Minimum sample size for estimating the Bayes error at a predetermined level.

by Ryno Potgieter

Submitted in partial fulfilment of the requirements for the degree
MSc: Mathematical Statistics
In the Faculty of Natural & Agricultural Sciences
University of Pretoria
Pretoria
(November 2013)

Contents

List of Figures

Declaration

I, Ryno Potgieter, declare that the dissertation, which I hereby submit for the degree MSc: Mathematical Statistics at the University of Pretoria, is my own work and has not previously been submitted by me for a degree at this or any other tertiary institution.

SIGNATURE: ...

DATE: .....

Acknowledgements

I would like to thank Dr Frans Kanfer and Mr Sollie Millard who have acted as my supervisors during all of my postgraduate studies. Their shared knowledge, commitment and guidance over the past few years has moulded my analytical curiosity and ignited my passion for the subject. I would like to thank my family and friends for their unwavering support and genuine interest.

Abstract

Determining the correct sample size is of utmost importance in study design. Large samples yield classifiers or parameters with more precision and conversely, samples that are too small yield unreliable results. Fixed sample size methods, as determined by the specified level of error between the obtained parameter and population value, or a confidence level associated with the estimate, have been developed and are available. These methods are extremely useful when there is little or no cost (consequences of action), financial and time, involved in gathering the data. Alternatively, sequential sampling procedures have been developed specifically to obtain a classifier or parameter estimate that is as accurate as deemed necessary by the researcher, while sampling the least number of observations required to obtain the specified level of accuracy.

This dissertation discusses a sequential procedure, derived using Martingale Limit Theory, which had been developed to train a classifier with the minimum number of observations to ensure, with a high enough probability, that the next observation sampled has a low enough probability of being misclassified. Various classification methods are discussed and tested, with multiple combinations of parameters tested. Additionally, the sequential procedure is tested on microarray data. Various advantages and shortcomings of the sequential procedure are pointed out and discussed.

This dissertation also proposes a new sequential procedure that trains the classifier to such an extent as to accurately estimate the Bayes error with a high probability. The sequential procedure retains all of the advantages of the previous method, while addressing the most serious shortcoming. Ultimately, the sequential procedure developed enables the researcher to dictate how accurate the classifier should be and provides more control over the trained classifier.

Summary

Determining the correct sample size is of utmost importance in study design. Samples that are large can more accurately estimate unknown parameters, whilst underestimating the required sample size will yield unreliable results. Therefore the sample needs to be large enough for the results obtained to be representative. Quite often there are multiple costs involved, including the financial cost of acquiring the data, and the time needed to obtain or physically sample the observations.

Multiple fixed-sample size methods have been developed to calculate the minimum required number of observations in a sample. One such method is Stein's two stage sampling method. In Stein's method, the first stage samples observations to obtain estimates of parameters needed to calculate the final sample size. The second stage then entails sampling the remainder of the required observations. The sample size required in study design, is often limited by cost, time or the availability of data. Consequently, sequential sampling procedures have been developed to sample the least number of observations necessary to accurately train a classifier or derive an estimate. These procedures sample only the minimum number of observations to ensure the classifier or parameter obtained is as accurate as need be.

In this dissertation, a sequential procedure previously proposed, using Martingale Limit Theory, is discussed. This sequential procedure had been developed to sample the minimum number of observations required to train a classifier to such an extent as to ensure, with a probability of at least $1-a$, that the probability of the next observation being misclassified is acceptably low. For the purpose of this dissertation, various simulation studies were conducted to assess the predictive ability of the classifiers derived from the sequential procedure. Some of the classification methods used include linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), K-nearest neighbours (KNN) and linear regression of an indicator matrix (LRIM). Additionally, multiple combinations of input parameters are tested. The sequential procedure is then also tested on a microarray dataset containing data on the breast cancer prognosis of 295 patients. The dataset has been used extensively in the medical field, primarily to derive a 70-gene signature that could more accurately predict a patient's prognosis. The primary disadvantage of this sequential procedure is its inability to account for Bayes errors larger than the desired maximum rate of error as specified by the researcher, which ultimately results in the sequential procedure continuing ad infinitum. Other shortcomings of the sequential procedure are also briefly discussed.

In conclusion, a new sequential procedure that samples the minimum number of observations required to derive a classifier that estimates the Bayes error at a predetermined level is also proposed. Therefore the classifier is trained until the error rate obtained estimates the Bayes error within a predetermined level. The classifier will never try and obtain an unfeasibly low misclassification rate. The sequential procedure retains all of the advantages (being independent of the classification method used, assessing stopping criteria after each iteration, etc.) of the previous method, while addressing the most serious shortcoming. Various simulation studies are conducted, and the two sequential methods are compared. The proposed sequential procedure is also tested on the microarray dataset. Ultimately, the sequential procedure developed enables the researcher to dictate how accurate the classifier should be and provides more control over the trained classifier.

Future work could comprise of testing the proposed sequential procedure on other classification methods not specifically tested in this dissertation, such as Classification and regression trees (CART) or Neural networks. Additionally, the sequential procedure can be enhanced in such a manner that it automatically estimates the values for the input parameters α (the level of significance required to obtain a coverage probability of $100(1-\alpha)\%$) and a (a parameter used to change the bounds of the rate of misclassification from $[0;1]$ to $(0;1)$).

Chapter 1

Introduction

In study design, it is of utmost importance to determine the correct sample size. Large samples yield greater precision when unknown parameters are estimated, and therefore, the obtained results tend to be reliable. There could, however, be multiple costs involved in sampling the required data. These costs often include, and are not necessarily limited to, the actual time needed to obtain the required number of observations and the financial impact associated. When the sample size is too small, the results obtained are often unreliable and inaccurate. This makes a sound or optimal decision, based on these results, impossible. Therefore, it is imperative that the size of the sample be large enough to attain a prespecified level of accuracy.

In this Chapter, fixed-sample size methods of calculating the sample size required to train a classifier or obtain an estimate of an unknown parameter at a particular level of accuracy are briefly discussed. Stein (1945) proposed a two stage sampling procedure to calculate the required sample size. If the cost of sampling data is a limiting factor, a sequential sampling procedure can be used to sample only the minimum number of observations required. Sequential sampling procedures will also be briefly discussed in this chapter, with an emphasis on a sequential procedure derived to train a classifier that would yield a predetermined rate of misclassification. The sequential procedure discussed was originally tested only with one method of classification, however, different methods of classification are presented and discussed in this chapter.

1.1 Fixed-sample size methods and Stein's two-stage sampling method

In most scenarios the sample size needed to train a classifier or derive a sample estimate for some unknown parameter, can be calculated theoretically by specifying an allowed level of error between the obtained parameter estimate and the actual population's value and a level of confidence associated with the estimate. Prescribing the power of the test may also be used to determine the sample size needed.

1.1. FIXED-SAMPLE SIZE METHODS AND STEIN'S TWO-STAGE SAMPLING METHOD

Assuming that there is a random sample of size n generated from a $N(\mu,\ \sigma^2)$ distribution, $X \sim N(\mu,\ \sigma^2)$ with σ^2 known, a $100(1-\alpha)\,\%$ confidence interval for the unknown population mean μ is given by

$$\left(\overline{x} - z_{1-\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}};\ \overline{x} + z_{1-\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}}\right)$$

where $\overline{x}$ is the sample estimate for μ, $z_{1-\frac{\alpha}{2}}$ is the $100\left(1-\frac{\alpha}{2}\right)^{th}$ percentile of the standard normal distribution and n is the sample size required. Solving for n in the equation would yield the sample size needed, for a given value of α, to obtain a confidence interval of half-width $z_{1-\frac{\alpha}{2}}\frac{\sigma}{\sqrt{n}}$.

If the value of σ is not known, there is no fixed-sample size procedure that can be used to derive a $100(1-\alpha)\,\%$ confidence interval for the population mean μ. Stein (1945) proposed a two stage sampling method that addresses this by drawing two samples sequentially. The first sample is used to estimate σ, which in turn is used to calculate the required sample size needed for a particular level of accuracy, and the second sample comprises the remaining observations of the total sample required.

Stein (1945) states that it was proven by Dantzig (1940) that there was no fixed sample-size test for Student's Hypothesis where the power of the test was independent of the variance. Stein (1945) proposed a two-sample test to test the null-hypothesis $H_0 : \mu = \mu_0$, where the power of the test is only dependent on $\mu - \mu_0$ and not on σ^2. In this context, Stein (1945) defines x_i, $i = 1, 2, ...$ to be a series of independent observations from a $N(\mu, \sigma^2)$ distribution.

For $n \geq 2$ the sample variance is defined as $s_n^2 = \frac{\sum_1^n (x_i - \overline{x}_n)^2}{n-1}$. In the first stage of Stein's sampling method, an initial sample of size n_0 is randomly selected and the sample variance $s_{n_0}^2$ is calculated. This serves as an estimate for the population variance. Once the population variance has been estimated with $s_{n_0}^2$, the second stage of Stein's sampling method estimates the required sample size as

$$N = \max\left\{n_0,\ \left[\frac{\left(t_{1-\frac{\alpha}{2},n_0-1}\right)^2 s_{n_0}^2}{d^2}\right] + 1\right\}$$

where d is the half-width of the confidence interval and $[z]$ denotes the greatest integer less than z.

In the derivation, Stein (1945) defined the required sample size as

$$n = \max\left\{n_0 + 1,\ \left[\frac{s_{n_0}^2}{\varpi}\right] + 1\right\}$$

and the statistic

$$\begin{aligned} t' &= \frac{\sum_1^n a_i x_i - \mu_0}{\sqrt{\varpi}} = \frac{\sum_1^n a_i (x_i - \mu)}{\sqrt{\varpi}} + \frac{\mu - \mu_0}{\sqrt{\varpi}} \\ &= u + \frac{\mu - \mu_0}{\sqrt{\varpi}} \end{aligned}$$

where ϖ is a predefined positive constant and a_1, ..., a_n is a series of real numbers chosen in such a manner that

$$\begin{aligned} \sum_1^n a_i &= 1 \\ \text{and} & \\ s^2 \sum_1^n a_i^2 &= \varpi \end{aligned}$$

u then has a t-distribution with $n_0 - 1$ degrees of freedom. The optimal selection of a and ϖ will not be discussed as it falls outside the scope of this dissertation.

Stein (1945) defines the random variable $t_{n_0-1} = y/s$ as a normally distributed random variable with mean 0 and variance σ^2, and states that the random variable $(n_0 - 1)\, s^2/\sigma^2$ has a $\chi^2_{n_0-1}$ distribution. Furthermore, the conditional distribution of u, provided s, is $N\left(0,\ \frac{\sigma^2}{s^2}\right)$ and both t_{n_0-1} and u have the same distribution. This theorem is then used by Stein (1945) to derive an unbiased test for $H_0 : \mu = \mu_0$ where the power of the test is independent of σ^2. Stein (1945) notes that using the proposed tests and confidence intervals result in too many required observations to ensure that the power of the test or the confidence intervals are truly independent of σ^2.

This fact then induced Stein (1945) to redefine the required sample size as

$$n = \max\left\{n_0,\ \left[\frac{s^2_{n_0}}{\varpi}\right] + 1\right\}$$

and the statistic t'' as

$$\begin{aligned} t'' &= \frac{\left(\frac{1}{n}\sum_1^n x_i - \mu_0\right)\sqrt{n}}{s} \\ &= \frac{\frac{1}{n}\sum_1^n (x_i - \mu)}{s}\sqrt{n} + \frac{\mu - \mu_0}{s}\sqrt{n} \\ &= u' + \frac{\mu - \mu_0}{s}\sqrt{n} \end{aligned}$$

It is proven that u' has a t-distribution with $n_0 - 1$ degrees of freedom.

Assuming that

$$n = \max\left\{n_0 + 1, \quad \left[\frac{s_{n_0}^2}{\varpi}\right] + 1\right\}$$

Stein (1945) concluded that $n \geq s^2/\varpi$, leading to

$$\left|\frac{\mu - \mu_0}{s}\sqrt{n}\right| \geq \left|\frac{\mu - \mu_0}{\sqrt{\varpi}}\right|$$

Therefore, if $\left|t''\right| > t_{1-\frac{\alpha}{2},n_0-1}$ or $t'' > t_{1-\alpha,n_0-1}$ the power of the test will always be increased.

A confidence interval for the population mean μ is then provided by

$$\left(\overline{x} - \frac{l}{2},\ \overline{x} + \frac{l}{2}\right)$$

where the probability that the true mean μ is covered is a function of σ, however, it is always larger than $1-\alpha$ with only a minor difference from $1-\alpha$ if $\sigma^2 > n\frac{d^2}{\left(t_{1-\frac{\alpha}{2},n_0-1}\right)^2}$. In this context l is defined as

$$l = \frac{2\sigma t_{1-\frac{\alpha}{2},n_0-1}}{\sqrt{E\left(n\right)}}$$

Since $s_{n_0}^2$ is an estimate of the population variance, the value of N derived is also an estimate. Therefore, there is a level of dependency between the two samples taken in the two stages. Mukhopadhyay (1980) notes that "Stein's procedure is not 'asymptotically efficient'". As cited by Mukhopadhyay (1980), Chow and Robbins (1965) and Ray (1957) proposed a rule that would address the asymptotic inefficiency, resulting in the new estimate for N as:

$$N = \inf\left\{n : n \geq n_0 \text{ and } n \geq \frac{a^2 s_n^2}{d^2}\right\}$$

where a is such that $\Phi\left(a\right) = 1 - \frac{\alpha}{2}$.

1.2 Sequential sampling methods

Sequential sampling is a sampling technique where no predetermined sample size is specified. Observations are either sampled individually or in groups, and after each single or group of observations has been randomly sampled, the researcher's study is conducted and the pre-defined stopping rule or hypothesis is evaluated. Depending on the outcome, another observation or group of observations might need to be sampled and the relevant rules and tests are again evaluated and conducted.

In certain scenarios, the cost associated with sampling observations can be severe, and a researcher might wish to sample the minimum number of observations possible while still being able to make the best informed decision at a prescribed level of accuracy. The greatest benefit, perhaps, of sequentially sampling observations is the impact it has on the study's cost - both financial and time. Sampling fewer observations will have a direct impact on the financial cost of a study. If the process of sampling observations is laborious, a sequential procedure can significantly reduce the amount of time required to obtain enough information for a sound decision to be made.

Fu et al. (2005) proposed a sequential approach to determine the sample size needed to build a classifier. The object of the article was to derive stopping criteria for a sequential procedure so that, with a probability of at least $100\,(1-\alpha)\,\%$, the classifier obtained would yield a probability of a misclassification less than or equal to ε. Therefore, the approach seeks to find the minimum number of observations needed for training a classifier in such a way that there is a high probability, $100\,(1-\alpha)\,\%$, that the probability of the next sampled observation being misclassified is acceptably low, ε.

In this dissertation, the sequential procedure proposed by Fu et al. (2005) will be discussed and the shortcomings of the procedure will be examined. An alternative sequential procedure, addressing the largest shortcoming of the sequential procedure proposed by Fu et al. (2005), is proposed and will be discussed.

1.3 Methods of classification

Classification entails the identification of which group a new observation should be assigned to, based on a number of observed attributes. A separate training dataset, whose observations and corresponding classes are known, is used to derive criteria for the allocation of random observations to any particular class. The attributes, used for assigning an observation to a particular class, can be categorical (blood type or hair colour), ordinal ("High", "Average" or "Low"), integer-valued (the number of people who own dogs) or real valued (a person's height). To group any observation into a specific group, a rule or function is required that will take the inherent information contained in the attributes into account and assign the observation into the correct predefined group. The function is known as the classifier.

Fu et al. (2005) chose to use linear discriminant analysis (LDA) as the classifier in the simulation studies, stating that the sequential procedure is independent of the classifier. For the purposes of this dissertation, LDA, quadratic discriminant analysis (QDA), K-nearest neighbours (KNN) and linear regression of an indicator matrix (LRIM) were

tested. It stands to reason that a classifier that is performing well, or alternatively more efficiently, will on average stop retraining the classifier quicker as the stopping criteria will be met earlier. In the particular case where the predictor takes on values in a discrete set (i.e. the predictor can be any of a list of discrete values), the input space can be divided into regions depending on the classification. A decision boundary is a partition or hypersurface in p-dimensional vector space that divides the vector space into two or more response regions. Depending on which classification method is used, the decision boundaries between the regions can either be smooth (as is the case with a linear decision boundary for example) or rough (as is the case with a decision boundary like K-nearest neighbours). A brief summary of the underlying theory for each of the above mentioned classification procedures follows to familiarise the reader with the methods proposed.

1.3.1 Linear discriminant analysis (LDA)

Linear discriminant analysis is a popular method used for classification. From statistical decision theory, it is known that the optimal classification can be obtained if the class posteriors are known, i.e. if $P(class\ G|X = x)$ is known. If $f_k(x)$ is the class-conditional density of X for class $G = k$, the prior probability of class k is denoted by π_k and $\Sigma_{k=1}^{K}\pi_k = 1$, where K is the total number of classes, using the Bayes Theorem the class posterior for class k can be written as:

$$P(G = k|X = \underline{x}) = \frac{f_k(\underline{x})\pi_k}{\Sigma_{l=1}^{K} f_l(\underline{x})\pi_l}$$

If each class density is then modelled as a multivariate Gaussian, the density function of class k is given by:

$$f_k(\underline{x}) = \frac{1}{(2\pi)^{p/2}\,|\mathbf{\Sigma}_k|^{1/2}} e^{-\frac{1}{2}\left(\underline{x}-\underline{\mu}_k\right)^T \mathbf{\Sigma}_k^{-1}\left(\underline{x}-\underline{\mu}_k\right)}$$

where $\underline{\mu}_k$ is a $p \times 1$ matrix that denotes the population mean for class k, $\mathbf{\Sigma}_k$ is a $p \times p$ matrix denoting the population covariance matrix for class k, $\underline{v}^T$ denotes the transpose of $\underline{v}$ and $\underline{x}$ is a p-dimensional row vector.

LDA considers the specific case where all the classes have a common covariance matrix $\mathbf{\Sigma}_k = \mathbf{\Sigma} \forall k$.

Hastie, Tibshirani and Friedman (2001) note that looking at the log ratio of the class posteriors is sufficient for comparing any two classes, k and l. Therefore the following holds:

$$\begin{aligned}\log\frac{P\left(G=k|X=\underline{x}\right)}{P\left(G=l|X=\underline{x}\right)} &= \log\frac{\frac{f_k(\underline{x})\pi_k}{\Sigma_{j=1}^K f_j(\underline{x})\pi_j}}{\frac{f_l(\underline{x})\pi_l}{\Sigma_{j=1}^K f_j(\underline{x})\pi_j}} \\ &= \log\left(\frac{\pi_k}{\pi_l}\right)-\frac{1}{2}\left(\underline{\mu}_k+\underline{\mu}_l\right)^T\mathbf{\Sigma}^{-1}\left(\underline{\mu}_k-\underline{\mu}_l\right)+\underline{x}^T\mathbf{\Sigma}^{-1}\left(\underline{\mu}_k-\underline{\mu}_l\right)\end{aligned}$$

which is a linear equation in x. The linear discriminant functions can be defined as:

$$\delta_k\left(\underline{x}\right)=\underline{x}^T\mathbf{\Sigma}^{-1}\underline{\mu}_k-\frac{1}{2}\underline{\mu}_k^T\mathbf{\Sigma}^{-1}\underline{\mu}_k+\log\pi_k$$

If the parameters are not known, sample estimates for $\underline{\pi_k}$, $\underline{\mu}_k$ and $\mathbf{\Sigma}$ can be used. Therefore, if N_k is the total number of observations sampled from class k, the parameters can be estimated as:

1. $\widehat{\pi}_k=\frac{N_k}{\Sigma_{l=1}^K N_l}$
2. $\widehat{\underline{\mu}}_k=\frac{\Sigma_{x\in k}\underline{x}}{N_k}$
3. $\widehat{\mathbf{\Sigma}}=\Sigma_{k=1}^K\Sigma_{x\in k}(\underline{x}-\widehat{\underline{\mu}}_k)(\underline{x}-\widehat{\underline{\mu}}_k)^T/(N-K)$

as defined in Hastie, Tibshirani and Friedman (2001).

When comparing two classes, say k and l, an observation would be classified as a member of l if $\delta_l\left(\underline{x}\right)>\delta_k\left(\underline{x}\right)$. A complete derivation of the linear discriminant function and the decision boundary between any two classes is provided in Appendix A.

1.3.2 Quadratic discriminant analysis (QDA)

One of the assumptions of LDA is $\mathbf{\Sigma}_k=\mathbf{\Sigma}\forall k$. However, if this is not the case, the discriminant functions are no longer linear as many of the factors no longer cancel in the derivations, and the resulting discriminant functions are quadratic. This leads to quadratic discriminant analysis. Again looking at the log ratio of the class posteriors, the following results:

$$\begin{aligned}\log\frac{P\left(G=k|X=\underline{x}\right)}{P\left(G=l|X=\underline{x}\right)} &= \log\frac{f_k(\underline{x})\pi_k}{f_l(\underline{x})\pi_l} \\ &= \left(\log\pi_k-\frac{1}{2}\log|\mathbf{\Sigma}_k|-\frac{1}{2}\left(\underline{x}-\underline{\mu}_k\right)^T\mathbf{\Sigma}_k^{-1}\left(\underline{x}-\underline{\mu}_k\right)\right) \\ &\quad-\left(\log\pi_l-\frac{1}{2}\log|\mathbf{\Sigma}_l|-\frac{1}{2}\left(\underline{x}-\underline{\mu}_l\right)^T\mathbf{\Sigma}_l^{-1}\left(\underline{x}-\underline{\mu}_l\right)\right)\end{aligned}$$

leading to the quadratic discriminant functions:

$$\delta_k(\underline{x}) = \log \pi_k - \frac{1}{2}\log|\mathbf{\Sigma}_k| - \frac{1}{2}\left(\underline{x} - \underline{\mu}_k\right)^T \mathbf{\Sigma}_k^{-1}\left(\underline{x} - \underline{\mu}_k\right)$$

In general the estimates derived from QDA should be similar and slightly better than those obtained using LDA, with the only real difference being that separate covariance matrices need to be estimated for each class. If the number of dimensions p in R^p is large the number of parameters can increase substantially. If the necessary parameters are not known, sample estimates for π_k, $\underline{\mu}_k$ and $\mathbf{\Sigma}_k$ can again be used.

Therefore:

1. $\widehat{\pi}_k = \frac{N_k}{\Sigma_{l=1}^K N_l}$
2. $\widehat{\underline{\mu}}_k = \frac{\Sigma_{x\in k}\underline{x}}{N_k}$
3. $\widehat{\mathbf{\Sigma}}_k = \Sigma_{x\in k}(\underline{x} - \widehat{\underline{\mu}}_k)(\underline{x} - \widehat{\underline{\mu}}_k)^T/(N-1)$

When comparing two classes, say k and l, an observation would be classified as a member of l if $\delta_l(\underline{x}) > \delta_k(\underline{x})$. A complete derivation of the quadratic discriminant function and the decision boundary between any two classes is provided in Appendix B.

1.3.3 K-nearest neighbours (KNN)

K-nearest neighbours is an intuitively simple classification method that uses the K observations in the training set nearest to the input observation to derive the classifier $\widehat{Y}$. Nearest in this case implies some form of a distance metric and it is usually assumed to be Euclidean distance. The Euclidean distance between two vectors $\underline{a}$ and $\underline{b}$ is defined as:

$$d(\underline{a},\underline{b}) = \sqrt{(b_1 - a_1)^2 + ... + (b_p - a_p)^2} = \sqrt{(\underline{b} - \underline{a}) \bullet (\underline{b} - \underline{a})} = \|\underline{a} - \underline{b}\|$$

If the set $\mathcal{N}$ is defined to be the K observations nearest to the input vector, based on the chosen distance metric, the K-nearest neighbour classifier is defined as:

$$\widehat{Y} = \frac{1}{K}\Sigma_{y_i\in\mathcal{N}} y_i$$

In the case where the y_i's are binary observations (i.e. $y_i \in \{0,1\}$), $0 \leq \widehat{Y} \leq 1$. If $\widehat{Y} < 0.5$ the observation would be classified as a 0, otherwise if $\widehat{Y} \geq 0.5$ the observation would be classified as a 1. This is equivalent to taking a "majority vote" of the response variable for the K observations in $\mathcal{N}$.

