X_{kNN}^{(i)})\} = \frac{\sqrt{2}\,\Gamma(k+1/n)\Gamma(N_i+1)\Gamma^{1/n}(1+n/2)}{\Gamma(k)\Gamma(N_i+1+1/n)(1-1/n)^{n/2}} . \tag{7}$$

For $n=8$ and $N_i=100$, this gives an expected 1-NN distance of 1.9. These large nearest neighbor distances necessitate the use of large values of h. The need for such large values of h clearly demonstrates that biases associated with Parzen density estimates must be considered when these estimates are used in practice. A possible means of compensating for these biases is presented in Section 3.4.

The behavior of the k-NN error as k increases is not so easily explained. However, the k-NN rule may be viewed as a Parzen density estimate with uniform kernel and adaptable kernel size. Varying the kernel size h for the

Parzen classifier is analogous to changing the value of k for the k-NN classifier. It is known that for infinite sample sizes, larger values of k give tighter bounds on the Bayes error. However, with a finite number of design samples, one expects that the k-NN leave-one-out error will pass through some minimum as k increases. It is this minimum which gives us the best bound on the Bayes error.

3.4 The Decision Threshold t

If $\hat{p}_1(X)$ and $\hat{p}_2(X)$ were accurate estimates of $p_1(X)$ and $p_2(X)$, then using $t=\ln(P(\omega_1)/P(\omega_2))$, where $P(\omega_i)$ represents the priori probability of ω_i, would give the optimal Bayes classifier. In practice though, our estimates may be heavily biased, particularly when h or k is large. In order to evaluate the error performance, we must design our classifier based on $\hat{p}_1(X)$ and $\hat{p}_2(X)$, and test the samples drawn from the true densities $p_1(X)$ and $p_2(X)$. In this case, the optimal value of t to use in the decision rule (6) is not clear. We would like to select the threshold which gives performance as close to the Bayes error as possible.

3.4.1 The Gaussian Case

In the Gaussian case, for the Parzen classifier, the threshold may be solved for explicitly for large h. Recall from (3) the expected value of the estimate $\hat{p}_i(X)$ is given by $(1/h^n)k_i(X/h)*p_i(X)$. When $p_i(X)$ and $(1/h^n)k_i(X/h)$ are normal densities with covariances Σ_i, and $h^2\Sigma_i$ respectively, this convolution gives another normal density with mean M_i and covariance $(1+h^2)\Sigma_i$. For larger values of h, the variance of $\hat{p}_1(X)$ and $\hat{p}_2(X)$ decreases, and the estimates approach their expected values. Substituting the expected values into the estimated log likelihood ratio, one obtains

$$-\ln\frac{\hat{p}_1(X)}{\hat{p}_2(X)} \approx \frac{1}{2}(X-M_1)^T\frac{\Sigma_1^{-1}}{1+h^2}(X-M_1) - \frac{1}{2}(X-M_2)\frac{\Sigma_2^{-1}}{1+h^2}(X-M_2) + \frac{1}{2}\ln\frac{|\Sigma_1|}{|\Sigma_2|} .$$

Except for the $1/(1+h^2)$ factors on the covariance matrices, this expression is identical to the actual log likelihood ratio, $-\ln\dfrac{p_1(X)}{p_2(X)}$. In fact, the two may be related by

$$-\ln\frac{\hat{p}_1(X)}{\hat{p}_2(X)} \simeq \frac{1}{1+h^2}(-\ln\frac{p_1(X)}{p_2(X)}) + \frac{1}{2}(\frac{h^2}{1+h^2})\ln\frac{|\Sigma_1|}{|\Sigma_2|} \tag{8}$$

The actual Bayes decision rule is given by $-\ln[p_1(X)/p_2(X)] \gtrless \ln[P(\omega_1)/P(\omega_2)]$. Using (8), an equivalent test may be expressed in terms of the estimated densities:

$$-\ln\frac{\hat{p}_1(X)}{\hat{p}_2(X)} \gtrless t$$

where

$$t = \frac{1}{1+h^2}(\ln\frac{P(\omega_1)}{P(\omega_2)}) + \frac{1}{2}(\frac{h^2}{1+h^2})\ln\frac{|\Sigma_1|}{|\Sigma_2|} \tag{9}$$

In all of our experiments, we assume $P(\omega_1) = P(\omega_2) = 0.5$, so that the first term of (9) may be neglected. (9) gives the appropriate threshold to use when using the Parzen classifier with Gaussian kernel function on Gaussian data. When the class covariances are equal, as in the data for case 1, the appropriate threshold is $t=0$ for all h. Thus in Figure 3.1(a), setting $t=0$ happened to be the correct choice, and relatively good performance was obtained even for large values of h. When the covariance determinants are different, as for the data in cases 2 and 3, the value of the decision threshold becomes a function of h. In this case, if the value of t is simply held constant at zero, as is commonly done, the performance of the Parzen classifier degrades sharply for larger values of h, as evidenced in Figures 3.1(b) and 3.1(c). Figure 3.2 shows the behavior of the Parzen classifier for cases 1, 2, and 3 with t given by (9). For low values of h, the classifiers give similar performance to that shown in Figure 3.1, since the appropriate value of t given in (9) is close to zero. As h increases, if t is held fixed the performance begins to degrade sharply for cases 2 and 3, while if the value of t is set according to (9) relatively good performance is obtained for all values of h. Thus, by allowing the decision threshold to vary with h, we are able to make the Parzen classifier much less sensitive to the value of h. As a byproduct, we are able to use larger values of h to obtain tighter bounds on the Bayes error.

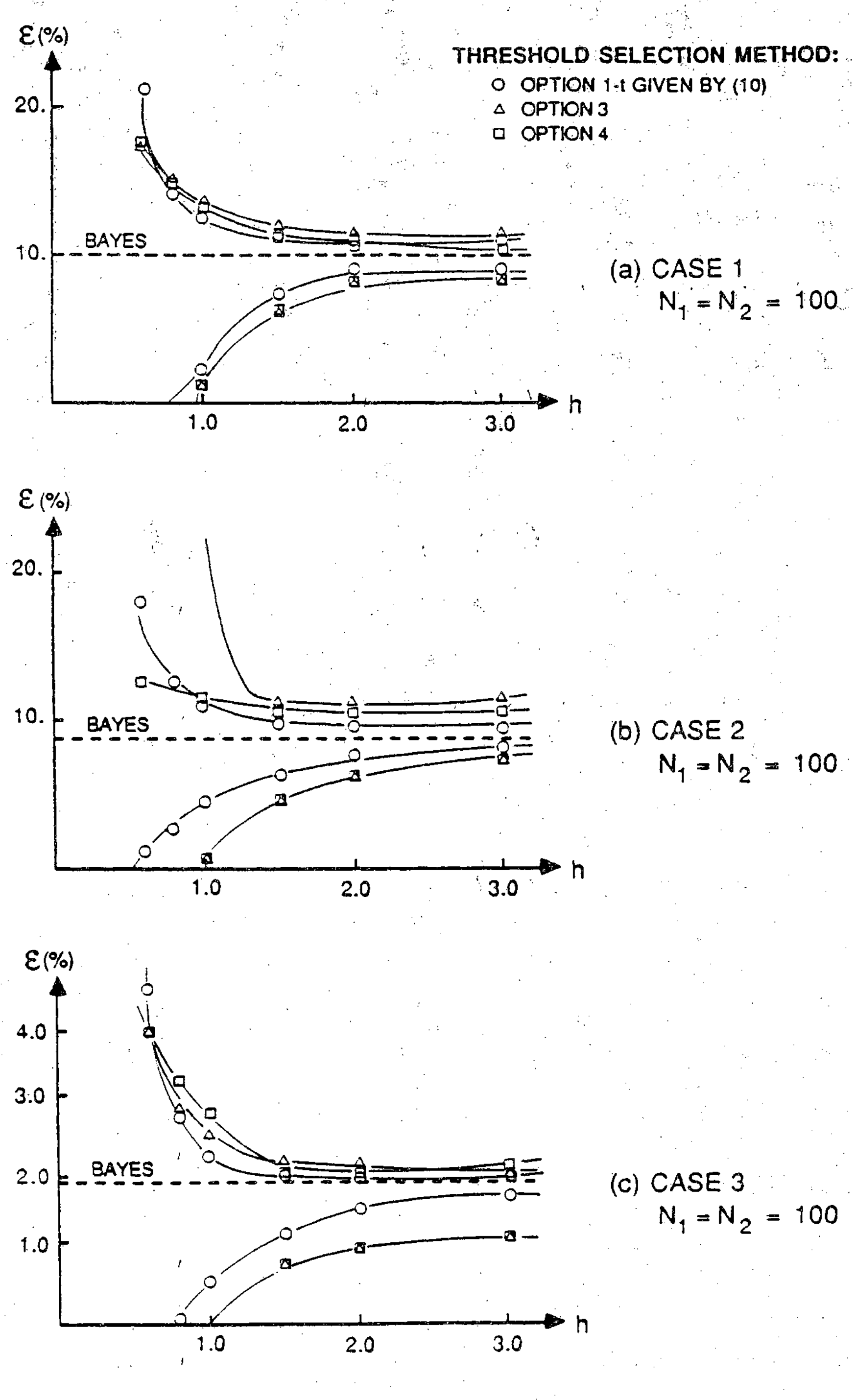


Figure 3.2 Threshold selection for Parzen classifiers.

3.4.2 The Non-Gaussian Case

The decision threshold as used here is simply a means of compensating for the bias which is inherent to the density estimation procedure. When the data and the kernel functions are Gaussian, we have shown that the bias may be completely compensated for by choosing the value of t given in (9). In the non-Gaussian case, we cannot hope to obtain a decision rule equivalent to the Bayes classifier simply by varying t. However, by choosing an appropriate value of t we can hope to compensate to some extent the bias of the density estimates in a region close to the Bayes decision region, giving significant improvement in the performance of the Parzen classifier. Procedures are therefore needed for determining the best value of t to use when non-Gaussian data is encountered. We have investigated four possible options. These options, and a brief discussion of their motivation, as given below.

Option 1: Use the threshold as calculated under the Gaussian assumption (9). Since for larger values of h the decision rule is dominated by the functional form of the kernels, this procedure may give satisfactory results when the kernels are Gaussian, even if the data is not Gaussian.

Option 2: For each value of h, find the value of t which minimizes the leave-one-out and resubstitution errors. This option involves finding and sorting the leave-one-out and resubstitution estimates of the likelihood ratio, and incrementing the values of t through these sorted lists. The error rate used as the estimate is the minimum error rate obtained over all values of t. This option makes no assumption of the densities of the data or the shape of the kernel function. However, since the value of the threshold is customized to the data being tested, using this option will consistently bias the results low. This is not objectionable in the case of resubstitution errors, since the resubstitution error is used as a lower bound of the Bayes error. However, using this procedure can give erroneous results for the leave-one-out error. Options 3 and 4 are designed to alleviate this problem.

Option 3: For each value of h, find the value of t which minimizes the resubstitution error, and then use this value of t to find the leave-one-out error. Since the selection of the threshold has been isolated from the actual values of the leave-one-out estimates of the likelihood ratio, using this method does in fact help reduce the bias encountered in option 2. Our experimental results will show that this method does give reliable results as long as h is relatively large. When h is small, however, the resubstitution estimates of the likelihood ratio are heavily biased, and use of these estimates to determine the threshold may give far from optimal results. An advantage of this option is that it requires no more computation time than option 2.

Option 4: Under this option, the resubstitution error is found exactly as in option 2, by finding the value of t which minimizes the resubstitution error, and using this error rate. In order to find the leave-one-out error, we use a leave-one-out procedure to determine the value of t to use for each sample. Hence, under option 4, we use a different threshold to test each of the N_1+N_2 samples, determining the threshold for each sample from the other N_1+N_2-1 samples in the design set. The exact procedure is as follows.

1) Find the leave-one-out density estimates at all samples,

$$\hat{p}_m(X_j^{(i)}),\ m=1,2,\ \ i=1,2,\ \ j=1,2,\dots N_i\ .$$

2) To test sample $X_k^{(\ell)}$:

a) Modify the density estimates by removing the effect of $X_k^{(\ell)}$ from all estimates

$$\tilde{p}_m(X_j^{(i)}) = \begin{cases} \hat{p}_m(X_j^{(i)}) & m\neq\ell \\ \dfrac{1}{N_m-2}[(N_m-1)\hat{p}_m(X_j^{(i)})-(1/h^n)k_m((X_j^{(i)}-X_k^{(\ell)})/h)] & m=\ell,\ i=\ell \\ \dfrac{1}{N_m-1}[N_m\hat{p}_m(X_j^{(i)}) - (1/h^n)k_m((X_j^{(i)} - X_k^{(\ell)})/h)] & m=\ell,\ i\neq\ell \end{cases} \quad (10)$$

b) Calculate the likelihood ratio estimates at all samples $X_j^{(i)}\neq X_k^{(\ell)}$ based on the modified density estimates.

$$\ell(X_j^{(i)}) = -\ln\frac{\tilde{p}_1(X_j^{(i)})}{\tilde{p}_2(X_j^{(i)})}\ ,\ X_j^{(i)}\neq X_k^{(\ell)} \quad (11)$$

c) Find the value of t which minimizes the error among the N_1+N_2-1 samples (don't include $X_k^{(\ell)}$), under the decision rule

$$\ell(X_j^{(i)}) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} t\ . \quad (12)$$

This is best accomplished by first sorting the likelihood ratio estimates $\ell(X_j^{(i)})$, and then incrementing the value of t through this list, keeping track of the number of errors for each value of t.

d) Classify the sample $X_k^{(\ell)}$ using the original density estimates and the value of t found in step c:

$$\ell(X_k^{(\ell)}) = -\ln \frac{\hat{p}_1(X_k^{(\ell)})}{\hat{p}_2(X_k^{(\ell)})} \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} t \tag{13}$$

Count an error if the decided class is not ω_ℓ.

3) Repeat Step 2 for each sample, counting the resulting number of classification errors.

Although this procedure is by far the most complex computationally, it is the only true leave-one-out procedure, and gave the most reliable results particularly for small values of h.

Figure 3.2 shows the results of applying options 1, 3, and 4 to the three test cases listed in Table 3.1. In each case, Gaussian kernel functions were used, and the estimates are obtained using 100 design samples per class. In all of our experiments, using the threshold calculated under the Gaussian assumption (option 1) gave the best performance. This is as expected, since both the data and the kernel function are in fact Gaussian. It is notable, however, that the use of option 4 gave performance nearly equal to that of option 1. Option 3 gave good results also, but performance degraded sharply for small h, particularly for the data in case 2, where the covariance determinants are extremely different. We believe that the most reliable procedure when the form of the densities is not known is option 4.

3.4.3 Threshold Selection for k-NN Classifiers

Until this point, only the Parzen error estimate has been considered in the discussion of threshold selection. However, most of the arguments regarding selection of the threshold may be directly applied to k-NN estimation as well. The k-NN density estimates are known to be biased when the size of the design set is limited, and by choosing an appropriate threshold, one might hope to reduce or eliminate the effect of that bias when classification is performed. We have not found usable expressions for t even in the Gaussian case. However, each of the non-Gaussian methods for threshold selection (options 2, 3 and 4) are directly applicable to the k-NN problem.

Insight into the use of the k-NN procedures proposed here may be gained by viewing the procedures in the nonparametric data display framework introduced by Fukunaga and Mantock [46]. Under this framework, a two dimensional display is created with coordinates related to the volume to the kth-NN to each class. Human intervention may then be used to determine the best classification rule. Under our procedure, we restrict our decision boundaries to those of the form given in (6), i.e. to lines with slope one in the data display. By restricting ourselves to this set of decision boundaries, we gain the ability to automate the process, and hence find the errors for a wide variety of values for k. More complicated boundaries could be used, but only at the cost of significantly more complex procedures. In this chapter, we are recognizing this shift of the decision boundary as a means of compensating for biases associated with different values of k.

One comment is in order regarding the application of option 4 to k-NN estimation. In step 2 of option 4, in the Parzen case it is fairly simple to remove the effect of X_j (the test sample) from the density estimates of all the other samples using (10). There is no analogous simple modification in the k-NN case. In order to remove the effect of X_j from all other density estimates, one must remove X_j from the table of nearest neighbors, rearrange the NN table, and recalculate all of the density estimates. This procedure would have to be repeated to test each of samples in the design set, resulting in a fairly drastic increase in computation time. In practice we have found that modifying each of the density estimates to remove the effect of X_j is not nearly as important as is finding the threshold by minimizing the error among the remaining N_1+N_2-1 samples. That is, modifying the estimates of the likelihood ratios in step 2 is not necessary to get reliable results - we do it in the Parzen case primarily because it is easy. Thus for k-NN estimation, step 2 of option 4 involves finding and sorting $\ell(X_i)$ for all samples $i \neq j$, finding the value of t which minimizes the error among these N_1+N_2-1 samples, and finally using this value of t to classify X_j.

Figure 3.3 shows the results of applying option 4 to the k-NN estimation problem. For comparison, the results obtained using $t=0$ are also shown. As in the Parzen case, we find that the threshold plays its most significant role when the covariances of the data are different, and particularly when the covariance determinants are different. In test case 1 ($\Sigma_1=\Sigma_2=I$, $\epsilon^* = 10\%$) the bias of the density estimates for ω_1 and ω_2 are

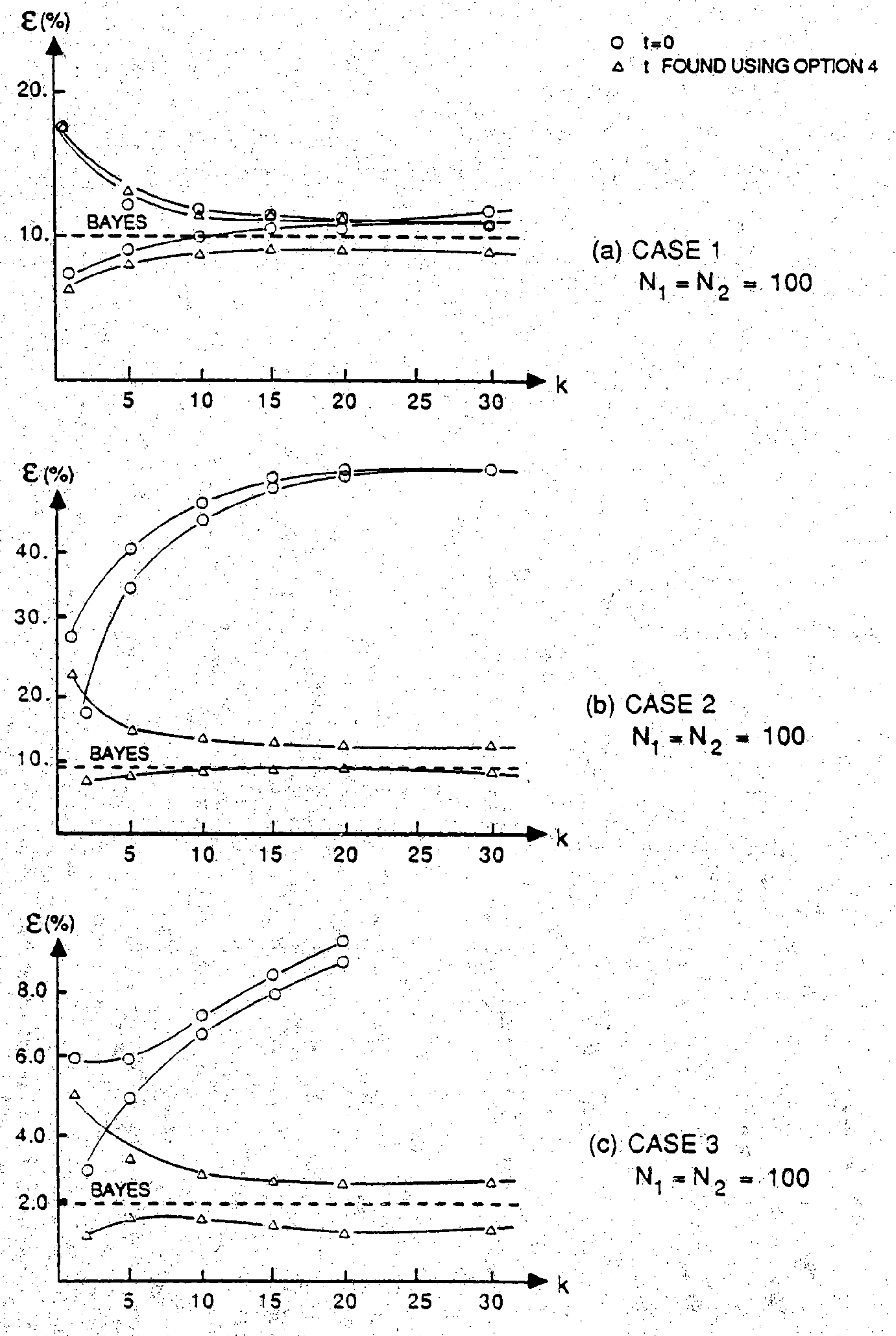


Figure 3.3 Threshold selection for k-NN classifiers.

nearly equal near the Bayes decision boundary, and hence good results are obtained using $t=0$.

3.5 Parzen Kernel Shape.

While (with the threshold adjustment) the Parzen classifier provides excellent upper bounds on the Bayes error, for reasonable values of h the lower bounds provided by the resubstitution error seem much too conservative. This is especially true when these bounds are compared with the resubstitution k-NN error rates. This tends to indicate that the kernel function places too much weight on the sample being tested in the resubstitution estimate. Hence, one possible approach to improving the lower bound from the Parzen estimate is to use a kernel function which places less weight on the test sample and more weight on the neighboring samples than does the Gaussian kernel. The uniform kernel function, with constant value inside a specified region, is one such kernel function. However, if a uniform kernel function is employed one must decide on the decision to be made when the density estimates from the two classes are equal, and the Parzen procedure becomes even more complex.

In order to examine the effect of kernel shape on the Parzen classifier performance, we used the kernel functions given by

$$\frac{1}{h^n}k_i(X/h) = \frac{c}{h^n}\exp\left[-\left(\frac{\Gamma(\frac{n+2}{2m})}{n\Gamma(\frac{n}{2m})}\right)^m ((1/h^2)X^T\Sigma_i^{-1}X)^m\right] \tag{14}$$

$$c = \frac{m\Gamma(\frac{n}{2})\Gamma^{n/2}(\frac{n+2}{2m})}{(n\pi)^{n/2}\Gamma^{n/2+1}(\frac{n}{2m})|\Sigma|^{1/2}}$$

where m is a parameter determining the shape of the kernel. It may be verified that for any value of m, the covariance of the kernel density (14) is given by $h^2\Sigma_i$, just as it has been throughout this chapter. The parameter m determines the rate at which the kernel function drops off. For $m=1$, (14) reduces to a simple Gaussian kernel. As m becomes large, (14) approaches a uniform (hyperelliptical) kernel, always with a smooth roll-off, (for finite m) and always with covariance $h^2\Sigma_i$. Using this kernel allows us to use kernel functions close to the uniform kernel, without having to worry about the problem of equal density estimates.