Smaller values of K yield a localised classifier, usually leading to smaller misclassification error. However, this can lead to very irregular or rough decision boundaries.

1.3.4 Linear regression of an indicator matrix (LRIM)

In Simple linear regression, the aim is to derive a linear function that models the relationship between the dependent variable and the independent variable(s), whilst minimising the total sum of squared errors. Assuming that the relationship between the dependent variable and the independent variables is linear, and the elements of $\underline{Y}$ are independent $N(\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip}, \sigma^2)$ distributed, the following holds: $E[\underline{y}] = \mathbf{X}^T \underline{\beta}$, where $\mathbf{X}^T$ denotes a $N \times (p+1)$ matrix consisting of N rows denoting the p-dimensional input observations. The first column of $\mathbf{X}^T$ is entirely populated with 1's, corresponding to a dummy variable for the intercept. $\underline{\beta} = \left(\beta_0, \beta_1, ..., \beta_p\right)$ is a $(p+1) \times 1$ matrix, β_0 denoting the intercept and β_1, ..., β_p denoting the coefficients of the p input variables. If the error term $\underline{e}$ is defined as $\underline{e} = \underline{y} - \mathbf{X}\underline{\beta}$, the minimum sum of squared errors is obtained for $\underline{\widehat{\beta}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\underline{y}$. A complete derivation of this is provided in Appendix C.

In the case where the response variable can be an element of any one of a multiple of G classes, $\mathbf{Y}$ is defined as a $n \times G$ matrix with each column populated either with a 1 (if the response variable is an element of that respective class) or a zero. Each observation of the response variable can be an element of only one class. In this case, the $(p+1) \times G$ parameter matrix is defined as $\underline{\widehat{\beta}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y}$. Any new observation is classified to the class corresponding to the largest value of $\underline{y}_{new} = \underline{x}\underline{\widehat{\beta}}$, which is equivalent to the column of $\underline{y}_{new}$ with the largest value.

Chapter 2

A sequential procedure to attain a predetermined probability of a future misclassification

2.1 Introduction

In this chapter a sequential procedure, that ultimately samples the minimum number of observations required to train a classifier that attains a predetermined probability of misclassifying the next sampled observation, will be discussed. Fu et al. (2005) proposed a procedure that sequentially samples an observation, classifies the observation accordingly, and then evaluates the stopping criteria proposed. The stopping criteria depends on a theorem derived using Martingale Limit Theory and, when satisfied, yields a classifier that, with a probability of at least $100\,(1-\alpha)\,\%$, has a maximum probability ε of misclassifying the next observation sampled. A series of simulations were run to observe the sequential procedure's performance. Various classification methods were tested (LDA, QDA, KNN and LRIM) as well as the effect of changing one of the underlying distributions' variance for all classification methods. The effect of a change in other input parameters was also tested using LDA, as well as the sequential procedure's predictive ability on a microarray dataset. Furthermore, shortcomings of the sequential procedure are discussed and a general conclusion is provided.

2.2 Sequential procedure

In formulating the stopping rule, Fu et al. (2005) defined $\mathcal{Y}_i = \{Y_1, ..., Y_i\}, i = 1, 2, 3, ...$ to be a set of independent and uncorrelated binary observations; Q_i to be an indicator function that Y_i is misclassified based on the $i-1$ previous observations, i.e. $Q_i = 1$ if Y_i is misclassified and 0 otherwise; and $\pi_i = P(Q_i = 1|\mathcal{Y}_{i-1})$, or the conditional probability of misclassifying observation Y_i provided the prior $i-1$ results are known.

2.2. SEQUENTIAL PROCEDURE

In the derivation of their stopping rule, two assumptions are made:

- π_n is weakly monotonically decreasing. Therefore, $\pi_{n+j} \leq \pi_n \; \forall \; j \geq n, \; j \geq 1$; i.e. as the number of observations available for classifier training increases, the conditional probability of a misclassification will either decrease, or remain constant. π_n will therefore converge weakly towards $\pi_\infty \geq 0$ as $n \to \infty$.
- $\pi_\infty > 0$. Therefore, a positive probability of misclassification exists. For a given classification problem with a non-zero Bayes error, the probability $\pi_\infty > 0$ regardless of the type of classifier.

In a scenario where there is a non-zero Bayes error, no trained classifier should be able to perform better and achieve a rate of error lower than the Bayes error. In this context, if the Bayes error is denoted by π_{Bayes} then $\pi_\infty \geq \pi_{Bayes}$.

The stopping rule proposed by Fu et al. (2005) depends on a theorem derived in Fu et al. (2005). Theorem 1, as provided below, depends on the Martingale Central Limit Theorem. Theorem 1 will not be derived or expatiated in this dissertation, but is provided for reference purposes to the reader.

Theorem 1 *For* $0 < \alpha < 1$

$$P(\pi_N \leq N^{-1}\Sigma_{i=1}^{N} Q_i + z_{1-\alpha}\widehat{\kappa}_N / N^{1/2}) \to 1 - \alpha \text{ as } N \to \infty$$

where $z_{1-\alpha}$ is the $(1-\alpha)$ quantile of the standard normal distribution $N(0,1)$.

$$\widehat{\kappa}_N = N^{-1}\Sigma_{i=1}^{N} \widehat{\pi}_i(1-\widehat{\pi}_i)$$
and
$$\widehat{\pi}_i = i^{-1}\Sigma_{j=1}^{i} Q_j$$

$\widehat{\pi}_i$ is an estimator for the conditional probability of a misclassification, π_i, as it is the proportion of observations that have been misclassified from the i observations sampled.

Theorem 1 seeks to minimise the sample size needed to obtain $P(\pi_N \leq \varepsilon) \geq 1-\alpha$. Setting

$$\begin{aligned} \varepsilon &= N^{-1}\Sigma_{i=1}^{N} Q_i + z_{1-\alpha}\widehat{\kappa}_N/N^{1/2} \\ \Rightarrow \quad \varepsilon - N^{-1}\Sigma_{i=1}^{N} Q_i &= z_{1-\alpha}\widehat{\kappa}_N/N^{1/2} \\ \Rightarrow \quad \frac{z_{1-\alpha}\widehat{\kappa}_N}{\varepsilon - N^{-1}\Sigma_{i=1}^{N} Q_i} &= N^{1/2} \\ \Rightarrow \quad N &= \left(\frac{z_{1-\alpha}\widehat{\kappa}_N}{\varepsilon - N^{-1}\Sigma_{i=1}^{N} Q_i}\right)^2 \end{aligned}$$

with $\varepsilon > N^{-1}\Sigma_{i=1}^{N} Q_i$ and $\widehat{\kappa}_N > 0$.

An additional rule is provided by Fu et al. (2005) to stop the sequential procedure should a large number (N_0) of consecutive correct classifications occur. This would typically be the case if the classifier is performing admirably, or if there are very large Bayes errors. Bayes errors larger than the specified acceptable level of error, ε, result in the sequential procedure continuing ad infinitum. In both cases it is better to induce an early stop to the procedure, as this will save processing time and more importantly, resources. Fu et al. (2005) suggested $N_0 = \log(\alpha)/\{\log(1-\varepsilon)\}$ be used as a limit to stop the sequential procedure should N_0 consecutive perfect classifications occur.

The number of required sequential observations in the stopping rule therefore is:

$$N \geq \min\left\{\left(\frac{z_{1-\alpha}\widehat{\kappa}_N}{\varepsilon - N^{-1}\Sigma_{i=1}^{N} Q_i}\right)^2, N_0\right\} \text{ where } 0 < N^{-1}\Sigma_{i=1}^{N} Q_i < \varepsilon \text{ and } \widehat{\kappa}_N > 0$$

.

Assume now that the requirement for $\widehat{\kappa}_N > 0$ was not present. If any number of consecutive perfect classifications occur from the first observation sampled in the sequential procedure, $\widehat{\pi}_i$ would be equal to 0, resulting in $\widehat{\kappa}_N$ equalling 0. If $\widehat{\kappa}_N = 0$ then $\left(\frac{z_{1-\alpha}\widehat{\kappa}_N}{\varepsilon - N^{-1}\Sigma_{i=1}^{N} Q_i}\right)^2 = 0$ and the stopping rule would simplify to $N \geq \min\{0, N_0\}$. Therefore, the sequential procedure would stop at the very first observation sampled if that observation were correctly classified. The same would occur if no observations were correctly classified - $\widehat{\pi}_i$ would equal 1, resulting in $\widehat{\kappa}_N$ equalling 0. If no observations were correctly classified the stopping criteria would also not evaluate against N_0.

Consequently, the stopping criteria is dependent on two factors:

- Firstly, there must be at least one correct classification. This holds for both parts of the rule (i.e. N_0 and $\left(\frac{z_{1-\alpha}\widehat{\kappa}_N}{\varepsilon - N^{-1}\Sigma_{i=1}^{N} Q_i}\right)^2$)
- Secondly, the first parameter in the rule ($\left(\frac{z_{1-\alpha}\widehat{\kappa}_N}{\varepsilon - N^{-1}\Sigma_{i=1}^{N} Q_i}\right)^2$) depends on at least one incorrect classification.

If $\varepsilon = 0$, the $N_0 = \log(\alpha)/\{\log(1-\varepsilon)\}$ equality would not be defined. It is also required that $\varepsilon > N^{-1}\Sigma_{i=1}^{N} Q_i$. This would be the case only if there is at least one correct classification.

Taking the above into consideration, suggests that the stopping criteria should be slightly amended to reflect these rules. The following is therefore proposed for ease of reference:

- If $\max(\underline{Q}) = 1$ and $\min(\underline{Q}) = 0$ (i.e. there are both correct and incorrect classifications) the number of sequential observations in the stopping rule is

  $$N \geq \min\left\{\left(\frac{z_{1-\alpha}\widehat{\kappa}_N}{\varepsilon - N^{-1}\Sigma_{i=1}^{N} Q_i}\right)^2, N_0\right\}$$

- If $\max(\underline{Q}) = 0$ (i.e. there are no incorrect classifications) the number of sequential observations in the stopping rule is $N \geq N_0$

If $\min(\underline{Q}) = 1$ (i.e. there are no correct classifications) the sequential procedure will continue ad infinitum as the classifier has not been successfully trained. If, however, a maximum sample size is specified, the sequential procedure can at least be stopped. Therefore,

- If $\min(\underline{Q}) = 1$ (i.e. there are no correct classifications) the number of sequential observations in the stopping rule is $N = M$, where M is a selected maximum sample size.

Fu et al. (2005) defined M to be the maximum allowed number of sequential observations sampled. It is important to note that this is an artificial upper bound on the number of observations available for sampling. The following sequential procedure, as defined by Fu et al. (2005), can be used to determine the minimum number of observations needed to train the classifier and to obtain the final trained classifier:

An initial sample of size S_0 is chosen and the variable N_0, denoting the number of consecutive correct classifications, is set to 0. At the i^{th} step of the sequential procedure,

the particular chosen classifier is trained. An additional observation is randomly and independently sampled and classified using the trained classifier. If a successful classification was observed, Q_i is set to 0 and N_0 is set equal to $N_0 + 1$. Alternatively, if the classification was incorrect Q_i is set to 1 and N_0 is set equal to 0. $\widehat{\kappa}_i$ is now calculated using the observed values of $\underline{Q}$. Depending on which of the stopping rules are applicable, the respective rule is evaluated, and if the rule is satisfied then the procedure is stopped. If the selected maximum sample size (M) has not yet been reached, another observation should be randomly sampled and the process of training the classifier, classifying the sampled observation and evaluating the applicable stopping rule, should be repeated. If, however, a stopping rule has not yet been satisfied and the number of observations chosen (N) is equal to the maximum number of predefined observations (M), the procedure should also be stopped. The sequential procedure can be summarised as follows:

1. Start with the initial sample S_0 and set $N_0 = 0$.
2. At the i^{th} step, train the classifier using all observations available.
3. Randomly sample a new observation and classify the observation using the classifier trained in step 2.
4. If a correct classification is observed, set $Q_i = 0$ and $N_0 = N_0 + 1$. Otherwise, set $Q_i = 1$ and $N_0 = 0$.
5. Calculate $\widehat{\pi}_i$ and $\widehat{\kappa}_N$.
6. Evaluate the stopping criteria and if any of the stopping criteria is met, or if $N = M$, stop the recursive procedure. If this is not the case, return to step 2 and repeat steps 2 to 6.

It is important to note that the number of sequential steps in training the classifier is reported, as was done by Fu et al. (2005). This is different to the number of observations needed to train the classifier, as the number of sequential steps do not include the initial sample S_0.

2.3 Simulation study

A series of simulations were run to observe the sequential procedure's performance by comparing the observed misclassification rate of the trained classifier with the theoretical error, i.e. the Bayes error. The minimum possible error, or the Bayes error, depends on the underlying distributions of the data, the associated parameters of these distributions, and the respective probabilities of drawing a random observation from each respective distribution.

For the simulations however, the underlying distributions and their respective parameters can be specified, and the probability to select a random observation from each distribution is a function of the initial random sample size of each distribution. The trained classifier can then be used to classify observations in a hold-out sample, and the observed error rate can be compared to the Bayes error. A classifier that is performing well will tend to the Bayes error. Therefore, testing the different classification methods, as well as the effect that a change in the underlying distributions' parameters have, should highlight some of the benefits of using a specific classification method (such as the stopping criteria being satisfied quicker, smaller observed misclassification rates or more/less sensitivity to a shift in the underlying distributions) or possible flaws of the sequential procedure. A classifier that is performing well is expected to satisfy the stopping criteria quicker, or alternatively should yield lower rates of misclassification.

2.3.1 Bayes error

The Bayes error rate is the lowest achievable error rate for any given classifier. Consider a scenario where data is generated from two different overlapping independent normal distributions. The Bayes error is the probability of a misclassification, i.e. the sum of the probabilities to misclassify an observation as being an element from one distribution when, in fact, it is from the other. To calculate the Bayes error, assume a random sample is generated from a $N(\mu_1, \sigma_1^2)$ distribution with probability p, and another random sample is generated from a $N(\mu_2, \sigma_2^2)$ distribution with probability $(1-p)$. Assume the classifier is denoted by λ. The probability of a misclassification is then calculated as:

$$\begin{aligned}
\text{Bayes error} &= (1-p)P[X \leq \lambda | X \sim N(\mu_2, \sigma_2^2)] + pP[X > \lambda | X \sim N(\mu_1, \sigma_1^2)] \\
&= (1-p)P\left[Z \leq \frac{\lambda - \mu_2}{\sqrt{\sigma_2^2}}\right] + pP\left[Z > \frac{\lambda - \mu_1}{\sqrt{\sigma_1^2}}\right] \\
&= (1-p)\Phi\left(\frac{\lambda - \mu_2}{\sqrt{\sigma_2^2}}\right) + p\left(1 - \Phi\left(\frac{\lambda - \mu_1}{\sqrt{\sigma_1^2}}\right)\right)
\end{aligned}$$

2.3. SIMULATION STUDY

Consider the case of equal probability of selection, then ...

$$\begin{aligned}\text{Bayes error} &= (1-p)\Phi\left(\frac{\lambda-\mu_2}{\sqrt{\sigma_2^2}}\right)+p\left(1-\Phi\left(\frac{\lambda-\mu_1}{\sqrt{\sigma_1^2}}\right)\right)\\ &= \frac{1}{2}\left[\Phi\left(\frac{\lambda-\mu_2}{\sqrt{\sigma_2^2}}\right)+\left(1-\Phi\left(\frac{\lambda-\mu_1}{\sqrt{\sigma_1^2}}\right)\right)\right]\end{aligned}$$

Consider the case where $\mu_1 = 0$, $\mu_2 = \Delta$, $\sigma_1^2 = 1$, $\sigma_2^2 = 1$.

$$\begin{aligned}\text{Bayes error} &= \frac{1}{2}\left[\Phi\left(\frac{\lambda-\mu_2}{\sqrt{\sigma_2^2}}\right)+\left(1-\Phi\left(\frac{\lambda-\mu_1}{\sqrt{\sigma_1^2}}\right)\right)\right]\\ &= \frac{1}{2}\left[\Phi\left(\frac{\lambda-\Delta}{\sqrt{1}}\right)+\left(1-\Phi\left(\frac{\lambda}{\sqrt{1}}\right)\right)\right]\\ &= \frac{1}{2}[\Phi(\lambda-\Delta)+(1-\Phi(\lambda))]\end{aligned}$$

Consider the case where $\lambda = \frac{\Delta}{2}$.

$$\begin{aligned}\text{Bayes error} &= \frac{1}{2}[\Phi(\lambda-\Delta)+(1-\Phi(\lambda))]\\ &= \frac{1}{2}\left[\Phi\left(-\frac{\Delta}{2}\right)+\left(1-\Phi\left(\frac{\Delta}{2}\right)\right)\right]\\ &= \frac{1}{2}\left[\left(1-\Phi\left(\frac{\Delta}{2}\right)\right)+\left(1-\Phi\left(\frac{\Delta}{2}\right)\right)\right]\\ &= 1-\Phi\left(\frac{\Delta}{2}\right)\end{aligned}$$

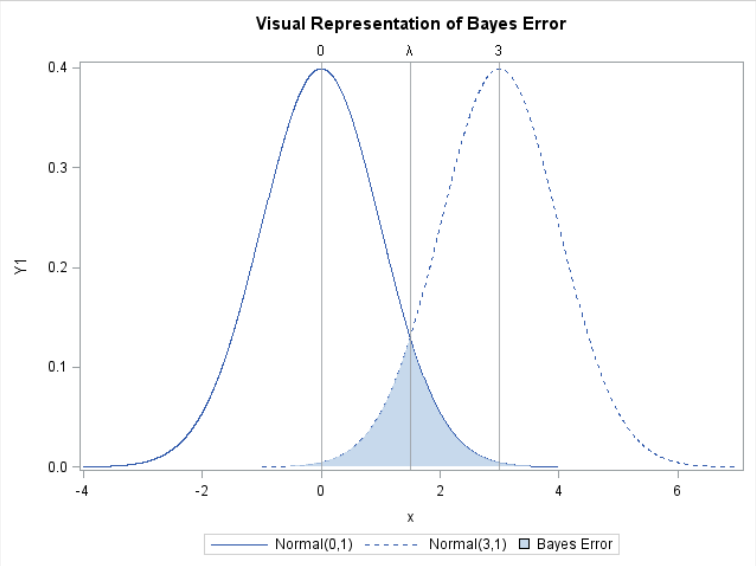


Figure 2.1: *Visual Representation of Bayes error*

Figure 2.1 provides a visual representation of two normal distributions with unit variance, one centered at 0 and the other centered at 3. If λ is considered to be the classifier, an observation from the $N(3,1)$ distribution would be misclassified if it were smaller or equal to λ, and an observation from the $N(0,1)$ distribution would be misclassified if it were larger than λ.

In this particular case the Bayes error is calculated as follows:

$$\begin{aligned}
\text{Bayes error} &= P\left[X \leq \lambda | X \sim N(3,1)\right](1-p) + \\
&\quad P[X > \lambda | X \sim N(0,1)]p \\
&= 0.5\Phi(\lambda - 3) + 0.5(1 - \Phi(\lambda)) \\
&= 1 - \Phi\left(1.5\right) \\
&= 0.06681
\end{aligned}$$

2.3.2 Simulation design

For this particular series of simulations, 5 observations were generated from a $N(\mu_1, \sigma_1^2)$ distribution and 5 observations were generated from a $N(\Delta, \sigma_2^2)$ distribution, with probabilities $\widehat{p}_1 = \frac{n_1}{n_1+n_2} = 0.5$ and $\widehat{p}_2 = 0.5$ respectively. For each simulation the effect of a shift in the underlying distributions is tested, and therefore $\Delta \in \{1,\ 1.3,\ 1.5,\ 2,\ 2.3,\ 2.5,\ 3,\ 4\}$. A change in the allowed level of error, ε, is also tested and therefore $\varepsilon \in \{0.05,\ 0.10,\ 0.15,\ 0.20\}$. The maximum allowed number of sequential steps is set to $M = 90$ and $M = 40$ respectively. To keep track of the number of consecutive correct or perfect classifications, a count variable, denoted N_0, is set equal to 0. In the case of LDA, QDA and LRIM, the initial 10 observations are then used to train the initial classifier $\widehat{\lambda}$.

Note that the same simulation design as that employed in Fu et al. (2005) is used in order for the results to be comparable. The initial sample sizes of 5 and 5 records respectively are the same as employed in Fu et al. (2005).

A single random observation, denoted o_1 in this case, is generated either from a $N(\mu_1, \sigma_1^2)$ distribution with a probability $\widehat{p}_1$, or from a $N(\Delta, \sigma_2^2)$ distribution with a probability $\widehat{p}_2$ and is then evaluated against $\widehat{\lambda}$. In the specific scenarios where either the LDA or QDA classifier is used, o_1 was generated from the $N(\mu_1, \sigma_1^2)$ distribution and $o_1 > \widehat{\lambda}$, the observation is considered to be misclassified. If, however, $o_1 \leq \widehat{\lambda}$ the observation is considered correctly classified. A similar rule applies if it is assumed that o_1 was generated from a $N(\Delta, \sigma_2^2)$ distribution: if $o_1 \leq \widehat{\lambda}$ then the observation is considered to be misclassified, otherwise it is considered to be correctly classified. If the KNN classifier is used, the K observations considered "nearest" to o_1 are used to train the classifier. If o_1 was generated from the $N(\mu_1, \sigma_1^2)$ distribution and $\widehat{\lambda} = 0$, or o_1 was generated from the $N(\mu_2, \sigma_2^2)$ distribution and $\widehat{\lambda} = 1$ the observation is considered correctly classified. In all other scenarios o_1 would be considered incorrectly classified. When using linear regression of an indicator matrix, o_1 is considered to be correctly classified if the largest value in $o_1\widehat{\beta}$ is in the first column - provided o_1 was generated from the $N(\mu_1, \sigma_1^2)$ distribution. If o_1 was generated from the $N(\mu_2, \sigma_2^2)$ distribution and the largest value in $o_1\widehat{\beta}$ is in the second column, o_1 is also considered to be correctly classified. In any other scenario the observation is deemed to be incorrectly classified.

Correctly classifying the first observation would result in the perfect-classification variable N_0 being incremented with 1, i.e. setting $N_0 = N_0 + 1$. This would also result in the value of Q_1 being set equal to 0. However, a misclassification would result in setting N_0 equal to 0 and $Q_1 = 1$. Using these variables, $\widehat{\kappa}_1$ can be calculated and the stopping rules evaluated. This procedure is continued until $N = M$ or the stopping rules are satisfied.

After each repetition of the simulation was finalised, the final value of $\widehat{\lambda}$ was kept in memory and used to classify a random holdout sample to gauge classifier performance and the actual observed error rate. For the holdout sample, 5000 observations were generated from a $N(0, \sigma_1^2)$ distribution and 5000 observations were generated from a $N(\Delta, \sigma_2^2)$ distribution. Depending on the specific classification method used, it was possible to classify the 10000 observations randomly generated accordingly, and the misclassification rate was calculated. The simulations were iterated 1000 times. Afterwards, the average and standard deviation of the observed misclassification rates were calculated, and the minimum, maximum, average and standard deviation of the number of sequential steps required were also calculated.

It is important to note that the method used for classifier testing in this dissertation is different to the method used by Fu et al. (2005). The method used by Fu et al. (2005) is provided below.

"To examine classifier performance, we drew random samples of 10,000 data points, 5000 from $N(0,1)$ *and 5000 from* $N(\Delta,1)$*, and tested the LDA classifier to estimate its cutoff value* λ*. Thus a data point is categorized as class 0 if less than* λ*, or class 1 otherwise. We repeated the drawing of random samples 50 times to obtain accurate estimation of* λ*. The error of the LDA classifier was then calculated with* $e(C_N) = \{1-\Phi(\lambda)+\Phi(\lambda-\Delta)\}/2$ *for each fixed value* Δ*. The sequential training and testing procedure was repeated 1000 times for each fixed pair* (Δ,ε) *to compare the LDA error with the Bayes error, which was calculated with* $\{1-\Phi(\Delta/2)\}$*."*

The reason for using a different testing method is that the method used by Fu et al. (2005) does not use the classifier derived using the sequential procedure for testing classifier performance (calculating the misclassification rate). Therefore the misclassification results provided by Fu et al. (2005) do not completely correspond to what would have been achieved by the proposed sequential procedure. Attempts to contact the author of the article for clarification proved unsuccessful.

The purpose of the initial LDA simulation study was, to an extent, to imitate the results obtained by Fu et al. (2005). The original article proved quite vague in certain respects, and comparing the results obtained and provided in this dissertation to those presented by Fu et al. (2005), to an extent, validates the correctness of the coding. Minor differences, however, are expected as the methods for testing the derived classifier are different. The results, however, compare favourably.

Large sections of pages have deliberately been left empty to facilitate reading within a section and to keep the relevant tables and discussions grouped together.

2.3.3 LDA simulation

The purpose of the initial LDA simulation study was to imitate the results obtained by Fu et al. (2005). For the initial simulation study the results can be found in Table 2.3.1 and Table 2.3.2. The results for $M = 90$ are provided in Table 2.3.1 and Table 2.3.2 provides the results for $M = 40$.

TABLE 2.3.1 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and* 5 *additional observations were generated from a* $N(\Delta,1)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 90 *(i.e.* $M = 90$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.30854	Error (Std)	0.31285 (0.008577)	0.31353 (0.010736)	0.31391 (0.012429)	0.3158 (0.015819)
		Min; Max	30; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.94; 1.8974	86.211; 16.706	77.613; 28.7596	64.031; 36.7603
1.3	0.25785	Error (Std)	0.26124 (0.006895)	0.26207 (0.008684)	0.26336 (0.011322)	0.26426 (0.013017)
		Min; Max	23; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.933; 2.1187	81.487; 23.9947	67.721; 35.1793	50.69; 38.028
1.5	0.22663	Error (Std)	0.22982 (0.006442)	0.23049 (0.00771)	0.23177 (0.00996)	0.23356 (0.011494)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	88.831; 8.6495	77.771; 27.8412	63.872; 35.8741	42.472; 36.4448
2	0.15866	Error (Std)	0.16116 (0.00541)	0.16267 (0.007501)	0.16441 (0.00912)	0.16733 (0.013879)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	84.815; 17.6119	64.702; 34.6758	40.695; 34.5987	22.144; 23.7994
2.3	0.12507	Error (Std)	0.12746 (0.005157)	0.12984 (0.009084)	0.13126 (0.010644)	0.13253 (0.011599)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	80.384; 22.864	51.86; 35.6802	27.003; 26.2447	15.99; 15.2084
2.5	0.10565	Error (Std)	0.10817 (0.005146)	0.11035 (0.007881)	0.11243 (0.010427)	0.11262 (0.00991)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	76.252; 25.9617	44.843; 33.1998	24.203; 22.5269	14.495; 12.9519
3	0.06681	Error (Std)	0.06921 (0.004828)	0.07131 (0.007756)	0.07233 (0.008137)	0.07321 (0.009446)
		Min; Max	21; 90	11; 90	8; 90	6; 67
		$\bar{n}$; Sd	61.15; 30.2236	29.956; 23.5537	16.907; 11.2557	11.783; 5.0827
4	0.02275	Error (Std)	0.02424 (0.002595)	0.02493 (0.003738)	0.02554 (0.004493)	0.02581 (0.00516)
		Min; Max	21; 90	11; 90	8; 28	6; 27
		$\bar{n}$; Sd	44.574; 18.6972	23.972; 8.3582	16.261; 4.073	12.694; 2.5121

For each value of Δ in Table 2.3.1, specifying a smaller value of ε does not lead to a significant reduction in the average misclassification rate. The maximum absolute reduction of 0.00617 is observed where $\Delta = 2$ and ε decreases from 0.2 to 0.05. This, however, leads to an increase in the average number of steps necessary to train the classifier from 22.144 to 84.815. If it is very costly or takes a very long time to sample additional observations, it would not be practical to specify a small ε.

Note that the maximum number of steps, M, was reached at least once in 29 of the 32 scenarios. This indicates cases where either the procedure did not train the classifier successfully (i.e. the procedure was stopped because $N = M$ and not because the probability of the next observation being misclassified was sufficiently small) or it took 90

sequential steps to ensure that the probability of the next observation being misclassified is sufficiently small.

Large values for the average number of sequential steps indicate that the maximum was reached many times. For $\varepsilon = 0.05$, the average number of sequential steps is larger than 80 for all values of Δ up to 2.3. Even for $\Delta = 2.5$ the average number of steps is still larger than 75. This indicates that M was reached more often than not. For all these cases ($\varepsilon = 0.05$ and $\Delta \leq 3$) the Bayes error is larger than ε. Therefore the sequential procedure is trying to train the classifier to adhere to a maximum error rate ε that is impossible to achieve as it is smaller than the Bayes error, the smallest possible error. To successfully train the classifier with an error rate $\varepsilon = 0.05$, the Bayes error must be smaller than 0.05. This is the case only for a tested $\Delta = 4$. Note that for $\Delta \in \{1, 1.3, 1.5\}$ the values of ε tested were too small as the Bayes error was larger than ε in all cases.

The sequential procedure frequently trains the classifier to obtain a rate of misclassification lower than the specified acceptable error ε. For $\Delta = 4$, the Bayes error is lower than ε and the classifier should have been trained to yield a rate of misclassification of at most ε. For all the values of ε tested where $\Delta = 4$, the observed rate of misclassification is lower than ε and is closer to the Bayes error. Therefore the sequential procedure has overtrained the classifier and the classifier does not yield a rate of misclassification near ε, but rather near the Bayes error. Overtraining the classifier results in too many observations sampled to derive a classifier that yields the specific rate of misclassification.

TABLE 2.3.2 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated from a* $N(\Delta,1)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 40 *(i.e.* $M = 40$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.30854	Error (Std)	0.31836 (0.01892)	0.31798 (0.017691)	0.31777 (0.018296)	0.3199 (0.021928)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.883; 1.4083	39.07; 4.9693	36.24; 9.9424	31.856; 13.7534
1.3	0.25785	Error (Std)	0.26455 (0.010733)	0.26433 (0.010924)	0.26512 (0.012389)	0.26661 (0.016065)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.879; 1.4479	37.412; 7.9402	33.072; 12.5064	26.845; 15.1226
1.5	0.22663	Error (Std)	0.23182 (0.008526)	0.23284 (0.010207)	0.23367 (0.011749)	0.23519 (0.015349)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.82; 1.7206	36.06; 9.4627	31.002; 13.4306	23.204; 14.9484
2	0.15866	Error (Std)	0.1633 (0.007031)	0.16423 (0.009126)	0.16565 (0.010787)	0.16598 (0.011324)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.184; 3.4983	32.098; 11.7571	23.712; 13.749	17.322; 12.474
2.3	0.12507	Error (Std)	0.12908 (0.006612)	0.13059 (0.008686)	0.13182 (0.010737)	0.1332 (0.013106)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	37.962; 5.2923	29.214; 12.4479	21.282; 12.6159	14.408; 9.8875
2.5	0.10565	Error (Std)	0.10958 (0.006624)	0.11071 (0.008069)	0.11222 (0.009899)	0.11296 (0.010682)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	37.433; 5.7521	26.502; 12.1397	19.034; 11.6084	13.396; 8.5505
3	0.06681	Error (Std)	0.06985 (0.005286)	0.07113 (0.00717)	0.07234 (0.008239)	0.07235 (0.008693)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	34.435; 7.2197	23.894; 10.6262	15.787; 7.5031	11.952; 4.6989
4	0.02275	Error (Std)	0.02438 (0.002865)	0.02511 (0.003857)	0.02558 (0.004484)	0.02622 (0.005359)
		Min; Max	21; 40	11; 40	8; 40	6; 27
		$\bar{n}$; Sd	34.013; 7.0158	23.38; 7.1047	16.523; 4.1098	12.611; 2.6767

Similar to the results seen for $M = 90$ in Table 2.3.1, for small values of ε in Table 2.3.2, the mean misclassification rate does not decrease substantially as ε is decreased and the average number of sequential steps also does not increase substantially as ε is decreased. For larger values of Δ the average misclassification rate decreases more and the average number of sequential steps increases substantially as ε is decreased.

It is interesting to note that the maximum number of sequential steps allowed, $M = 40$, was reached for nearly all combinations of Δ and ε tested. Considering that the maximum M was reached for most combinations of ε and Δ where $M = 40$ and for most of the combinations where $M = 90$ in Table 2.3.1, it seems as if M should be increased to see how well the sequential procedure performs - this could be a case where the chosen M is

just too small. It is also interesting to note that the observed rate of misclassification was again lower than ε for large values of Δ, resulting in an overtrained classifier.

Comparing the results from Table 2.3.1 to Table 2.3.2, the average rate of misclassification is mostly larger when $M = 40$ compared to when $M = 90$. This is mainly because there are less observations available for training the classifier. The differences are very small though, and in some cases the LDA classifier trained for $M = 40$ performs better (has a lower misclassification rate) than the LDA classifier trained for $M = 90$. One example of this is where $\Delta = 3$ and $\varepsilon = 0.2$.

Various simulations were also conducted where the variance σ_2^2 was allowed to vary ($\sigma_2^2 \in \{2,\ 3,\ 4\}$) for both $M = 90$ and $M = 40$. The results are provided in Appendix D.

Increasing σ_2^2 resulted in an increase in the Bayes error as there is a greater overlap in the underlying distributions. Increasing σ_2^2 resulted in the average misclassification rate increasing. For each value of σ_2^2 tested, however, no significant change in the average rate of misclassification was observed when ε was decreased, and the average number of sequential steps again increases significantly as ε is decreased for large values of Δ. The average number of sequential steps has also increased in nearly all of the scenarios tested, but the standard deviation in the number of sequential steps has decreased mostly for those scenarios where the Bayes error is larger than ε. This indicates that the observations are more clustered around the high average.

Considering that M seems arbitrarily chosen, and no evidence supporting the choice of M was provided by Fu et al. (2005), it should be tested how often the sequential procedure reaches the maximum without actually successfully training the classifier to such an extent that there is a high probability that the probability of the next observation being misclassified is acceptably low. In nearly all of the scenarios tested, the maximum number of allowed sequential steps M was reached at least once. The high averages and low standard deviations in the observed number of sequential steps for some scenarios indicate that the sequential procedure reaches the maximum M often.

2.3.4 QDA simulation

For the initial simulation study done by Fu et al. (2005) the only classification method used was LDA. QDA is a natural extension of LDA and should be investigated. A better, or more efficient, classification method should result in the sequential procedure stopping earlier, due to more correct classifications. This was investigated with the use of the QDA classifier. The results for $M = 90$ are presented in Table 2.3.3 and the results for $M = 40$ are presented in Table 2.3.4.

TABLE 2.3.3 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the QDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated from a* $N(\Delta,1)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 90 *(i.e.* $M = 90$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.30854	Error (Std)	0.31397 (0.00923)	0.31414 (0.01024)	0.31603 (0.015517)	0.31901 (0.021923)
		Min; Max	24; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.871; 2.8839	86.851; 15.2427	80.946; 25.178	66.654; 35.5186
1.3	0.25785	Error (Std)	0.26132 (0.006525)	0.26226 (0.008705)	0.26409 (0.012111)	0.26769 (0.020528)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.726; 4.3262	84.16; 20.2321	72.089; 32.566	57.337; 37.0855
1.5	0.22663	Error (Std)	0.22973 (0.00644)	0.23128 (0.010826)	0.23321 (0.012781)	0.2353 (0.017312)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.535; 5.545	80.467; 25.1086	65.884; 35.0633	45.707; 36.6818
2	0.15866	Error (Std)	0.16133 (0.00577)	0.16338 (0.011158)	0.16679 (0.015677)	0.16981 (0.019561)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	85.455; 16.5387	67.449; 33.5311	43.151; 34.8855	24.382; 25.0485
2.3	0.12507	Error (Std)	0.12763 (0.005282)	0.13097 (0.011992)	0.13333 (0.01531)	0.13655 (0.018565)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	82.003; 21.1957	56.117; 35.418	31.179; 28.9003	17.212; 16.7096
2.5	0.10565	Error (Std)	0.10846 (0.005396)	0.11168 (0.011747)	0.11475 (0.016952)	0.11815 (0.022259)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	78.76; 24	47.48; 34.2825	24.633; 22.7084	14.782; 12.6707
3	0.06681	Error (Std)	0.07025 (0.006341)	0.07445 (0.014003)	0.07626 (0.015157)	0.07906 (0.021901)
		Min; Max	21; 90	11; 90	8; 90	6; 57
		$\bar{n}$; Sd	65.312; 29.5199	30.507; 23.3342	17.629; 11.6041	11.787; 5.657
4	0.02275	Error (Std)	0.02649 (0.006099)	0.02836 (0.00919)	0.03023 (0.012759)	0.03255 (0.018398)
		Min; Max	21; 90	11; 84	8; 59	6; 28
		$\bar{n}$; Sd	45.591; 20.4195	23.893; 8.8553	16.119; 4.6282	12.253; 2.9766

Similar to the LDA results, for a given Δ in Table 2.3.3, the mean misclassification rate does not decrease substantially as ε is decreased. The maximum absolute reduction of 0.00969 is observed where $\Delta = 2.5$ and ε decreases from 0.2 to 0.05. This leads to an increase in the average number of steps necessary to train the classifier from 14.782 to 78.76, which is slightly higher than the LDA results. In general the average number of steps needed to train the classifier is slightly higher for the QDA method when compared to the LDA method.

The maximum number of steps, M, was reached at least once in 28 of the 32 scenarios, compared to the 29 times of the LDA classifier. Consider those results where ε is considerably smaller than the Bayes error. The standard deviation of the number of sequential

steps required is smaller for the QDA classifier than it is for the LDA classifier. This indicates a bit less variation in the number of steps observed, and taking into account that the averages are slightly lower for the LDA classifier compared to the QDA classifier, it shows that the QDA classifier tends to reach the maximum number of steps, M, more often than the LDA classifier. This in turn indicates that the LDA classifier satisfies the stopping criteria at an earlier stage when compared to the QDA classifier. This is to be expected, however, as the underlying data was generated from two distributions with equal variances, and optimal separation would occur with a linear decision boundary.

TABLE 2.3.4 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the QDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated from a* $N(\Delta,1)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 40 *(i.e.* $M = 40$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.30854	Error (Std)	0.32127 (0.021685)	0.32144 (0.021382)	0.32221 (0.022308)	0.32377 (0.025475)
		Min; Max	26; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.986; 0.4427	39.002; 5.1098	36.253; 9.8887	32.695; 13.196
1.3	0.25785	Error (Std)	0.26611 (0.012922)	0.26729 (0.016484)	0.26811 (0.016903)	0.27057 (0.024011)
		Min; Max	22; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.931; 1.0906	37.868; 7.2336	33.341; 12.3194	29.067; 14.5582
1.5	0.22663	Error (Std)	0.2336 (0.013372)	0.234 (0.013442)	0.23525 (0.014998)	0.23869 (0.022481)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.856; 1.5018	37.13; 8.212	31.792; 13.0979	25.104; 15.0572
2	0.15866	Error (Std)	0.16384 (0.008868)	0.1656 (0.012193)	0.16716 (0.014617)	0.1716 (0.022827)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.164; 3.5414	32.904; 11.4121	25.525; 13.9239	17.772; 12.8796
2.3	0.12507	Error (Std)	0.12938 (0.007179)	0.13147 (0.011205)	0.13435 (0.015214)	0.13624 (0.018431)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	38.609; 4.4036	30.538; 12.0179	21.587; 12.9199	15.577; 10.761
2.5	0.10565	Error (Std)	0.10983 (0.006733)	0.11228 (0.011063)	0.11522 (0.014401)	0.11842 (0.020734)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	37.777; 5.343	28.195; 12.0778	19.503; 11.7862	13.941; 9.0266
3	0.06681	Error (Std)	0.07129 (0.00818)	0.07374 (0.012444)	0.07666 (0.017198)	0.07865 (0.021983)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	35.241; 6.9009	23.423; 10.6314	15.949; 7.8427	12.116; 5.2397
4	0.02275	Error (Std)	0.02667 (0.006709)	0.02913 (0.012181)	0.03024 (0.011991)	0.0306 (0.014022)
		Min; Max	21; 40	11; 40	8; 40	6; 28
		$\bar{n}$; Sd	34.089; 6.9336	23.301; 7.5656	16.146; 4.5236	12.38; 2.7938

Similar to those results seen where $M = 90$ in Table 2.3.3, the average rate of misclassification does not decrease substantially as ε is decreased in Table 2.3.4. The results are very similar to those seen where the LDA classifier's maximum number of steps was decreased from 90 to 40. The average misclassification rate for QDA where $M = 40$ is generally also slightly larger than it is for QDA with $M = 90$, and the maximum $M = 40$ was reached at least once for 31 of the 32 scenarios.

Various simulations were also conducted where the variance σ_2^2 was allowed to vary ($\sigma_2^2 \in \{2,\ 3,\ 4\}$) for both $M = 90$ and $M = 40$. The results are provided in Appendix D.

Increasing σ_2^2 resulted in the average misclassification rate increasing by a similar value when compared to the increase in variance for the LDA classifier. As ε is decreased there

is no significant change in the misclassification rate, and for large values of Δ the average number of sequential steps increases significantly as the allowed error ε decreases. Similar to the LDA results, the average number of steps increased while the standard deviation in the number of sequential steps decreased for those scenarios where the Bayes error is larger than ε. In the scenarios where the Bayes error is smaller than ε the standard deviation increased as was seen for the LDA classifier.

TABLE 2.3.5 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the QDA classifier is given below. 5 initial observations were generated from a $N(0,1)$ distribution and 5 additional observations were generated a $N(\Delta,3)$ distribution and $\alpha = 0.05$. The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max, $\bar{n}$ and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32919	Error (Std)	0.32764 (0.009462)	0.32816 (0.010577)	0.32921 (0.01703)	0.33253 (0.019323)
		Min; Max	24; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.934; 2.0871	87.219; 14.4033	82.343; 23.4365	69.098; 34.5981

Increasing σ_2^2 from 1 to 3 for $M = 90$ resulted in the average misclassification rate increasing while the corresponding standard deviations remained relatively stable. The maximum number of steps was also reached at least once in 31 of the 32 scenarios tested. It is, however, very interesting to note that the observed average misclassification rate for $\Delta = 1$ and $\varepsilon \in \{0.05,\ 0.1\}$ is actually lower than the Bayes error, which theoretically should not be the case as the Bayes error is the theoretical minimum error. This result is provided in Table 2.3.5. Increasing σ_2^2 from 1 to 4 for both $M = 90$ and $M = 40$ resulted in multiple cases where the Bayes error was larger than the average misclassification rate. These seem to be consigned to the scenarios where there is a significant overlap in the generating distributions and the specified error rate, ε, is much smaller than the Bayes error. This is explained in more detail in Section 2.6.2.

As the variance σ_2^2 is increased, the average rate of misclassification for the QDA classifier is generally lower than it is for the corresponding LDA classifier. LDA classification assumes that the underlying distributions have equal variances and therefore the pooled variance is used as an estimate for the population variance. QDA assumes that the underlying distributions have unequal variances, therefore QDA is more applicable in this specific scenario where $\sigma_2^2 \neq \sigma_1^2$. Consequently, the observed misclassification rates are lower for QDA than observed for LDA as optimal separation of the distributions occur with a quadratic decision boundary.

2.3.5 5-nn simulation

TABLE 2.3.6 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the* 5*-nn classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and* 5 *additional observations were sampled from a* $N(\Delta,1)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 90 *(i.e.* $M = 90$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.30854	Error (Std)	0.35454 (0.024912)	0.35241 (0.024755)	0.35162 (0.026369)	0.35063 (0.028713)
		Min; Max	22; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.798; 3.6847	87.545; 13.5176	80.497; 25.8278	71.401; 33.3352
1.3	0.25785	Error (Std)	0.29387 (0.021254)	0.29535 (0.022765)	0.2919 (0.023932)	0.2901 (0.027223)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.864; 3.0399	85.077; 18.4837	72.929; 32.1678	57.788; 37.8642
1.5	0.22663	Error (Std)	0.2602 (0.021338)	0.25781 (0.022441)	0.25712 (0.023622)	0.25608 (0.029484)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.468; 5.9309	80.934; 24.7025	66.538; 35.4507	49.108; 37.9617
2	0.15866	Error (Std)	0.18069 (0.018189)	0.17957 (0.019988)	0.1786 (0.022296)	0.17806 (0.023966)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	86.55; 14.6248	67.479; 33.9346	43.989; 35.7108	27.319; 28.6949
2.3	0.12507	Error (Std)	0.14245 (0.016034)	0.14055 (0.018397)	0.14223 (0.020074)	0.1418 (0.022865)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	83.716; 19.1087	55.223; 36.0437	32.254; 30.5644	19.336; 20.7586
2.5	0.10565	Error (Std)	0.11899 (0.013705)	0.11936 (0.016974)	0.12 (0.019556)	0.12045 (0.020212)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	77.677; 25.037	47.99; 34.5804	26.738; 25.7567	16.22; 16.3111
3	0.06681	Error (Std)	0.0758 (0.011213)	0.07652 (0.01345)	0.07657 (0.014421)	0.0776 (0.014715)
		Min; Max	21; 90	11; 90	8; 90	6; 89
		$\bar{n}$; Sd	64.719; 29.8551	32.401; 25.5257	16.809; 11.2941	12.534; 7.3361
4	0.02275	Error (Std)	0.02648 (0.005645)	0.02711 (0.006628)	0.02746 (0.006985)	0.028 (0.00778)
		Min; Max	21; 90	11; 66	8; 52	6; 22
		$\bar{n}$; Sd	44.369; 19.7192	23.855; 7.6815	16.522; 4.62	12.586; 2.6225

To test whether a localised classifier used in the sequential procedure would yield smaller misclassification rates and stop the sequential procedure quicker, K-nearest neighbours was used as a classification method. A small value of K should yield a localised classifier and slightly lower misclassification errors. A better, or more efficient, classification method should result in the sequential procedure stopping earlier, due to more correct classifications. This was investigated with the use of the K-nearest neighbours classifier, testing the 3-nearest neighbours and 5-nearest neighbours respectively. The results for $M = 90$ and $M = 40$, where $K = 5$, are provided in Table 2.3.6 and Table 2.3.7 respectively. The results for $M = 90$ and $M = 40$, where $K = 3$, are provided in

Table 2.3.9 and Table 2.3.10 respectively.

The mean misclassification rate in Table 2.3.6 does not exhibit much variation and only shows minor increases or decreases, contrary to the results seen for the LDA and QDA classifiers where the average rate of misclassification decreases as ε is decreased. For cases where the Bayes error is much larger than ε the misclassification rate is larger than it is for both the LDA and QDA classifiers, and as Δ is further increased and the Bayes error decreases, the misclassification rate tends towards the Bayes error. As ε increases and Δ increases, the 5-nearest neighbour classifier performs better than the QDA classifier as it obtains a lower average rate of misclassification. This, however, only occurs for large values of Δ. In the specific scenarios where the Bayes error is larger than the prespecified minimum acceptable error, ε, the 5-nearest neighbour classifier required, on average, more sequential observations to satisfy the stopping criteria as the average number of sequential steps is slightly higher. The maximum M is still reached at least once for 28 of the 32 different scenarios tested.