Figure 3.4 shows the performance of the Parzen estimates with m=1 (Gaussian kernel), 2 and 4. In each case, the estimates were based on 100 design samples per class, and the threshold was determined using option 4. In all cases, using higher values of m (more uniform kernel functions) did improve the lower bound while having little effect on the upper bounds on the error.

3.6 Effect of Sample Size and Dimension

In order to determine the effect of sample size on the error estimates, k-NN and Parzen error estimates were obtained for test case 1 using sample sizes $N_1=N_2=25$, 50, 100, 200, and 400. Gaussian kernels were used for the Parzen estimates and in all cases option 4 was used to find the best threshold. Performance of the Parzen and k-NN estimators is graphed in Figures 3.5(a) and 3.5(b) respectively. As may be seen in the figures, both procedures give reasonable bounds using sample sizes as low as 25, and good bounds for sample sizes of 50 per class. The variance of the two estimates has been graphed in Figure 3.6. Note that the Parzen and k-NN procedures give comparable variances. Similar results were obtained for test cases 2 and 3.

These results show no clear preference between the Parzen and k-NN procedure as used here. Both procedures yield usable bounds on the Bayes error, with very comparable variances. This is not surprising, since the k-NN procedure may be viewed as a Parzen classifier with kernel size determined adaptively (according to NN distances). It should be stressed, however, that we are not comparing the conventional k-NN and Parzen procedures, but rather then improved procedures as introduced in this chapter. Recall that our k-NN procedure involves (possibly) using different metrics for each class in the design set, and for both the k-NN and Parzen procedures the threshold is varied according to h or k. We believe that these modifications must be made to form a valid comparison between the two procedures, and should also be made in practice to obtain reliable bounds on the Bayes error.

The proposed Parzen and k-NN procedures were applied to real data with 60 dimensional feature vectors*. Figure 3.7 shows the results of both

*The results are provided by R. Han, G. Green and R. McCoy of Martin Marietta Aerospace, Orlando, FL.

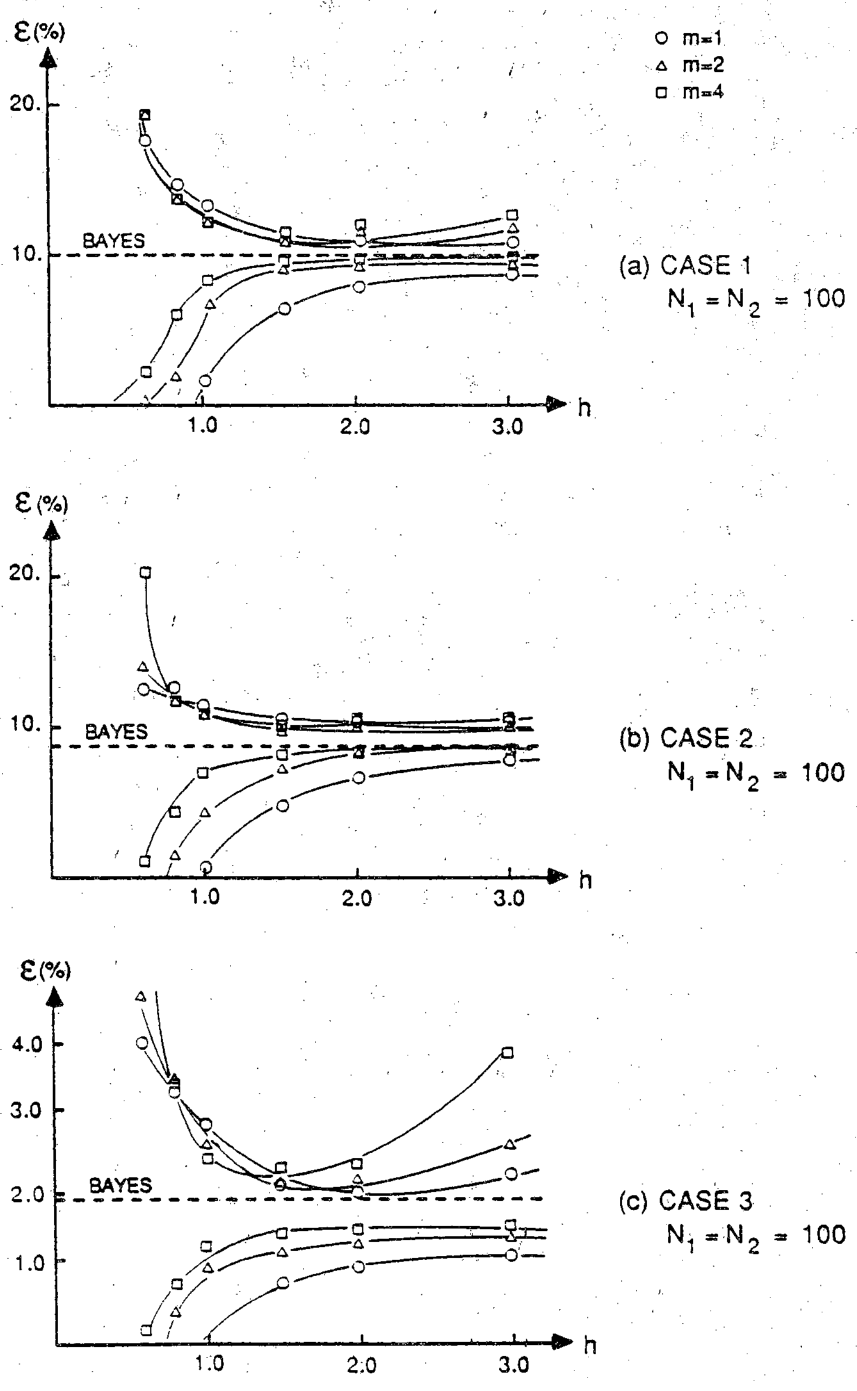
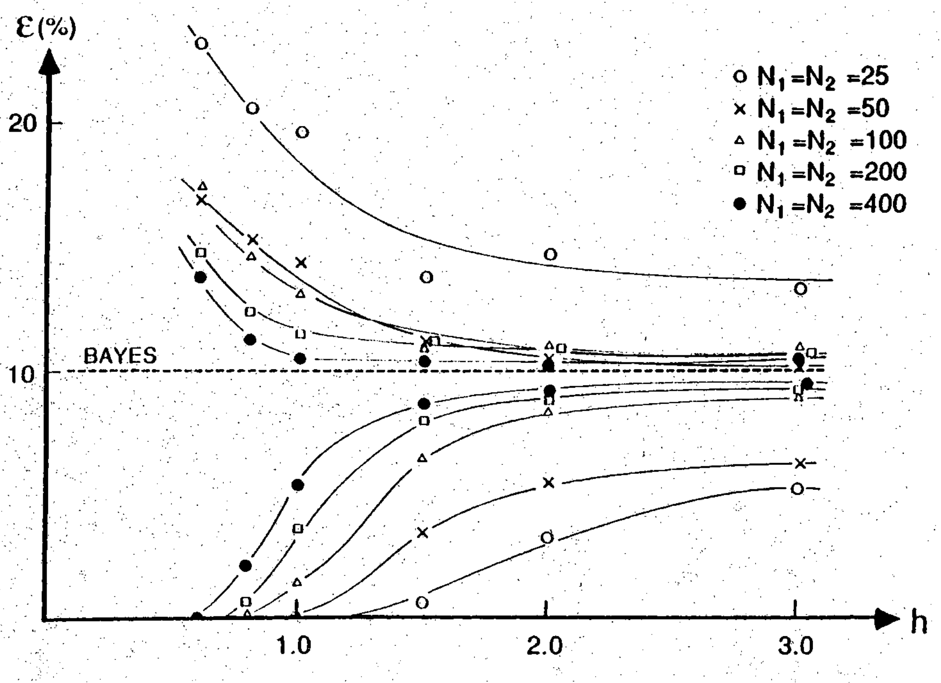
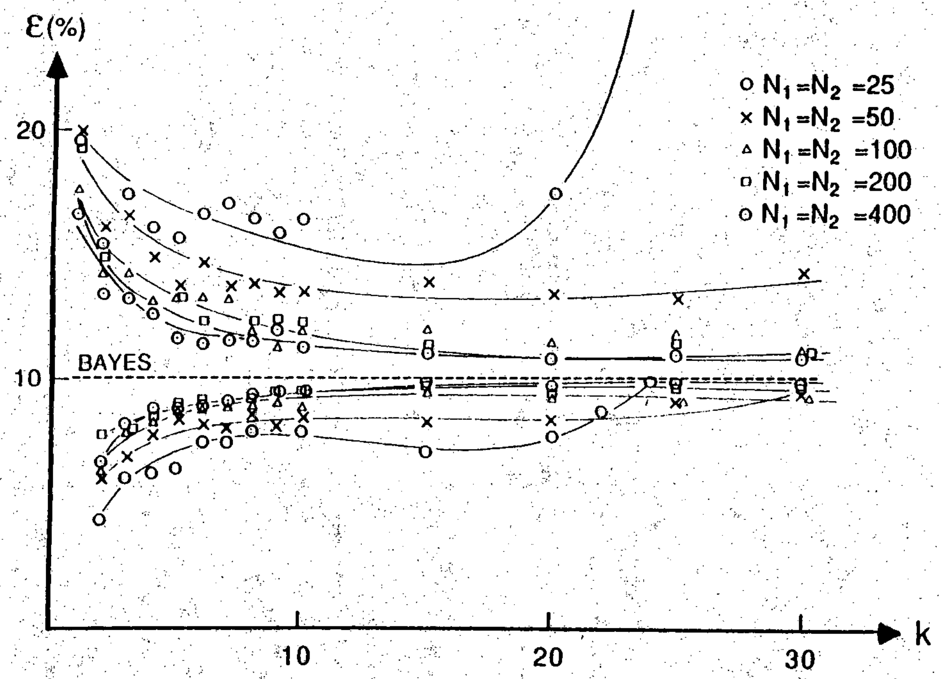


Figure 3.4 Effect of kernel shape on Parzen error.

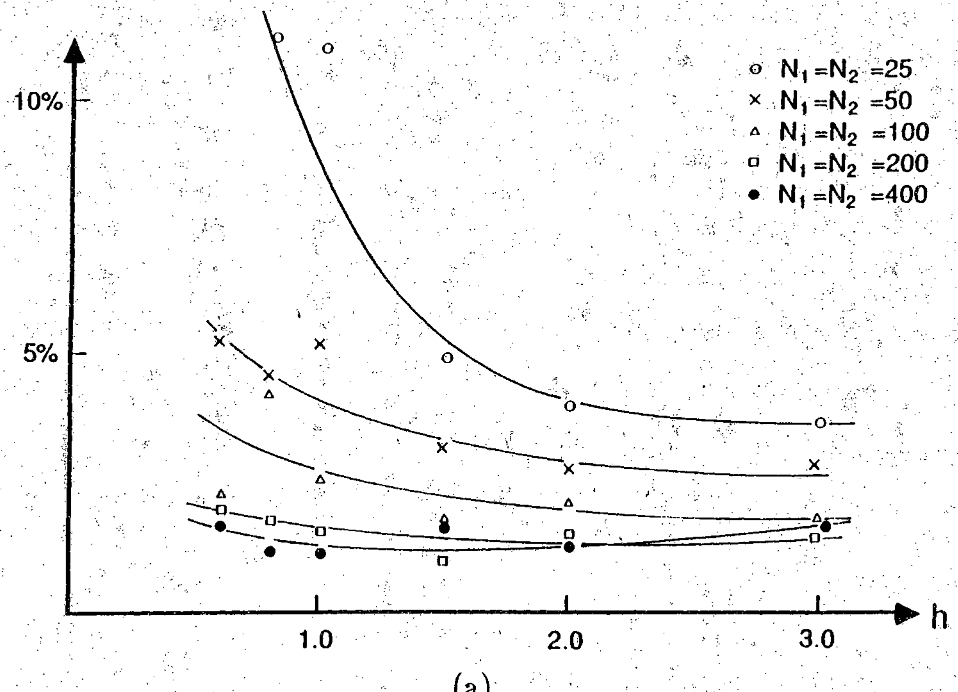


(a)



(b)

Figure 3.5 (a) Parzen error for various sample sizes. (b) k-NN error for various sample sizes.



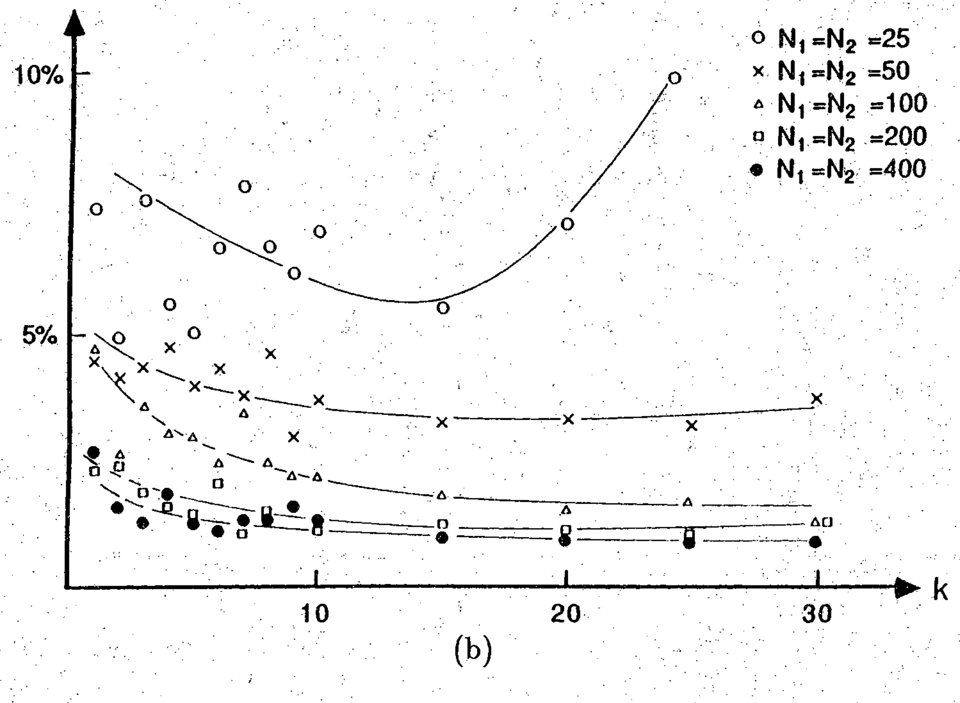


Figure 3.6 (a) Standard deviation of Parzen leave-one-out error for various sample sizes. (b) Standard deviation of k-NN leave-one-out error for various sample sizes.

Parzen and k-NN error estimates using sample sizes $N_1=N_2=115$ and 230, and using option 4 to determine the threshold. Class covariance matrices were used to determine the kernel functions for the Parzen and the metrics for the k-NN.

Although the true Bayes error is not known in this experiment, a quadratic classifier was designed from the data (using 5000 samples) and found to have an error of 8%. This value is significantly above the error rates obtained by both the Parzen and k-NN classifiers, indicating the suboptimality of the quadratic classifier (that is, the distributions of the data are not Gaussian.) Notable is the fact that both the k-NN and Parzen procedures give nearly identical error estimates. This suggests that both procedures are working well, even in this experiment where the sample sizes are extremely small for the dimensionality considered.

Note that the value of h used for the Parzen classifier is large, ranging from 5 to 9. These large values are required for the kernel functions to reach a reasonable number of neighboring samples. The expected distance to the first NN is 7.35 by (7) for a 60-dimensional Gaussian distribution with covariance matrix I and $N_1=N_2=230$ [16].

3.7 A Non-Gaussian Test Case

It is of interest to examine the behavior of the Parzen and k-NN classifiers in a case in which the optimal Bayes classifier is not a quadratic classifier. Toward this end, the classifiers we tested on data drawn from the following distributions:

$$\begin{aligned} p_1(X) &= 0.5N(M_{11},I) + 0.5N(M_{12},I) \\ p_2(X) &= 0.5N(M_{21},I) + 0.5N(M_{22},I) \end{aligned} \tag{15}$$

where $N(M,\Sigma)$ is a normal density with mean M and covariance Σ. Like our previous test cases, the dimension, n, was set to 8. The first component of the mean vectors are given by $M_{11}(1) = 0.0$, $M_{12}(1) = 6.58$, $M_{21}(1) = 3.29$, and $M_{22}(1) = 9.87$, and all other components of the mean vectors were set to zero. These parameters give a Bayes error of 7.5%. In our experiment, Gaussian kernel functions were used with kernel covariance given by h^2I. The euclidean metric was used for the k-NN classification. For both Parzen and k-NN classifiers, 100 design samples per class were used.

Figure 3.8 shows the results for both the k-NN and Parzen classifier. As expected, as h grows the error rate for the Parzen classifier drops to a minimum and then increases as the classifier converges to the quadratic

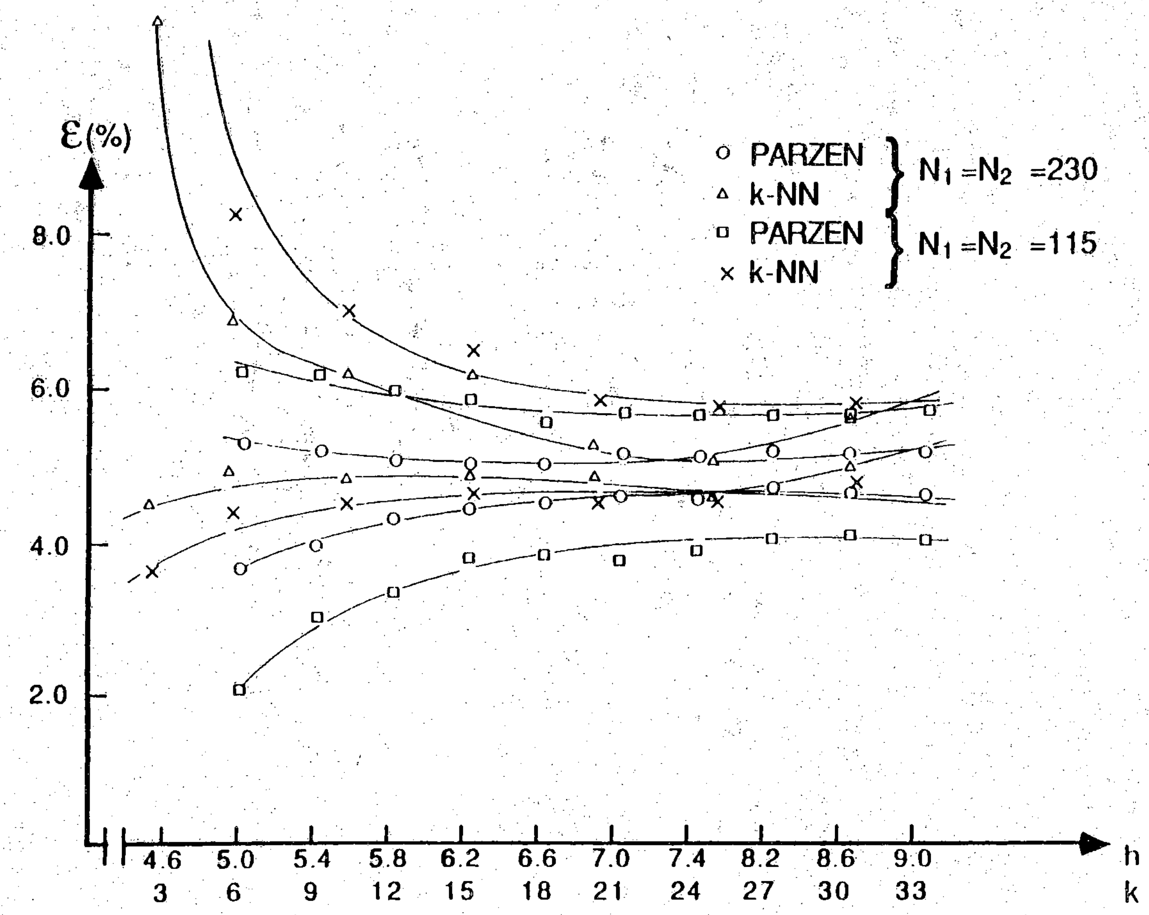
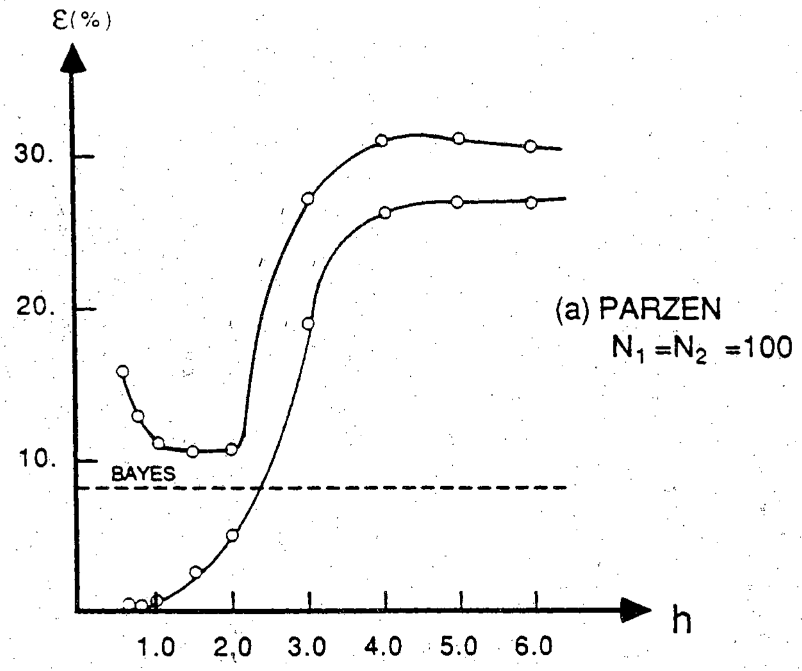


Figure 3.7 k-NN and Parzen error rates for a 60-dimensional feature space.