TABLE 2.3.7 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the 5-nn classifier is given below. 5 initial observations were generated from a $N(0,1)$ distribution and 5 additional observations were generated from a $N(\Delta,1)$ distribution, $\alpha = 0.05$ and the maximum number of sequential steps is 40 (i.e. $M = 40$). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max, $\bar{n}$ and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.30854	Error (Std)	0.35489 (0.037584)	0.35553 (0.037201)	0.35453 (0.038088)	0.35468 (0.038407)
		Min; Max	23; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.967; 0.7379	39.244; 4.4885	36.487; 9.649	32.508; 13.4186
1.3	0.25785	Error (Std)	0.2951 (0.031845)	0.29555 (0.031743)	0.29327 (0.032009)	0.29242 (0.035234)
		Min; Max	22; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.951; 0.8986	37.998; 7.0863	33.927; 11.9413	29.322; 14.6241
1.5	0.22663	Error (Std)	0.25929 (0.027827)	0.25879 (0.029235)	0.25804 (0.030713)	0.25797 (0.034809)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.903; 1.2622	36.767; 8.7228	31.714; 13.1978	26.453; 15.0632
2	0.15866	Error (Std)	0.18018 (0.023782)	0.17997 (0.023326)	0.17966 (0.024959)	0.18198 (0.03001)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.208; 3.4671	32.718; 11.5026	24.706; 14.1394	18.685; 13.467
2.3	0.12507	Error (Std)	0.14183 (0.020223)	0.14232 (0.020739)	0.14148 (0.022718)	0.14264 (0.023288)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	38.316; 4.9068	30.204; 12.0718	21.506; 12.973	15.413; 11.026
2.5	0.10565	Error (Std)	0.12032 (0.017918)	0.12117 (0.019682)	0.11978 (0.019052)	0.12212 (0.022606)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	37.669; 5.5325	27.84; 12.3494	19.676; 11.8988	13.318; 8.6156
3	0.06681	Error (Std)	0.07597 (0.012831)	0.07677 (0.014552)	0.07756 (0.015463)	0.07747 (0.015617)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	35.388; 6.8662	24.215; 10.8472	16.458; 8.3848	12.008; 4.9879
4	0.02275	Error (Std)	0.02682 (0.006048)	0.0277 (0.007403)	0.02777 (0.00798)	0.02819 (0.00863)
		Min; Max	21; 40	11; 40	8; 40	6; 21
		$\bar{n}$; Sd	34.152; 7.0013	23.275; 7.281	16.28; 4.2696	12.466; 2.684

The results presented in Table 2.3.7 are similar to the results seen for the 5-nearest neighbour classifier with $M = 90$. As the value of ε is decreased, the mean misclassification rate does not decrease substantially. The misclassification rate is nearly equal to those results when $M = 90$. The average number of sequential steps is roughly equal to the average number observed with the LDA classifier, and the maximum M was reached at least once for 31 of the 32 different scenarios tested.

It is also interesting to note that the standard deviation in the number of sequential steps is lower than it is when $M = 90$. This indicates that there is less variation in the number of steps observed, and the observations are closer to the observed averages. Considering that the average number of sequential steps is relatively high in many of the

scenarios tested, it suggests that the maximum M might have been reached more often than liked.

Various simulations were also conducted where the variance σ_2^2 was allowed to vary ($\sigma_2^2 \in \{2,\ 3,\ 4\}$) for both $M = 90$ and $M = 40$. The results are provided in Appendix D.

Increasing σ_2^2, for both $M = 90$ and $M = 40$, results in the average misclassification rate increasing for all combinations of Δ and ε. The average number of sequential steps required to train the classifier has also increased for nearly all combinations of Δ and ε tested, and the maximum M was reached at least once for the majority of the different combinations tested. The standard deviation in the number of sequential steps has increased for nearly all of the different scenarios tested, indicating more spread in the observed values. In those cases where the standard deviation has decreased, the average number of sequential steps has increased towards the maximum M. Taking into account that the maximum was reached for all of these scenarios indicates that the observations are clustered more closely to the maximum, with more of the observations being equal to the maximum.

Increasing σ_2^2 to 3 results in an increase in the average rate of misclassification, however the average rate of misclassification does no longer decrease as substantially when Δ is increased. As was seen in Table D.3.1 the standard deviation in the number of sequential steps has increased for nearly all of the scenarios tested. In those cases where the standard deviation has decreased, the average number of steps has increased towards the maximum M.

TABLE 2.3.8 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the 5-nn classifier is given below. 5 initial observations were generated from a $N(0,1)$ distribution and 5 additional observations were generated a $N(\Delta,4)$ distribution, $\alpha = 0.05$ and the maximum number of sequential steps is 90 (i.e. $M = 90$). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max, $\bar{n}$ and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32743	Error (Std)	0.46677 (0.01336)	0.46384 (0.01958)	0.45815 (0.03408)	0.4502 (0.04189)
		Min; Max	90; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	90; 0	88.295; 11.3756	83.282; 22.2223	75.013; 30.679

It is very interesting to note the specific results for $(\Delta,\ \varepsilon) = (1,\ 0.05)$ where $M = 90$ and $\sigma_2^2 = 4$, as provided in provided in Table 2.3.8. The maximum M was reached in all 1000 repetitions of the simulation. This indicates that the stopping procedure reaches the maximum M too often and it is very possible that the classifiers obtained have not been trained to such an extent to ensure, with a probability of at least $100\,(1-\alpha)\,\%$, that the probability of a misclassification on the next observation is acceptably low - ε. It is necessary to test what the impact will be if the maximum M is increased.

2.3.6 3-nn simulation

TABLE 2.3.9 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the 3-nn classifier is given below. 5 initial observations were generated from a $N(0,1)$ distribution and 5 additional observations were sampled from a $N(\Delta,1)$ distribution, $\alpha = 0.05$ and the maximum number of sequential steps is 90 (i.e. $M = 90$). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max, $\bar{n}$ and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.30854	Error (Std)	0.50001 (0.003822)	0.49982 (0.006826)	0.50001 (0.010108)	0.50007 (0.01617)
		Min; Max	23; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.933; 2.1187	87.698; 13.1105	81.371; 24.7328	71.088; 33.7997
1.3	0.25785	Error (Std)	0.4999 (0.004537)	0.49984 (0.008761)	0.49963 (0.015131)	0.50042 (0.020688)
		Min; Max	22; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.736; 4.1737	83.947; 20.6926	73.757; 31.8854	59.877; 37.9165
1.5	0.22663	Error (Std)	0.5003 (0.004934)	0.50047 (0.010315)	0.49981 (0.018173)	0.49952 (0.026773)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.406; 6.2434	81.595; 23.8542	68.79; 34.5188	51.754; 38.3739
2	0.15866	Error (Std)	0.50029 (0.006131)	0.50031 (0.017051)	0.50043 (0.026352)	0.50263 (0.040599)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	86.685; 14.3701	69.961; 32.6527	48.792; 37.2594	29.298; 30.4767
2.3	0.12507	Error (Std)	0.49994 (0.007648)	0.49914 (0.023401)	0.50127 (0.034256)	0.5005 (0.046521)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	83.615; 19.2003	57.039; 35.5977	35.672; 32.7518	20.655; 22.4392
2.5	0.10565	Error (Std)	0.49966 (0.008003)	0.49813 (0.026024)	0.50074 (0.035897)	0.50215 (0.047212)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	81.143; 21.9089	47.279; 34.9445	29.026; 28.5033	17.143; 17.7356
3	0.06681	Error (Std)	0.50054 (0.01438)	0.49946 (0.030915)	0.4977 (0.043238)	0.49756 (0.055892)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	63.474; 30.5024	32.262; 26.4758	18.31; 13.9272	12.388; 7.5503
4	0.02275	Error (Std)	0.49992 (0.017517)	0.49817 (0.034344)	0.49897 (0.048283)	0.49817 (0.059611)
		Min; Max	21; 90	11; 90	8; 44	6; 42
		$\bar{n}$; Sd	44.958; 19.9403	24.163; 9.0895	16.323; 4.2662	12.476; 2.9967

For a given value of Δ the mean misclassification rate does not vary much and does not decrease as the value of ε is decreased. The average rate of misclassification is much higher than it is for the 5-nearest neighbour classifier or any of the other classifiers tested.

It is also interesting to note that the average rate of misclassification does not decrease or increase significantly as the value of Δ is increased, nor does the average misclassification rate tend towards the Bayes error as is the case for the 5-nearest neighbour classifier. The maximum number of steps is still reached at least once for 30 of the 32 different scenarios tested. Due to the much localised nature of the classifier, the average rate of misclassification is nearly equal to the probability of sampling a random observation from

TABLE 2.3.10 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the* 3*-nn classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and* 5 *additional observations were generated from a* $N(\Delta,1)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 40 *(i.e.* $M = 40$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.30854	Error (Std)	0.37042 (0.033812)	0.36978 (0.036017)	0.36723 (0.036705)	0.36647 (0.038177)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.981; 0.6008	39.065; 4.93	37.397; 8.4857	33.334; 12.8725
1.3	0.25785	Error (Std)	0.31134 (0.032428)	0.31072 (0.034552)	0.30777 (0.035396)	0.30807 (0.039892)
		Min; Max	22; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.954; 0.8465	38.107; 6.7995	34.72; 11.4129	29.825; 14.4063
1.5	0.22663	Error (Std)	0.27375 (0.031387)	0.27307 (0.0339)	0.27031 (0.03459)	0.26957 (0.03649)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.782; 1.9147	36.778; 8.6834	32.47; 12.8421	26.707; 15.2458
2	0.15866	Error (Std)	0.1922 (0.027093)	0.18868 (0.027018)	0.18777 (0.028954)	0.19084 (0.033597)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.206; 3.4365	32.988; 11.4715	25.991; 14.1064	20.695; 14.2897
2.3	0.12507	Error (Std)	0.15228 (0.025653)	0.14855 (0.023536)	0.14975 (0.027648)	0.14898 (0.030125)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	38.29; 4.9153	30.767; 12.1085	22.638; 13.4216	16.392; 11.966
2.5	0.10565	Error (Std)	0.12736 (0.022359)	0.12663 (0.023176)	0.12737 (0.026785)	0.12646 (0.026262)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	37.866; 5.3508	28.394; 12.2265	20.398; 12.0902	14.215; 9.943
3	0.06681	Error (Std)	0.08096 (0.017869)	0.08027 (0.017573)	0.08119 (0.01864)	0.08198 (0.02091)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	35.246; 6.961	23.748; 10.6441	16.96; 9.1041	12.388; 5.9794
4	0.02275	Error (Std)	0.02776 (0.007788)	0.02864 (0.008598)	0.0293 (0.010103)	0.03001 (0.012156)
		Min; Max	21; 40	11; 40	8; 40	6; 27
		$\bar{n}$; Sd	34.261; 7.1004	23.498; 7.2707	16.539; 4.4215	12.478; 2.7527

either of the underlying distributions. Due to the unexpected results obtained, the simulations were repeated multiple times and all results verified.

The results obtained in Table 2.3.10 are very similar to those obtained in Table 2.3.7. For a given Δ the average misclassification rate does not decrease or increase substantially as the value of ε is decreased. Contrary to the results obtained in Table 2.3.9 the average rate of misclassification decreases as Δ is increased, a result comparable to the other classification methods tested. The maximum M is reached at least once for 31 of the 32 different combinations tested.

It is important to note here, that the classifier obtained from the sequential procedure is used for testing purposes against the holdout sample, irrespective of whether the classifier has been correctly trained or not. The unexpected results for the K-nearest neighbour

classifier, particularly the fact that the misclassification rate in Table 2.3.9 does not decrease as Δ is increased, but the misclassification rate in Table 2.3.10 does decrease as Δ is increased, could be related to the classifiers being unsuccessfully trained.

Various simulations were also conducted where the variance σ_2^2 was allowed to vary ($\sigma_2^2 \in \{2,\ 3,\ 4\}$) for both $M = 90$ and $M = 40$. The results are provided in Appendix D.

Increasing σ_2^2 from 1 to 2 and from 1 to 4 for $M = 90$ results in the average rate of misclassification decreasing. The additional spread in the observations and the very local nature of the trained classifier has resulted in the trained classifiers being able to predict better on the testing set. The average number of sequential steps has increased and the maximum M was reached at least once for all the scenarios tested. In those scenarios where the Bayes error is substantially larger than ε the standard deviation of the number of sequential steps has decreased, indicating that the observations are located nearer to the average, and consequently nearer to the maximum M.

Increasing σ_2^2 from 1 to 3 for $M = 90$ results in the average rate of misclassification increasing, albeit to a level lower than observed where $\sigma_2^2 = 1$. Contrary to the results observed for $\sigma_2^2 \in \{1,\ 2\}$, for a given Δ an increase in ε results in the average rate of misclassification decreasing. These results are similar to those seen for the 5-nearest neighbour classifier where $\sigma_2^2 = 3$ and $M = 90$. The average number of sequential steps has increased, and for those particular scenarios where the Bayes error is larger than ε the standard deviation of the number of sequential steps has decreased.

Increasing σ_2^2 for $M = 40$, results in an increase in the average rate of misclassification. The increase is more severe as σ_2^2 increases. The average number of sequential steps has also increased and for most cases where the Bayes error is much larger than ε the standard deviation in the average number of sequential steps has decreased. The maximum number of steps M was reached at least once for all the different scenarios tested.

Due to the much localised nature of the classifier, and the artificial upper bounds placed on the number of observations available to train the classifier, the results obtained using 3-nearest neighbours as a classifier are very mixed. In some scenarios the average rate of misclassification would increase as σ_2^2 is increased, and in other scenarios the average rate of misclassification would decrease. This behaviour was also apparent for the maximum level of acceptable error ε - in some scenarios a decrease in ε would result in the average rate of misclassification increasing substantially, whilst in other scenarios negligible changes were observed. The only consistent observation was that the average number of sequential steps required to "train" the classifier increased as σ_2^2 increased. This particular classifier is very sensitive to the observations already sampled, and to a far lesser extent the underlying distribution of the sampled data.

2.3.7 LRIM simulation

TABLE 2.3.11 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LRIM classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and* 5 *additional observations were generated from a* $N(\Delta,1)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 90 *(i.e.* $M = 90$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.30854	Error (Std)	0.31304 (0.008768)	0.31302 (0.00858)	0.31508 (0.013732)	0.31585 (0.01381)
		Min; Max	25; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.873; 2.8392	87.389; 13.9298	77.631; 28.8084	64.965; 36.4448
1.3	0.25785	Error (Std)	0.26137 (0.006592)	0.26172 (0.008215)	0.2634 (0.010747)	0.26481 (0.015812)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.599; 5.1677	82.812; 22.3402	69.775; 33.7926	52.109; 38.1097
1.5	0.22663	Error (Std)	0.22964 (0.006087)	0.23036 (0.006932)	0.2325 (0.010924)	0.23337 (0.011981)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.401; 6.2903	79.57; 25.9278	61.626; 36.7391	42.967; 36.5363
2	0.15866	Error (Std)	0.16108 (0.005268)	0.16277 (0.008838)	0.16472 (0.010922)	0.1669 (0.012821)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	86.059; 15.5265	64.283; 34.9277	42.141; 35.6687	23.443; 24.7779
2.3	0.12507	Error (Std)	0.12745 (0.005018)	0.12931 (0.008816)	0.13151 (0.009714)	0.1335 (0.013364)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	81.993; 20.9127	54.653; 35.67	29.204; 28.4085	15.837; 14.7272
2.5	0.10565	Error (Std)	0.10792 (0.004539)	0.11075 (0.009306)	0.11204 (0.010535)	0.11272 (0.011007)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	77.935; 24.7681	44.781; 33.9002	22.55; 21.5898	14.206; 11.6808
3	0.06681	Error (Std)	0.06935 (0.004978)	0.07124 (0.008202)	0.07229 (0.008603)	0.0728 (0.009045)
		Min; Max	21; 90	11; 90	8; 88	6; 56
		$\bar{n}$; Sd	62.715; 29.8291	28.742; 22.1182	16.388; 9.7278	11.994; 5.1344
4	0.02275	Error (Std)	0.02419 (0.002681)	0.02491 (0.003775)	0.0255 (0.004829)	0.02541 (0.004168)
		Min; Max	21; 90	11; 90	8; 36	6; 33
		$\bar{n}$; Sd	45.162; 19.3027	23.779; 7.8708	16.423; 4.1332	12.658; 2.6714

To test whether the sequential procedure takes longer to stop with a less efficient classification method, linear regression of an indicator matrix is used. The results for $M = 90$ and $M = 40$ are provided in Table 2.3.11 and Table 2.3.12 respectively.

Similar to the results for the LDA classifier, for a given Δ in Table 2.3.11 the mean misclassification rate does not decrease substantially as ε is decreased from 0.2 to 0.05. It is however very interesting to note how stable the misclassification rate remains as the values of ε are varied. The results are comparable to those obtained using the LDA classifier. When the variances of the underlying distributions are equal, the QDA classifier

is much more sensitive to a change in ε than either LDA or LRIM.

The average misclassification rate is lower for this particular classifier than it is using QDA as the classifier. The average rate of misclassification is considerably lower for those cases where the specified error rate, ε, is higher than the Bayes error, and the observed misclassification rate is much closer to the Bayes error than it is for QDA.

The average number of steps required to train the classifier is also lower in nearly all of the scenarios than the average number of sequential steps required when QDA is used as a classifier. The maximum number of sequential steps, M, was still reached at least once in 28 of the 32 scenarios. Using LRIM as a classifier results in lower misclassification rates, and the classifier needs less sequential steps on average than the QDA classifier to be trained to a satisfactory level.

TABLE 2.3.12 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LRIM classifier is given below. 5 initial observations were generated from a $N(0,1)$ distribution and 5 additional observations were generated from a $N(\Delta,1)$ distribution, $\alpha = 0.05$ and the maximum number of sequential steps is 40 (i.e. $M = 40$). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max, $\bar{n}$ and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.30854	Error (Std)	0.31774 (0.016796)	0.31748 (0.01834)	0.31889 (0.017421)	0.31995 (0.021251)
		Min; Max	22; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.982; 0.5692	38.975; 5.1762	35.557; 10.7193	31.384; 13.9984
1.3	0.25785	Error (Std)	0.2643 (0.01233)	0.26585 (0.013546)	0.26602 (0.015285)	0.26685 (0.016189)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.876; 1.4795	37.384; 7.9589	33.523; 12.2513	26.7; 15.0979
1.5	0.22663	Error (Std)	0.23273 (0.009912)	0.23308 (0.010232)	0.23377 (0.012186)	0.23488 (0.013774)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.865; 1.5116	36.198; 9.228	30.15; 13.8879	25.021; 15.121
2	0.15866	Error (Std)	0.16346 (0.008319)	0.16451 (0.010249)	0.16525 (0.010291)	0.16673 (0.012453)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	38.895; 4.0331	31.46; 12.0157	24.437; 13.6192	17.537; 12.6661
2.3	0.12507	Error (Std)	0.12914 (0.006487)	0.13073 (0.009225)	0.13208 (0.011092)	0.13299 (0.011991)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	37.951; 5.3823	28.519; 12.4319	20.46; 12.375	15.182; 10.5985
2.5	0.10565	Error (Std)	0.10948 (0.006176)	0.11088 (0.008229)	0.11209 (0.00977)	0.11396 (0.013201)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	37.19; 5.9603	26.637; 12.1424	18.425; 11.0815	13.338; 8.5283
3	0.06681	Error (Std)	0.06993 (0.005081)	0.07145 (0.007643)	0.07248 (0.008286)	0.07306 (0.009119)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	35.058; 7.0125	23.451; 10.0324	16.207; 7.5478	11.916; 4.7228
4	0.02275	Error (Std)	0.02448 (0.00284)	0.02493 (0.003587)	0.02506 (0.003754)	0.02598 (0.005258)
		Min; Max	21; 40	11; 40	8; 40	6; 26
		$\bar{n}$; Sd	33.729; 7.0338	23.67; 6.9984	16.46; 4.2363	12.568; 2.6729

The results obtained by limiting the maximum number of sequential steps to 40 are very similar to those obtained for this particular classifier when the maximum number of sequential steps was set at $M = 90$. The average rate of misclassification is lower for this classifier than it is for QDA classifier, and in some cases the average rate of misclassification is lower than it is for the LDA classifier, albeit slightly. The classifier performs quite well and only estimates marginally worse than when $M = 90$ as the average rate of misclassification is only slightly higher.

Various simulations were also conducted where the variance σ_2^2 was allowed to vary ($\sigma_2^2 \in \{2,\ 3,\ 4\}$) for both $M = 90$ and $M = 40$. The results are provided in Appendix D.

Increasing σ_2^2 results in the average rate of misclassification increasing for all combinations of Δ and ε. The average number of sequential steps required to "train" the classifier has also increased, with the increases being significant for those cases where the Bayes error is larger than ε. For those specific cases where the Bayes error is significantly larger than ε, the standard deviation of the number of sequential steps decreases slightly. There are multiple scenarios as well where the maximum allowed number of sequential steps allowed, M, is reached at least once. The decrease in the standard deviation of the number of sequential steps and the increase observed in the average number of sequential steps again indicates that the model reaches the maximum specified (M) far more often.

The results obtained from using LRIM are comparable to the results obtained using LDA. The two sets of results should be comparable though as both classification methods use linear decision boundaries for optimal separation of the data. However, care must be taken when using the regression approach as it easily masks other classes if the response variable Y has more than two classes (i.e. is not binary). This happens when the boundaries found by LRIM pass through one of the classes. Consequently that particular class will never dominate the output from the regression model as there is only 1 linear decision boundary. LDA, on the other hand, has $p-1$ linear decision boundaries when the response variable has p classes.

2.4 Influence of parameter changes

To test how sensitive the procedure is to a change in parameters, various input parameters were changed. Additional simulations were run where the assumption of equal proportions for the sampled data $\widehat{p}_1 = \widehat{p}_2$ was no longer adhered to ($\widehat{p}_2 \in \{0.1, 0.2, 0.4\}$) with all other assumptions kept constant; and where the maximum number of sequential steps allowed was increased to 300 with all other parameters kept constant. For the purpose of this section, the only classification method tested was LDA. As seen in the results provided, the sequential procedure often reaches the maximum allowed number of sequential steps M. This was not limited to any specific classification method tested (i.e. in all methods of classification tested, the sequential procedure reached M at least once), and it therefore seems necessary to test the effects of the parameter changes on only one classification method. LDA was chosen as the variances from the underlying distributions are equal (thereby negating the need for QDA), LRIM yields similar results (the response variable has only two classes and therefore only one linear decision boundary will be obtained) and the K-nearest neighbour classifiers are too computationally intensive for the current purpose.

2.4.1 Changing sampling probabilities

To test how sensitive the procedure is to a change in the sampling probabilities, the input parameter $\widehat{p}_2$ was varied ($\widehat{p}_2 \in \{0.1, 0.2, 0.4\}$). For this section, 3 separate simulations were conducted. To keep the total initial sample size equal to 10 observations, 9 observations were randomly generated from a $N(0,\ 1)$ distribution and an additional 1 observation was randomly generated from a $N(\Delta, 1)$ ($\Delta \in \{1, 1.3, 1.5, 2, 2.3, 2.5, 3, 4\}$) distribution. For the other two simulations the initial sample size combinations were $(8; 2)$ and $(6; 4)$ respectively. The initial sample sizes were kept small to limit the total number of initial observations to 10 for comparative purposes. The maximum number of sequential steps allowed was set at 90 (*i.e.* $M = 90$). The results are available in Appendix D in Table D.6.1, Table D.6.2 and Table D.6.3.

Similar to the results presented for the LDA classifier with equal sampling probabilities, for each value of Δ, specifying a smaller value of ε does not lead to a significant reduction in the average misclassification rate. The average rate of misclassification is, however, lower than observed when sampling with equal probabilities. Additionally, the Bayes errors have been greatly reduced, therefore many of the scenarios can now be correctly tested as the maximum error threshold for the model, ε, is larger than the Bayes error. The average number of sequential steps has been greatly reduced in many scenarios where the separation between the samples is small, i.e. Δ is small. This is mainly due to the fact that in the simulation with equal sampling probabilities, the maximum error threshold specified was too small, forcing the sequential procedure to try and correctly train the classifiers to an error rate that is impossible to achieve. The standard deviation in the number of sequential steps has also increased significantly, indicating a much greater spread in the observed number of sequential steps. When the majority of the data is generated from the $N(0,\ 1)$ distribution, the average number of sequential steps is more dependent on ε than it is on a change in Δ, but as the difference in the sampling probabilities decreases the sensitivity towards Δ increases substantially. It is still very important to note that in all three simulations, the maximum allowed number of sequential steps M was reached at least once for the majority of the $(\Delta,\ \varepsilon)$ combinations tested. The sequential procedure is unable to account for Bayes errors larger than the prespecified level of acceptable error ε. Therefore, the sequential procedure tries to train the classifier to obtain an unfeasibly low rate of misclassification.

2.4.2 Changing maximum number of sequential steps

In all results presented, the maximum allowed number of sequential steps M was reached at least once for the majority of the $(\Delta,\ \varepsilon)$ combinations tested. As there was no evidence supporting the particular choice of M, an increase could result in the sequential

procedure successfully training the classifier more often before $N = M$. If the average number of sequential steps required increases dramatically while the standard deviation in the number of sequential steps remains relatively low, it is indicative of how often the recursive procedure reached the maximum allowed. To test how sensitive the procedure is to a change in the maximum number of sequential steps allowed, a simulation was run where the maximum allowed was set at 300 observations. 5 observations were generated from a $N(0, 1)$ distribution and 5 additional observations were generated from a $N(\Delta, 1)$ distribution ($\Delta \in \{1, 1.3, 1.5, 2, 2.3, 2.5, 3, 4\}$). The effect of a change in ε was also investigated ($\varepsilon \in \{0.05,\ 0.1,\ 0.15,\ 0.2\}$). The level of significance was kept constant at 0.05 (*i.e.* $\alpha = 0.05$). The results are available in Appendix D in Table D.7.1.

Similar to the LDA results observed for $M = 90$ and $\sigma_2^2 = 1$, for each value of Δ, specifying a smaller value of ε does not lead to a significant reduction in the average misclassification rate. The maximum absolute reduction of 0.03566 is observed where $\Delta = 3$ and ε decreases from 0.2 to 0.05. This, however, leads to an increase in the average number of sequential steps necessary to train the classifier from 11.982 to 210.269. This suggests that the maximum number of steps allowed, $M = 300$, was again reached multiple times. For small values of Δ ($\Delta \in \{1,\ 1.3\ , 1.5\ ,\ 2\}$) and ε ($\varepsilon \in \{0.05,\ 0.1\}$) the average number of sequential steps remained above 200. In all of these cases the maximum $M = 300$ was reached at least once. For all of these cases though the Bayes error is much larger than the error rate used in training the classifier, i.e. ε. This would suggest that the sequential procedure is trying to train the classifier to an unrealistic error rate that can not be achieved. The maximum is still reached at least once for 26 of the 32 scenarios tested. It therefore seems plausible to investigate how often the procedure reaches the maximum M, and how often the classifier is successfully trained.

2.5 Microarray sample data application

To test the sequential procedure's performance on real-world data, the sequential procedure was applied to a dataset containing breast-cancer patient prognosis data. In Van't Veer et al. (2002) a gene expression signature that is highly predictive of disease outcome is derived. The sample data is freely available and is widely used in the Biometric field and the sample size is relatively large. It is also possible to compare the results obtained from this analysis to results obtained in other experiments. The dataset consists of 295 observations in total, with measurements on the 70 genes that were found to be highly predictive of the prognosis. The prognosis variable had to be derived from 2 separate variables (conservFlag and C1used) as stated in the original article. The total number of "good prognosis" observations derived equalled the number provided in the article.

TABLE 2.5.1 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LDA classifier on the sample data. The initial sample sizes are provided.* $\alpha = 0.05$ *and the minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier on the* 3 *genes most highly correlated to the prognosis variable is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Initial Sample	Maximum Allowed		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
$(5, 5)$	50	Error (Std)	0.25007 (0.030842)	0.25493 (0.034387)	0.26314 (0.037582)	0.26707 (0.038849)
		Min; Max	8; 50	6; 50	5; 50	4; 50
		$\bar{n}$; Sd	41.622; 15.9589	33.248; 19.478	23.597; 19.0874	16.363; 15.5845
$(5, 5)$	80	Error (Std)	0.24659 (0.031182)	0.25301 (0.034455)	0.26198 (0.038551)	0.26865 (0.041054)
		Min; Max	8; 80	6; 80	5; 80	4; 80
		$\bar{n}$; Sd	62.399; 29.4154	47.478; 33.0945	32.155; 29.8622	18.147; 20.486
$(10, 10)$	50	Error (Std)	0.24483 (0.025483)	0.24804 (0.026697)	0.25123 (0.029396)	0.25442 (0.028819)
		Min; Max	8; 50	6; 50	5; 50	4; 50
		$\bar{n}$; Sd	39.706; 17.1464	30.746; 19.5965	22.935; 18.4514	14.966; 14.3404
$(10, 10)$	80	Error (Std)	0.24172 (0.026975)	0.24592 (0.028477)	0.24985 (0.027322)	0.25288 (0.0296)
		Min; Max	8; 80	6; 80	5; 80	4; 80
		$\bar{n}$; Sd	61.146; 29.9166	45.144; 33.0221	28.472; 28.2134	17.95; 19.9879

For the application to sample data, LDA, 5-nearest neighbours and 3-nearest neighbours were tested. For all scenarios, the sampling probabilities were kept equal sampling either 5 initial observations from both the good prognosis and the bad prognosis groups or 10 initial observations from either. The maximum allowed number of sequential steps was set at 50 and 80 respectively, and the values of ε tested were $\varepsilon \in (0.15,\ 0.2,\ 0.25,\ 0.3)$. To test each respective classifier's predictive ability the remainder of the sample was used as a testing set, comparing the predicted outcome to the clinical outcome. The sequential procedure is repeated 1000 times and the level of significance is kept at $\alpha = 0.05$. The results for the LDA classifier are available in Table 2.5.1, the results for the 3-nearest neighbour classifier are available in Table 2.5.2 and the results for the 5-nearest neighbour classifier are available in Table 2.5.3.

The LDA classifier was trained on the 3 genes most highly correlated to the prognosis variable. Similar to what was observed by Fu et al. (2005), as the minimum allowed error ε decreases from 0.3 to 0.15 the average rate of misclassification decreases and the minimum number and average number of sequential steps increases. This holds for all tested combinations of the different initial sample sizes and the maximum allowed number of sequential steps. For all tested scenarios the maximum number of sequential steps allowed M was reached at least once. Increasing the number of observations in the

TABLE 2.5.2 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the 3-nn classifier on the sample data. The initial sample sizes are provided.* $\alpha = 0.05$ *and the minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier on the* 5 *genes most highly correlated to the prognosis variable is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Initial Sample	Maximum Allowed		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
$(5, 5)$	50	Error (Std)	0.27078 (0.028696)	0.27594 (0.033705)	0.2786 (0.037529)	0.27915 (0.037641)
		Min; Max	8; 50	6; 50	5; 50	4; 50
		$\bar{n}$; Sd	41.428; 16.216	32.992; 19.774	25.794; 19.843	19.126; 17.6267
$(5, 5)$	80	Error (Std)	0.26728 (0.29908)	0.27129 (0.033772)	0.27503 (0.03662)	0.27865 (0.041416)
		Min; Max	8; 80	6; 80	5; 80	4; 80
		$\bar{n}$; Sd	63.303; 27.9452	48.792; 33.4317	36.047; 31.4503	21.581; 24.4526
$(10, 10)$	50	Error (Std)	0.26799 (0.028542)	0.27156 (0.030456)	0.27299 (0.03094)	0.27276 (0.032012)
		Min; Max	8; 50	6; 50	5; 50	4; 50
		$\bar{n}$; Sd	41.26; 16.305	34.326; 19.3775	25.053; 19.513	18.429; 17.1498
$(10, 10)$	80	Error (Std)	0.263 (0.027597)	0.26798 (0.029094)	0.27203 (0.030933)	0.27318 (0.030399)
		Min; Max	8; 80	6; 80	5; 80	4; 80
		$\bar{n}$; Sd	65.011; 28.1259	48.615; 33.3866	33.572; 30.4035	21.057; 23.1812

initial sample results in slightly lower average number of sequential steps as well as slightly lower misclassification rates. Increasing the number of observations in the initial sample does little, however, to change the standard deviation in the number of average steps. This indicates that the distribution of the observed number of sequential steps is shifted slightly. The classifier therefore seems to be better trained, which in general should result in earlier stopping.

The 3-nearest neighbour classifier was trained on the 5 genes most highly correlated to the prognosis variable. As the value of ε is decreased the average rate of misclassification decreases for all scenarios tested. The results are not directly comparable to those obtained from the LDA classifier as that particular classifier was trained on the 3 genes most highly correlated with the prognosis variable. Increasing the number of observations in the initial sample again leads to a minor reduction in the average misclassification rate and the average number of sequential steps needed to train the classifier.

TABLE 2.5.3 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the 5-nn classifier on the sample data. The initial sample sizes are provided.* $\alpha = 0.05$ *and the minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier on the 3 genes most highly correlated to the prognosis variable is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Initial Sample	Maximum Allowed		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
$(5, 5)$	50	Error (Std)	0.26499 (0.035146)	0.26984 (0.035986)	27.137 (0.038206)	0.27457 (0.040335)
		Min; Max	8; 50	6; 50	5; 50	4; 50
		$\bar{n}$; Sd	41.711; 15.9212	34.297; 19.2945	26.395; 19.5876	19.169; 17.5116
$(5, 5)$	80	Error (Std)	0.26146 (0.030778)	0.26456 (0.036375)	0.27366 (0.041771)	0.2749 (0.044655)
		Min; Max	8; 80	6; 80	5; 80	4; 80
		$\bar{n}$; Sd	62.567; 29.5083	50.842; 33.1719	31.633; 29.6903	22.45; 24.2241
$(10, 10)$	50	Error (Std)	0.26114 (0.027926)	0.26365 (0.028544)	0.26573 (0.031888)	0.27009 (0.033119)
		Min; Max	8; 50	6; 50	5; 50	4; 50
		$\bar{n}$; Sd	41.395; 16.2267	33.375; 19.5803	24.379; 19.1291	16.937; 15.8617
$(10, 10)$	80	Error (Std)	0.25843 (0.027138)	0.26057 (0.029336)	0.26498 (0.031295)	0.26885 (0.033495)
		Min; Max	8; 80	6; 80	5; 80	4; 80
		$\bar{n}$; Sd	63.137; 29.109	49.277; 33.1269	32.988; 30.4162	20.722; 23.0017

The 5-nearest neighbour classifier was also trained on the 5 genes most highly correlated to the prognosis variable. Similar to the results reported for the 3-nearest neighbour classifier, decreasing the value of ε results in a decrease in the average rate of misclassification, while the average and minimum number of sequential steps needed to train the classifier increases. The 5-nearest neighbour classifier, although being less localised than the 3-nearest neighbour classifier, performs slightly better with lower average misclassification rates and lower average values for the number of sequential steps required.

In all of the scenarios tested, the maximum allowed number of sequential steps was reached. The same result was observed in all of the simulation studies. This indicates that the sequential procedure did not train the classifiers quickly enough. Taking into account that the procedure was specifically developed to minimise the number of observations needed for a sample to train a classifier that yields an acceptably low rate of misclassification, it is quite worrying that there are only a few simulated scenarios where the maximum was not reached. This leads to the question of what would happen if the maximum were to be increased. Increasing the maximum number of sequential steps allowed to 300 proved ineffective as the classifier was still not successfully trained. Increasing the maximum number of allowed sequential steps surely is one possible solution to ensure that the maximum is not reached, but it seems inconsistent with the goal of minimising the amount of data needed.

2.6 Sequential procedure shortcomings and suggestions

In the results presented in the simulation study, it was evident that the sequential procedure reached the maximum allowed number of sequential steps M at least once for the majority of the parameter combinations tested. This occurrence was not limited to any particular classifier either. Furthermore, for the QDA simulation study it was noted that the observed average misclassification rates for some of the combinations of $(\Delta,\ \varepsilon,\ \sigma_2^2)$ were lower than the Bayes error. This should not be case, as the Bayes error is the theoretical minimum achievable rate of error. Additionally, the classifier obtained from the sequential procedure often yields a rate of misclassification lower than the specified maximum rate of error ε. This results in an overtrained classifier. These shortcomings will be discussed in the following sub-sections.

2.6.1 Artificial upper bound

While testing the sequential procedure's performance using LDA as the classifier, it was noted that the procedure reached the maximum selected number of sequential steps allowed, M, at least once for most of the combinations of $(\Delta,\ \varepsilon)$. This result is repeated for all the other classifiers tested. Changing the input variance of one of the samples had no effect on this, and changing the sampling probabilities also did not remedy this. When the maximum number of sequential steps allowed, M, was increased to 300 the maximum was still reached at least once for more than 80% of the scenarios tested. In many of the different scenarios tested, high average and low standard deviation values for the observed number of sequential steps were observed, indicating that most of the actual contributing results are very near to the average. Increasing the maximum number of sequential steps allowed, M, from 90 to 300 for the LDA classifier decreased the average misclassification rate for all tested combinations of (Δ, ε). However this also brought to light that the sequential procedure reaches M quite often. This leads to the question of whether the sequential procedure actually successfully trained the classifier by the time $N = M$ or whether the sequential procedure was stopped because the maximum M was reached.

Suppose the rule to stop the sequential procedure if $N = M$ were not applicable. If the classifier is not successfully trained, none of the other stopping criteria would be met and the procedure would continue training the classifier, sampling more and more observations. This particular scenario was tested as well. After more than a week of processing the commands on the server, the procedure was manually forced stop. If the procedure were allowed to continue like this the number of steps required before the stopping criteria is met would increase drastically, possibly resulting in an infinite loop.

Arbitrarily choosing a stopping point M and then using those unsuccessfully trained classifiers (obtained classifier value when $N = M$) to estimate classifier performance or report the average number of steps needed to "train" the classifier seems flawed, as the classifiers obtained in such a manner to do not conform to the theoretical requirements derived for the procedure. It is therefore necessary to test how many times the classifier is successfully trained.

To test how many of the 1000 iterations stopped due to the classifier being successfully trained, another series of 1000 simulations were conducted where the classifier would only be recorded if:

- The stopping criteria for $\widehat{\kappa}$ etc. were met and $N \neq M$.
- The stopping criteria for $\widehat{\kappa}$ etc. were met and $N = M$.

where $\widehat{\kappa}$ is defined as on page 11.

Those scenarios where the sequential procedure was stopped purely because $N = M$ were not taken into account.

TABLE 2.6.1 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated from a* $N(\Delta,1)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 300 *(i.e.* $M = 300$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively). For this particular series of simulations the number of sequential steps was not recorded in the case where the stopping criteria had not been met but* $N = M$*. Also provided is the number of times the classifier was successfully trained, denoted by Classifiers Correctly Trained.*

Δ	Bayes error			$\epsilon = 0.05$		$\epsilon = 0.10$		$\epsilon = 0.20$	
1	0.30854	Error (Std) Min; Max $\bar{n}$; Sd	Classifiers Correctly Trained	0.3336 (.) 23; 23 23; .	1	0.33412 (0.040835) 11; 282 21.8333; 49.2244	30	0.32725 (0.031165) 6; 300 80.4625; 91.3745	493
1.3	0.25785	Error (Std) Min; Max $\bar{n}$; Sd	Classifiers Correctly Trained	0.31268 (0.050675) 21; 25 23; 1.633	4	0.28119 (0.031684) 11; 227 25.3263; 43.0337	95	0.27505 (0.030343) 6; 300 80.3699; 88.4056	784
1.5	0.22663	Error (Std) Min; Max $\bar{n}$; Sd	Classifiers Correctly Trained	0.25832 (0.023908) 21; 28 23.8333; 2.4833	6	0.25557 (0.04693) 11; 296 48.9597; 71.3845	124	0.25024 (0.04034) 6; 297 67.5142; 78.8897	914
2	0.15866	Error (Std) Min; Max $\bar{n}$; Sd	Classifiers Correctly Trained	0.18279 (0.03293) 21; 42 25.0357; 4.9627	28	0.18313 (0.044645) 11; 297 68.9559; 84.7527	408	0.1884 (0.04551) 6; 294 31.9759; 43.9518	995
2.3	0.12507	Error (Std) Min; Max $\bar{n}$; Sd	Classifiers Correctly Trained	0.15815 (0.052234) 21; 274 34.4024; 39.4032	82	0.14816 (0.038196) 11; 300 72.1351; 82.8633	666	0.16371 (0.056179) 6; 206 21.831; 26.6442	1000
2.5	0.10565	Error (Std) Min; Max $\bar{n}$; Sd	Classifiers Correctly Trained	0.13359 (0.036073) 21; 281 44.6242; 53.8841	149	0.13181 (0.04461) 11; 297 67.3628; 77.0527	827	0.14407 (0.05527) 6; 164 16.473; 18.8261	1000
3	0.06681	Error (Std) Min; Max $\bar{n}$; Sd	Classifiers Correctly Trained	0.09137 (0.045484) 21; 297 79.9362; 83.0275	392	0.0949 (0.050693) 11; 299 46.2181; 54.7282	986	0.11365 (0.068465) 6; 66 12.607; 7.6007	1000
4	0.02275	Error (Std) Min; Max $\bar{n}$; Sd	Classifiers Correctly Trained	0.042 (0.037184) 21; 296 65.9722; 58.72	971	0.0566 (0.057581) 11; 180 24.767; 17.939	1000	0.06785 (0.07258) 6; 49 11.915; 3.7552	1000

Provided in Table 2.6.1 is the number of times the classifier was successfully trained (denoted classifiers correctly trained). Note that for $\varepsilon = 0.05$ and $\Delta \in \{1, 1.3, 1.5, 2, 2.3, 2.5\}$ the classifier was successfully trained in less than 15% of the cases, and for $\varepsilon = 0.05$ and $\Delta = 1$ the classifier was successfully trained only once. This corresponds to a correct result in only 0.1% of the repetitions.

In calculating the average number of sequential steps, Fu et al. (2005) seem to have taken those cases into account where the classifier was not successfully trained but $N = M$. This seems flawed as the classifier has not been trained to such an extent to ensure that, with a probability of at least $100\,(1-\alpha)\,\%$, the probability of the next observation being misclassified is smaller or equal to ε. For all purposes those trained classifiers should not be used as enough evidence has not yet been obtained to show that the classifiers should perform adequately.

In those cases where the Bayes error is larger than ε the sequential procedure often fails to correctly train the classifier. When $\Delta = 1$ the classifier is correctly trained at most nearly 50% of the time ($\varepsilon = 0.2$). Only in those cases where ε is substantially larger than the Bayes error is the classifier correctly trained every time. The sequential procedure aims to derive a classifier that yields, with a high probability, an acceptably low rate of misclassification. If the specified rate of misclassification is unfeasible, the sequential procedure is unable to account for this and continues ad infinitum. What is required, however, is a sequential procedure that will train the classifier to obtain a feasible error rate. The researcher need not specify the feasible error rate, merely how accurately he/she wishes to estimate this error rate. This will cater specifically for those scenarios where the rate of misclassification that the researcher wishes to achieve is lower than the Bayes error.

2.6.2 Misclassification rates smaller than the Bayes error

The Bayes error is the theoretical minimum rate of achievable error for a given classifier. To calculate the Bayes error, the underlying distribution of the data, the associated parameters of these distributions, as well as the respective probabilities of sampling a random observation from each respective distribution needs to be known, and in practice this is often not the case. Therefore the Bayes error often needs to estimated rather than calculated.

In the simulation studies, particularly in Table D.2.2, Table D.2.3 and Table D.2.6 the observed misclassification rate was lower than the Bayes error. This was observed in the specific scenarios where one of the two underlying distributions exhibited more variation. The misclassification rate reported is the average of the observed misclassification rates over all iterations for each of the variable combinations of (Δ, ε) tested.

It is possible, due to sampling variation, that a calculated error rate for any particular iteration is lower than the Bayes error. A level of confidence or a measure of the error associated with the estimated error rate is required. This measure of error has been defined in the literature as Monte Carlo Error (MCE).

LDA QDA

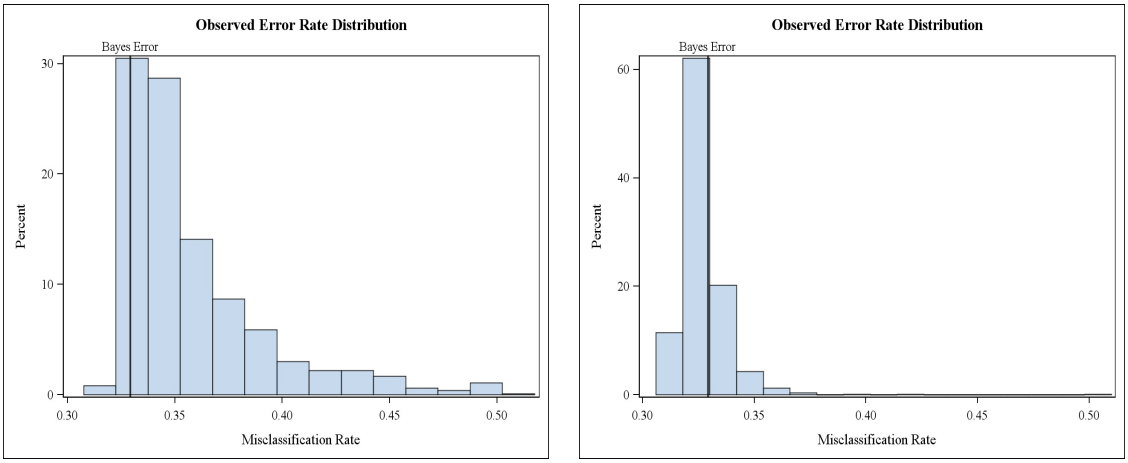


Figure 2.6.2: *Observed Misclassification Rate for LDA and QDA classifiers. Results correspond to Table D.1.2 and Table D.2.2.*

A visual representation of the distribution of the observed classification error is provided in Figure 2.6.2. The images display the observed misclassification rates for Table D.1.2 and Table D.2.2 respectively, for $\Delta = 1$ and $\varepsilon = 0.05$. In both scenarios, data was generated from a $N(0,1)$ and a $N(1,3)$ distribution. Although some of the observed misclassification rates for the LDA classifier is smaller than the Bayes error, the majority of the observations contributing to the average rate of observed misclassification are larger than the Bayes error, resulting in the reported value being larger than the Bayes error.

For the QDA classifier however, the majority of the contributing observations are smaller than the Bayes error. It is interesting to note how skewed both of the distributions are. Furthermore, it is also important to take into account that both sets of results were obtained from simulations where the maximum acceptable rate of error was unfeasibly low (i.e. smaller than the Bayes error). Consequently the classifiers and all parameters used in calculating the Bayes error are incomplete and the classifiers have not been successfully trained to ensure, with a high probability, a sufficiently low probability of misclassification.

Provided in Figure 2.6.3 are the distributions of the observed misclassification rate if the maximum number of allowed sequential steps in training both the LDA and the QDA classifier (for a $N(0,1)$ distribution and a $N(1,3)$ distribution, $\varepsilon = 0.05$ and $\alpha = 0.05$) is increased from $M = 90$ to $M = 1000$.