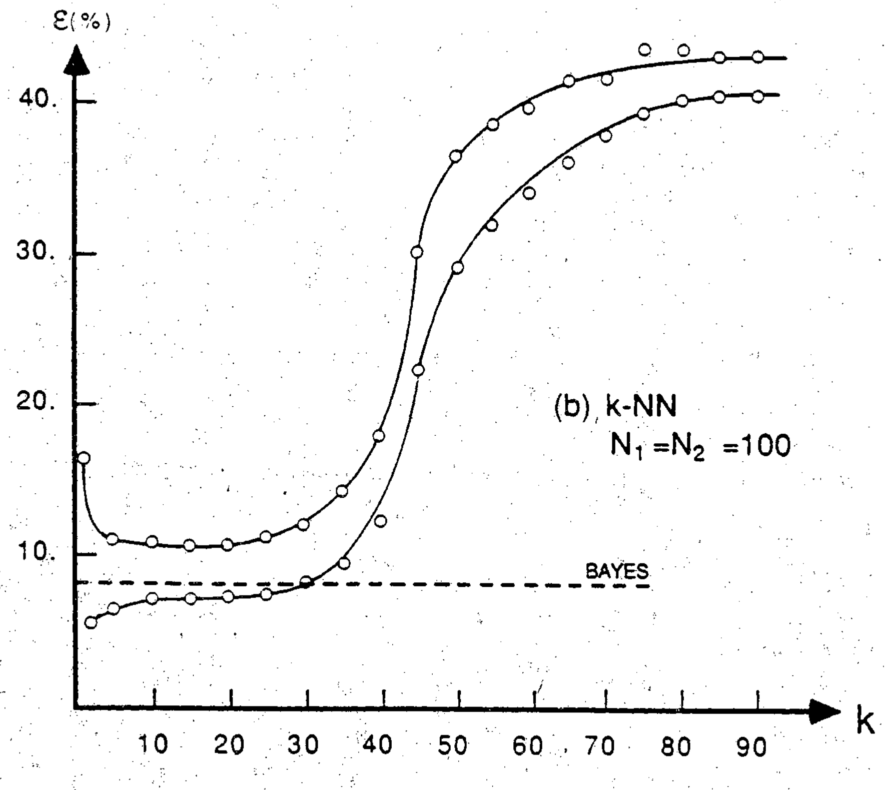


Figure 3.8 Parzen and k-NN error rates for a non-Gaussian test set.

classifier. The degradation in performance does not seem as severe for the k-NN classifier over the values of k plotted, although a slight degradation can be detected for $k > 20$. In both cases, however, both the k-NN and Parzen classification do provide usable bounds on the Bayes error.

3.8 Comparison With Existing Methods

As mentioned earlier, the Parzen and k-NN procedures developed in this chapter have several differences from the conventional procedures commonly in use. Probably the most important difference is that in the procedures used here the decision threshold is allowed to vary as h and k change. This provides a means of compensating for otherwise unaccountable biases in the Parzen and k-NN density estimates, and hence results in a dramatic improvement in the performance of the estimates under finite design set conditions. As a result of varying the threshold t, the performance of the k-NN or Parzen classifier is made much less sensitive to the value of k or h.

While the determination of the decision threshold is the primary difference in the Parzen error estimation procedures, another important difference exists in our k-NN procedure. Under the density estimation approach used in this chapter, different metrics may be used to measure distances to samples from each class allowing further improvement in the performance of the k-NN procedure. This is not possible in methods in which the sample is classified by the majority class among its k nearest neighbors from the mixture density.

Figures 3.9, 3.10 and 3.11 summarize the performance of both the new and conventional procedures for the three test cases listed in Table 3.1. In all experiments, 100 design samples per class were used, and Gaussian kernels we used for the Parzen estimates. Under the conventional procedures, the threshold t was set to zero, while under the new Parzen and k-NN procedures, option 4 was used to find the threshold for each value of h. The Euclidean metric was used in the conventional k-NN procedure.

Figure 3.9 shows that the new procedure provides virtually no improvement over the conventional procedures for test case 1. This is not particularly surprising since using $t=0$ in the conventional procedure happens to be the optimal value of t given in (9) for this test case. The real improvement in the k-NN and Parzen procedure is clearly demonstrated in Figures 3.10 and 3.11, where cases 2 and 3 are presented. In both of these

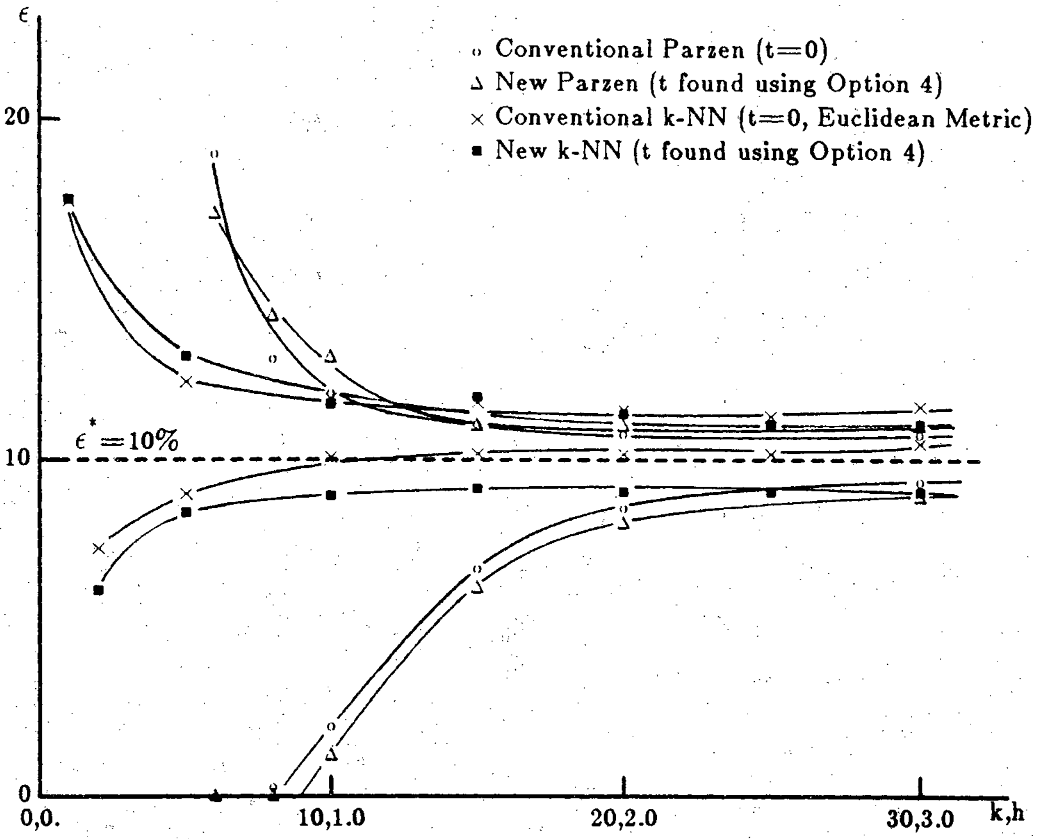


Figure 3.9 Comparison of conventional and modified procedures (Case 1)

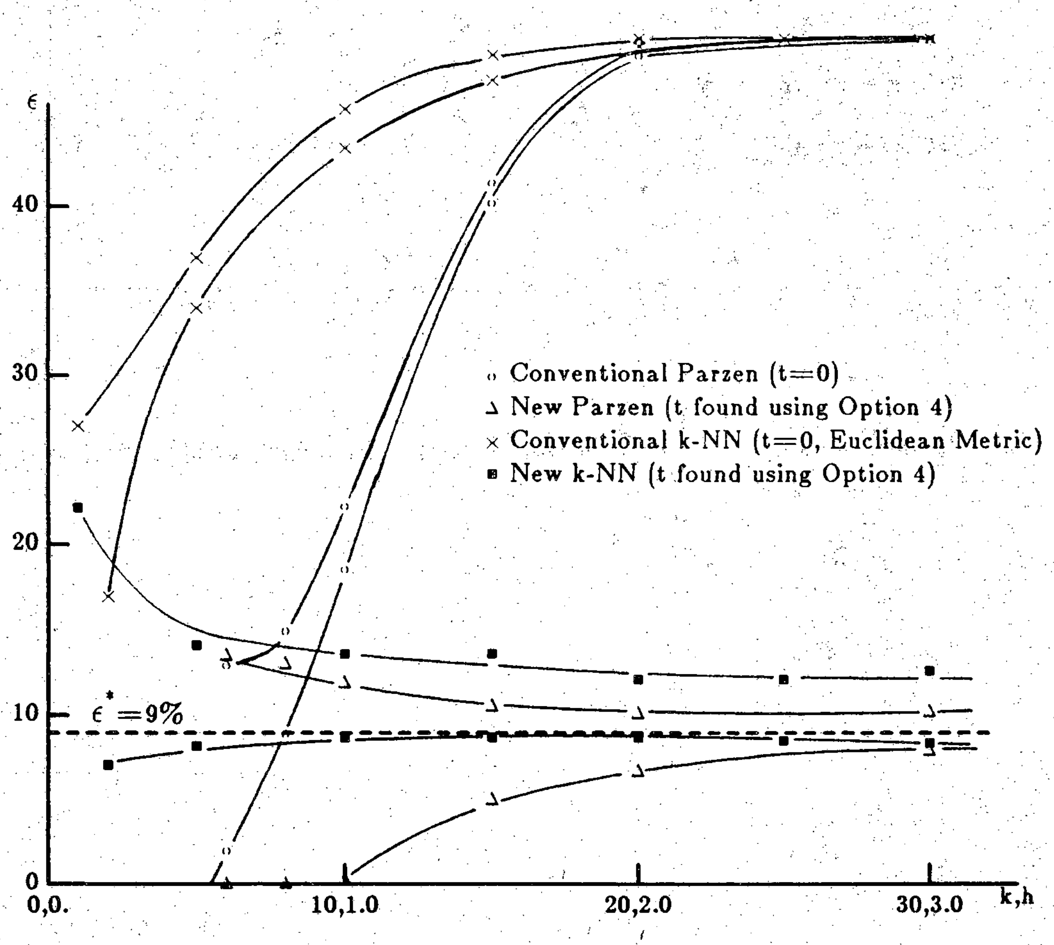


Figure 3.10 Comparison of conventional and modified procedures (Case 2)

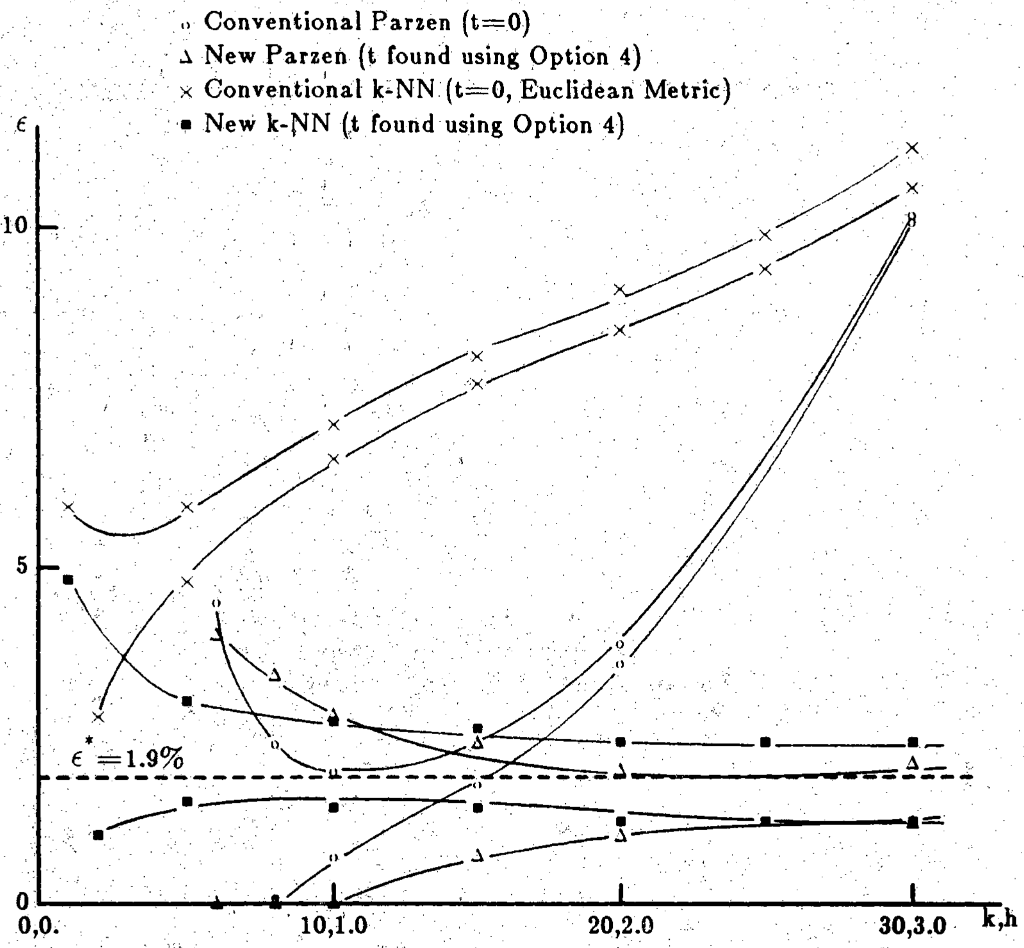


Figure 3.11 Comparison of conventional and modified procedures (Case 3)

cases, the conventional k-NN error rate is virtually unusable as a bound on the Bayes error, while the new k-NN estimates converge nicely, even for large values of k. The values of the conventional Parzen estimates are also questionable, since they seem to be extremely sensitive to the value of h. By simply changing the value of t as h is increased, this sensitivity can be nearly eliminated.

3.9 Summary

In this chapter we have examined the use of k-NN and Parzen error estimation procedures under conditions in which the number of design samples are limited. The k-NN and Parzen procedures have been developed in parallel so that valid comparisons could be made between them. The resulting k-NN error estimation procedure allows different metrics to be used for each class, and is thus a generalization of the k-NN counting procedure commonly in use. The most significant discovery of this chapter is the potential role of the decision threshold in error estimation and classifier design. The decision threshold is seen as a means of compensating for biases in the k-NN and Parzen density estimates under finite sample size conditions. We have found that by appropriately selecting the decision threshold, the error estimates obtained using the Parzen or k-NN procedure may be made much less sensitive to the values of k or h.

The experimental results here show no clear preference toward either the Parzen or k-NN procedure. Computationally, the k-NN errors are slightly easier to calculate since the error rate may be found for all values of k simultaneously, whereas the Parzen error rate must be completely recalculated for each value of h. As far as performance is concerned, neither procedure exhibits a clear advantage over the other. This is not surprising, given the close link between k-NN and Parzen procedures.

CHAPTER 4
LEAVE-ONE-OUT PROCEDURES FOR NONPARAMETRIC ERROR ESTIMATES

4.1. Introduction

Nonparametric Bayes error estimation plays an important role in pattern recognition, providing a means of evaluating various feature sets and forming a basis with which to evaluate various classifier designs. In Chapter 3, modified k-nearest neighbor (k-NN) and Parzen error estimation procedures were introduced which give significant improvement over the procedures commonly in use. These procedures provide estimates which are much less dependent on k (for k-NN estimate) and h (the Parzen smoothing parameter) than are the conventional estimates. However, in Chapter 3, knowledge of the class covariance matrices is assumed, ignoring the effects of estimating the covariance from the data.

We have found that in high-dimensional spaces, using the same data to estimate the kernel covariances and form the nonparametric classifier may lead to optimistically biased error estimates. This problem was brought to our attention while dealing with 66 dimensional feature vectors obtained by coherently sampling millimeter wave radar returns of various targets. The data was obtained by placing a Camero and van on a turn table, and time sampling the resulting radar returns at 10000 different angles ranging from 0 to 360 degrees. This large sample set was then reduced to sample sets of size 720 (samples taken every $1/2^{\circ}$) and 360 (samples taken every 1°). When the Parzen estimation procedures given in Chapter 3 were employed using 360 design samples per class (and estimating the kernel covariances from the same 360 samples), lower and upper bounds on the Bayes error of 3.9% and 5.1% respectively were obtained. However, when the sample size is increased to 720 samples per class, bounds of 7.0% to 8.2% resulted. Using good estimates of the covariance (estimates obtained using 10000 samples per class) gave the most realistic error bounds for this data — 16.4% to 17.6%, again using 720 samples per class to form the nonparametric error estimates. These inconsistent results demonstrate that the upper bounds on the Bayes error (given by the leave-one-out nonparametric procedures) may be severely (optimistically) biased if the class covariances are estimated from the same data as is used to form the error estimates.

If possible, then, to avoid this bias one should estimate the class covariances using a large number of independent samples as was done above. Once the covariances are estimated accurately, we have found that we may use a relatively small sample size for the nonparametric procedures given in Chapter 3 to produce reliable results. However, if additional samples for estimation of the covariance matrices are not available, then in order to obtain reliable upper bounds on the Bayes error one must use leave-one-out type estimates of the kernel covariances when forming the leave-one-out error estimate. This implies the use of a different covariance matrix for each sample being tested. This chapter gives an efficient method of calculating this estimate, requiring little addition computational effort over the procedures given in Chapter 3.

4.2 The Leave-one-out Procedure

Under the leave-one-out error estimation procedures given in Chapter 3, all samples $X_j^{(k)}$, $j=1,\ldots,N$ from class ω_k are tested using a nonparametric classifier designed using the N-1 samples $X_i^{(k)}$, $i=1,\ldots,N$, $i \neq j$. For both the k-NN and Parzen estimates, we are concerned with the calculation of the normalized distance estimates

$$\hat{d}^2(X_i,X_j) = (X_i - X_j)^T \hat{\Sigma}_k^{-1} (X_i - X_j) \tag{1}$$

where in this chapter we assume that the covariance matrix Σ_k has been estimated using the unbiased estimate

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

$$\hat{M}_k = \frac{1}{N} \sum_{i=1}^{N} X_i^{(k)}$$

Note that $\hat{\Sigma}_k$ and $\hat{M}_k$ represent the covariance and mean estimates formed using all samples, including the sample being tested, $X_j^{(k)}$. In order to obtain a valid upper bound for the error rate, we must remove the affect of $X_j^{(k)}$ on the covariance estimate $\hat{\Sigma}_k$. Let $\hat{\Sigma}_{kj}$ denote the estimate of the covariance estimate for class ω_k using all samples except $X_j^{(k)}$. Then [18]

$$\hat{\Sigma}_{kj}^{-1} = (\frac{N-2}{N-1}) \left[\hat{\Sigma}_k^{-1} + \frac{N\,\hat{\Sigma}_k^{-1}(X_j^{(k)}-\hat{M}_k)(X_j^{(k)}-\hat{M}_k)^T\,\hat{\Sigma}_k^{-1}}{(N-1)^2 - N(X_j^{(k)}-\hat{M}_k)^T\,\hat{\Sigma}_k^{-1}\,(X_j^{(k)}-\hat{M}_k)} \right] \tag{3}$$

Hence, when $X_j^{(k)}$ is removed from the covariance estimate, the desired normalized distance, formed using the covariance estimated using the remaining N-1 samples, becomes

$$(X_i^{(k)}-X_j^{(k)})^T\,\hat{\Sigma}_{kj}^{-1}(X_i^{(k)}-X_j^{(k)}) = (\frac{N-2}{N-1}) \left[\hat{d}^2(X_i^{(k)},X_j^{(k)}) + \frac{N[(X_i^{(k)}-X_j^{(k)})^T\,\hat{\Sigma}_k^{-1}(X_j^{(k)}-\hat{M}_k)]^2}{(N-1)^2 - N\hat{d}^2(X_j^{(k)},\hat{M}_k)} \right]$$

$$= (\frac{N-2}{N-1}) \left[\hat{d}^2(X_i^{(k)},X_j^{(k)}) + \frac{N[\hat{d}^2(X_i^{(k)},\hat{M}_k) - \hat{d}^2(X_j^{(k)},\hat{M}_k) - \hat{d}^2(X_i^{(k)},X_j^{(k)})]^2}{4[(N-1)^2 - N\hat{d}^2(X_j^{(k)},\hat{M}_k)]} \right] \tag{4}$$

Equation (4) relates the leave-one-out distance using $\hat{\Sigma}_{kj}$ to the distances $\hat{d}^2(X_i^{(k)},X_j^{(k)})$, $\hat{d}^2(X_j^{(k)},\hat{M}_k)$, and $\hat{d}^2(X_i^{(k)},\hat{M}_k)$ which are formed as in (1) using the full covariance and mean estimates given in (2). Hence, assuming that the N distances $\hat{d}^2(X_i^{(k)},\hat{M}_k)$ $i=1,\ldots,N$ have been saved, calculation of the leave-one-out distances $(X_i^{(k)}-X_j^{(k)})^T\,\hat{\Sigma}_{kj}^{-1}(X_i^{(k)}-X_j^{(k)})$ requires little additional effort over the calculation of $\hat{d}^2(X_i^{(k)},X_j^{(k)})$. Note that the conversion expression (4) involves only one-dimensional quantities, requiring no vector or matrix manipulation.

Also required for k-NN and Parzen estimates are the covariance determinants. The determinant of $\hat{\Sigma}_{kj}$ may be expressed in terms of $|\hat{\Sigma}_k|$ as follows [18]:

$$|\hat{\Sigma}_{kj}| = (\frac{N-1}{N-2})^n\,|\hat{\Sigma}_k|\,(1 - \frac{N}{(N-1)^2}\,\hat{d}^2(X_j^{(k)},\hat{M}_k)) \tag{5}$$

where n is the dimension of the feature vectors. Hence, assuming $|\hat{\Sigma}_k|$ has been calculated, a straightforward calculation gives $|\hat{\Sigma}_{kj}|$, provided the distances $\hat{d}^2(X_j^{(k)},\hat{M}_k)$ have been calculated and saved.

The modified nonparametric error estimate which employs an estimated covariance matrix may be summarized by the following steps:

1) Find $\hat{\Sigma}_k$ and $\hat{M}_k$ using all available samples as in (2). Use $\hat{\Sigma}_k$ to calculate $|\hat{\Sigma}_k|$ and $\hat{\Sigma}_k^{-1}$. Note that, as pointed out in Chapter 3, the computational complexity of the nonparametric procedures is greatly reduced if the data is first transformed so that $\hat{\Sigma}_k$ is diagonal. This makes the calculation of the N^2 distance terms, $\hat{d}(X_i^{(k)},X_j^{(k)})$, very simple.

2) Find and save $\hat{d}^2(X_i^{(k)},\hat{M}^{(k)})$ for all samples $(i=1,\ldots,N)$

3) When testing sample $X_j^{(k)}$:

   i) Find $\hat{d}(X_i^{(k)},X_j^{(k)})$ using $\hat{\Sigma}_k^{-1}$, $i=1,\ldots,N$. Use these values in forming the resubstitution error rates.

   ii) Remove the affect of $X_j^{(k)}$ on the covariance estimate using the (one-dimensional) expressions (4) and (5). Use these values in forming the leave-one-out density estimates at $X_j^{(k)}$

   iii) Use the Parzen or k-NN procedures as developed in Chapter 3. Note that the distance correction of step (ii) need only be performed when measuring the distances to samples of the same class as sample $X_j^{(k)}$, since $X_j^{(k)}$ does not enter into the covariance estimates of the other classes.

4.3 Experimental Results

The three experimental test cases used in Chapter 3 were repeated here, this time using estimated covariance matrices rather than the theoretical covariance matrices as in Chapter 3. All three cases are two class problems involving 8-dimensional Gaussian data. For test case 1, the class covariances are equal and the mean vectors are separated to give a theoretical Bayes error of $\epsilon^* = 10.0\%$. Test case 2 is an equal mean test case with class 1 and class 2 covariances of I and 4I respectively where I is the identity matrix. These parameters yield a Bayes error of 9.0%. Test case 3 is a complex distribution in which both the means and covariances are different, giving a Bayes error of $\epsilon^* = 1.9\%$. The reader is referred to Chapter 3 for details of the distribution parameters. For the experiments presented here, 100 samples per class were used to form the error estimates. Table 4.1 shows the leave-one-out and resubstitution error estimates obtained using both k-NN (for $k=10$) and Parzen (for $h=1.5$) procedures. Also shown in Table 4.1 are the corresponding results obtained in Chapter 3, which use the true covariance matrices. The results were averaged over ten trials, and the corresponding standard deviations, σ_{Loo} and σ_R are listed. To test the validity of the experiments, several of the test cases were repeated using 50 (rather than 10) trials. Changes in all the estimates were small (for example, using the Parzen estimate for case 1, the estimate of σ_{Loo} changed from 2.6% to 2.52%, and the estimate of σ_R from 1.3% to 1.25%). Comparison of the new results with those presented in Chapter 3 shows only

Table 4.1 Parzen and k-NN error estimates for the three experimental test cases used in Chapter 3. All results are based on 8-dimensional data with 100 samples per class forming the estimate.

(a) 10-NN error estimates

Test Case	Bayes Error	Covariance Used	L.o.o. Error	σ_{Loo}	Resubst. Error	σ_{R}
1	10%	True	11.9%	2.2%	8.7%	1.8%
		Estimated	13.6%	3.2%	8.2%	1.8%
2	9%	True	13.6%	2.8%	9.2%	2.6%
		Estimated	17.7%	5.0%	9.0%	2.1%
3	1.9%	True	2.7%	1.0%	1.4%	0.7%
		Estimated	3.2%	1.3%	1.3%	0.6%

(b) Parzen (h=1.5) error estimates

Test Case	Bayes Error	Covariance Used	L.o.o. Error	σ_{Loo}	Resubst. Error	σ_{R}
1	10%	True	11.0%	1.8%	6.4%	1.3%
		Estimated	12.6%	2.6%	5.8%	1.3%
2	9%	True	10.6%	2.9%	4.8%	1.0%
		Estimated	11.0%	3.2%	4.5%	1.3%
3	1.9%	True	1.7%	1.2%	1.1%	0.9%
		Estimated	2.3%	0.9%	0.8%	0.6%

slight degradation, this is due to the use of estimated, rather than true covariance matrices. Both the resubstitution and the new leave-one-out procedures continue to provide useful bounds on the Bayes error.

In order to more clearly demonstrate the behavior of the estimate in higher dimensional spaces, the procedures were tested using the 66-dimensional radar signature data referred to in the introduction. The results are shown in Table 4.2. The first two entries of the table show the performance of the nonparametric classifiers when good covariance estimates (formed using 10,000 samples per class) are used with limited sample sizes (360 and 720 per class) used as references to form the classifier. The second two entries indicate the performance of the procedures presented here, in which the same data is used to estimate the covariances and form the reference sets. The true Bayes error for this set of data is not known. Note that even as the sample size used to estimate the covariance matrices becomes small, the leave-one-out error rates continue to provide reasonable and consistent bounds. This is in contrast to the results given in the introduction in which the estimated covariances are blindly used without employing the leave-one-out procedures of this chapter. As expected, the bounds become worse as the sample sizes decrease. This is particularly the case for the resubstitution error rates, which become heavily biased as fewer samples are used to form the covariance estimates.

4.4 Summary

When the nonparametric procedures presented in Chapter 3 are applied in high dimensional spaces, the estimation of the class covariances (i.e. the determination of the kernel/metric shape) plays a very significant role. Specifically, if the same data is used to estimate the kernel shapes and form the error estimates, severely biased results may be obtained. This chapter has provided an efficient leave-one-out algorithm which may be used to eliminate this optimistic bias, while making effective use of all available design samples. This procedure requires only a slight increase in computational complexity over the procedures presented in Chapter 3, while still giving reliable bounds when covariance estimates, rather than the true covariances, are to be used to form the nonparametric classifiers.

Table 4.2 Error estimates obtained using 66 dimensional feature vectors derived by sampling millimeter wave radar returns of two targets. N_{cov} represents the number of samples (per class) used to estimate the covariance matrices, and N represents the number of samples used to form the nonparametric error estimates.

N_{cov}	N	Parzen (h = 9.0)		10-NN	
		Leave-one-out	Resubst.	Leave-one-out	Resubst.
10000	720	17.6%	16.4%	22.4%	18.0%
10000	360	19.4%	15.8%	22.2%	18.8%
720	720	23.0%	7.0%	24.2%	10.1%
360	360	27.5%	3.9%	29.3%	6.4%

CHAPTER 5
PERFORMANCE OF NONPARAMETRIC CLASSIFIERS

5.1 Introduction

In this chapter, we examine the performance of classifiers designed using Parzen and k-nearest neighbor (k-NN) density estimates. When the number of design samples is infinite, the convergence of the Parzen on k-NN error rates to the Bayes error is well understood. Here, we derive expressions which give the expected error performance of the classifiers when a finite number of design samples is available. These expressions relate the observed error rate to the true value of the Bayes error, and show how that error rate is affected by changes in the sample size or design parameters of the density estimates.

We begin in Section 5.2 by deriving the expected error performance in terms of the mean and variance of the density estimates used to form the classifier. This section is primarily an application of the techniques presented in [47] to the problem of classifier design. The results of this section are very general and have application beyond the scope of this thesis. In [47], similar expressions are used to examine the behavior of classifiers designed using parametric techniques. Sections 5.3 and 5.4 evaluate these expressions for the Parzen and k-NN classifiers respectively. The results show how the Parzen error rate is affected by the sample size, the size and shape of the kernel function, and the value of the decision threshold. The importance of the decision threshold was demonstrated in Chapter 3, and this chapter provides some theoretical justification for these results. Similarly, the performance of the k-NN classifier is expressed in terms of the metric, the value of k, and the sample size. These expressions are then used in Sections 5.5 and 5.6 to improve the performance of these classifiers. Section 5.5 introduces a method of estimating the true value of the Bayes error from the observed error rates for the nonparametric classifiers. Currently nonparametric error rates are used to form upper bounds on the Bayes error, and may in some cases be quite biased. We believe that the techniques of Section 5.5 more effectively use the available information and represent a significant improvement over the current techniques. Section 5.6 presents some guidelines regarding the selection of the kernel shape to improve the error performance of the classifiers. These results nicely complement results already in existence regarding the selection of the optimal metric for k-NN classifiers.

5.2 Effect of Finite Design Set on Classifier Performance

In this section, we develop the relationship between the accuracy of a density estimation procedures and the corresponding error rate obtained when the density estimates are used to form a classifier. Our development closely follows the work presented in [47], which gives similar results for linear and quadratic classifiers.

We begin by considering the probability of error, ϵ, provided by the classifier

$$h(x) \underset{\omega_2}{\overset{\omega_1}{\lessgtr}} 0 \tag{1}$$

where x is an n-dimensional random vector with density for class ω_i given by $p_i(x)$ (i=1,2). The probability of error for samples from each class is given by

$$\epsilon_1 = \int_0^\infty q_1(h)dh = \frac{1}{2} + \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{1}{j\omega}\phi_1(\omega)d\omega \tag{2}$$

$$\epsilon_2 = \int_{-\infty}^{0} q_2(h)dh = \frac{1}{2} - \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{1}{j\omega}\phi_2(\omega)d\omega \tag{3}$$

where $q_i(h)$ is the density of h(x) for class ω_i and $\phi_i(\omega)$ is the corresponding characteristic function:

$$\phi_i(\omega) = E\{e^{j\omega h(x)}|\omega_i\} = \int_{R^n} e^{j\omega h(x)} p_i(x)dx \tag{4}$$

The error rate for the classifier, $\epsilon = P_1\epsilon_1 + P_2\epsilon_2$, may now be expressed.

$$\epsilon = \frac{1}{2} + \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{R^n} \frac{1}{j\omega} e^{j\omega h(x)}(P_1p_1(x) - P_2p_2(x))dx\, d\omega \tag{5}$$

Here, P_i is the a priori probability of class ω_i.

In practice, h(x) is generally unknown and must be estimated using a set of N preclassified design samples. We define a random variable $\Delta h(x)$

which relates the true value of h(x) to the estimate.

$$\hat{h}(x) = h(x) + \Delta h(x) \tag{6}$$

The probability of error using $\hat{h}(x)$ may be obtained by substituting $\hat{h}(x)$ for h(x) in (5). When $\Delta h(x)$ is small, the exponential of (5) may be approximated as

$$\begin{aligned} e^{j\omega\hat{h}(x)} &= e^{j\omega h(x)}e^{j\omega\Delta h(x)} \\ &\cong e^{j\omega h(x)}\left[1 + j\omega\Delta h(x) + \frac{1}{2}(j\omega\Delta h(x))^2\right] \end{aligned} \tag{7}$$

The expected degradation of the error rate, $\Delta\epsilon$, may now be obtained by substituting (7) into (5) and taking the expectation over the set of N design samples.

$$E\{\Delta\epsilon\} = \frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{R^n} E\{\Delta h(x) + \frac{j\omega}{2}\Delta h^2(x)\}e^{j\omega h(x)}(P_1p_1(x) - P_2p_2(x))dxd\omega \tag{8}$$

This expression approximates the expected error degradation in terms of the bias and mean square error of $\hat{h}(x)$.

In this chapter, we investigate the estimation of the Bayes error using classifiers for which the estimated log likelihood ratio $\hat{h}(x) = -\ln(\hat{p}_1(x)/\hat{p}_2(x))$ is used. We wish to relate the error rate for this classifier to the Bayes error. Assuming $\hat{p}_i(x) = p_i(x) + \Delta p_i(x)$, then

$$\Delta h(x) = \ln\left(1 + \frac{\Delta p_2(x)}{p_2(x)}\right) - \ln\left(1 + \frac{\Delta p_1(x)}{p_1(x)}\right) - t \tag{9}$$

The parameter t in (9) is included to allow for the possible use of a non-zero threshold in (1). Expanding the log terms of (9), and dropping third or higher order terms gives

$$\Delta h(x) \cong \frac{\Delta p_2(x)}{p_2(x)} - \frac{1}{2}\left(\frac{\Delta p_2(x)}{p_2(x)}\right)^2 - \frac{\Delta p_1(x)}{p_1(x)} + \frac{1}{2}\left(\frac{\Delta p_1(x)}{p_1(x)}\right)^2 - t. \tag{10}$$

Squaring this result, and again dropping the higher order terms,

$$\Delta h^2(x) \cong \left(\frac{\Delta p_2(x)}{p_2(x)}\right)^2 + \left(\frac{\Delta p_1(x)}{p_1(x)}\right)^2 - 2\left(\frac{\Delta p_1(x)}{p_1(x)}\right)\left(\frac{\Delta p_2(x)}{p_2(x)}\right) + t^2$$

$$-2t\left[\frac{\Delta p_2(x)}{p_2(x)} - \frac{1}{2}\left(\frac{\Delta p_2(x)}{p_2(x)}\right)^2 - \frac{\Delta p_1(x)}{p_1(x)} + \frac{1}{2}\left(\frac{\Delta p_1(x)}{p_1(x)}\right)^2\right] \tag{11}$$

The expected error deviation can now be obtained by taking the expectation of (10) and (11) and substituting into (8). The actual evaluation depends upon the particular density estimates used. The parametric case assuming Gaussian distributions is treated in [47]. Sections 3 and 4 will evaluate these expressions for Parzen and k-NN classifiers respectively.

5.3 Degradation of the Parzen Classifier

5.3.1 Evaluation of $E\{\Delta\epsilon\}$:

We now assume that Parzen, or kernel type density estimates are used. The general Parzen density estimate is given by

$$\hat{p}_i(x) = (1/N) \sum_{j=1}^{N} (1/h^n)k_i((x - x_j^{(i)})/h) \tag{12}$$

In (12), $k_i(x)$ is a non-negative symmetric kernel function with $\int k_i(x)dx = 1$, h is a scaling parameter which determines the size of the kernel function and the smoothness of the density estimate, and $x_j^{(i)}(j=1,2,\ldots,N)$ are the N design samples from class ω_i. In order to simplify the notation, and to more clearly show the relationships between the Parzen and k-NN classifiers, we will assume a uniform kernel function:

$$(1/h^n)k_i((x - x_j^{(i)})/h) = \begin{cases} 1/v_i & x_j^{(i)} \in S_x \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

where $S_x = \{Y{:}d_i(x,Y) < h\sqrt{n+2}\}$, v_i is the volume of S_x, and $d_i(x,Y)$ is a metric measuring the distance from x to Y. This assumption significantly simplifies the discussion, while not appreciably altering the final result. A derivation in terms of general symmetric kernel functions is given in Appendix B. For the uniform kernel function the density estimate of (12) may be rewritten as

$$\hat{p}_i(x) = \frac{k}{Nv_i} \tag{14}$$

where k is the number of design samples which fall in the region S_x. For the Parzen estimate, v_i is held fixed and k is random and has a binomial distribution with parameter u_i, where

$$u_i = \Pr\{x_j^{(i)} \in S_x | \omega_i\} = \int_{S_x} p_i(Y)dY \tag{15}$$

The dependence of u_i on x is understood. The moments of k are then given in terms of u_i by

$$E\{k\} = Nu_i \tag{16}$$

$$E\{k^2\} = Nu_i + N(N-1)u_i^2 \tag{17}$$

The expectation of the error terms of (10) and (11) may now be calculated by combining (14), (16) and (17):

$$E\left\{\left(\frac{\Delta p_i(x)}{p_i(x)}\right)\right\} = \frac{u_i - p_i(x)v_i}{p_i(x)v_i} \tag{18}$$

$$E\left\{\left(\frac{\Delta p_i(x)}{p_i(x)}\right)^2\right\} = \left(\frac{u_i - p_i(x)v_i}{p_i(x)v_i}\right)^2 + \frac{1}{N}\left[\frac{u_i}{p_i^2(x)v_i^2} - \left(\frac{u_i}{p_i(x)v_i}\right)^2\right] \tag{19}$$

In order to express the error degradation in terms of the kernel size and shape, u_i and v_i must be expressed in terms of these quantities. For the density estimate of (14) to be accurate, v_i (and hence h) must be small. That is, S_x should represent a local region centered at x. Under these conditions, the integral of (15) may be approximated using a second order Taylor series about x:

$$p_i(Y) \cong p_i(x) + \nabla^T p_i(x)(Y-x) + \frac{1}{2}(Y-x)^T \nabla^2 p_i(x)(Y-x). \tag{20}$$

In order to evaluate the integral of (15), we must also specify the shape of the region S_x. For simplicity and practicality, we will assume that a quadratic metric is used to specify S_x:

$$d_i^2(x,Y) = (x-Y)^T \textstyle\sum_{k_i}^{-1}(x-Y) \tag{21}$$

This choice gives a scaled kernel function with covariance $h^2\sum_{k_i}$, which is consistent with notation used in previous papers as well as with the results for general kernel functions given in Appendix B. The integration of (15) may now be evaluated. Note that the first order term of the expansion in (20) vanishes as a result of the symmetry of the region S_x. The result is

$$u_i \cong p_i(x)v_i + \frac{1}{2}\mathrm{tr}\left[\nabla^2 p_i(x) \int_{S_x} (Y-x)(Y-x)^T dY\right]$$

$$= p_i(x)v_i + \frac{1}{2(1 + n/2)\alpha_i^{2/n}} v_i^{1 + 2/n} p_i(x)c_i(x) \tag{22}$$

where

$$c_i(x) = \mathrm{tr}\left[\frac{\nabla^2 p_i(x)}{p_i(x)} \Sigma_{k_i}\right] \tag{23}$$

$$\alpha_i = \frac{(2\pi)^{n/2}|\Sigma_{k_i}|^{1/2}}{\Gamma(1 + n/2)} \tag{24}$$

and tr[A] represents the trace of the matrix A. The relationship between v_i and h is given by

$$v_i = \alpha_i(1 + n/2)^{n/2}h^n \tag{25}$$

Combining (18) and (19) with (22) and (25) gives the desired error quantities in terms of the kernel size and shape, and the number of design samples.

$$E\left\{\frac{\Delta p_i(x)}{p_i(x)}\right\} \cong \frac{1}{2}h^2c_i(x) \tag{26}$$

$$E\left\{\left(\frac{\Delta p_i(x)}{p_i(x)}\right)^2\right\} \cong \frac{1}{4}h^4c_i^2(x) + \frac{1}{N}\left[\frac{1 + (1/2)h^2c_i(x)}{p_i(x)\alpha_i(1+n/2)^{n/2}h^n} - (1+\frac{1}{2}h^2c_i(x))^2\right] \tag{27}$$

Appendix B shows that virtually identical results may be obtained for arbitrary kernel functions.

The error degradation given in (8) is in terms of the expectation of (10) and (11). These terms may now be calculated using (26) and (27).

$$E\{\Delta h(x)\} = \frac{1}{2}h^2(c_2(x) - c_1(x)) + \frac{1}{8}h^4(c_1^2(x) - c_2^2(x)) - t$$
$$+ \frac{1}{2N}\left[\frac{1 + (1/2)h^2c_1(x)}{p_1(x)\alpha_1(1 + n/2)^{n/2}h^n} - \frac{1 + (1/2)h^2c_2(x)}{p_2(x)\alpha_2(1 + n/2)^{n/2}h^n}\right.$$
$$\left. + h^2(c_2(x) - c_1(x)) + (1/4)h^4(c_2^2(x) - c_1^2(x))\right] \tag{28}$$

$$E\left\{(\Delta h(x))^2\right\} \cong \left[\frac{1}{2}h^2(c_2(x) - c_1(x)) - t\right]^2 - \frac{1}{4}th^4(c_1^2(x) - c_2^2(x))$$

$$+ \frac{1}{N}\left[\frac{(1-t)(1+(1/2)h^2c_1(x))}{p_1(x)\alpha_1(1+n/2)^{n/2}h^n} + \frac{(1+t)(1+(1/2)h^2c_2(x))}{p_2(x)\alpha_2(1+n/2)^{n/2}h^n}\right.$$

$$- ((1+t)c_2(x) + (1-t)c_1(x))h^2$$

$$\left. - \frac{1}{4}((1+t)c_2^2(x) + (1-t)c_1^2(x))h^4 - 2\right] \tag{29}$$

5.3.2 Effect of N and h:

Substituting (28) and (29) into (8), we obtain the error degradation in terms of the sample size N and the value of h:

$$E\{\Delta\epsilon\} = a_1h^2 + a_2h^4 + \frac{1}{N}\left[a_3h^{-n} + a_4h^{-n+2} + a_5h^2 + a_6h^4\right] \tag{30}$$

Here, the constants $a_1,\ldots,a_6$ are obtained by evaluation of the indicated integral expression in (8). In this section, we have assumed for simplicity that the decision threshold t has been set to zero. Because of the complexity of the expressions, explicit evaluation is not possible. However the constants are functions only of the distributions and the kernel shapes, $\sum_{k_i}$, and are completely independent of the sample size or the smoothing parameter, h. Hence, (30) gives an expression showing how changes in h or N effect the error performance of the classifier. The a_1h^2 and a_2h^4 terms show how biases in the density estimates influence the performance of the classifier, while the a_3h^{-n}/N and a_4h^{-n+2}/N terms reflect the role of the variance of the density estimates. For small values of h, the variance terms dominate (30) and the observed error rates are significantly above the Bayes error. As h grows, however, the variance terms decrease while the a_1h^2 and a_2h^4 terms play an increasingly significant role. Thus, for a typical plot of the observed error rate versus h, ϵ decreases for small values of h until a minimum point is reached, and then increases as the bias terms of the density estimates becomes more significant. This behavior has repeatedly been observed by many researchers, and is accurately predicted in our expression for $E\{\Delta\epsilon\}$.

It should be noted that although explicit evaluation of a_1 through a_6 is not possible in general, it is reasonable to expect that these constants are positive. It is certainly true that $E\{\Delta\epsilon\}$ must be positive for any value of h, since the Bayes decision rule is optimal in terms of error performance. Also, close examination of (28), (29), and (8) show that the constants a_5 and a_6 are of the same order of magnitude as a_1 and a_2, so that for reasonable values of N (say, $N > 50$) the influence of the a_5 and a_6 terms is small in comparison with the a_1 and a_2 terms, and for practical purposes may be neglected.