The distribution of the misclassification rates for $M = 1000$ is much more symmetric than it is for $M = 90$. For the LDA classifier, however, the distribution is still slightly skewed. Similar to Figure 2.6.2, the majority of the observations contributing to the LDA distribution are larger than the Bayes error, resulting in the observed average misclassification rate being larger than the Bayes error. For the QDA classifier, however, the

LDA QDA

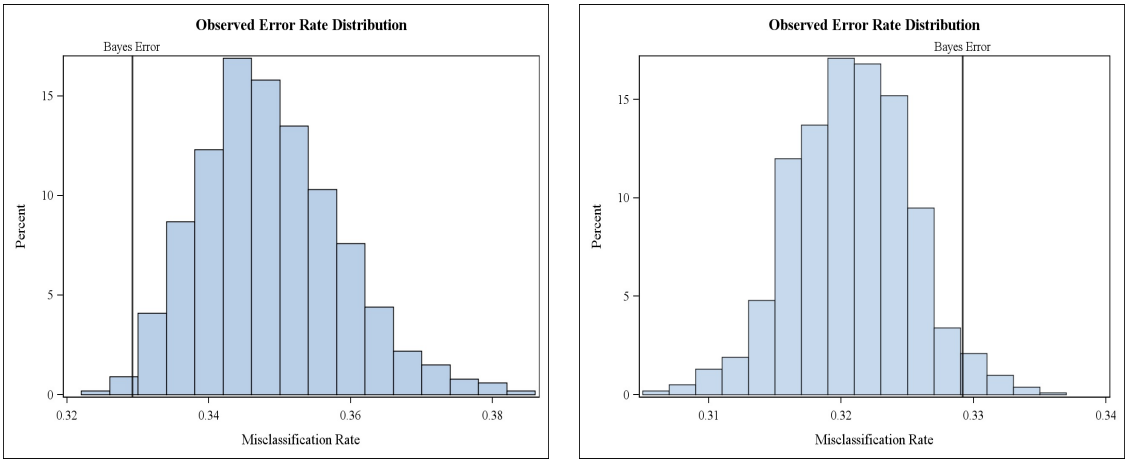


Figure 2.6.3: *Observed Misclassification Rate while training the LDA and QDA classifier. 5 initial observations were generated from a $N(0,1)$ distribution and 5 observations were generated from a $N(1,3)$ distribution, $\alpha = 0.05$, $\Delta = 0.05$.*

majority of contributing observations are smaller than the Bayes error. The distribution is also centered around a value lower than the Bayes error. This is caused mainly by the over- and under-estimation of the parameters (the mean and covariance matrices) that are used when calculating $\widehat{\lambda}$. As these matrices differ from the actual population matrices for the mean and covariance, their over- and under-estimation result in unexpectedly low misclassification rates. For a fixed value of $\widehat{\lambda}$, a sample covariance matrix Σ_2 that is larger than the $\widehat{\mathbf{\Sigma}}_2$ used in calculating $\widehat{\lambda}$ for example results in a lower than expected probability of misclassification. Since the classifier $\widehat{\lambda}$ that is used in the Bayes error calculation is the $\widehat{\lambda}$ obtained from the sequential procedure, the small misclassification rates indicate that the sample covariance and mean matrices are either too large or too small.

2.6.3 Overtrained classifier

In the results presented, the observed average rate of misclassification is often lower than the specified level of acceptable error ε. This is mostly prevalent in the scenarios where there is more separation between the underlying distributions (i.e. Δ is large). In the specific scenarios where a feasible acceptable rate of error is specified, i.e. the Bayes error is smaller than ε, the classifier is trained to such an extent that the rate of misclassification it obtains is smaller than ε. Therefore the classifier is performing better than required. Usually this would not pose a problem, but in a sequential procedure this implies that more observations were sampled than needed to obtain a particular level of accuracy, and consequently the classifier is overtrained. The sequential procedure was proposed to sample only the minimum number of observations required to obtain a classifier that

would yield a small enough rate of misclassification, but unfortunately samples more records than required and this would result in additional costs. Therefore the sequential procedure does not perform according to it's intended design.

2.7 Conclusion

In this chapter, a sequential procedure, developed to train a classifier that would yield, with a high probability, an acceptably low rate of misclassification, was discussed. The sequential procedure proposed by Fu et al. (2005) has been developed to sample the minimum number of observations necessary to train a classifier to such an extent that there is a probability of at least $100\,(1-\alpha)\,\%$ that the probability of misclassifying the next randomly sampled observation is at most ε. Therefore the sequential procedure tries to adhere to a prespecified probability of misclassifying the next randomly sampled observation, while still sampling only the minimum required number of observations.

The sequential procedure has multiple advantages: it evaluates the stopping criteria after each iteration, thereby ensuring the procedure will not sample observations unnecessarily once the stopping criteria have been met; it is not dependent on one method of classification, but rather depends only on a series of binary input variables, i.e. variables indicating whether a sampled observation was incorrectly (denoted 1) or correctly (denoted 0) classified; it recursively obtains a trained classifier that yields a specified rate of misclassification with a high probability.

Unfortunately, the sequential procedure can not account for Bayes errors larger than the prespecified maximum rate of allowable error ε. In these scenarios the sequential procedure tries to train the classifier to obtain an unfeasibly low rate of misclassification. Considering that the Bayes error is often not known, it is easy to specify a desired maximum level of acceptable error lower than the Bayes error. This results in the sequential procedure continuing ad infinitum. Additionally, the sequential procedure occasionally samples too many observations and trains the classifier to obtain an error level smaller than the specified error level. Multiple cases were observed where the average observed rate of misclassification is smaller than the prespecified acceptable level of error, ε. Therefore the classifier is trained to obtain a level of error which is smaller than required or specified.

The sequential procedure was proposed to aid in scenarios where data is not freely available, possibly due to high costs (in gathering data or the cost of a misclassification) or for some other reason, so that the classifier used could be trained with the least amount of data to ensure, with a high enough probability of at least $100\,(1-\alpha)\,\%$, that after

the classifier had been successfully trained the probability of the very next observation sampled being incorrectly classified is acceptably low ε. As stated, the procedure can not account for unfeasibly low levels of specified error and continues indefinitely. Rather than trying to attain a specified rate of error, which could be impossibly low, the minimum feasible rate of error should be pursued. Therefore, a sequential procedure that ultimately yields a classifier that has been trained to accurately estimate the minimum feasible rate of error is proposed. This sequential procedure will not continue indefinitely, provided an acceptable level of accuracy is specified, and will always yield a classifier that estimates the minimum rate of error - the Bayes error.

Chapter 3

A sequential procedure for estimating the Bayes error at a predetermined level of accuracy

3.1 Introduction

In this chapter a sequential procedure, that ultimately samples the minimum number of observations required to derive a classifier that estimates the Bayes error at a predetermined level, is proposed and will be discussed. Fu et al. (2005) proposed a sequential method that samples the minimum number of observations required to train a classifier that yields prespecified rate of misclassification with a high probability. The sequential procedure, unfortunately, is unable to account for Bayes errors larger than the acceptable rate of error and continues ad infinitum. A sequential procedure that yields a classifier that can, as accurately as deemed necessary, estimate the Bayes error is proposed. Using the same approach of sequentially training a classifier, randomly sampling an observation and classifying the observation accordingly, a series of binary observations is obtained. These binary observations are used to obtain the sample rate of misclassification, which is an estimate of the Bayes error. Therefore, a confidence interval for the proportion of misclassified observations can be derived, and a sequential procedure is proposed that utilises these confidence intervals to accurately estimate the Bayes error within a halfwidth h.

A series of simulations were run to observe the sequential procedure's performance using LDA as the classification method. Furthermore, the sequential procedure is also tested on the microarray dataset. A general conclusion is provided.

3.2 Obtaining input observations

In practice it is nearly never an easy task to estimate or calculate the Bayes error. If the underlying distribution of the data is not available this task could prove very time consuming and cost-ineffective. The sequential procedure discussed by Fu et al. (2005),

unfortunately, can not take into account cases where the specified level of acceptable error, ε, is smaller than the Bayes error.

As shown in the results obtained, the sequential procedure proposed by Fu et al. (2005) frequently reaches the prespecified maximum. Also, in simulations where the allowed maximum number of sequential steps had been removed the procedure continued ad infinitum. In the specific scenarios where the maximum is reached, the classifier obtained from the sequential procedure will, in most cases, not have been trained to such an extent to ensure, with a probability of at least $100\,(1-\alpha)\,\%$, that the probability of the next observation being misclassified is less than or equal to ε. No additional evidence supports the selected maximum and it could be some arbitrarily chosen number. Allowing the sequential procedure to continue ad infinitum, is not practical and is of no meaningful value.

An alternative approach to use would be to sample observations and train the classifier until the observed rate of misclassification converges towards a feasible value. This suggests an alternative approach to train the classifier. Each respective method of classification yields it's own Bayes error, provided the underlying distribution(s) and related parameters are kept constant. Since the Bayes error is the minimum feasible error, an optimally trained classifier will yield a rate of misclassification $\widehat{p}$ that estimates the Bayes error. Therefore, the proposed approach is to obtain a classifier that yields a rate of misclassification comparable to the Bayes error.

As the Bayes error is typically not known, the ideal would be for the sequential procedure to continue until the rate of misclassification stabilises within a certain range of the Bayes error, or an accurate estimate thereof. Therefore the classifier would be trained until, with a probability of at least $100\,(1-\alpha)\,\%$, the rate of misclassification obtained from the classifier is within h of the Bayes error, *i.e.* $P\left(MR \in [BE-h\,;BE+h]\right) \geq 1-\alpha$, with BE denoting the Bayes error and MR denoting the misclassification rate. In essence this involves finding a fixed width confidence interval for a proportion p.

Let a correct classification be denoted by $Q_i = 0$ and an incorrect classification denoted by $Q_i = 1$. Therefore a series of binary observations are obtained from the sequential procedure, irrespective of which classification method is used. The ratio of incorrect classifications to all records classified, $\widehat{p} = \Sigma\frac{Q_i}{n}$, is an estimate of the global rate of misclassification - the Bayes error (π). The recursive process of sampling observations and training a classifier that yields, with a high probability, a rate of misclassification near the Bayes error can be thought of as a "Wrapping" Procedure.

The Wrapping Procedure can be summarised as follows:

1. Obtain an initial sample of data.
2. Train the classifier with all available data.
3. Sample one or more additional observations.
4. Classify the observation(s) sampled in step 3 accordingly using the trained classifier. Denote a correct classification with a 0 and an incorrect classification with a 1.
5. Calculate $\widehat{p}$ and evaluate the respective stopping criteria.
6. If the stopping criteria are not satisfied, return to step 2. Otherwise exit the procedure.

The Wrapping Procedure is independent of the method of classification used and produces a list of Bernoulli observations which can then be used to obtain an estimate for π within a particular accuracy level.

3.3 A sequential procedure for the estimation of a proportion

The sequential procedure proposed derives a confidence interval for the proportion of observations misclassified, as the proportion of observations misclassified is an estimate of the Bayes error. Deriving a confidence interval for a proportion p from a series of independent and identically distributed Bernoulli observations is one of the best studied problems in statistics. Many fixed-sample-size and variable-sample-size methods have already been developed and are available. Training a classifier to estimate the Bayes error at a certain level of accuracy, however, depends on a fixed-width confidence interval. A fixed width confidence interval will enable the researcher to specify to what level of accuracy he/she would like the Bayes error to be estimated, while using the minimum number of required observations. In Frey (2010) various methods for obtaining a fixed-width confidence interval for a proportion for sequential procedures are discussed. Specifics of all the methods discussed and specifics of analyses done will not be provided in this dissertation. However, an outline of the method proposed will be provided in what follows as it is the method employed in the proposed sequential approach.

In the derivation of the fixed-width confidence interval estimator, Frey (2010) notes that "fixed-sample-size methods" for obtaining a confidence interval for a proportion p

often "tend to be narrower when p is near 0 or 1 than when p is near $\frac{1}{2}$", stating that these methods require too many observations to estimate p at a predetermined level. Therefore, he proposed a sequential procedure that would ensure a coverage probability of at least $100\,(1-\alpha)\,\%$ for p. Deriving the confidence interval, the author defined h to be the halfwidth of the confidence interval and the confidence interval is centered at $\widehat{p}$ as it is the maximum likelihood estimate of p. If all values outside the interval $[0,1]$ are excluded, a confidence interval for π is provided by $[\max(0,\widehat{p}-h),\min(1,\widehat{p}+h)]$. With the confidence interval defined, a stopping rule needs to be derived.

The first stopping rule considered in Frey (2010) is based on the Wald's confidence interval. Assume X is a binomial random variable denoting the number of successes observed in n independent trails and $\widehat{p}$ then denotes the probability to observe a success. Therefore $X \sim Bin(n,\ \widehat{p})$. From the central limit theorem it is known that $X \sim N(n\widehat{p},\ n\widehat{p}(1-\widehat{p}))$ and $p = \frac{X}{n} \sim N\left(\widehat{p},\ \frac{\widehat{p}(1-\widehat{p})}{n}\right)$ provided n is large enough (Steyn et al., 1982). Therefore

$$Z = \frac{p-\widehat{p}}{\sqrt{\frac{\widehat{p}(1-\widehat{p})}{n}}}$$

and $Z \dot{\sim} N(0,1)$ provided n is large enough.

The $100\,(1-\alpha)\,\%$ Wald's confidence interval for a proportion p is $\widehat{p} \pm z_{\alpha/2}\sqrt{\widehat{p}\,(1-\widehat{p})\,/n}$, $z_{\alpha/2}$ being the upper $\alpha/2^{th}$ percentile of a Standard Gaussian distribution, and $\widehat{p} = \frac{x}{n}$ with x being the number of successes observed. In this particular scenario a success is defined as a misclassification, i.e. $Q_i = 1$. In the specific cases where $x \in \{0,\ n\}$ the confidence interval always has a length of 0 as $\sqrt{\widehat{p}\,(1-\widehat{p})\,/n} = 0$. This is addressed by replacing the estimate of the variance with a non-zero constant $\widetilde{p}_a\,(1-\widetilde{p}_a)\,/n$, where $\widetilde{p}_a \equiv (x+a)\,(n+2a)$, effectively pulling $\widehat{p}$ towards 0.5. For a chosen interval half-width of h and $a > 0$, the sequential procedure is stopped if the adapted Wald's confidence interval $\widehat{p} \pm z_{\alpha/2}\sqrt{\widetilde{p}_a\,(1-\widetilde{p}_a)\,/n}$ in its entirety is contained in the interval $\widehat{p} \pm h$. This is analogous to stopping the sequential procedure when $\widetilde{p}_a\,(1-\widetilde{p}_a)\,/n \leq \left(h/z_{\alpha/2}\right)^2$.

If $a = 0$ the resulting confidence interval is just the nominal Wald's confidence interval for a proportion. Khan (1969) (as cited in Frey, 2010) concluded that stopping rules based on estimates of the Fisher Information worked well when fixed-width confidence intervals were required in continuous settings with nuisance parameters. In this particular scenario the Fisher Information is $n/\left((1-p)\,p\right)$, for which an estimate is $\left(\widetilde{p}_a\,(1-\widetilde{p}_a)\,/n\right)^{-1}$. The stopping criteria for the sequential procedure proposed in this dissertation is based on this adjusted version of the fixed-sample size confidence interval. If this interval is entirely contained in $[\widehat{p}-h;\ \widehat{p}+h]$, where h is the halfwidth of the interval around the Bayes error, the sequential procedure is stopped.

Some of the other stopping rules tested by Frey (2010) were based on :

- The Wilson confidence interval for a proportion p

$$\frac{\widehat{p} + z^2_{\alpha/2}/(2n)}{1 + z^2_{\alpha/2}/n} \pm \frac{z_{\alpha/2}\sqrt{\widehat{p}(1-\widehat{p})/n + z^2_{\alpha/2}/(4n^2)}}{1 + z^2_{\alpha/2}/n}$$

- The Clopper and Pearson confidence interval for a proportion p

For $0 < x < n$

$$[p_l,\ p_u]$$
where
$$\sum_{i=x}^{n} \binom{n}{i} p_l^i (1-p_l)^{n-i} = \alpha/2$$
$$\sum_{i=0}^{x} \binom{n}{i} p_u^i (1-p_u)^{n-i} = \alpha/2$$

For $x = 0$

$$[0,\ p_u]$$

For $x = n$

$$[p_l,\ 1]$$

Frey (2010) notes that the methods used for developing the stopping rules have flaws as fixed-sample size methods. While the Wald's confidence interval actually has a confidence coefficient of 0, the Clopper-Pearson interval's true confidence coefficient often exceeds the nominal level. Consequently, many refined methods have been developed that have a less pronounced difference between the actual confidence coefficient and the nominal confidence coefficient. Since a sequential method negates the greater control of the confidence coefficient, the author chose his specified rules.

To determine which of the sequential confidence interval methods yield the best results, Frey (2010) conducted multiple studies testing different values for h and the coverage probability $100(1-\alpha)\,\%$, calculating the mean average run length, the mean coverage probability, as well as the restricted mean coverage probability.

Frey (2010) notes that the width of the confidence intervals are always $2h$, except when $\widehat{p}$ tends towards either 0 or 1, and consequently the expected length of the confidence intervals is nearly equal for all methods. An alternative measure to consider when gauging performance of a particular method is Average Run Length, and this is defined

as $ARL(p) = \sum_{i=1}^{k} H(x_i,\ n_i)\, p^{x_i}(1-p)^{n_i-x_i}\, n_i$. Integrating this function with regards to p from 0 to 1 yields the average of $ARL(p)$ over all values of p, i.e. $\int_0^1 ARL(p)\, dp$.

The mean coverage probability is defined as $\int_0^1 T(p;\alpha)\, dp$ and is the average $T(p;\ \alpha)$ over all values of p. Frey (2010) notes that the mean coverage probability is close to 1 when $p \in [0,\ h]$ and $p \in [1-h,\ 1]$. Due to this, the restricted mean coverage probability, defined as $(1-2h)^{-1}\int_h^{1-h} T(p;\alpha)\, dp$, is also considered. The integrals are approximated using numerical integration. Frey (2010) found that, while no approach proved to be consistently better than any other over every value of h and a tested, the adapted Wald's confidence interval provided the most flexibility. Adjusting the value of a accordingly always led to better results. Consequently this is the sequential confidence interval method that will be used further, as it is intuitively simple to understand and implement.

Frey (2010) also noted that if an actual coverage probability of $100(1-\alpha)\,\%$ is desired, α should not be chosen equal to the desired value (e.g. $\alpha = 0.05$) as this would result in a confidence coefficient lower than $1-\alpha$. Using the path-counting ideas published by Girshik, Mosteller and Savage (2006) and Schultz et al. (1973) as cited in Frey (2010), based on the fact that the confidence coefficient of α usually decreases as α is increased, Frey (2010) was able to search for a value of α that would yield a confidence coefficient tending towards $1-\alpha$, however not being less than it. New values for a or α will not be derived in this dissertation and therefore the values obtained in Frey (2010) were used. The appropriate values for a and α are also provided for reference purposes in Table D.8.1 of Appendix D.

To derive the critical values, Frey (2010) defined $(x_1,\ n_1)$, ..., $(x_k,\ n_k)$ as the stopping points for a fixed α and a fixed stopping rule, $H(x,\ n)$ (where $n \geq 1$ and $0 \leq x \leq n$) as the number of ways that x successes can be obtained in n trials without reaching or equalling a stopping point prior to the n^{th} trial. Therefore, $H(x_i,\ n_i)\, p^{x_i}(1-p)^{n_i-x_i}$ is the probability to end the procedure at the point $(x_i,\ n_i)$, and $\sum_{i=1}^{k} H(x_i,\ n_i)\, p^{x_i}(1-p)^{n_i-x_i} = 1 \forall p \in [0,\ 1]$. If the stopping point is $(x_i,\ n_i)$, the confidence interval is provided by $\left[\max\left(0, \frac{x_i}{n_i} - h\right),\ \min\left(1, \frac{x_i}{n_i} + h\right)\right]$, with the coverage probability for p defined as

$$T(p;\ \alpha) \equiv \sum_{i=1}^{k} H(x_i,\ n_i)\, p^{x_i}(1-p)^{n_i-x_i}\, I\left(|p - \frac{x_i}{n_i}| \leq h\right)$$

where $I(A)$ is an indicator function for the event A, and the confidence coefficient is derived as $CP(\alpha) = \inf_{p\in[0,1]} T(p;\alpha)$.

As defined by Frey (2010), $T(p;\alpha)$ is a piecewise polynomial in p, with jumps at points in the set $C = \left\{\frac{x_i}{n_i} \pm h,\ i = 1,\ ...,\ k\right\} \cap [0,\ 1]$.

Frey (2010) used a two-step procedure to derive exact critical values. The first step entails, for a fixed α, approximating $CP(\alpha)$ by obtaining the minimum value of $T(p;\ \alpha)$

over a set of values $p = c \pm \epsilon$ where $c \in C$ and $\epsilon = 10^{-10}$. Using the bisection root-finding algorithm, a value for α is obtained that ensures $CP(\alpha)$ tends towards $1 - \alpha$ while never being less than it. The second step entails ensuring that $T(p;\ \alpha)$ is never less than $1 - \alpha$ for the particular choice of α. This is done using a checking algorithm derived in Frey (2010). The algorithm will not be provided in this dissertation.

3.4 Proposed sequential procedure for estimating the Bayes error at a predetermined level

Using this adapted Wald's confidence interval enables the researcher to sequentially sample and train the classifier until the misclassification rate obtained is within h of the Bayes error with a certain probability. Rather than training a classifier to attain a certain misclassification rate, a classifier should be trained to attain the minimum feasible error rate. This will ensure that the classifier is always predicting as well as possible, and the sequential procedure will not continue ad infinitum. Using the adapted Wald's confidence interval derived by Frey (2010), a sequential procedure is proposed where the observed rate of misclassification converges to within h of the Bayes error. The researcher therefore has the ability to specify to what level he/she wishes to estimate the minimum feasible error, i.e. the Bayes error.

A sequential procedure is proposed where the recursive sampling, training and classifying steps are continued until the observed rate of misclassification stabilises within h of the Bayes error. The following algorithm can be used to determine the number of observations needed:

1. Draw an initial random sample of size N_0 and train the relevant classifier.
2. At the i^{th} iteration sample an additional observation and classify the sampled observation accordingly.
3. If the complete adapted Wald's confidence interval is contained in $\widehat{p} \pm h$ the sequential procedure can be stopped, i.e. if $\widehat{p} \pm z_{\alpha/2}\sqrt{\widetilde{p}_a\left(1 - \widetilde{p}_a\right)/n} \in [\widehat{p} - h;\ \widehat{p} + h]$ the sequential procedure can be stopped. This is equivalent to stopping the procedure if $z_{\alpha/2}\sqrt{\widetilde{p}_a\left(1 - \widetilde{p}_a\right)/n} \leq h$.
4. Otherwise retrain the classifier with all observations sampled thus far and thereafter return to step 2 and continue.

Note that this is essentially the Wrapping Procedure previously discussed. Therefore the proposed sequential approach is independent of the method of classification used and only depends on a set of binary input observations.

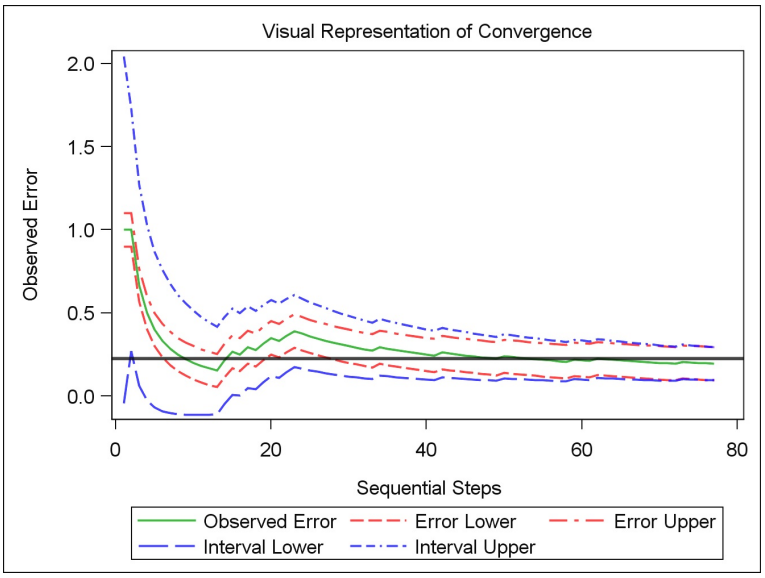


Figure 3.1: *Proposed Sequential Procedure Convergence for* $\Delta = 1.5$, $\alpha = 0.05$ *and* $h = 0.1$.

The convergence process of the adjusted Wald's interval towards the interval $[p - h;\ p + h]$ is provided in Figure 3.1. The solid horisontal black line indicates the theoretical Bayes error, with the two outermost spiked blue curves representing the upper and lower bounds of the adapted Wald's confidence interval, denoted by Interval Upper and Interval Lower respectively. The sequential procedure continues until the two spiked red curves, denoted Error Lower and Error Upper, completely contain the adapted Wald's confidence interval's blue curves. The observed classifier error is denoted by the solid spiked green curve.