The role of the sample size, N, in (30) is seen as a means of reducing the terms corresponding to the variance of the density estimates. Hence the primary effect of the sample size is seen at the smaller values of h, where the a_3 and a_4 terms of (30) dominate. As h grows, and the a_1 and a_2 terms become dominant, changing the sample size has a decreasing effect on the resulting error rate. These observations were verified experimentally. Figure 5.1 shows the observed Parzen error rates obtained for a particular set of Gaussian distributions for N ranging from 25 to 200 design samples per class, h ranging from 0.6 to 2.4. The distributions were 8-dimensional Gaussian with $M_1 = 0, \sum_1 = I$, $\sum_2 = \Lambda$ and M_2 as specified in Table 5.1. Here M_i and $\sum_i$ represent the mean vector and covariance matrix respectively for class ω_i, I is the identity matrix, and Λ is a diagonal matrix with diagonal elements $\lambda_k(k=1,...8)$. The chosen values of λ_k are also shown in Table 5.1. These parameters yield a Bayes error of 1.9%. For each combination of N and k, N independent samples per class were generated and used to form $\hat{h}(x) = -\ln(\hat{p}_1(x)/\hat{p}_2(x))$ where $\hat{p}_i(x)$ is as given in (12). Gaussian kernel functions were used with covariance $\sum_{k_i} = \sum_i$. The decision rule of (1) was then evaluated using 1000 test samples per class independently generated from the two distributions. The number of classification errors was counted and divided by the number of test samples to give an estimate of the probability of error for each design set. These results were averaged over ten independent design sets to provide an estimate of the expected error rate. Figure 5.1 shows that for each value of N, the Parzen classifier behaves as predicted by (30), decreasing to a minimum point, and then increasing as the biases of the density estimates become significant for larger values of h. Also note that the sample size

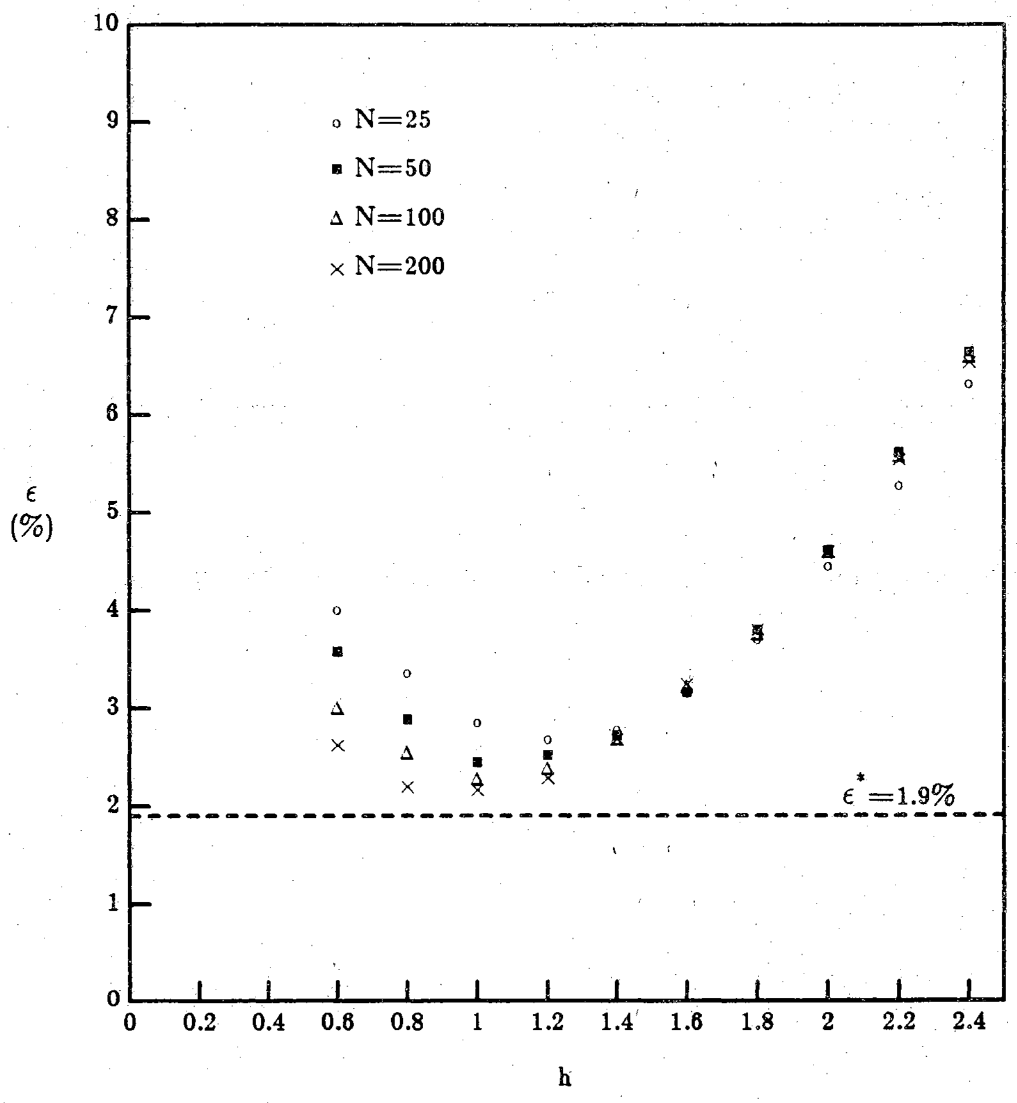


Figure 5.1. Parzen error rates for various sample sizes and values of h.

Table 5.1 Diagonal elements of Λ, and the corresponding components of the mean vector.

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

plays its primary role for small values of h, where the a_3 and a_4 terms are most significant, and has almost no effect at the larger values of h.

Note that in order to have $E\{\Delta\epsilon\} \rightarrow 0$ as $N \rightarrow \infty$, our error approximation implies that h must be chosen as a function of N such that $h \rightarrow 0$ and $\frac{1}{N} h^{-n} \rightarrow 0$. This is the well known condition fo the consistency of the Parzen density estimate, and lends confidence to the approximations which we used to obtain (30).

5.3.3 Effect of the Decision Threshold t:

In Chapter 3, the we introduced the use of the decision threshold as a possible means of correcting for the biases of density estimates and improving the performance of the Parzen classifier. The arguments presented were rather intuitive in nature. However, the experimental results indicate that threshold adjustment is a very effective tool toward this end, particularly for large values of h. To some extent, (28) and (29) provide some theoretical justification for the claims made in Chapter 3, showing the role which the decision threshold plays in terms of error performance.

Convergence of the observed error rate to the Bayes error may be improved by selecting t to compensate for the first two terms of (28).

$$t = (1/2)h^2(c_2(x) - c_1(x)) + (1/8)h^4(c_1^2(x) - c_2^2(x)) \qquad (31)$$

This selection eliminates the bias terms of (28), and reduces the bias terms of (29) to only higher order terms (h^6 and h^8). All other terms of (28) and (29) decrease as 1/N, and may be eliminated by choice of a large enough sample size. Further, because the bias terms are greatly reduced, it may be possible to use a much larger value of h, thereby reducing the h^{-n} and h^{-n+2} terms which are related to the variance of the density estimate.

In practice, use of (31) to determine t may not be possible. Since $c_i(x)$ is a complex function of x depending on the second order properties of the distribution, $c_i(x)$ is generally unknown and hard to estimate. In many cases, however, it may be possible to approximately satisfy (31) at least in the region close to the Bayes decision boundary. Since the integral expression of (8) depends primarily on the behavior of $\hat{h}(x)$ near the decision boundary, such an approach yields a significant improvement in the resulting error probability for the classifier.

Insight may be gained by examining the Gaussian case in which the shape of the kernel is chosen to be proportional to the covariance matrix for class ω_i, $\Sigma_{k_i} = \Sigma_i$. In this case $c_i(x)$ may be explicitly calculated and has the form

$$c_i(x) = (x - M_i)^T \Sigma_i^{-1}(x - M_i) - n = d_i^2(x,M_i) - n. \tag{32}$$

The threshold t may easily be chosen to compensate for the h^2 terms of (28) and (29) at least on the Bayes decision boundary:

$$\begin{aligned} t &= \frac{1}{2}h^2(c_2(x) - c_1(x)) \\ &= \frac{1}{2}h^2(d_2^2(x,M_2) - d_1^2(x_2M_1)) \\ &\cong \frac{1}{2}h^2\ln(|\Sigma_1|/|\Sigma_2|) \end{aligned} \tag{33}$$

The last line of (33) holds with equality only on the Bayes decision boundary. Thus in this case, our results indicate that a constant value of t may be used to improve the performance of the Parzen classifier. This observation was experimentally verified in Chapter 3. It is interesting to note that in Chapter 3 the appropriate value of the threshold was found to be $(1 + h^2)^{-1}$ times the value shown in (33). Thus the approximations used in this chapter yield consistent results with those of Chapter 3, at least for small values of h.

In general, an expression of the form of (33) cannot be found since the location of the Bayes decision boundary is unknown. However, (33) shows that use of a constant, nonzero threshold may result in a significant improvement of the performance of the Parzen classifier. In Chapter 3, general techniques are given for determining an appropriate threshold which do not depend on assumptions about the distributions. The results of this chapter strengthen the case for the use of these methods.

5.4 Degradation of the k-NN Classifier

5.4.1 Evaluation of E{$\Delta\epsilon$}:

A derivation similar to that of the previous section may be applied to k-NN classifiers. The k-NN density estimate is given by

$$\hat{p}_i(x) = \frac{k-1}{Nv_{ik}} \tag{34}$$

where k is an integer, v_{ik} is the volume of the set $S_{xk} = \{Y:d_i(x,Y)<d_i(x,x_{k-NN}^{(i)})\}$ and $x_{k-NN}^{(i)}$ is the kth NN to x from the N design samples of class ω_i. When $d_i(x,Y)$ is quadratic as in (21), then $v_{ik} = \alpha_i d_i^n(x,x_{k-NN}^{(i)})$. Note that in contrast to (14), v_{ik} is random in (34) and k is held fixed. To take the expectation of (34) we define $u_{ik} = \Pr\{x_j^{(i)}\epsilon S_{xk}|\omega_i\}$ and express v_{ik} in terms of u_{ik} by solving an expression similar to (22)

$$v_{ik} \cong \frac{u_{ik}}{p_i(x)(1+\beta_i(x)(p_i(x)v_{ik})^{2/n}} \tag{35}$$

where

$$\beta_i(x) = \frac{c_i(x)}{2(1+n/2)\alpha_i^{2/n}p_i^{2/n}(x)} \tag{36}$$

The v_{ik} term in the denominator of (35) may be eliminated by using the first order approximation $u_{ik} \cong p_i(x)v_{ik}$. This gives

$$1/v_{ik} \cong p_i(x)(u_{ik}^{-1} + \beta_i(x)u_{ik}^{2/n-1}) \tag{37}$$

$$(1/v_{ik})^2 \cong p_i^2(x)(u_{ik}^{-2} + 2\beta_i(x)u_{ik}^{2/n-2} + \beta_i^2(x)u_{ik}^{4/n-2}). \tag{38}$$

The distribution of u_{ik} is known to be Beta. The moments of u_{ik} are given by

$$E\{u_{ik}^{\alpha}\} = \frac{\Gamma(k+\alpha)\Gamma(N+1)}{\Gamma(k)\Gamma(N+1+\alpha)} \quad (\alpha+k>0) \tag{39}$$

Combining (39) with (34), (37), and (38), the desired moments for the density estimate may be obtained:

$$E\{\frac{\Delta p_i(x)}{p_i(x)}\} \cong \beta_i(x)g(N,k,2/n) \tag{40}$$

$$E\left\{\left(\frac{\Delta p_i(x)}{p_i(x)}\right)^2\right\} \cong \left[\left(\frac{k-1}{k-2}\right)\left(\frac{N-1}{N}\right) - 1\right]$$

$$+ 2\left[\left(\frac{k-1}{k-2-2/n}\right)\left(\frac{N-1+2/n}{N}\right) - 1\right]\beta_1(x)g(N,k,2/n)$$

$$+ \left[\left(\frac{k-1}{k-2-4/n}\right)\left(\frac{N-1+4/n}{N}\right)\right]\beta_1^2(x)g(N,k,4/n) \qquad (41)$$

where

$$g(N,k,\delta) = \frac{\Gamma(k-1+\delta)\Gamma(N)}{\Gamma(k-1)\Gamma(N+\delta)} . \qquad (42)$$

Note that for larger values of N (say, N>50), g may be accurately approximated by

$$g(N,k,\delta) \cong \frac{\Gamma(k-1+\delta)}{\Gamma(k-1)} N^{-\delta} \cong (k/N)^{\delta} \qquad (43)$$

so that for positive values of δ, $g(N,k,\delta)$ decreases with N. The second approximation of (43) holds only for large k. Taking the expectation of (10) and (11), and substituting (40) and (41) into this result gives the expectations of the Δh terms of the error expression (8):

$$E\{\Delta h(x)\} \cong \left[\left(\frac{k-1}{k-2+2/n}\right)\left(\frac{N-1+2/n}{N}\right) - 2\right](\beta_1(x) - \beta_2(x))g(N,k,2/n)$$

$$+ \frac{1}{2}\left(\frac{k-1}{k-2+4/n}\right)\left(\frac{N-1+4/n}{N}\right)\left(\beta_1^2(x) - \beta_2^2(x)\right)g(N,k,4/n) - t \qquad (44)$$

$$E\{(\Delta h(x))^2\} = 2\left[\left(\frac{k-1}{k-2}\right)\left(\frac{N-1}{N}\right) - 1\right]$$

$$+ \left[(\beta_2(x) - \beta_1(x))g(N,k,2/n) - t\right]^2 - (\beta_2^2(x) + \beta_1^2(x))g^2(N,k,2/n)$$

$$+ 2\left[\left(\frac{k-1}{k-2+2/n}\right)\left(\frac{N-1+2/n}{N}\right) - 1\right]\left[(1+t)\beta_2(x) + (1-t)\beta_1(x)\right]g(N,k,2/n)$$

$$+ \left(\frac{k-1}{k-2+4/n}\right)\left(\frac{N-1+4/n}{N}\right)\left[(1+t)\beta_2^2(x)+(1-t)\beta_1^2(x)\right]g(N,k,4/n) \qquad (45)$$

These expressions with (8) yield the behavior of $E\{\Delta\epsilon\}$ in terms of the sample size, the value of k, and the distributions.

For a fixed k, as N is increased all terms of (44) and (45) approach zero except for the first line of (45). This term corresponds to the finite variance of the k-NN density estimate under infinite sample set conditions, and results in the asymptotic k-NN error rate which is greater than the Bayes error. In order to guarantee convergence of the k-NN error to the Bayes error, k must be chosen as a function of N. To make all terms of (44) and (45) vanish we must require that $k \to \infty$ and $k/N \to 0$ as $N \to \infty$. These conditions are identical to the conditions required for asymptotic consistency of the k-NN density estimate provided by Loftgaarden [29], a result which lends confidence to the approximations used in this chapter.

5.4.2 Relationship Between k-NN and Parzen Procedures:

While the Parzen classifier with uniform kernel counts the number of design samples which fall within a specified volume, the k-NN classifier adjusts that volume to include a given number of design samples. For this reason, many researchers have suggested that the two procedures should give similar performance if the Parzen kernel size is determined adaptively, using larger kernel functions in regions in which the density is smaller. In this section, we show that this claim is largely supported by our error expressions.

We begin by rewriting the k-NN expressions (40) and (41) using large N and large k approximations. Using (43), dropping terms which behave as 1/Nk, we obtain

$$E\left\{\frac{\Delta p_i(x)}{p_i(x)}\right\} \cong \beta_i(x)(k/N)^{2/n} \tag{46}$$

$$E\left\{\left(\frac{\Delta p_i(x)}{p_i(x)}\right)^2\right\} \cong \left[1/k - 1/N\right]$$
$$+2(1-2/n)(1/k-1/N)\beta_i(x)(k/N)^{2/n}$$
$$+\left[1+(1-4/n)(1/k-1/N)\right]\beta_i^2(x)(k/N)^{4/n} \tag{47}$$

We now examine the behavior of the Parzen classifier when the size of the kernel is chosen adaptively, with h^n proportional to $1/p_i(x)$. In order to clearly exhibit the relationship between the two classifiers, we choose the particular functional form of h as

$$h = (1 + n/2)^{-1/2} \left[\ell/(\alpha_i N p_i(x))\right]^{1/n} \tag{48}$$

where the parameter ℓ (as with k for the k-NN classifier) is in general a function of N to be chosen by the designer. Substituting (48) into (26) and (27) gives the result for the Parzen classifier.

$$E\left\{\frac{\Delta p_i(x)}{p_i(x)}\right\} \cong \beta_i(x)(\ell/N)^{2/n} \tag{49}$$

$$E\left\{\left(\frac{\Delta p_i(x)}{p_i(x)}\right)^2\right\} \cong \left[1/\ell - 1/N\right]$$

$$+ (1/\ell - 2/N)\beta_i(x)(\ell/N)^{2/n}$$

$$+ [1 - 1/N]\beta_i^2(x)(\ell/N)^{4/n} \tag{50}$$

Comparison of (46) and (47) with (49) and (50) shows that the behavior of the density estimates is very similar, not only in its behavior with N and h, but also in absolute value. There are slight differences between (47) and (50), which may be due in part to the different approximations for u_i used in the two derivations. These results, when substituted into (10), (11), and (8) confirm the often made claim that the k-NN classifier gives similar performance to the Parzen classifier with adaptable h, at least under the large N, large k approximations used here.

5.5. Estimation of the Bayes Error

Nonparametric classifiers are often used to provide estimates of or bounds on the Bayes error, ϵ^*. Sections 5.3 and 5.4 relate the performance of these classifiers under finite design set conditions to the true Bayes error. In this section, we utilize these results to provide a method of estimating the Bayes error from the observed error rates for these classifiers.

The procedure for the Parzen classifier is as follows. We first rewrite the expected error rate in terms of h for a fixed value of t and N by combining (28), (29) and (8):

$$E\{\epsilon\} = \epsilon^* + b_1h^2 + b_2h^4 + b_3h^{-n} + b_4h^{-n+2} \tag{51}$$

Here, the constants b_1,b_2,b_3,b_4, and the desired value of ϵ^* are unknown and must be determined experimentally. An estimate of ϵ^* may be obtained by observing the Parzen error rate for a variety of values of h, and finding the set of constants which best fit the observed data. Any data fitting technique could be used, however the linear least squares approach is straightforward and easy to implement.

This approach has several intuitive advantages over the common procedure of accepting the lowest error rate over the various values of h. First, it provides a direct estimate of ϵ^* rather than simply an upper bound on the value. We have found in many cases that the bounds provided by Parzen or k-NN procedures may be biased far away from the true Bayes error. Another advantage is that this procedure provides a means of combining the observed error rates for a variety of values of h. Hence we may be utilizing certain information concerning the higher order properties of the distributions which is ignored by the previous procedures.

As mentioned earlier, it is reasonable to expect that all five of the constants (51) are positive since the observed error must remain above ϵ^* for any value of h. In order to ensure stability in the estimate of ϵ^*, we have found it advisable to restrict the constants to positive values during the curve fit procedure.

Our procedure is illustrated in Figure 5.2. Here the N=100 data of Figure 5.1 has been replotted, and the best fit of the form given in (51) has been drawn as a solid line. The resulting estimate of $\hat{\epsilon}^* = 1.96\%$ is extremely close to the true value of ϵ^* of 1.9%. Note the closeness of the fit, indicating that the observed error rates are in fact following the trends predicted in this chapter.

An identical procedure may be applied to the k-NN error estimates by combining (44), (45) and (8)

$$\begin{aligned} E\{\epsilon\} = \epsilon^* &+ b_1((N-k+1)/(N(k-2))) + b_2g(N,k,2/n) \\ &+ b_3g^2(N,k,2/n) + b_4((k-1)/(k-2+2/n))g(N,k,2/n) \\ &+ b_5((k-1)/(k-2+4/n))g(N,k,4/n) \end{aligned} \tag{52}$$

For the k-NN estimate, we observe the error rate for various values of k, and then solve for the best set of positive constants in (52) to fit the

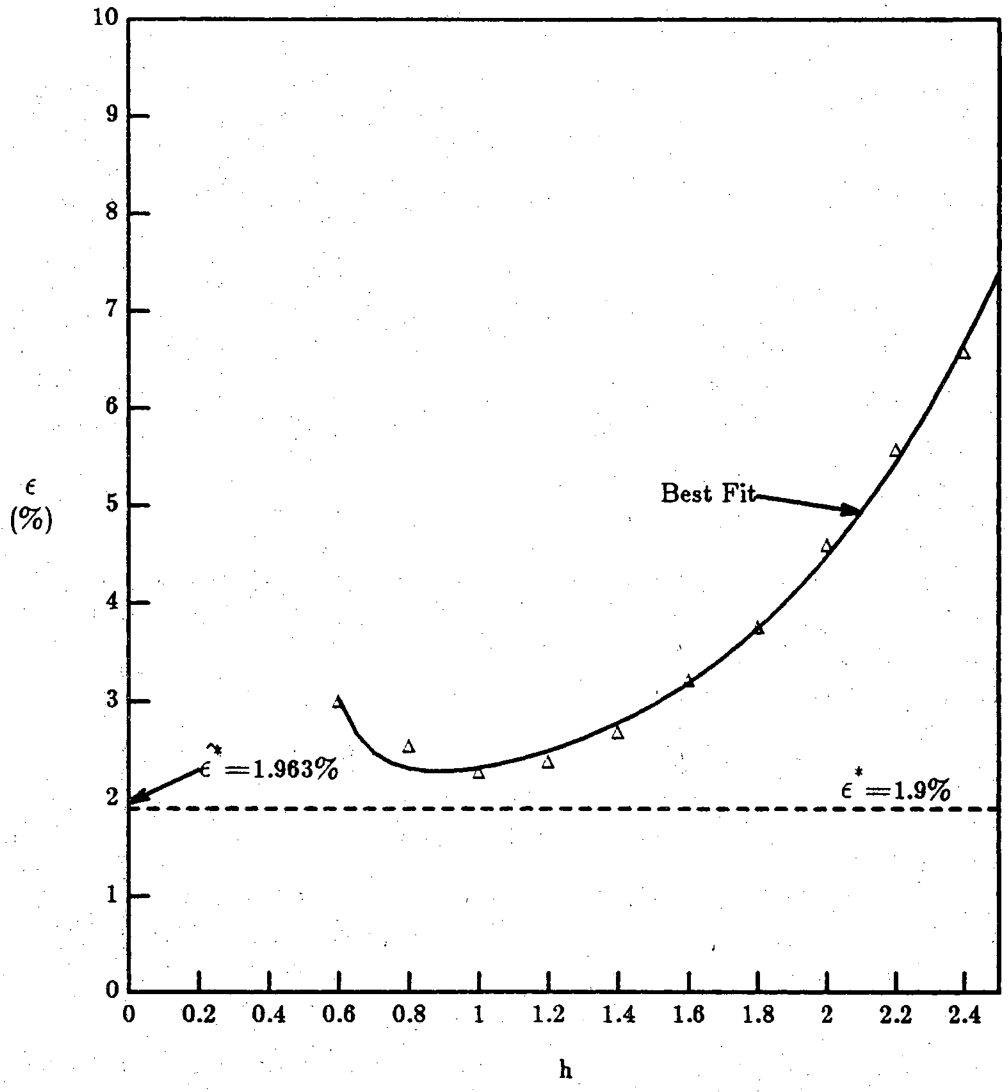


Figure 5.2 Estimation of ϵ^* from observed Parzen error rates.

observations. Figure 5.3 shows the results of this procedure applied to the k-NN error rates for the same data as was used in Figure 5.2. Note that a reasonable estimate of ϵ^* is obtained even though the observed error rates at this sample size (N=100) are well above the true Bayes error.

5.6 Selection of the Kernel Shape

While results are available regarding the selection of the distance metric for use with k-NN classifiers [12,13], little work has been done regarding the selection of the functional form of the kernel for the Parzen classifier under limited design set conditions. Our results using second order approximations of the density functions show that the performance of the Parzen classifier depends primarily on the covariance of the kernel function, Σ_{k_i}. A common choice is to select Σ_{k_i} equal to the ω_i covariance matrix Σ_i. While this choice ensures that the resulting density estimate has second order moments which are proportional to the true second order statistics, nothing is implied concerning the resulting error rate. In this section, we use our error expression to suggest a possible modification of this procedure which results in a more stable error estimate for larger values of h.

From (28) and (29) we see that if the kernel covariance is selected such that $c_1(x) = c_2(x)$, then all terms which are independent of the sample size may be eliminated from the error expression. Hence we must find positive definite matrices Σ_{k_1} and Σ_{k_2} such that

$$\mathrm{tr}\left(\frac{\nabla^2 p_1(x)}{p_1(x)}\Sigma_{k_1}\right) = \mathrm{tr}\left(\frac{\nabla^2 p_2(x)}{p_2(x)}\Sigma_{k_2}\right) \tag{53}$$

In general, however, the second order properties of the density are not known. When the data is Gaussian, however, we may obtain a solution of (53) in terms of the mean vectors and covariance matrices.