The primary advantage of this approach is that it enables the researcher to have more control over the feasible rate of misclassification, at a predetermined level. The classifier can be trained to within h of the minimum error rate obtainable, thereby lending more credibility to an obtained classifier. The researcher therefore has control to decide when the sequential procedure has to terminate, given their experimental specifications or design.

3.5 Simulation study

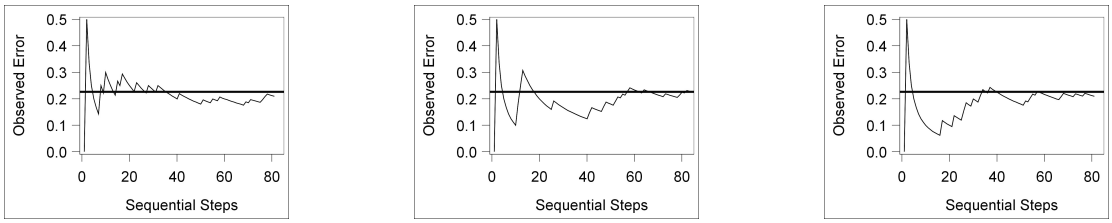
A series of simulations were run to observe the proposed sequential procedure's performance by comparing the observed misclassification error of the trained classifier with the theoretical error, i.e. the Bayes error. The results obtained are not directly comparable to those in Table 2.3.1 as the criteria for stopping the sequential procedures are different, and the method proposed by Fu et al. (2005) had an artificial upper bound imposed on

the number of observations available for sampling. It should still suffice, however, to show how the proposed sequential procedure performs.

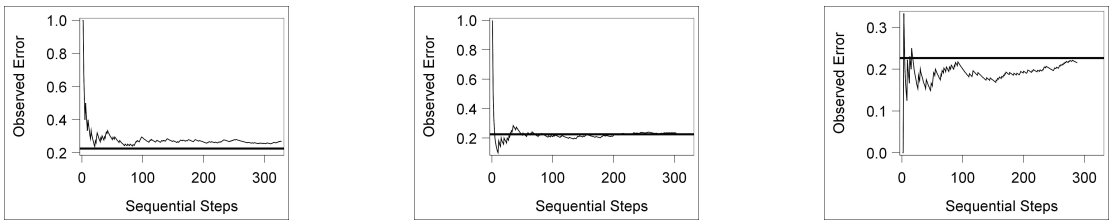
The sequential approach proposed by Fu et al. (2005) evaluates whether the probability of a misclassification is sufficiently small, whereas the proposed approach evaluates whether the adjusted confidence interval for the observed error rate is sufficiently small. It should still suffice however to show the benefits of the proposed sequential procedure. The results of the simulations can be found in Tables 3.5.2 to Tables 3.5.4.

To ensure that the simulation environments were as similar as possible, the same simulation design as employed in Chapter 2 was used. 5 observations were randomly generated from a $N(0,\ 1)$ distribution and 5 additional observations were randomly generated a $N(\Delta,\ 1)$ distribution. Once a sampled observation had been classified accordingly, the adapted Wald's confidence interval was obtained and if the complete interval was contained in $[\widehat{p} - h;\ \widehat{p} + h]$ the sequential procedure was stopped and the number of sequential steps required was recorded. Simulations were run for $h \in \{0.1,\ 0.05,\ 0.01\}$ and $\alpha \in \{0.01,\ 0.05,\ 0.1\}$, and the only classification method used was LDA. The results obtained from LDA and LRIM should be comparable, and as the underlying data is generated from two distributions with equal variances the need to use QDA is negated. If a very small value for h is chosen, the sequential procedure will most likely sample many observations. KNN is not tested in this chapter as it would be very computationally intensive to sort such large matrices for multiple iterations.

h = 0.10



h = 0.05



h = 0.01

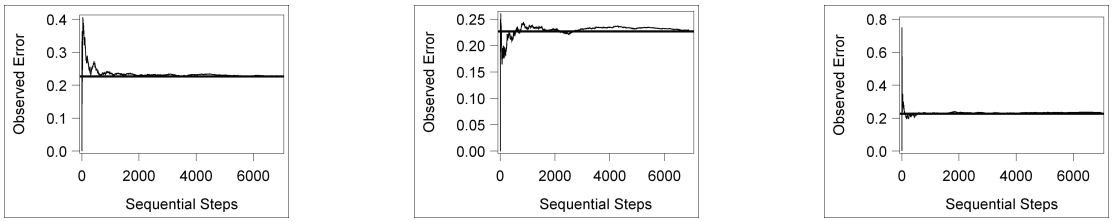


Figure 3.5.1: *Observed classifier error for various values of* h *and* $\alpha = 0.05$

A visual representation of the error rate observed in training the LDA classifier is provided in Figure 3.5.1. The observed classifier error is displayed against the total number of observations classified. The graphs are displayed for 3 iterations of the sequential procedure, for $\alpha = 0.05$, $\Delta = 1.3$ and for $h \in \{0.1, 0.05, 0.01\}$. The solid horisontal line denotes the theoretical Bayes error. As is evident in the graphs, there is a lot of initial variation in the observed rate of misclassification, but as the number of sequential steps increase (and the classifier is better trained) the rate of misclassification converges towards the Bayes error. For a large value of $h = 0.1$ the sequential procedure is stopped before 100 observations are sampled, but as the value of h is decreased to $h = 0.01$ the number of sequential steps required has increased to more than 6000. in all of these cases, though, the classifier is successfully trained and estimates the Bayes error at a predetermined level.

TABLE 3.5.2 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated from a*$N(\Delta,1)$ *distribution and* $\alpha = 0.1$. *The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$h = 0.1$	$h = 0.05$	$h = 0.01$
1	0.3085	Error (Std)	0.3135 (0.009259669)	0.31021 (0.005192205)	0.30863 (0.004592991)
		Min; Max	45; 80	195; 282	5677; 6071
		$\bar{n}$; Sd	70.302; 5.4826	254.73; 12.6219	5879.49; 62.822
1.3	0.2578	Error (Std)	0.26209 (0.008078367)	0.2595 (0.004834442)	0.25811 (0.004376857)
		Min; Max	34; 78	163; 274	5016; 5495
		$\bar{n}$; Sd	65.184; 6.86887	230.503; 16.1511	5280.47; 79.022
1.5	0.2266	Error (Std)	0.23075 (0.006569723)	0.22817 (0.00454338)	0.22666 (0.004228055)
		Min; Max	30; 77	133; 254	4439; 5096
		$\bar{n}$; Sd	61.194; 7.21793	212.825; 17.371	4840.79; 86.876
2	0.1587	Error (Std)	0.16239 (0.006593837)	0.15984 (0.004088071)	0.15871 (0.003642087)
		Min; Max	30; 71	100; 216	3319; 4114
		$\bar{n}$; Sd	52.038; 7.28626	169.932; 19.6716	3698.84; 110.875
2.3	0.1251	Error (Std)	0.12846 (0.005875874)	0.12649 (0.003868138)	0.12512 (0.00331319)
		Min; Max	30; 68	71; 197	2489; 3512
		$\bar{n}$; Sd	47.273; 7.09809	146.095; 20.5516	3052.72; 123.276
2.5	0.1056	Error (Std)	0.10906 (0.005922502)	0.10689 (0.00364893)	0.10601 (0.003167984)
		Min; Max	30; 64	71; 186	2221; 3049
		$\bar{n}$; Sd	44.246; 7.05445	132.634; 19.7482	2652.39; 120.216
3	0.0668	Error (Std)	0.06954 (0.004779777)	0.06792 (0.002998095)	0.06681 (0.002481202)
		Min; Max	30; 59	63; 157	1246; 2247
		$\bar{n}$; Sd	39.116; 5.65991	104.686; 16.7073	1805.66; 138.479
4	0.0228	Error (Std)	0.02426 (0.002619354)	0.02346 (0.001840469)	0.022809 (0.00152069)
		Min; Max	30; 48	63; 118	512; 1162
		$\bar{n}$; Sd	33.037; 3.3134	76.099; 9.726	846.005; 102.639

For $h = 0.1$ in Table 3.5.2, the maximum number of sequential steps previously imposed ($M = 90$) is not exceeded once, yet the classifier is successfully trained for all 1000 iterations. The average number of sequential steps required to successfully train the classifier decreases from 70.302 for $\Delta = 1$ to less than half of that (33.037) for $\Delta = 4$. The average rate of misclassification is fractionally higher than the Bayes error, never being less than it. The standard deviation of the rate of misclassification decreases from 0.0092 for $\Delta = 1$ to 0.0026 for $\Delta = 4$, and are significantly lower than those observed in the initial simulation study using the method proposed by Fu et al. (2005), indicating that the observed rates of misclassification display much less variance.

For $h = 0.05$ the maximum number of sequential steps previously imposed ($M = 90$) is exceeded at least once for all Δ, and is exceeded in all iterations for $\Delta \leq 2$. For $h = 0.01$

the maximum number of sequential steps previously imposed ($M = 90$) is exceeded in all iterations for all values of Δ tested. In all scenarios the average rate of misclassification is near the Bayes error, and as h decreases the standard deviation in the rate of misclassification decreases. Additionally, as h decreases from 0.1 to 0.01 the average number of sequential steps required to successfully train the classifier increases, with the most significant increase evident for $\Delta = 1$.

In all of the scenarios tested, the proposed sequential procedure successfully trained the classifier to yield, with a high probability, a rate of misclassification within h of the Bayes error. The maximum number of sequential steps required to successfully train the classifier was 6071 and was observed for $\Delta = 1$ and $h = 0.01$. Therefore the sequential procedure required at most 6071 observations to ensure, with a probability of at least 90%, that the trained classifier will yield a misclassification rate within only 0.01 of the Bayes error, and on average only about 5880 observations were needed.

TABLE 3.5.3 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated from a*$N(\Delta,1)$ *distribution and* $\alpha = 0.05$. *The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$h = 0.1$	$h = 0.05$	$h = 0.01$
1	0.3085	Error (Std)	0.31251 (0.007410005)	0.30952 (0.004871182)	0.30865 (0.004613307)
		Min; Max	68; 111	307; 392	8067; 8514
		$\bar{n}$; Sd	96.818; 6.67663	354.477; 14.0576	8297.83; 76.344
1.3	0.2578	Error (Std)	0.26154 (0.00698033)	0.25867 (0.004668742)	0.25795 (0.004299132)
		Min; Max	41; 108	247; 373	7135; 7728
		$\bar{n}$; Sd	88.796; 8.37248	321.256; 17.547	7452.12; 97.559
1.5	0.2266	Error (Std)	0.22966 (0.006291187)	0.22767 (0.004460538)	0.22669 (0.004227758)
		Min; Max	45; 106	235; 351	6463; 7157
		$\bar{n}$; Sd	83.048; 9.50316	296.959; 20.2312	6830.21; 107.814
2	0.1587	Error (Std)	0.16159 (0.0056926)	0.15974 (0.00397432)	0.15881 (0.003733498)
		Min; Max	41; 95	152; 301	4763; 5592
		$\bar{n}$; Sd	69.544; 9.40045	236.982; 22.5843	5227.51; 132.366
2.3	0.1251	Error (Std)	0.12786 (0.005471891)	0.1261 (0.003517343)	0.12521 (0.00323402)
		Min; Max	37; 87	143; 273	3894; 4663
		$\bar{n}$; Sd	62.197; 9.22894	205.718; 22.9336	4306.72; 140.696
2.5	0.1056	Error (Std)	0.10832 (0.004958729)	0.10654 (0.003442556)	0.10563 (0.002998168)
		Min; Max	37; 81	117; 256	3161; 4184
		$\bar{n}$; Sd	57.992; 8.88566	186.194; 22.203	3743.74; 154.259
3	0.0668	Error (Std)	0.06908 (0.004303877)	0.06766 (0.002716346)	0.06681 (0.002484673)
		Min; Max	37; 76	91; 217	2001; 3064
		$\bar{n}$; Sd	49.609; 7.45182	147.322; 19.4157	2541.23; 154.344
4	0.0228	Error (Std)	0.02416 (0.002407678)	0.02336 (0.001695299)	0.02286 (0.001523874)
		Min; Max	37; 58	91; 152	747; 1535
		$\bar{n}$; Sd	41.021; 4.21762	108.245; 11.8624	1155.86; 126.71

Changing the level of significance from $\alpha = 0.1$ to $\alpha = 0.05$, as provided in Table 3.5.3, has resulted in an increase in the average number of sequential steps required to successfully train the classifier. The average observed rate of misclassification is slightly lower for $\alpha = 0.05$ than for $\alpha = 0.1$ as the classifier is trained to estimate the Bayes error better. The standard deviation of the observed rate of misclassification is also slightly lower than it is for $\alpha = 0.1$.

For $h = 0.1$ the maximum number of sequential steps previously imposed ($M = 90$) is not exceeded once for $\Delta \geq 2.3$, yet the classifier is trained successfully for all 1000 iterations. The average number of sequential steps required to successfully train the classifier decreases from 96.818 for $\Delta = 1$ to 41.021 for $\Delta = 4$. The average rate of misclassification

is fractionally higher than the Bayes error, albeit lower than the misclassification rate observed for $\alpha = 0.1$. The standard deviation of the rate of misclassification decreases from 0.0074 for $\Delta = 1$ to 0.0023 for $\Delta = 4$.

For $h = 0.05$ the maximum number of sequential steps previously imposed ($M = 90$) is exceeded in all iterations for all Δ tested. For $h = 0.01$ the maximum number of sequential steps previously imposed ($M = 90$) is exceeded in all iterations for all values of Δ tested. In all scenarios the average rate of misclassification is again near the Bayes error, and as h decreases the standard deviation in the rate of misclassification decreases. Additionally, as h decreases from 0.1 to 0.01 the average number of sequential steps required to successfully train the classifier increases, with the most significant increase evident for $\Delta = 1$.

In all of the scenarios tested, the proposed sequential procedure successfully trained the classifier to yield, with a high probability, a rate of misclassification within h of the Bayes error. The maximum number of sequential steps required to successfully train the classifier was 8514 and was observed for $\Delta = 1$ and $h = 0.01$. Therefore the sequential procedure required at most 8514 observations to ensure, with a probability of at least 95%, that the trained classifier will yield a misclassification rate within only 0.01 of the Bayes error, and on average only about 8298 observations were needed.

TABLE 3.5.4 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated from a*$N(\Delta,1)$ *distribution and* $\alpha = 0.01$. *The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$h = 0.1$	$h = 0.05$	$h = 0.01$
1	0.3085	Error (Std)	0.31125 (0.005851223)	0.30916 (0.0046575)	0.3087 (0.004685672)
		Min; Max	122; 179	492; 646	13954; 14604
		$\bar{n}$; Sd	161.021; 8.6423	602.282; 18.5281	14288.45; 97.1696
1.3	0.2578	Error (Std)	0.26004 (0.00532608)	0.25873 (0.004457055)	0.25805 (0.004212218)
		Min; Max	105; 175	456; 609	12329; 13209
		$\bar{n}$; Sd	147.58; 10.8953	544.552; 23.6328	12821.81; 124.198
1.5	0.2266	Error (Std)	0.22867 (0.004922768)	0.22721 (0.004310056)	0.22665 (0.004269492)
		Min; Max	82; 169	421; 577	11344; 12118
		$\bar{n}$; Sd	138.121; 11.3515	502.668; 25.0011	11745.68; 135.154
2	0.1587	Error (Std)	0.16046 (0.004505461)	0.15937 (0.003687151)	0.15873 (0.003740153)
		Min; Max	71; 147	286; 487	8185; 9451
		$\bar{n}$; Sd	113.804; 12.3818	397.828; 30.5143	8985.55; 174.79
2.3	0.1251	Error (Std)	0.127 (0.004437459)	0.12555 (0.003460164)	0.1252 (0.00330591)
		Min; Max	58; 134	224; 435	6804; 7921
		$\bar{n}$; Sd	101.765; 12.6856	341.235; 32.0364	7383.71; 183.92
2.5	0.1056	Error (Std)	0.10725 (0.003938885)	0.10615 (0.003140704)	0.10579 (0.003090732)
		Min; Max	63; 129	190; 384	5692; 7053
		$\bar{n}$; Sd	94.909; 11.5608	306.228; 31.0325	6402.76; 206.229
3	0.0668	Error (Std)	0.06822 (0.003246723)	0.0673 (0.002588731)	0.06685 (0.002584759)
		Min; Max	58; 116	170; 332	3583; 4887
		$\bar{n}$; Sd	80.664; 10.2337	237.479; 27.5531	4332.95; 206.855
4	0.0228	Error (Std)	0.02358 (0.001939936)	0.02312 (0.001592048)	0.022905 (0.001521382)
		Min; Max	58; 89	138; 229	1317; 2433
		$\bar{n}$; Sd	65.255; 5.8563	168.922; 16.243	1917.24; 174.239

Changing the level of significance from $\alpha = 0.05$ to $\alpha = 0.01$, as provided in Table 3.5.4, has resulted in an increase in the average number of sequential steps required to successfully train the classifier, and for the specific scenarios where there is little separation between the underlying distributions the increases are rather significant. The average observed rate of misclassification is generally only slightly lower for $\alpha = 0.01$ than for $\alpha = 0.05$ as the classifier is trained to estimate the Bayes error better. The standard deviation of the observed rate of misclassification is also slightly lower than it is for $\alpha = 0.1$.

For $h = 0.1$ the maximum number of sequential steps previously imposed ($M = 90$) is not exceeded once for $\Delta = 4$, but the maximum is exceeded at least once for all other values of Δ tested. The average number of sequential steps required to successfully train

the classifier decreases from 161.021 for $\Delta = 1$ to 65.255 for $\Delta = 4$. The standard deviation of the rate of misclassification decreases from 0.0058 for $\Delta = 1$ to 0.0019 for $\Delta = 4$.

For $h = 0.05$ the maximum number of sequential steps previously imposed ($M = 90$) is exceeded in all iterations for all Δ tested. For $h = 0.01$ the maximum number of sequential steps previously imposed ($M = 90$) is exceeded in all iterations for all values of Δ tested. In all scenarios the average rate of misclassification is again near the Bayes error, and as h decreases the standard deviation in the rate of misclassification decreases. Additionally, as h decreases from 0.1 to 0.01 the average number of sequential steps required to successfully train the classifier increases, with the most significant increase evident for $\Delta = 1$.

In all of the scenarios tested, the proposed sequential procedure successfully trained the classifier to yield, with a high probability, a rate of misclassification within h of the Bayes error. The average number of sequential steps required to successfully train the classifier has increased significantly in some scenarios, while the corresponding average rate of misclassification has decreased only slightly. Therefore, it seems unnecessary to use a 1% level of significance as the improvement in the rate of misclassification is somewhat out-weighed by the increase in the required number of observations.

3.6 Microarray sample data application

The sequential procedure proposed is also applied to the Microarray dataset used by Van't Veer et al. (2002) and Van de Vijver et al. (2002). The proposed sequential approach will truly train the classifier to be as feasibly accurate as the researcher decides. The classifier will be trained sequentially until there is a high probability that the obtained classifier yields an error rate that is within a predetermined level of the minimum possible error.

To test the proposed sequential procedure's performance on the sample data, the same study design as noted previously was used. The only classification method tested was LDA and the results are provided in Table 3.6.1. The results presented in Table 3.6.1 are not directly comparable to those presented in Table 2.5.1 as the latter have an artificially imposed upper limit on the allowable number of observations to be sampled ($M \in \{50;\ 80\}$). The results presented in Table 3.6.1 should still display the proposed procedure's ability to optimally train the classifier using the least number of observations. As there are no artificial upper limits imposed on the allowable number of sequential steps, the procedure is allowed to train the classifier to estimate the Bayes error to within a certain level.

TABLE 3.6.1 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LDA classifier on the sample data. The initial sample sizes are provided.* $\alpha = 0.05$ *and the minimum, maximum, mean and standard deviation of the number of sequential steps required to train the LDA classifier on the* 3 *genes most highly correlated to the prognosis variable is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Initial Sample Sizes	h		$\alpha = 0.1$	$\alpha = 0.05$	$\alpha = 0.01$
$(5, 5)$	0.1	Error (Std)	0.23994 (0.02424)	0.23694 (0.02541)	0.23409 (0.03176)
		Min; Max	30; 77	52; 104	105; 164
		$\bar{n}$; Sd	63.331; 6.0271	86.804; 7.1331	141.414; 8.7104
$(5, 5)$	0.05	Error (Std)	0.23502 (0.05111)	0.26718 (0.27862)	. (.)
		Min; Max	175; 244	267; 295	295; 295
		$\bar{n}$; Sd	219.452; 11.0024	294.424; 3.1327	295; 0
(5.5)	0.01	Error (Std)	. (.)	. (.)	. (.)
		Min; Max	295; 295	295; 295	295; 295
		$\bar{n}$; Sd	295; 0	295; 0	295; 0
(10.10)	0.1	Error (Std)	0.23842 (0.02422)	0.23688 (0.0272)	0.23427 (0.0332)
		Min; Max	40; 78	52; 105	105; 164
		$\bar{n}$; Sd	62.743; 5.9115	85.772; 7.6089	140.482; 9.3986
(10.10)	0.05	Error (Std)	0.23618 (0.0558)	0.22959 (0.23132)	. (.)
		Min; Max	170; 250	247; 295	295; 295
		$\bar{n}$; Sd	218.457; 10.733	294.814; 2.33728	295; 0
(10.10)	0.01	Error (Std)	. (.)	. (.)	. (.)
		Min; Max	295; 295	295; 295	295; 295
		$\bar{n}$; Sd	295; 0	295; 0	295; 0

For the same h and α in Table 3.6.1, the average number of sequential steps required to successfully train the classifier does not seem to be influenced by the number of observations initially sampled. A similar result was evident in Table 2.5.1. As h decreases, for the same α, the average rate of misclassification also decreases - provided an iteration of the simulation did not use all observations available to try and train the classifier.

In the specific example where the initial number of observations sampled are $(5,\ 5)$ and $\alpha = 0.05$, the sequential procedure reached the maximum number of observations available for training the classifier more than once during the simulation. Consequently - the classifiers were not successfully trained in these cases. Additionally - no observations remained to test against. It is also evident in scenarios where $h = 0.01$ that the sample was too small to successfully train any classifiers, irrespective of the initial samples selected or their respective sizes.

Comparing the results in Table 3.6.1 to Table 2.5.1, the average rate of misclassification is similar but the average number of sequential steps required to successfully train the classifier is significantly larger. This is due to the fact that no artificial upper limits were imposed on the allowable number of sampled observations.

In this particular scenario the proposed sequential procedure allows the researcher to optimally train the classifier based on the researcher's study design. It provides the researcher with control over how accurately the classifier should estimate the minimum feasible error. The proposed sequential procedure will also never continue to train the classifier indefinitely to try and obtain an unfeasibly low level of error.

3.7 Conclusion

In this chapter, a sequential procedure, developed to sample the minimum number of observations required to train a classifier that would yield an error rate comparable to the Bayes error, was proposed and discussed. The sequential procedure proposed by Fu et al. (2005) tries to train a classifier that yields a prespecified probability of misclassifying the next randomly sampled observation, while sampling only the minimum required number of observations. If the error rate specified is too low, as can easily be the case considering that the Bayes error is often not known, the sequential procedure proposed by Fu et al. (2005) will not stop. The sequential procedure proposed in this dissertation, however, trains the classifier to such an extent that the error rate obtained is within h of the minimum feasible error rate, i.e. the Bayes error, with a high probability. A series of binary observations are returned from each iteration of the sequential procedure and used to obtain the rate of misclassification - an estimate of the Bayes error. A variance adjusted confidence interval for the population rate of misclassification is obtained, and once the variance adjusted confidence interval is completely contained in a fixed width confidence interval for the rate of misclassification, the sequential procedure is stopped. Consequently, the researcher now has the ability to specify how accurate the classifier should be and the sequential procedure proposed can not train the classifier to yield an unfeasible rate of error.