When $p_i(x)$ is Gaussian, then

$$\frac{\nabla^2 p_i(x)}{p_i(x)} = \Sigma_i^{-1}(x - M_i)(x - M_i)^T\Sigma_i^{-1} - \Sigma_i^{-1} . \tag{54}$$

To solve (53), we begin by proposing a solution of the form

$$\Sigma_{k_i} = \Sigma_i + \gamma_i(x - M_i)(x - M_i)^T \tag{55}$$

We wish to determine the possible values of γ_1 and γ_2 so that (53) holds. Substituting (54) and (55) into (53) and simplifying gives

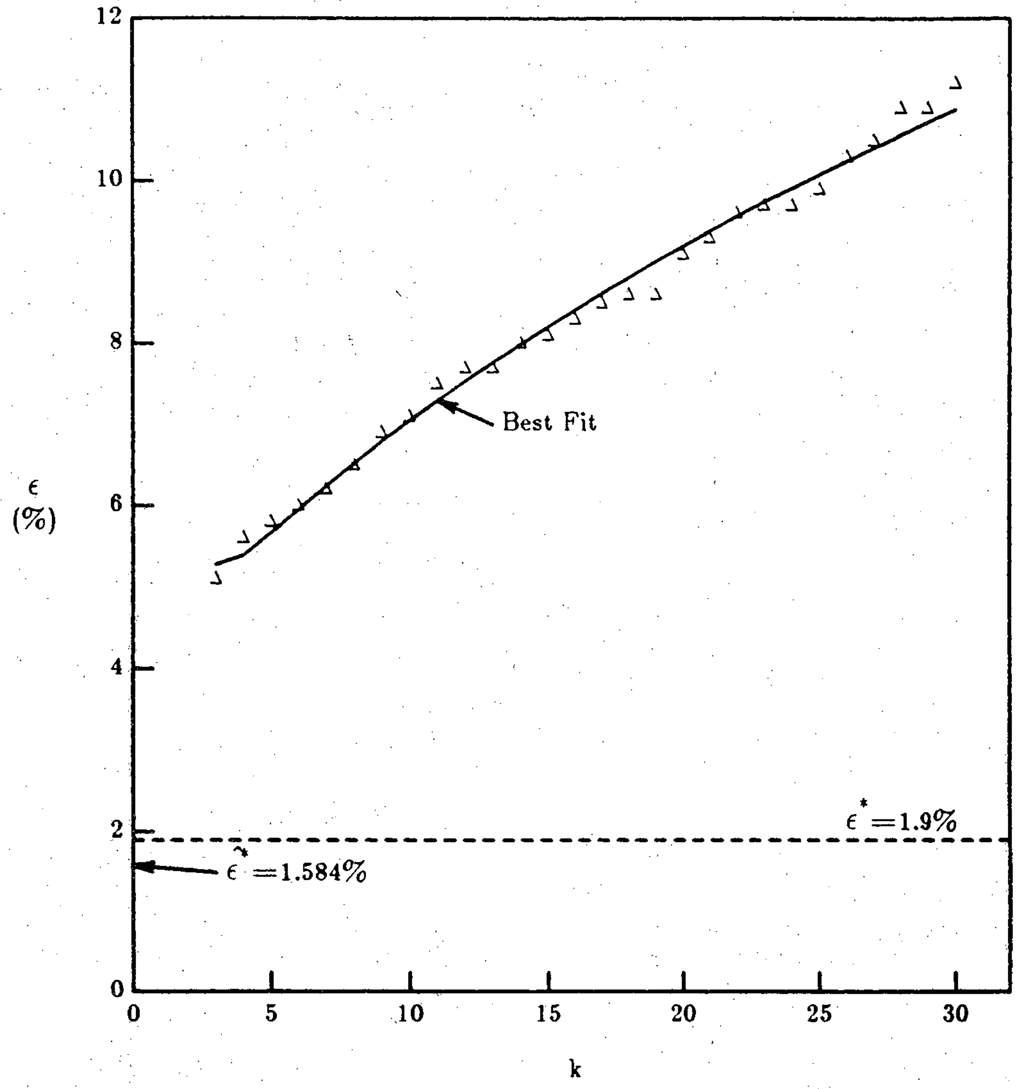


Figure 5.3 Estimation of ϵ^* from observed k-NN error rates.

$$y_1(d_1^2(x,M_1) - 1) - y_2(d_2^2(x,M_2) - 1) = d_2^2(x,M_2) - d_1^2(x,M_1) \quad (56)$$

where y_i is given by $\gamma_i d_i^2(x,M_i)$, and $d_i(x,Y)$ is as given in (21). In addition to satisfying (56), we must also choose $y_i > -1$ to ensure that $\sum_{k_i}$ of (55) remains positive definite. Hence valid choices for y_1 and y_2 lie in the region $y_1 > -1$, $y_2 > -1$, and satisfy the linear equation (56). This is illustrated in Figure 5.4. Equation (56) describes a line passing through the point $(y_1,y_2) = (-1,-1)$ with slope $(d_1^2(x,M_1) - 1)/(d_2^2(x,M_2) - 1)$. Hence, valid positive definite solutions exist provided this slope is positive. For practical purposes, it is convenient to use a fixed set of values for y_1 and y_2. In this case, a near solution may be used by choosing (y_1,y_2) close to the $(-1,-1)$ point. Such a choice comes close to satisfying (53) even in the case in which the slope of (56) is negative.

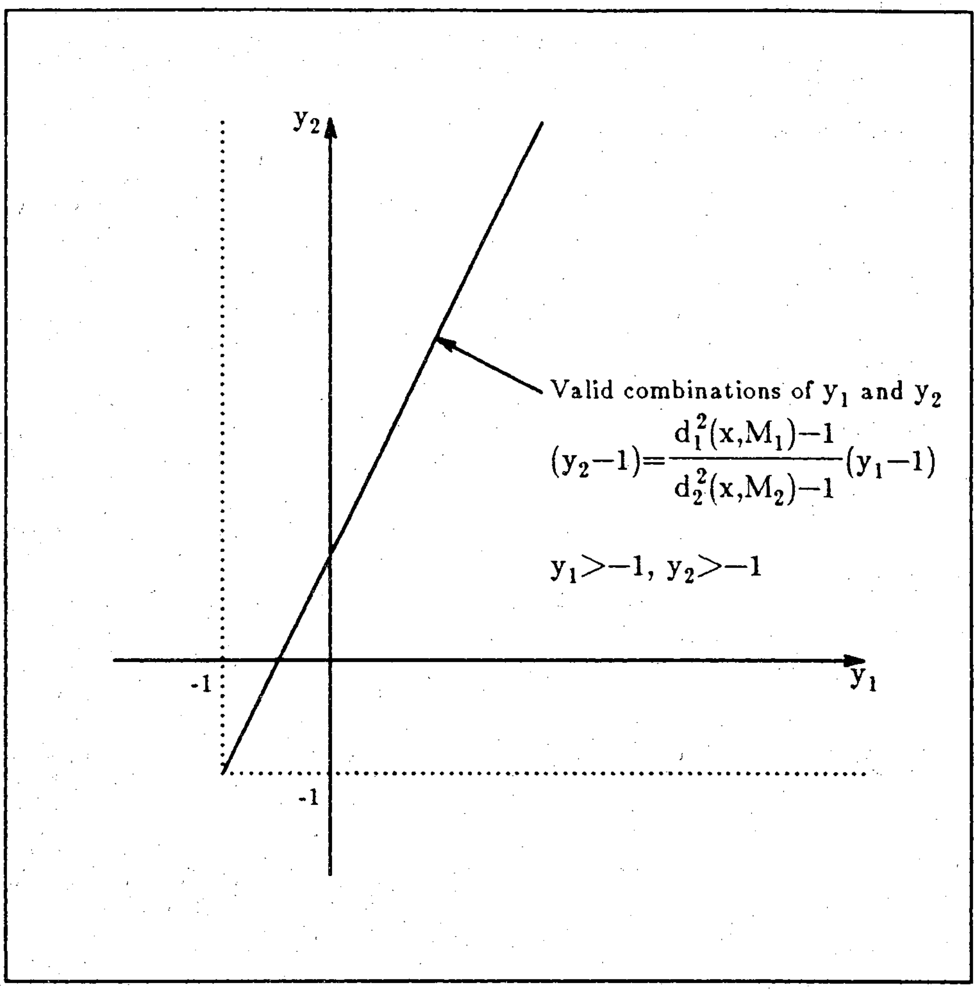
The data used in Figure 5.1 was tested using the kernel given by (55) using N=100 design samples per class. The results using $y_1 = y_2 = -0.8$ are shown in Figure 5.5. These results indicate that although the estimates seem less stable at smaller values of h, as h grows the results using (55) remain close to the Bayes error while the results using $\sum_{k_i} = \sum_i$ degrade rapidly. This implies that the h^2 and h^4 terms of (28) and (29) have been effectively eliminated.

These experimental results indicate the potential importance of the kernel covariance in designing Parzen classifiers.

5.7 Conclusion

This chapter has developed relationships between the expected error rates of nonparametric classifiers and the true Bayes error. These relationships explicitly show how the error performance changes as the size of the design set is varied, how the value of h affects the performance of the Parzen classifier, and how k affects the performance of the k-NN classifier. The results are related directly to the Bayes error, in contrast to Chapter 2 which relates the 1-NN and 2-NN performance to their asymptotic values.

Two direct applications of these results have been presented in Sections 5.5 and 5.6. The first is a curve fit technique which utilizes the observed error rates for many values of h or k to obtain an estimate of the true value of the Bayes error. The experimental results indicate that this value may be a significant improvement over the bounds provided by the observed error rates. The second application involves an improvement of the performance



y_2
Valid combinations of y_1 and y_2
$(y_2-1)=\frac{d_1^2(x,M_1)-1}{d_2^2(x,M_2)-1}(y_1-1)$
$y_1>-1,\ y_2>-1$
-1
y_1
-1

Figure 5.4 Valid combinations of y_1 and y_2

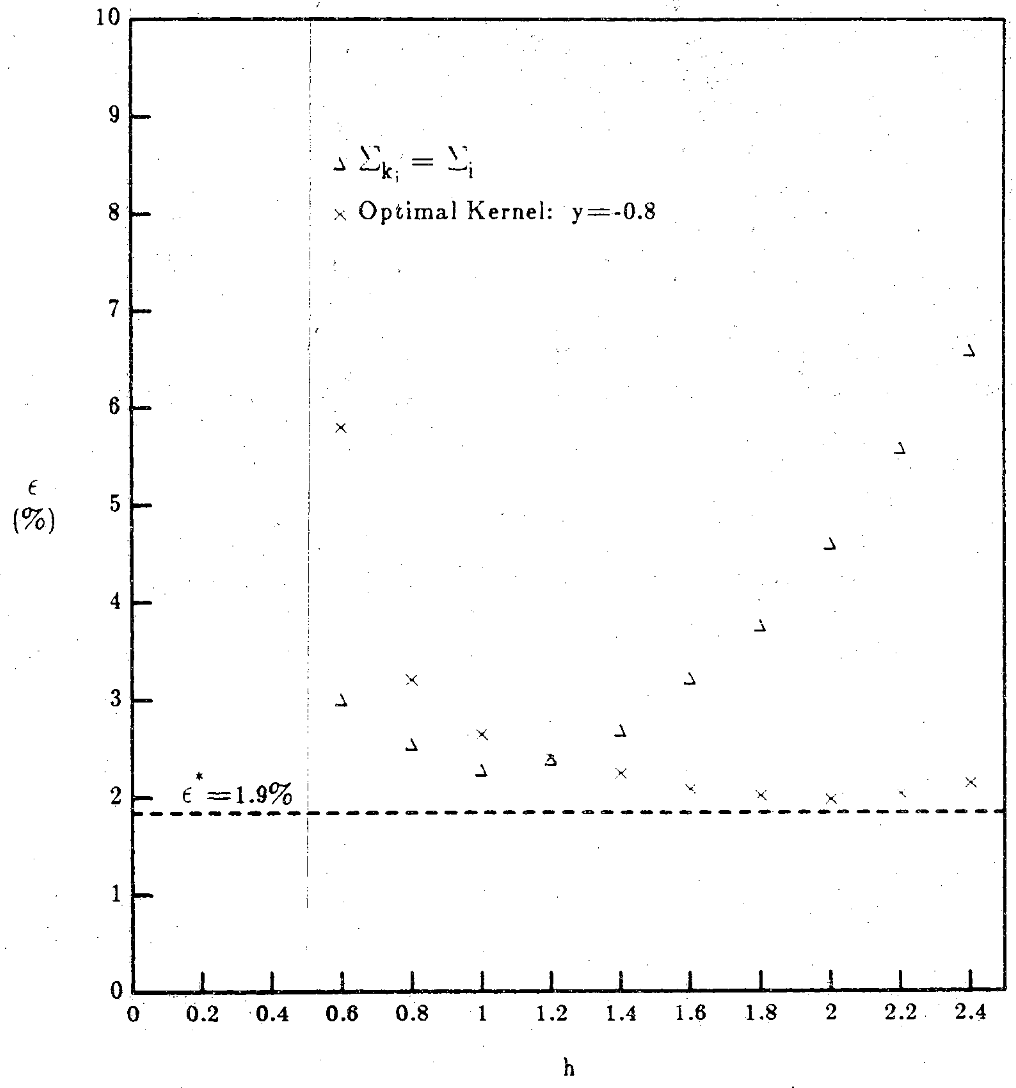


Figure 5.5 Parzen error rates for two different kernel functions.

of the Parzen classifier through the selection of the kernel covariance. Again, the experimental results indicate that a significant improvement may be possible, particularly for larger values of h.

Aside from the practical contributions, this chapter has also given some theoretical justification to the results concerning the decision threshold presented in Chapter 3. Careful selection of the decision threshold should play a key role in the design of nonparametric classifiers.

CHAPTER 6 BOOTSTRAP METHODS FOR NONPARAMETRIC ERROR ESTIMATES

6.1 Introduction

Since its introduction in 1977 [48], bootstrap techniques have been applied to a wide variety of estimation problems. Particular attention has been paid to the problem of estimating the error rate for various classifiers designed using a small number of preclassified samples. Traditionally, these error rates have been estimated using the apparent (resubstitution) error and cross-validation (leave-one-out) type estimates. The bootstrap has been proposed as a means of correcting for the negative bias of the resubstitution error, resulting in an estimate with lower variance than the leave-one-out error [49].

Bootstrap procedures refer to a general class of techniques which resample the given data in order to induce information about the sampling distribution of an estimator. As long as the resampling procedures are similar enough to the original sampling procedure, the bootstrap allows one to determine various statistical properties of an estimator even when very little is known about the underlying distributions. The choice of an appropriate resampling procedure is crucial to obtain accurate estimates of the true sampling statistics. Normally, the resampling is performed by drawing samples from the empirical distribution (that is, from a distribution with probability mass 1/N at each of the original N design samples). Many authors have found this to be an acceptable procedure for a wide variety of estimation problems [48-51]. However, all of these authors point out possible drawbacks of this procedure for some estimates due to the discrete nature of the empirical distribution. These problems are clearly manifested when the bootstrap is applied to estimate the error for nonparametric (e.g. Parzen or k-NN) classifiers. This is the case to be investigated in this chapter. Here, it is important that the samples which are drawn in the resampling procedure are different from those samples being tested in order that the sampling statistics of the bootstrap sample not be biased away from the true sampling statistics.

Several methods have been proposed in order to satisfy this requirement. Chernik *et al.* [50,51] perform experiments in which the bootstrap samples ar convex combinations of the original N samples. While this procedure does remove the bootstrap samples from the design set, it places convexity assumptions on the underlying distributions which may not be valid. This is particularly a problem when the technique is used to investigate nonparametric procedures, which are usually used to avoid making assumption about the distributions. Further, the experimental results in [50,51] suggest that this procedure is inferior to several others. Another possible procedure is to use the $\epsilon^{(0)}$ estimate of Efron [49]. The $\epsilon^{(0)}$ estimate is obtained by resampling the data from the empirical distribution to obtain the bootstrap samples, and then testing only those samples of the original data which are not part of the bootstrap sample set. The $\epsilon^{(0)}$ error estimate is the average error among these left out samples over many bootstrap trials. Efron introduces the $\epsilon^{(0)}$ estimate only as a heuristic motivation for his ".632" estimate, claiming that the $\epsilon^{(0)}$ error should be biased high since the samples being tested are "too far" from the design set. Jain *et al.* [52] investigate the use of the $\epsilon^{(0)}$ estimate in evaluating the error for a 1-NN classifier. They found that although a slight positive bias is observed, use of the $\epsilon^{(0)}$ estimate gave tighter confidence regions than did the standard leave-one-out or resubstitution errors.

In this chapter, we investigate a third alternative. Under our approach, the bootstrap samples are drawn not from the empirical distributions, but from a smoothed estimate of the true distribution. The degree of smoothing is chosen so that the leave-one-out error for the bootstrap sample is roughly equivalent to the observed error for the original data. The bootstrap is then used to form an estimate of the standard deviation of the actual leave-one-out error. This procedure is somewhat different from the conventional use of the bootstrap. In the past, most authors have used the bootstrap a a means of estimating and correcting for the negative bias which is observed in the apparent, or resubstitution, error. Here, instead of trying to estimate the bias of the resubstitution error, we are interested in estimating the standard deviation of our observed leave-one-out error, gaining an indication of the reliability of our Bayes error estimates for the particular distributions being considered.

Section 6.2 formally introduces the procedures used here, and Section 6.3 presents some experimental results for several test cases. Although the emphasis in this treatment is on the estimation of the variance for the Parzen classifier, application of the same techniques to the k-NN classifiers is straightforward.

6.2 The Bootstrap Procedure

We now adopt the notation of Efron [49]. We assume that the classifier is designed based on a set of N training vectors $\mathbf{x} = \{x_1, \cdots, x_N\}$ where each x_i represents a pair (t_i,y_i) consisting of a n-dimensional feature vector t_i and the known class y_i from which t_i was drawn. The prediction rule $\eta(t,\mathbf{x})$ is constructed which assign an unknown vector t to some class based on the training set $\mathbf{x}$. In this treatment, the prediction rule $\eta(t,\mathbf{x})$ will be the Parzen or k-NN classifier designed from $\mathbf{x}$, the set of preclassified samples. We define an indicator function $Q(y,\eta(t,\mathbf{x}))$ to be 0 of the decision rule $\eta(t,\mathbf{x})$ correctly assigns t to class y, and 1 otherwise.

$$Q(y,\eta(t,\mathbf{x})) = \begin{cases} 0 & \text{if } \eta(t,\mathbf{x}) = y \\ 1 & \text{if } \eta(t,\mathbf{x}) \neq y \end{cases} \tag{1}$$

We are interested in estimating the actual error rate, Err, which is encountered when new samples independent of $\mathbf{x}$ are encountered. Thus

$$\mathrm{Err} = E\left\{ Q(y_0, \eta(t_0,\mathbf{x})) \right\} \tag{2}$$

where $x_0 = (t_0,y_0)$ is a randomly selected independent test sample. There are two widely used estimates of Err: the apparent (resubstitution) error and the cross-validation (leave-one-out) error. The apparent error App is found by reclassifying each of the N samples, counting the number of misclassifications, and dividing by the total number of samples tested

$$\mathrm{App} = \frac{1}{N}\sum_{i=1}^{N} Q(y_i, \eta(t_i,\mathbf{x})). \tag{3}$$

Since the samples being tested are included in the design set $\mathbf{x}$, the apparent error tends to be lower that the true value of Err. This is particularly true for the nonparametric classifiers considered in this treatment which are highly dependent on the location of individual samples in the design set. In order to help eliminate the negative bias associated with the apparent error, cross-validation estimates may be used. Under this procedure, each sample in the design set is tested using a decision rule which is designed based on the remaining N-1 samples. The leave-one-out error, $\mathrm{Err}^{(CV)}$, is found by

$$\mathrm{Err}^{(CV)} = \frac{1}{N}\sum_{i=1}^{N} Q(y_i, \eta(t_i, \mathbf{x}_i)). \tag{4}$$

where $\mathbf{x}_i = \mathbf{x} - \{x_i\}$ is the original design set with the sample $x_i = (t_i, y_i)$ removed.

While $\mathrm{Err}^{(CV)}$ provides the designer with a nearly unbiased estimate of the true error, in general the standard deviation of the estimate is not known. In this chapter, we introduce a bootstrap technique to estimate this standard deviation from the given set of design samples. The procedure is as follows. From the given design samples x, we first form an estimate of the unknown underlying density, $\hat{F}$. Normally, the empirical distribution is used

$$\hat{F}\text{: mass } 1/N \text{ on } x_i, \quad i=1,2,\ldots,N \tag{5}$$

However, we will find that for k-NN and Parzen classifiers it is necessary to use a smoothed estimate of the density. From this estimate we repeatedly perform the following procedure:

1) Draw N "bootstrap samples" $x_1^{*(j)}, x_2^{*(j)}, \ldots, x_N^{*(j)}$ independently from the estimated distribution $\hat{F}$. We denote this sample set $\mathbf{x}^{*(j)}$.

2) Find and tabulate the leave-one-out error for the sample set $\mathbf{x}^{*(j)}$:

$$\mathrm{Err}_j^{(CV)*} = \frac{1}{N}\sum_{i=1}^{N} Q(y_i^{*(j)}, \eta(t_i^{*(j)}, \mathbf{x}_i^{*(j)})) \tag{6}$$

3) Repeat steps 1 and 2 for $j=1,2,\ldots,B$, where B is the number of bootstrap trials desired. The estimate of the standard deviation of $\mathrm{Err}^{(CV)}$ may now be formed by calculating the sample standard deviation over the B values of $\mathrm{Err}_j^{(CV)*}$.

If the resampling procedure is similar enough to the true sampling conditions (that is, if the density estimate $\hat{F}$ is close enough to the true density F) then the statistics of $\mathrm{Err}_j^{(CV)*}$, $j=1,2,\ldots,B$, should be close to the actual sampling statistics of $\mathrm{Err}^{(CV)}$.

The appropriate value of B depends largely on the sampling statistic to be estimated. In the past, authors have concentrated on estimating the entire sampling distribution of the estimated error rates, and hence have required B to be quite large. Jain *et al.* [52] claim that B should be greater than 100, while Efron [49] suggest that B need not be greater than 200. In this paper, the value of B is severely limited by computational concerns.

The nonparametric procedures considered here already require a large amount of computer time, and the bootstrap requires that the procedures be repeated B times. Because of this restriction, we only concentrate here on estimating the standard deviation of the error estimates (as opposed to the entire sampling distribution), and hence use a fairly small value of B (B=10 for the experiments presented in Section 6.3).

Application of the bootstrap is complicated by the fact that we are considering error rates of Parzen and k-NN classifiers. For these classifiers, use of $\hat{F}$ as given in (5) will result in values of $Err_j^{(CV)*}$ far below the values of $Err^{(CV)}$ obtained from the true distribution. This fact is clearly demonstrated in Table 6.1. Columns 2 and 3 of Table 6.1 show the average leave-one-out and the corresponding standard deviation for a Parzen classifier applied to a variety of distributions. In each case, 8 dimensional Gaussian distributions were used with covariance matrices $\Sigma_1=\Sigma_2=I$ and mean vectors separated to give the indicated Bayes error. All results are averaged over ten independent trials. Columns 4 and 5 of Table 6.1 show the average predicted values for the same experiments using the bootstrap procedures (B=10) and $\hat{F}$ as given in (5). These predictions are also averaged over the ten trials. The bootstrap predictions are clearly biased below the true sampling statistics of $Err^{(CV)}$, indicating that $\hat{F}$ is not an adequate estimate of the distribution. This bias is a result of the fact that that many of the bootstrap samples $x_i^{*(j)}$ appear multiple times in the bootstrap sample set $x^{*(j)}$. Samples which appear more than once will almost never be misclassified by the Parzen or k-NN classifier, and hence the leave-one-out error for the bootstrap sample is biased below the actual leave-one-out error. (This is most apparent when one considers the 1-NN classifier. Here, if a sample is included more than once it is always considered as its own nearest neighbor, and is hence never misclassified.)

This problem may be eliminated by choosing a smooth estimate of F rather than the empirical distribution given in (5). Toward this end, we use the Parzen density estimate for each class

$$\hat{F}(x_0|y_0) = \frac{1}{N_{y_0}} \sum_{y_i=y_0} (1/h_B^n)\,k_{y_0}((t_0-t_i)/h_B) \qquad (7)$$

where N_{y_0} is the number of design samples in **x** from y_0, $k_{y_0}(t_0)$ is a zero mean density function which may depend on y_0, and h_B is a parameter which determines the degree of smoothing. Note that as $h_B \rightarrow 0$, $(1/h_B^n)k_{y_0}((t_0-t_i)/h_B)$ approaches an impulse function and (7) becomes the

Table 6.1 Error rate statistics for the Parzen classifier using h=1.0 with 100 samples per class. Bootstrap samples were drawn from the empirical distribution.

Bayes Error	$\overline{\mathrm{Err}^{(CV)}}$	$\sigma^{(CV)}$	$\overline{\mathrm{Err}^{(CV)*}}$	$\sigma^{(CV)*}$
30%	34.2	3.9	17.1	2.5
20%	22.1	3.5	11.2	2.3
10%	10.9	2.4	5.8	1.7
5%	6.9	1.7	3.0	1.1
2%	2.2	1.0	1.1	0.8

empirical distribution given in (5). As h_B grows, the density estimate $\hat{F}(x_0|y_0)$ becomes smoother. To generate a sample $x_i^{*(j)}$ from the density of (7), one first randomly (all samples equally likely) selects a sample $x_k = (t_k, y_k)$ from x. A sample z is then generated (independent of t_k) from the scaled kernel density $(1/h_B^n)k_{y_k}(z/h_B)$. It may be verified that the random variable formed by $x_i^{*(j)} = (t_k + z, y_k)$ has the density given in (7). Since the density of z is continuous, no two samples of $x^{*(j)}$ will have the same value, and hence the normal problems associated with using the bootstrap on Parzen or k-NN error rates is eliminated. For convenience in the sample generation procedure, the kernel density $k_y(\cdot)$ was chosen to be a Gaussian density with covariance equal to the covariance for class y, Σ_y. The covariance of the scaled kernel density $(1/h_B^n)k_y(z/h_B)$ is then $h^2\Sigma_y$.

A key problem in the use of the above procedure is how to select the value of h_B to be used. It is desired that the value of the smoothing parameter h_B be selected so that the sampling statistics of $Err_j^{(CV)*}$ be as close as possible to those of $Err^{(CV)}$. For small values of h_B, the leave-one-out error of the bootstrap sample $Err_j^{(CV)*}$ is biased below that for the original design set, $Err^{(CV)}$. As h_B grows, however, the covariance associated with the generated samples grows and the leave-one-out error of the bootstrap samples becomes larger, eventually overtaking the leave-one-out error of the original data. A typical case is illustrated in Figure 6.1. We are interested in finding a value of h_B at which the sample statistics of $Err_j^{(CV)*}$ are similar to those of $Err^{(CV)}$. Hence, the proposed procedure is the measure the mean and variance of $Err^{(CV)*}$ for a variety of different values of h_B, and plot the mean value to obtain a graph similar to Figure 6.1. A reasonable value of h_B is then given by the value of h_B for which the mean value of $Err_j^{(CV)*}$ equals the measured value of $Err^{(CV)}$. The estimate of the standard deviation of $Err^{(CV)}$ is then the corresponding sample standard deviation of $Err_j^{(CV)*}$, $j=1,2,\ldots,B$, evaluated at this value of h_B.

6.3 Experimental Results

The procedure was tested using the Parzen classifier for two different test distributions. Both test cases are two-class problems with 8-dimensional Gaussian distributions. For test case 1, the class covariances are equal, and the mean vectors are separated to give a Bayes error of 10%. Test case 2 is a complex set of distributions in which the means and covariances are not equal, giving a Bayes error of 1.9%. The actual distribution parameters for

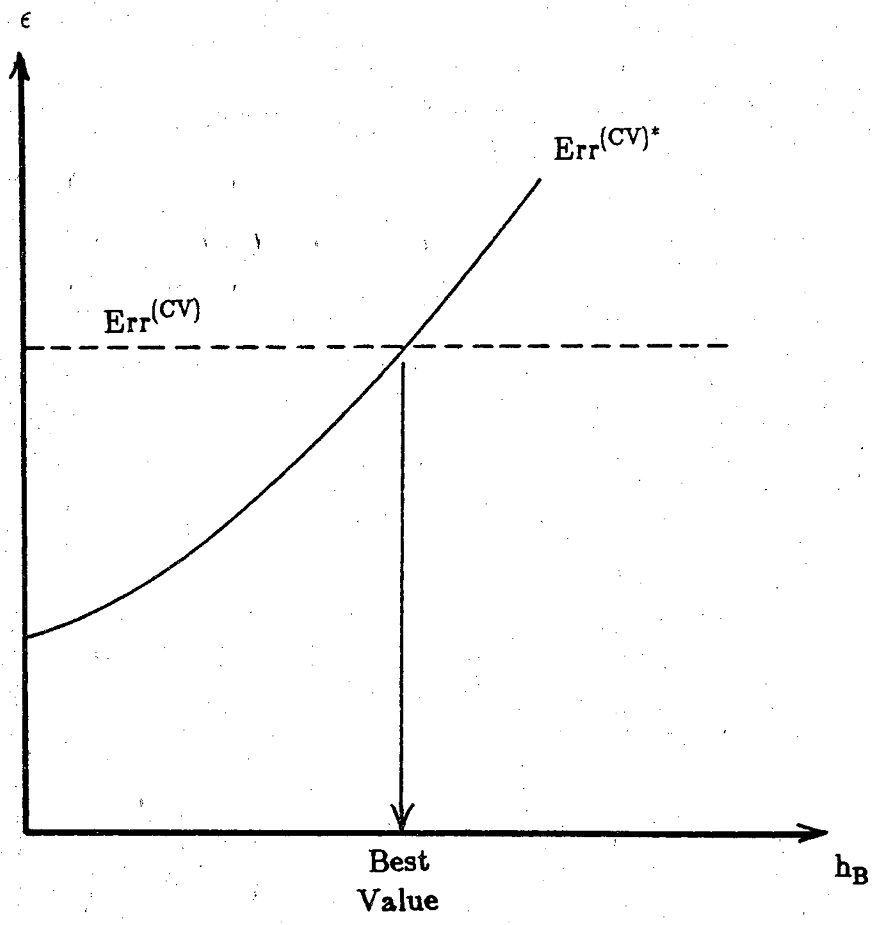


Figure 6.1 Typical trend of the bootstrap error $Err_j^{(CV)*}$ as the value of h_B increases.

test case 2 are shown in Tables 3.1 and 3.2 of Chapter 3. For each of the test cases, the Parzen smoothing parameter used to form the classifier, h_D, is allowed to vary from 0.6 to 3.0. The value of the smoothing parameter used to generate the bootstrap samples, h_B, was chosen so that the bootstrap average $Err^{(CV)*}$ was equal to the observed cross-validation error $Err^{(CV)}$ when $h_D = 1.0$. The resulting values were $h_B = 0.55$ for test case 1, and $h_B = 0.51$ for test case 2.

For each experiment, ten independent sets of samples with 100 samples per class were generated. For each set, $Err^{(CV)}$ and App were determined. The statistics (mean and standard deviation) of these values were calculated over the ten trials and serve as a reference in our evaluation of the bootstrap procedures. For each set of samples, the bootstrap procedure of Section 6.2 was applied (using $B = 10$) forming bootstrap estimates of the standard deviation of $Err^{(CV)}$ and App. These estimates are then averaged over the ten trials.

The results for test case 1 are shown in Figures 6.2 and 6.3. Figure 6.2 shows the mean value of $Err^{(CV)}$ and App over the ten independent trials for various values of h_D. The average predicted values of these variables using the bootstrap sample sets (generate with $h_B = 0.55$) are also shown. Ideally, if the $\hat{F}$ of (7) was close to the true distribution F, these ,curves would be identical. Figure 3 shows the calculated standard deviations of $Err^{(CV)}$ and App, as well as the average predicted standard deviations given by the bootstrap procedure. Similar results for test case 2 are shown in Figures 6.4 and 6.5.

Our results indicate a close correspondence between the actual standard deviations and the corresponding bootstrap predictions for both test cases. This relationship exists not only for $h_D = 1.0$, for which the value of h_B was selected, but also for the entire range of h_D from 0.6 to 3.0. This indicates that the sampling statistics of our bootstrap sample sets are in fact very similar to those for the true distributions, and gives us confidence in the estimates. For test case 1, the values of $Err^{(CV)*}$ and $Err^{(CV)}$ begin to diverge for large values of h_D in Figure 2. This indicates that at these values of h_D, it may be wise to employ a different value of h_B. However, even for these values, the bootstrap estimates of the standard deviations are very close to our estimated values.

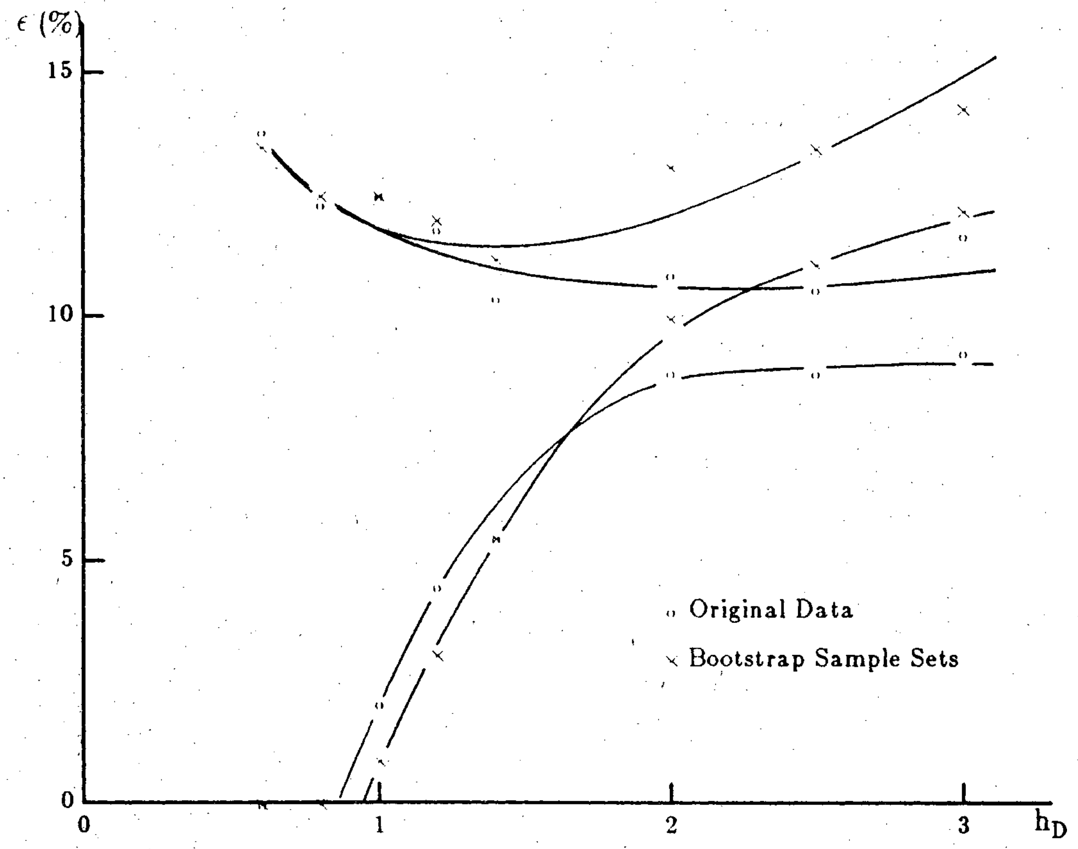


Figure 6.2 Results for test case 1. Average leave-one-out and resubstitution error rates for the original data and the generated bootstrap sample sets.

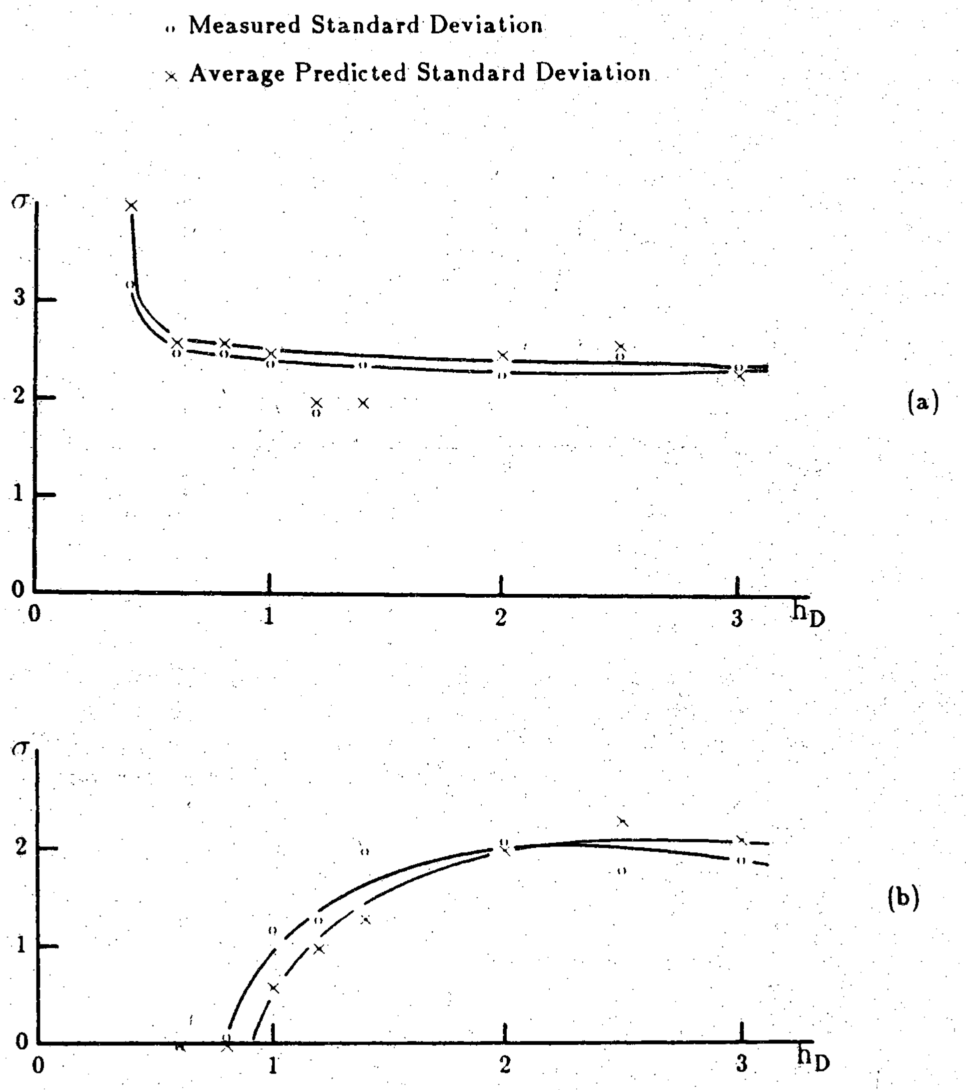


Figure 6.3 Results for test case 1. Standard deviation of the leave-one-out (a), and resubstitution (b), error rates for the original data and average predicted standard deviation from the the generated bootstrap sample sets.

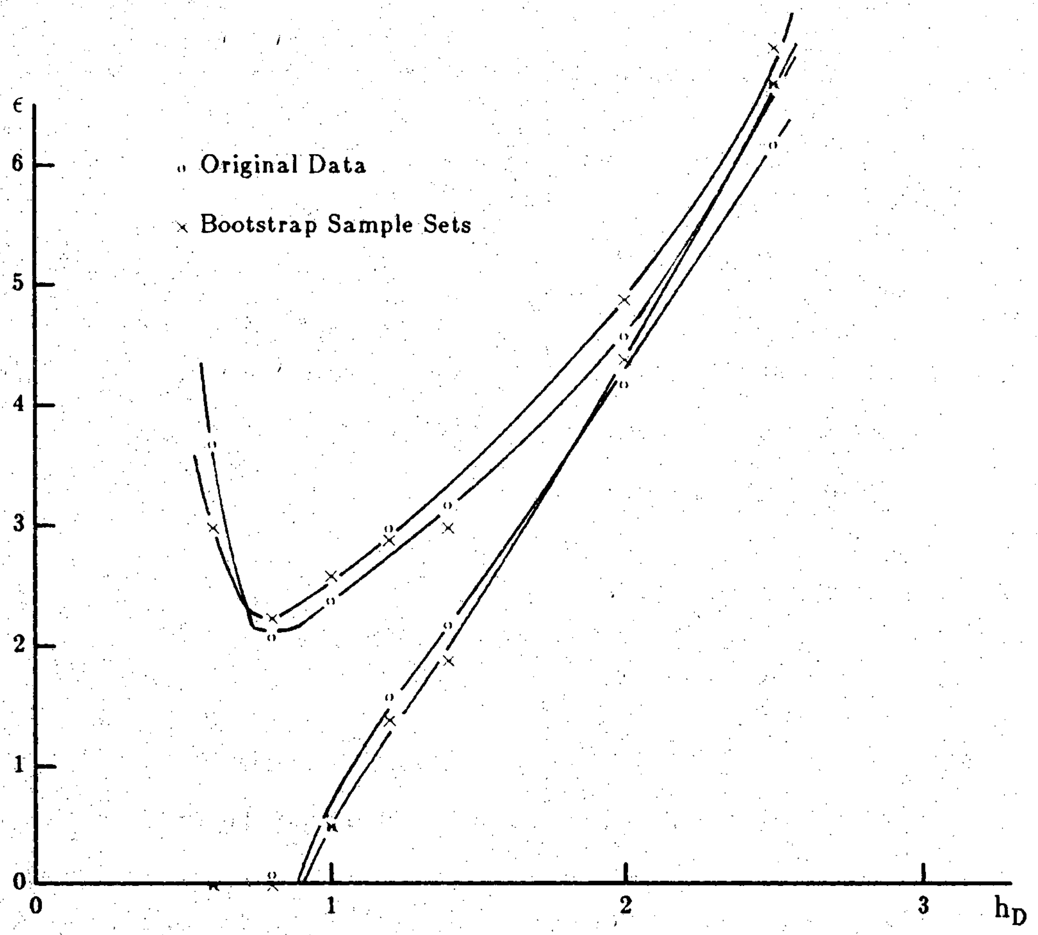


Figure 6.4 Results for test case 2. Average leave-one-out and resubstitution error rates for the original data and the generated bootstrap sample sets.

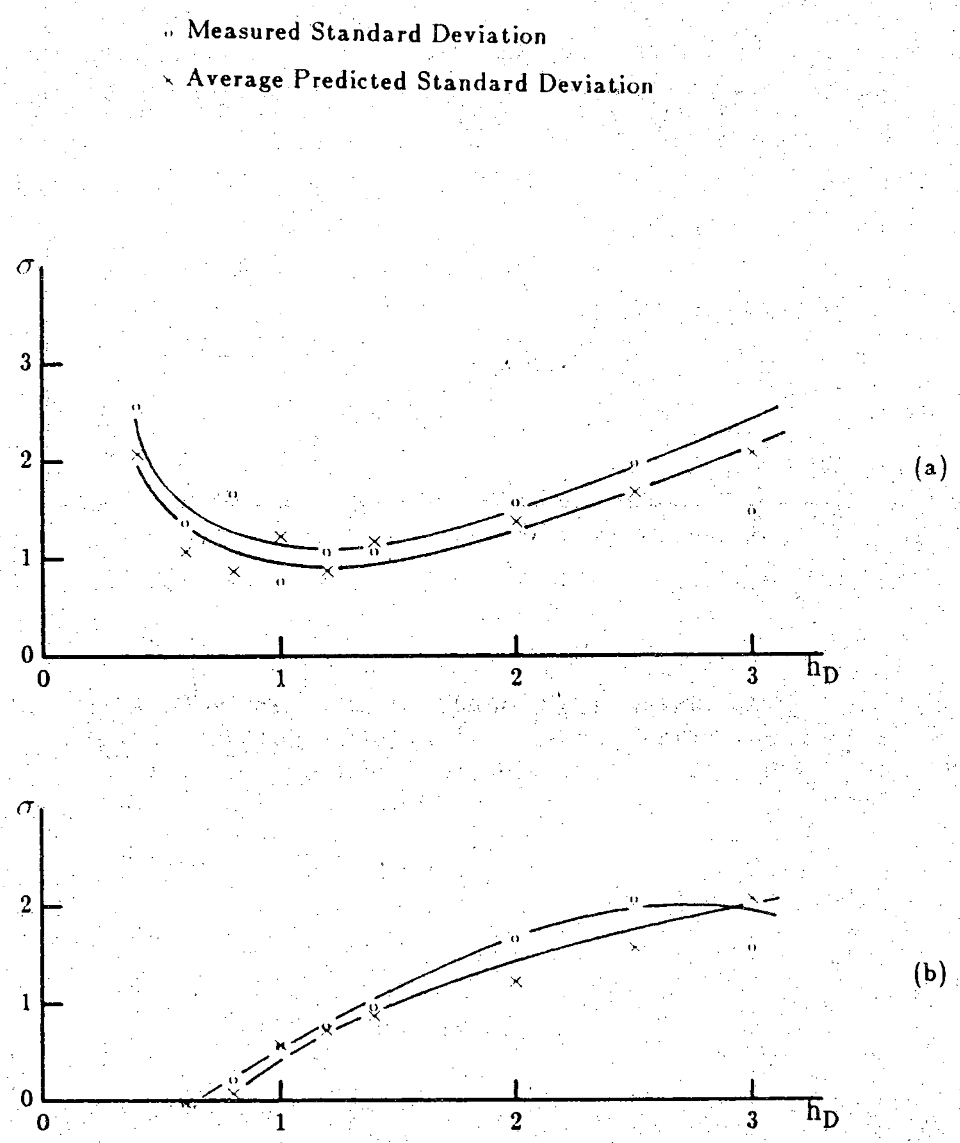


Figure 6.5 Results for test case 2. Standard deviation of the leave-one-out (a), and resubstitution (b), error rates for the original data and average predicted standard deviation from the the generated bootstrap sample sets.

6.4 Summary

This chapter has provided a means by which the standard deviation of an error estimate may be obtained from a single set of design samples. Our experiments indicate that a suitably modified bootstrap procedure provides a nearly unbiased estimate of the variances of the cross-validation and apparent error rates. It should be noted that although our experiments have concentrated on the Parzen classifier, and identical bootstrap procedure may be applied to k-NN classifiers. The approaches indicated in this chapter seem most appropriate for use with nonparametric classifiers, since most investigations have shown that for parametric classifiers the empirical distribution is an adequate estimate of F.

As a bonus, we have uncovered a possible means of generating artificial samples from a limited number of available design samples. Our procedure involves first finding the value of h_B to match a test statistic of the artificially generated samples to the observed statistic for the design set. Once h_B has been determined, one is free to generate an unlimited number of samples and be assured that in some sense, the statistical behavior of the generated samples will be close to that of the original samples. This procedure may have widespread application in many areas of pattern recognition, and is a possible subject of further investigation.

CHAPTER 7
SUMMARY AND RECOMMENDATIONS

7.1 Summary of Results

This thesis has examined the expected error of nonparametric classifiers and applied these results to the estimation of the Bayes classification error rate. Chapter 2 provides an analysis of 1-NN and 2-NN classifiers, isolating the effects of sample size, dimension, and distance metric from those of the distributions. These results are then used to provide an improved estimate of the asymptotic 1-NN and 2-NN error rates, which place bounds on the Bayes error. The Parzen classifier was examined in Chapter 3, and the k-NN classifier was reframed into an analogous density estimation viewpoint. Guidelines are developed regarding the selection of the kernel size and shape, the selection of k, and the selection of the k-NN distance metric. The most important discovery of Chapter 3 is the potential role that the decision threshold plays in compensating for the biases of the density estimates. These discoveries are shown to significantly improve the conventional procedures now in use.

A computationally efficient method of estimating the kernel covariance for leave-one-out estimates is presented in Chapter 4. In Chapter 5 we use the density estimation framework developed in Chapter 3 to derive the expected error performance of both Parzen and k-NN ($k>2$) classifiers. These results give explicit functional forms of the observed error rate in terms of the sample size, kernel size and shape, k, and the NN distance metric. These expressions relate the observed error rate directly to the Bayes error, and provide a means of estimating (rather than bounding) that value. The optimal selection of the kernel covariance is also discussed in Chapter 5. Finally, a bootstrap procedure has been developed which allows the designer to estimate the standard deviation of a nonparametric estimate of the Bayes error.

7.2 Recommendations for Further Research

There are several topics related to this thesis which merit further investigation.

It should be possible to extend the results of Chapter 5 to include resubstitution error rates, by taking into account the biases associated with the resubstitution density estimates. While the justification is strong for using resubstitution k-NN error rates as lower bounds on the Bayes error, analogous results have not been found regarding resubstitution Parzen error rates. Such an approach could provide some theoretical insight.

One problem associated with the algorithms presented in this thesis is the large computational burden which is assumed when the procedures are implemented. To some extent, this thesis has helped to alleviate some of this burden by reducing the number of samples which are required to obtain reliable estimates. However, computational concerns remain an important factor in high dimensional spaces, where the number of samples required remains quite large. Useful contributions could be made in this area, either by finding efficient algorithms of finding the k-NN and Parzen errors directly, or by finding methods of estimating the respective error rates without actually designing and implementing the classifier.

Finally, this thesis has not treated the class of nonparametric procedures based on the estimation of the density functions using orthogonal series. Application of these techniques in high dimensional spaces is often difficult in practice due to the large number of series coefficients which must be determined. However, several authors have obtained theoretical results which indicate that the rate of convergence of density estimates based on orthogonal series is independent of the dimension of the space. This is in sharp contrast to the results for both k-NN and Parzen estimates. An open question is the relationship between the orthogonal series estimates and the estimates presented in this thesis. Practical application of these methods to high dimensional problems is contingent on finding a means of keeping the required number of series coefficients reasonable, and this is also an area in which significant contributions could be made.

LIST OF REFERENCES

LIST OF REFERENCES

[1] T. M. Cover and P. E Hart, "Nearest neighbor pattern classification," IEEE Trans. Inform. Theory, Vol. IT-13, pp. 21-27, Jan. 1967.

[2] T. M. Cover, "Rates of convergence for nearest neighbor procedures," in Proc. Hawaii Int. Conf. on System Sciences, pp. 413-415, 1968.

[3] T. J. Wagner, "Convergence of the nearest neighbor rule," IEEE Trans. Inform. Theory, Vol. IT-17, pp. 566-571, 1971.

[4] J. Fritz, "Distribution-free exponential error bounds for nearest neighbor pattern classification," IEEE Trans. Inform. Theory, Vol. IT-21, pp. 552-557, 1975.

[5] C. J. Stone, "Consistent nonparametric regression," Ann. Statistics, Vol. 5, pp. 595-645, 1977.

[6] L. Devroye, "On the almost everywhere convergence of nonparametric regression function estimates". Ann. Statistics, Vol. 9, pp. 1310-1319, 1981.

[7] L. Devroye, "On the inequality of Cover and Hart in nearest neighbor discrimination," IEEE Trans. Pattern Anal. Machine Intell., Vol. PAMI-3, pp. 75-78, Jan. 1981.

[8] L. Devroye and T. J. Wagner, "Distribution free consistency results for nonparametric discrimination and regression function estimation". Ann. Statistics, Vol. 8, pp. 231-239, 1980.

[9]

W. H. Rogers and T. J. Wagner, "A finite sample distribution free performance bound for local discrimination rules," Ann. Statistics, Vol. 6, pp. 506-514, 1978.

[10] J. Beck, "The exponential rate of convergence of error for k_n—NN nonparametric regression and decision," Prob. Contr. Inform. Theory, Vol. 8, pp. 303-311, 1979.

[11] L. Györfi, "The rate of convergence of k_n—NN regression estimates and classification rules," IEEE Trans. Inform. Theory, Vol. IT-27, pp. 362-364, 1981.

[12] R. D. Short and K. Fukunaga, "The optimal distance measure for nearest neighbor classification," IEEE Trans. Inform. Theory, Vol. IT-27, pp. 622-627, Sept. 1981.

[13] K. Fukunaga and T. E. Flick, "An optimal global nearest neighbor metric," IEEE Trans. Pattern Anal. and Machine Intell., Vol. PAMI-6, pp. 314-318, May 1984.

[14] D. A. S. Fraser, Nonparametric Methods in Statistics, New York: Wiley, 1957, Chapter 4.

[15] K. W. Pettis, T. A. Bailey, A. K. Jain and R. C. Dubes, "An intrinsic dimensionality estimator from near-neighbor information," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. PAMI-1, pp. 25-37, Jan. 1979.

[16] K. Fukunaga and T. E. Flick, "Classification error for a very large number of classes," IEEE Trans. Pattern Anal. and Machine Intell., Vol. PAMI-6, pp. 779-788, Nov. 1984.

[17] K. Fukunaga and L. D. Hostetler, "Optimization of k-nearest neighbor density estimates," IEEE Trans. Inform. Theory, Vol. IT-19, pp. 320-326, May 1973.

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

[19] E. Fix and L. J. Hodges, "Discriminatory analysis, nonparametric discrimination, consistency properties," Report No. 4, Project 21-49-004, School of Aviation Medicine, Randolph Field, Texas, 1951.

[20] E. Fix and L. J. Hodges, "Nonparametric discrimination small sample performance," Report NO. 11, Project 21-49-004, School of Aviation Medicine, Randolph Field, Texas, 1952.

[21] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," Annals of Mathematical Statistics, Vol. 27, pp. 832-837, 1956.

[22] E. Parzen, "On the estimation of a probability density function and the mode," Annals of Mathematical Statistics, Vol. 33, pp. 1065-1076, 1962.

[23] T. Cacoullos, "Estimation of a multivariate density," Annals of the Institute of Statistical Mathematics, Vol. 18, pp. 178-189, 1966.

[24] L. Devroye, "The equivalence of weak, strong, and complete convergence in L_1 for kernel density estimates," Annals of Statistics, Vol. 11, pp. 896-904, 1983.

[25] L. Devroye and T. J. Wagner, "The L_1 convergence of kernel estimates," Annals of Statistics, Vol. 7, 1136-1139, 1979.

[26] P. Deheuvels, "Conditions nécessaires et suffisantes de convergence pon ctuelle presque sûre et uniforme presque sûre des estimateurs de la densité, C. R. Acad. Sci., Paris, Sér A 178, 1217-1220, 1974.

[27] L. Devroye and T. J. Wagner, "The strong uniform consistency of kernel density estimates," in Multivariate Analysis V, P.R. Krishnaiah (Ed.), North Hollad, New York, pp. 59-77, 1980.

[28] L. Devroye and L. Györfi, Nonparametric Density Estimation: The L_1 View. New York: John Wiley & Sons 1985.

[29] D. O. Loftsgaarden and C. P. Quesenberry, "A nonparametric estimate of a multivariate density function," Annals of Mathematical Statistics, Vol. 36, pp. 1049-1051, 1965.

[30] D. S. Moore and J. W. Yackel, "Consistency properties of nearest neighbor density estimates," Annals of Statistics, Vol. 5, pp. 143-154, 1977.

[31] L. Devroye and T. J. Wagner, "The strong uniform consistency of nearest neighbor density estimates," Annals of Statistics, Vol. 5, pp. 536-540, 1977.

[32] J. Van Ryzin, "Bayes risk consistency of classification procedures using density estimation," Sankhya Series A,Vol. 28, pp. 161-170, 1966.

[33] L. Rejtö and P. Révész, "Density estimation and pattern classification," Problems of Control and Information Theory, Vol. 2, pp. 67-80, 1973.

[34] N. Glick, "Sample based classification procedures derived from density estimators," J. Amer. Statist. Assoc., Vol. 67, pp. 116-122, 1972.

[35] N. Glick, "Sample based classification procedures related to empirical distributions," IEEE Trans. Inform. Theory, Vol. IT-22, pp. 454-461, 1976.

[36] W. Greblicki, "Asymptotically optimal procedures with density estimates," IEEE Trans. Inform. Theory, Vol. IT-24, pp. 250-251, 1978.

[37] C. Spiegelman and J. Sacks, "Consistent window estimation in nonparametric regression," Annals of Statistics, Vol. 8, p. 240-246, 1980.

[38] L. Devroye, "Necessary and sufficient conditions for the pointwise convergence of nearest neighbor regression function estimates," Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete, Vol. 61, pp. 467-481, 1982.

[39] R. P. W. Duin, "On the choice of smoothing parameters for Parzen estimators of probability density functions," IEEE Transactions on Computers, Vol. C-25, pp. 1175-1179, 1976.

[40] J.D.F. Habbema, J. Hermans, and K. Vandenbreock, "A stepwize discriminant analysis program using density estimation," in COMPSTAT 1974, G. Bruckmann (Ed.), Physica Verlag, Wien, pp. 101-110, 1974.

[41] J. VanNess, "On the dominance of non-parametric Bayes rule discriminant algorithms in high dimensions," Pattern Recognition, Vol. 12, pp. 355-368, 1980.

[42] P. Hall, "Large sample optimality of least squares cross-validation in density estimation," Annals of Statistics, Vol. 11, pp. 1156-1174, 1983.

[43] Y. S. Chow, S. German, and L. D. Wu, "Consistent cross-validated density estimation," Annals of Statistics, Vol. 11, pp. 25-38, 1983.

[44] T. J. Wagner, "Nonparametric estimates of probability densities," IEEE Trans. Inform. Theory, Vol. IT-21, pp. 438-440, 1975.

[45] B. W. Silverman, "Choosing the window width when estimating a density," Biometrika, Vol. 65, pp. 1-11, 1978.

[46] K. Fukunaga and J. M. Mantock, "A Nonparametric two-dimensional display for classification," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-4, pp. 427-436, July 1982.

[47] K. Fukunaga and R. R. Hayes, "Effects of sample size on classifier design," submitted to IEEE Trans. on Pattern Analysis and Machine Intelligence, 1987.

[48] B. Efron, "Bootstrap methods: another look at the jackknife," Annals of Statistics, Vol. 7, pp. 1-26, 1979.

[49] B. Efron, "Estimating the error rate of a prediction rule: improvement on cross-validation," Journal of the American Statistical Association, Vol. 78, pp. 316-331, 1983.

[50] M.R. Chernik, V.K. Murthy, C.D. Nealy, "Application of bootstrap and other resampling techniques: evaluation of classifier performance," Pattern Recognition Letters, Vol. 3, pp. 167-178, May 1985.

[51] M.R. Chernik, V.K. Murthy, C.D. Nealy, "Correction note to 'Application of bootstrap and other resampling techniques: evaluation of classifier performance'," Pattern Recognition Letters, Vol. 4, pp. 133-142, April 1986.

[52] A.K. Jain, R.C. Dubes, and C.C. Chen, "Bootstrap techniques for error estimation," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-9, pp. 628-633, Sept. 1987.

APPENDICES

Appendix A
B_1 of Chapter 2 for Gaussian Cases

When the density function of ω_i, $p_i(X)$, $(i = 1,2)$ is Gaussian with the expected vector M_i and the covariance matrix Σ_i,

$$-\ell n p_i(X) = \frac{1}{2}(X-M_i)^T\Sigma_i^{-1}(X-M_i) + \frac{1}{2}\ell n\,|\Sigma_i| + \frac{n}{2}\ell n 2\pi \qquad (1)$$

By taking the gradient of (1),

$$-\nabla p_i(X)/p_i(X) = \Sigma_i^{-1}(X-M_i) = V_i(X) \qquad (2)$$

The gradient of the mixture density, $p(X)=P_1p_1(X)+P_2p_2(X)$ with a prior probability P_i, is

$$\nabla p(X)/p(X) = \frac{P_1p_1(X)}{p(X)}\,\frac{\nabla p_1(X)}{p_1(X)} + \frac{P_2p_2(X)}{p(X)}\,\frac{\nabla p_2(X)}{p_2(X)}$$

$$= -\left[q_1(X)V_1(X) + q_2(X)V_2(X)\right] \qquad (3)$$

where $q_i(X) = P_ip_i(X)/p(X)$.

The gradients of $q_1(X)$ are:

$$\nabla q_1(X) = \nabla\left\{\frac{P_1p_1(X)}{p(X)}\right\} = q_1(X)q_2(X)\left[\frac{\nabla p_1(X)}{p_1(X)} - \frac{\nabla p_2(X)}{p_2(X)}\right]$$

$$= q_1(X)q_2(X)\left[V_2(X) - V_1(X)\right] \qquad (4)$$

and

$$\nabla^2 q_1(X) = q_1(X)q_2(X)\left[(q_2(X)-q_1(X))\right.$$

$$\left.\cdot\,[V_2(X)-V_1(X)]\,[V_2(X)-V_1(X)]^T + \Sigma_2^{-1} - \Sigma_1^{-1}\right] \qquad (5)$$

Substituting (3), (4) and (5), the B_1 of (17) in chapter 2 becomes

$$B_1(X) = \frac{1}{2} p^{-2/n}(X)\,(q_2(X) - q_1(X))q_1(X)q_2(X)$$

$$\cdot\,[V_1(X)V_1^T(X) - V_2(X)V_2^T(X) + \Sigma_2^{-1} - \Sigma_1^{-1}] \tag{6}$$

For the simplest case in which $\Sigma_1 = \Sigma_2 = I$, $P_1 = P_2 = 0.5$ and $A = I$, $\operatorname{tr} B_1(X)$ becomes

$$\operatorname{tr} B_1(X) = \frac{1}{2} p^{-2/n}(X)q_1(X)q_2(X)(q_2(X) - q_1(X))$$

$$\cdot\,[(X - M_1)^T(X - M_1) - (X - M_2)^T(X - M_2)] \tag{7}$$

In this case, $[\cdot] \underset{\omega_1}{\overset{\omega_2}{\lessgtr}} 0$ is the Bayes classification rule which also satisfies $q_2(X) - q_1(X) \underset{\omega_1}{\overset{\omega_2}{\lessgtr}} 0$. That is, $[\cdot]$ and $q_2(X) - q_1(X)$ share the same sign regardless of X. Since $p(X)$, $q_1(X)$ and $q_2(X)$ are all positive, $\operatorname{tr} B_1(X)$ and subsequently the bias of (16) in chapter 2 becomes positive.

The positive bias is not guaranteed in the more general case. Since $q_2(X) - q_1(X) \underset{\omega_1}{\overset{\omega_2}{\lessgtr}} 0$ is the Bayes classifier, $q_2(X) - q_1(X) > 0$ for the Gaussian case is equivalent to

$$(X - M_1)^T\Sigma_1^{-1}(X - M_1) - (X - M_2)^T\Sigma_2^{-1}(X - M_2) + \ln\frac{|\Sigma_1|}{|\Sigma_2|} - \ln\frac{P_1}{P_2} > 0 \tag{8}$$

On the other hand, from (6) $\operatorname{tr} A^{-1}\,[\cdot]$ becomes positive when

$$(X - M_1)^T\Sigma_1^{-1}A^{-1}\Sigma_1^{-1}(X - M_1) - (X - M_2)^T\Sigma_2^{-1}A^{-1}\Sigma_2^{-1}(X - M_2)$$

$$+ \operatorname{tr} A^{-1}(\Sigma_2^{-1} - \Sigma_1^{-1}) > 0 \tag{9}$$

Thus, these two terms share the same sign in some domain and have different signs in the other domain. Thus, the bias of (16) in chapter 2 could be either positive or negative, depending on Σ_i and M_i as well as A.

Appendix B
Evaluation of $E\left\{\frac{\Delta p_i(x)}{p_i(x)}\right\}$ and $E\left\{\left(\frac{\Delta p_i(x)}{p_i(x)}\right)^2\right\}$ for General Symmetric Kernel Functions

When $\hat{p}_i(x)$ is the Parzen density estimate as given in (12) of Chapter 5, the desired expectations for a general symmetric kernel function is given by

$$E\left\{\frac{\Delta p_i(x)}{p_i(x)}\right\} = \frac{E\{\hat{p}_i(x)\}}{p_i(x)} - 1$$

$$= \frac{h^{-n}k_i(x/h)*p_i(x)}{p_i(x)} - 1 \tag{1}$$

$$E\left\{\left(\frac{\Delta p_i(x)}{p_i(x)}\right)^2\right\} = (1/p_i^2(x))\left[(E(\Delta p_i(x)))^2 + \mathrm{Var}(\Delta p_i(x))\right]$$

$$= \left[\frac{h^{-n}k_i(x/h)*p_i(x)}{p_i(x)} - 1\right]^2$$

$$+ \frac{1}{N}\left[\frac{h^{-2n}k_i^2(x/h)*p_i(x)}{p_i^2(x)} - \left(\frac{h^{-n}k_i(x/h)*p_i(x)}{p_i(x)}\right)^2\right] \tag{2}$$

where * represents convolution in R^n. Equations (1) and (2) give exact expressions for the indicated moments. For a local kernel function, the convolution may be evaluated by approximating $p_i(Y)$ by

$$p_i(Y) \cong p_i(x) + (Y-x)^T\nabla p_i(x) + \frac{1}{2}(Y-x)^T\nabla^2 p_i(x)(Y-x) \tag{3}$$

where the approximation holds in the region in which $k_i((x-Y)/h)$ takes significant values (i.e. when Y is close to x). Under this approximation and using the symmetry of the kernel function, the convolution term becomes

$$\int_{R^n} h^{-n} k_i((x-Y)/h) \left[p_i(x) + (y-x)^T \nabla p_i(x) + \frac{1}{2}(Y-x)^T \nabla^2 p_i(x)(Y-x) \right] dY$$

$$= p_i(x) + \frac{1}{2} \mathrm{tr} \left[\nabla^2 p_i(x) \int (Y-x)(Y-x)^T h^{-n} k_i((Y-x)/h) dY \right]$$

$$= p_i(x) + \frac{1}{2} \mathrm{tr} \left[\nabla^2 p_i(x) h^2 \textstyle\sum_{k_i} \right] \tag{4}$$

Similar results hold for the convolution of $k_i^2(x/h)$ and $p_i(x)$. Substituting these results into (1) and (2) gives

$$E\left\{ \frac{\Delta p_i(x)}{p_i(x)} \right\} \cong \frac{1}{2} h^2 c_i(x) \tag{5}$$

$$E\left\{ \left(\frac{\Delta p_i(x)}{p_i(x)} \right)^2 \right\} \cong \frac{1}{4} h^4 c_i^2(x) + \frac{1}{N} \left[\frac{h^{-n} I(k_i)}{p_i(x)} (1 + \frac{1}{2} h^2 b_i(x)) - (1 + \frac{1}{2} h^2 c_i(x))^2 \right] \tag{6}$$

where

$$I(k_i) = \int k_i^2(x) dx \tag{7}$$

$$c_i(x) = \mathrm{tr} \left[\frac{\nabla^2 p_i(x)}{p_i(x)} \textstyle\sum_{k_i} \right] \tag{8}$$

$$b_i(x) = \mathrm{tr} \left[\frac{\nabla^2 p_i(x)}{p_i(x)} \textstyle\sum_{k_i^2} \right] \tag{9}$$

and $\sum_{k_i^2}$ is the covariance matrix for the density function given by $k_i^2(x)/I(k_i)$. For many popular kernel functions $c_i(x)$ is proportional to $b_i(x)$. For the uniform kernel, $b_i(x) = c_i(x)$, and for the Gaussian kernel $b_i(x) = \frac{1}{2} c_i(x)$.

VITA

VITA

Donald M. Hummels was born in Phoenix Arizona on June 15, 1961. At the age of 10, he moved to Manhattan Kansas where he was raised and eventually attended Kansas State University. He received his B.S. in electrical engineering at Kansas State in May of 1983. He then attended Purdue University, where he obtained his Masters degree in electrical engineering in May, 1985. Since then he has remained at Purdue in pursuit of the Ph.D. degree, also in electrical engineering. Mr. Hummels is a member of Tau Beta Pi, Eta Kappa Nu, and the IEEE computer Society.