Considering that the most frequently used level of significance is $\alpha = 0.05$, it is interesting to note, from the simulation study, that a mere 8300 observations are on average necessary to train a LDA classifier that will yield a rate of misclassification within just 0.01 of the Bayes error. Usually researchers prefer to use as much data as they have available (costs and time already taken into account) and implicitly trust that using most of the data will yield the most accurate classifier - the researcher does not specify how accurate the classifier should be. This result, however, shows that large amounts of data

are not necessarily needed to obtain an accurate classifier, and using less data to derive a classifier that is as accurate as need be is of great value. Not only will using less data save processing time, but it could have a significant impact on the cost of the study. Additionally, if the cost of sampling more observations is not as important, the observations not used in classifier training can be used as testing data. One of the most important features of the proposed sequential procedure is the ability it provides to the researcher to specify his/her own acceptable level of accuracy, and it removes the possibility of choosing an unfeasibly low level of error.

The sequential procedure has multiple advantages: it evaluates the stopping criteria after each iteration, thereby ensuring the procedure will not sample observations unnecessarily once the stopping criteria have been met; it is not dependent on one method of classification, but rather depends only on a series of binary input variables, i.e. variables indicating whether a sampled observation was incorrectly (denoted 1) or correctly (denoted 0) classified; it recursively obtains a trained classifier that yields an optimised rate of misclassification with a high probability; it enables the researcher to dictate how accurately the Bayes error should be estimated.

The sequential procedure proposed does, however, have a few minor shortcomings. Due to the recursive nature of the process, computationally intensive classification methods like KNN take substantially longer to complete. This, however, is true of any sequential procedure and does not only apply to the proposed sequential procedure. Another shortcoming of the procedure currently is that the critical values α and a need to be specified beforehand. It is, however, possible to include the necessary steps to derive these values beforehand. This would enable the researcher to only specify the values of h and α.

The sequential procedure was proposed to aid in scenarios where data is not freely available, possibly due to high costs (in gathering data or the cost of a misclassification) or for some other reason, so that the classifier could be trained with the least amount of data to ensure, with a high enough probability of at least $100\,(1-\alpha)\,\%$, that after the classifier had been successfully trained the probability of the very next observation sampled being incorrectly classified is comparable to the minimum possible error, i.e. the Bayes error. The proposed sequential procedure addresses the most serious shortcoming of the sequential procedure proposed by Fu et al. (2005), and the proposed sequential procedure will never attempt to train the classifier to obtain an unfeasible rate of error. Therefore, the sequential procedure will always yield a result - provided h is large enough.

Chapter 4

General conclusion

Determining the correct sample size is of utmost importance, as samples that are too small will yield unreliable results. Samples that are very large will often yield parameters or estimates that have greater predictive ability. In scenarios where there is no or little cost involved, the researcher dictates to what level of accuracy he/she would like the parameter to be estimated. Numerous fixed sample size methods have been developed, like Stein's two stage sampling procedure, to aid in estimating the necessary sample size required to obtain the parameter estimate within a prespecified level of accuracy.

In scientific studies, for example, there are often limits imposed on the number of observations available to train a classifier or estimate an unknown parameter. These limits are usually determined by the financial cost involved in obtaining the necessary observations, as well as the time involved. In these scenarios the researcher may wish to still obtain an accurate estimate for the unknown parameter while minimising the number of observations required to obtain the estimate. This could ultimately save both time and money. There are numerous sequential procedures available that will seek the minimum number of observations required to meet some predetermined stopping criteria. One such an approach had previously been developed that would continually train and test a classifier until there was a large enough probability that the probability of the next observation sampled being classified incorrectly was acceptably small.

The sequential procedure could train a classifier until there was a high level of certainty that an additionally sampled observation would most likely not be misclassified. The procedure is independent of the method of classification used. Therefore any classification method could be used, provided a series of dichotomous observations could be obtained. Unfortunately the sequential procedure can not account for scenarios where the maximum allowable rate of error (as dictated by the researcher) is unfeasibly low (i.e. lower than the Bayes error), ultimately resulting in the sequential procedure processing ad infinitum. The procedure occasionally samples too many observations and trains the classifier to obtain a level of error smaller than the specified value. In specific scenarios where a large cost is involved in sampling or false-positives, the procedure provides unsatisfactory results.

An alternative sequential procedure is developed that trains the classifier to an optimal level. The classifier is trained until the observed rate of error has converged to the Bayes error - the theoretical minimum error rate. The proposed sequential procedure is independent of the method of classification used and depends only on a series of Bernoulli observations. The proposed sequential procedure will never train the classifier to obtain an unfeasibly low rate of error. This procedure provides the researcher with a feasible way to train the classifier, knowing that the classifier obtained will perform as well as is feasibly possible from the underlying data.

The application for large datasets is just as interesting. The proposed sequential procedure provides the researcher with a measure of how accurate the obtained classifier actually is. Although more data is usually preferable, it might no longer be necessary to use as much data. The researcher is able to dictate how accurately he/she wishes the classifier be trained, and the obtained classifier will always perform at an optimal level - estimating the Bayes error.

Future work could comprise of including the necessary logic to automatically obtain the critical values α and a, as these values need to be specified currently. Additionally, the procedure could be tested on the other classification methods (QDA, KNN and LRIM) not tested here. A simulation could also be conducted using the sequential procedure proposed by Fu et al. (2005) to test the specific scenario where ε is marginally larger than the Bayes error. Currently, the artificial upper bounds imposed, both M and N_0, inhibit the sequential procedure from successfully training the classifier to ensure, with a probability of at least $100\,(1-\alpha)\,\%$, that the probability of the next observation sampled being misclassified is at most ε. The actual unimpeded number of required observations should be compared to the number obtained from the newly proposed sequential procedure.

Bibliography

BERGER, J. O. 1985. Statistical decision theory and Bayesian analysis, Springer.

CHATTERJEE, S. 2010. Spin glasses and Stein's method. Probability theory and related fields, 148, 567-600.

CHOW, Y. S. & ROBBINS, H. 1965. On the asymptotic theory of fixed-width sequential confidence intervals for the mean. The Annals of Mathematical Statistics, 36, 457-462.

DANTZIG, G. B. 1940. On the non-existence of tests of "Student's" hypothesis having power functions independent of sigma. The Annals of Mathematical Statistics, 11, 186-192

FREY, J. 2010. Fixed-width sequential confidence intervals for a proportion. The American Statistician, 64.

FU, W. J., DOUGHERTY, E. R., MALLICK, B. & CARROLL, R. J. 2005. How many samples are needed to build a classifier: a general sequential approach. Bioinformatics, 21, 63-70.

GIRSHICK, M., MOSTELLER, F. & SAVAGE, L. 2006. Unbiased estimates for certain binomial sampling problems with applications. Selected Papers of Frederick Mosteller. Springer.

HASTIE, T., TIBSHIRANI, R. & FRIEDMAN, J. J. H. 2001. The elements of statistical learning, Springer New York.

KHAN, R. A. 1969. A general method of determining fixed-width confidence intervals. The Annals of Mathematical Statistics, 704-709.

MUKHOPADHYAY, N. 1980. A consistent and asymptotically efficient two-stage procedure to construct fixed width confidence intervals for the mean. Metrika, 27, 281-284.

RAY, W. 1957. Sequential confidence intervals for the mean of a normal population with unknown variance. Journal of the Royal Statistical Society. Series B (Methodological), 133-143.

SCHULTZ, J., NICHOL, F., ELFRING, G. & WEED, S. 1973. Multiple-stage procedures for drug screening. Biometrics, 293-300.

STEIN, C. 1945. A two-sample test for a linear hypothesis whose power is independent of the variance. The Annals of Mathematical Statistics, 16, 243-258.

STEYN, A. G. W., SMIT, C. F. & DU TOIT, S. H. C. 1982. Moderne statistiek vir die praktyk, van Schaik

VAN'T VEER, L. J., DAI, H., VAN DE VIJVER, M. J., HE, Y. D., HART, A. A., MAO, M., PETERSE, H. L., VAN DER KOOY, K., MARTON, M. J. & WITTEVEEN, A. T. 2002. Gene expression profiling predicts clinical outcome of breast cancer. nature, 415, 530-536.

VAN DE VIJVER, M. J., HE, Y. D., VAN'T VEER, L. J., DAI, H., HART, A. A., VOSKUIL, D. W., SCHREIBER, G. J., PETERSE, J. L., ROBERTS, C. & MARTON, M. J. 2002. A gene-expression signature as a predictor of survival in breast cancer. New England Journal of Medicine, 347, 1999-2009.

Appendix A

Linear discriminant analysis (LDA)

A.1 Obtaining the linear discriminant function

From statistical decision theory, it is known that the optimal classification can be obtained if the class posteriors are known, i.e. if $P(class\ G|X = x)$ is known. If $f_k(x)$ is the class-conditional density of X for class $G = k$, the prior probability of class k is denoted by π_k and $\Sigma_{k=1}^{K}\pi_k = 1$, where K is the total number of classes, using the Bayes Theorem the class posterior for class k can be written as:

$$P(G = k|X = \underline{x}) = \frac{f_k(\underline{x})\pi_k}{\Sigma_{l=1}^{K} f_l(\underline{x})\pi_l}$$

If each class density is then modelled as a multivariate Gaussian, the density function of class k is given by:

$$f_k(\underline{x}) = \frac{1}{(2\pi)^{p/2}\,|\mathbf{\Sigma}_k|^{1/2}} e^{-\frac{1}{2}\left(\underline{x}-\underline{\mu}_k\right)^T \mathbf{\Sigma}_k^{-1}\left(\underline{x}-\underline{\mu}_k\right)}$$

where $\underline{\mu}_k$ is a $p \times 1$ matrix that denotes the population mean for class k, $\mathbf{\Sigma}_k$ is a $p \times p$ matrix denoting the population covariance matrix for class k, $\underline{v}^T$ denotes the transpose of $\underline{v}$ and $\underline{x}$ is a p-dimensional row vector.

Consider now the specific case where all the classes have a common covariance matrix $\mathbf{\Sigma}_k = \mathbf{\Sigma} \forall k$.

Hastie, Tibshirani and Friedman (2001) notes that looking at the log ratio of the class posteriors is sufficient for comparing any two classes, k and l.

Therefore the following holds:

$$\begin{aligned}
\log \frac{P\left(G=k|X=\underline{x}\right)}{P\left(G=l|X=\underline{x}\right)} &= \log \frac{\frac{f_k(\underline{x})\pi_k}{\Sigma_{j=1}^K f_j(\underline{x})\pi_j}}{\frac{f_l(\underline{x})\pi_l}{\Sigma_{j=1}^K f_j(\underline{x})\pi_j}} \\
&= \log \frac{f_k(\underline{x})\pi_k}{f_l(\underline{x})\pi_l} \\
&= \log\left(\frac{f_k(\underline{x})}{f_l(\underline{x})}\right) + \log\left(\frac{\pi_k}{\pi_l}\right) \\
&= \log\left(\frac{\pi_k}{\pi_l}\right) + \\
&\quad \log\left(\left(\frac{e^{-\frac{1}{2}\left(\underline{x}-\underline{\mu}_k\right)^T\mathbf{\Sigma}_k^{-1}\left(\underline{x}-\underline{\mu}_k\right)}}{(2\pi)^{p/2}\left|\mathbf{\Sigma}_k\right|^{1/2}}\right) / \left(\frac{e^{-\frac{1}{2}\left(\underline{x}-\underline{\mu}_l\right)^T\mathbf{\Sigma}_l^{-1}\left(\underline{x}-\underline{\mu}_l\right)}}{(2\pi)^{p/2}\left|\mathbf{\Sigma}_l\right|^{1/2}}\right)\right) \\
&= \log\left(\frac{\pi_k}{\pi_l}\right) + \log\left(\left(\frac{(2\pi)^{p/2}\left|\mathbf{\Sigma}_l\right|^{1/2}}{(2\pi)^{p/2}\left|\mathbf{\Sigma}_k\right|^{1/2}}\right)\left(\frac{e^{-\frac{1}{2}\left(\underline{x}-\underline{\mu}_k\right)^T\mathbf{\Sigma}_k^{-1}\left(\underline{x}-\underline{\mu}_k\right)}}{e^{-\frac{1}{2}\left(\underline{x}-\underline{\mu}_l\right)^T\mathbf{\Sigma}_l^{-1}\left(\underline{x}-\underline{\mu}_l\right)}}\right)\right) \\
&= \log\left(\frac{\pi_k}{\pi_l}\right) + \log\left(\left(\frac{(2\pi)^{p/2}\left|\mathbf{\Sigma}\right|^{1/2}}{(2\pi)^{p/2}\left|\mathbf{\Sigma}\right|^{1/2}}\right)\left(\frac{e^{-\frac{1}{2}\left(\underline{x}-\underline{\mu}_k\right)^T\mathbf{\Sigma}^{-1}\left(\underline{x}-\underline{\mu}_k\right)}}{e^{-\frac{1}{2}\left(\underline{x}-\underline{\mu}_l\right)^T\mathbf{\Sigma}^{-1}\left(\underline{x}-\underline{\mu}_l\right)}}\right)\right) \\
&\quad \text{since } \Sigma_k = \Sigma \forall k \\
&= \log\left(\frac{\pi_k}{\pi_l}\right) + \log\left(e^{-\frac{1}{2}\left(\underline{x}-\underline{\mu}_k\right)^T\mathbf{\Sigma}^{-1}\left(\underline{x}-\underline{\mu}_k\right)+\frac{1}{2}\left(\underline{x}-\underline{\mu}_l\right)^T\mathbf{\Sigma}^{-1}\left(\underline{x}-\underline{\mu}_l\right)}\right) \\
&= \log\left(\frac{\pi_k}{\pi_l}\right) - \\
&\quad \frac{1}{2}\left(\underline{x}-\underline{\mu}_k\right)^T\mathbf{\Sigma}^{-1}\left(\underline{x}-\underline{\mu}_k\right) + \frac{1}{2}\left(\underline{x}-\underline{\mu}_l\right)^T\mathbf{\Sigma}^{-1}\left(\underline{x}-\underline{\mu}_l\right) \\
&= \log\left(\frac{\pi_k}{\pi_l}\right) - \frac{1}{2}\left(\underline{x}^T\mathbf{\Sigma}^{-1}\underline{x} - \underline{x}^T\mathbf{\Sigma}^{-1}\underline{\mu}_k - \underline{\mu}_k^T\mathbf{\Sigma}^{-1}\underline{x} + \underline{\mu}_k^T\mathbf{\Sigma}^{-1}\underline{\mu}_k\right) \\
&\quad + \frac{1}{2}\left(\underline{x}^T\mathbf{\Sigma}^{-1}\underline{x} - \underline{x}^T\mathbf{\Sigma}^{-1}\underline{\mu}_l - \underline{\mu}_l^T\mathbf{\Sigma}^{-1}\underline{x} + \underline{\mu}_l^T\mathbf{\Sigma}^{-1}\underline{\mu}_l\right) \\
&= \log\left(\frac{\pi_k}{\pi_l}\right) - \frac{1}{2}\left(\underline{\mu}_k + \underline{\mu}_l\right)^T\mathbf{\Sigma}^{-1}\left(\underline{\mu}_k - \underline{\mu}_l\right) + \underline{x}^T\mathbf{\Sigma}^{-1}\left(\underline{\mu}_k - \underline{\mu}_l\right)
\end{aligned}$$

which is a linear equation in x. The linear discriminant functions can be defined as:

$$\delta_k\left(\underline{x}\right) = \underline{x}^T\mathbf{\Sigma}^{-1}\underline{\mu}_k - \frac{1}{2}\underline{\mu}_k^T\mathbf{\Sigma}^{-1}\underline{\mu}_k + \log\pi_k$$

If the parameters are not known, sample estimates for $\underline{\pi}_k$, $\underline{\mu}_k$ and $\mathbf{\Sigma}$ can be used. Therefore:

1. $\widehat{\pi}_k = \frac{N_k}{\Sigma_{l=1}^K N_l}$
2. $\underline{\widehat{\mu}}_k = \frac{\Sigma_{x\in k}\underline{x}}{N_k}$
3. $\widehat{\mathbf{\Sigma}} = \Sigma_{k=1}^K \Sigma_{x\in k}(\underline{x} - \underline{\widehat{\mu}}_k)(\underline{x} - \underline{\widehat{\mu}}_k)^T/(N-K)$

A.2 Determining the decision boundary between class 1 and 2 using LDA

LDA assumes that $\Sigma_k = \Sigma \ \forall k$ where k are the classes. Assume there are two classes, denoted 1 and 2.

The linear discriminant function for class 1 is given by:

$$\delta_1(\underline{x}) = \underline{x}^T \mathbf{\Sigma}^{-1} \underline{\mu}_1 - \frac{1}{2} \underline{\mu}_1^T \mathbf{\Sigma}^{-1} \underline{\mu}_1 + \log(\pi_1)$$

where $X \sim N(\underline{\mu}_1, \mathbf{\Sigma}_1^2)$ and π_1 is the probability of selecting a random variable from class 1.

Similar for class 2 the linear discriminant function for class 2 is given by :

$$\delta_2(\underline{x}) = \underline{x}^T \mathbf{\Sigma}^{-1} \underline{\mu}_2 - \frac{1}{2} \underline{\mu}_2^T \mathbf{\Sigma}^{-1} \underline{\mu}_2 + \log(\pi_2)$$

where $X \sim N(\underline{\mu}_2, \mathbf{\Sigma}_2^2)$ and π_2 is the probability of selecting a random variable from class 2.

For both class 1 and 2, assume the multidimensional space - R^p.
Setting $\delta_1(\underline{x}) = \delta_2(\underline{x})$ the following holds:

$$\begin{aligned}
\delta_1(\underline{x}) &= \delta_2(\underline{x}) \\
\underline{x}^T \mathbf{\Sigma}^{-1} \underline{\mu}_1 - \frac{1}{2} \underline{\mu}_1 \mathbf{\Sigma}^{-1} \underline{\mu}_1 + \log(\pi_1) &= \underline{x}^T \mathbf{\Sigma}^{-1} \underline{\mu}_2 - \frac{1}{2} \underline{\mu}_2^T \mathbf{\Sigma}^{-1} \underline{\mu}_2 + \log(\pi_2) \\
\underline{x}^T \mathbf{\Sigma}^{-1} \underline{\mu}_1 - \underline{x}^T \mathbf{\Sigma}^{-1} \underline{\mu}_2 &= \log(\pi_2) - \log(\pi_1) + \frac{1}{2} \left(\underline{\mu}_1^T \mathbf{\Sigma}^{-1} \underline{\mu}_1 - \underline{\mu}_2^T \mathbf{\Sigma}^{-1} \underline{\mu}_2 \right) \\
\underline{x}^T \mathbf{\Sigma}^{-1} \left(\underline{\mu}_1 - \underline{\mu}_2 \right) &= \log(\frac{\pi_2}{\pi_1}) + \frac{1}{2} \left(\underline{\mu}_1^T \mathbf{\Sigma}^{-1} \underline{\mu}_1 - \underline{\mu}_2^T \mathbf{\Sigma}^{-1} \underline{\mu}_2 \right)
\end{aligned}$$

where an observation is classified as coming from class 1 if

$$\underline{x}^T \mathbf{\Sigma}^{-1} \left(\underline{\mu}_1 - \underline{\mu}_2 \right) > \log(\frac{\pi_2}{\pi_1}) + \frac{1}{2} \left(\underline{\mu}_1^T \mathbf{\Sigma}^{-1} \underline{\mu}_1 - \underline{\mu}_2^T \mathbf{\Sigma}^{-1} \underline{\mu}_2 \right)$$

and from class 2 otherwise.

Alternatively, an observation is classified as coming from class 2 if

$$\underline{x}^T \mathbf{\Sigma}^{-1} \left(\underline{\mu}_2 - \underline{\mu}_1 \right) \geq \log(\frac{\pi_1}{\pi_2}) + \frac{1}{2} \left(\underline{\mu}_2^T \mathbf{\Sigma}^{-1} \underline{\mu}_2 - \underline{\mu}_1^T \mathbf{\Sigma}^{-1} \underline{\mu}_1 \right)$$

and from class 1 otherwise.

π_1 can be estimated by $\widehat{p}_1 = \frac{n_1}{n_1+n_2}$. Similarly π_2 can be estimated by $\widehat{p}_2 = \frac{n_2}{n_1+n_2}$. $\underline{\mu}_1$ and $\underline{\mu}_2$ can be estimated from the sample means for class 1 and class 2 respectively, namely $\widehat{\underline{\mu}}_1$ and $\widehat{\underline{\mu}}_2$ respectively. Similarly $\widehat{\mathbf{\Sigma}} = \Sigma_{k=1}^{2}\Sigma_{x\in k}(\underline{x} - \widehat{\underline{\mu}}_k)(\underline{x} - \widehat{\underline{\mu}}_k)^T/(N-2)$.

This leads to the following:

Classify an observation as coming from class 2 if:

$$\begin{aligned} \underline{x}^T\widehat{\mathbf{\Sigma}}^{-1}\left(\widehat{\underline{\mu}}_2 - \widehat{\underline{\mu}}_1\right) &\geq \log\left(\frac{\widehat{p}_1}{\widehat{p}_2}\right) + \frac{1}{2}\left(\widehat{\underline{\mu}}_2^T\widehat{\mathbf{\Sigma}}^{-1}\widehat{\underline{\mu}}_2 - \widehat{\underline{\mu}}_1^T\widehat{\mathbf{\Sigma}}^{-1}\widehat{\underline{\mu}}_1\right) \\ x &\geq \left(\log\left(\frac{\widehat{p}_1}{\widehat{p}_2}\right) + \frac{1}{2}\left(\widehat{\underline{\mu}}_2^T\widehat{\mathbf{\Sigma}}^{-1}\widehat{\underline{\mu}}_2 - \widehat{\underline{\mu}}_1^T\widehat{\mathbf{\Sigma}}^{-1}\widehat{\underline{\mu}}_1\right)\right)\left(\widehat{\mathbf{\Sigma}}^{-1}\left(\widehat{\underline{\mu}}_2 - \widehat{\underline{\mu}}_1\right)\right)^{-1} \\ &\quad \text{provided } \widehat{\mathbf{\Sigma}}^{-1}\left(\widehat{\underline{\mu}}_2 - \widehat{\underline{\mu}}_1\right) \text{ is invertible} \end{aligned}$$

and coming from class 1 otherwise.

In the case where the prior probabilities are known:

Let $\widehat{p}_1 = \widehat{p}_2 = 0.5$. Classify an observation as coming from class 2 if:

$$\begin{aligned} \underline{x}^T\widehat{\mathbf{\Sigma}}^{-1}\left(\widehat{\underline{\mu}}_2 - \widehat{\underline{\mu}}_1\right) &\geq \log\left(\frac{\widehat{p}_1}{\widehat{p}_2}\right) + \frac{1}{2}\left(\widehat{\underline{\mu}}_2^T\widehat{\mathbf{\Sigma}}^{-1}\widehat{\underline{\mu}}_2 - \widehat{\underline{\mu}}_1^T\widehat{\mathbf{\Sigma}}^{-1}\widehat{\underline{\mu}}_1\right) \\ &= \log(1) + \frac{1}{2}\left(\widehat{\underline{\mu}}_2^T\widehat{\mathbf{\Sigma}}^{-1}\widehat{\underline{\mu}}_2 - \widehat{\underline{\mu}}_1^T\widehat{\mathbf{\Sigma}}^{-1}\widehat{\underline{\mu}}_1\right) \\ &= \frac{1}{2}\left(\widehat{\underline{\mu}}_2^T\widehat{\mathbf{\Sigma}}^{-1}\widehat{\underline{\mu}}_2 - \widehat{\underline{\mu}}_1^T\widehat{\mathbf{\Sigma}}^{-1}\widehat{\underline{\mu}}_1\right) \end{aligned}$$

Consider now the case where $p = 1$. For ease of use $\widehat{\mathbf{\Sigma}}$ can be written as $\widehat{\sigma^2}$. Classify an observation as coming from class 2 if:

$$\begin{aligned} x &\geq \frac{\left(\widehat{\mu}_2^2\left(\widehat{\sigma^2}\right)^{-1} - \widehat{\mu}_1^2\left(\widehat{\sigma^2}\right)^{-1}\right)}{2\left(\widehat{\sigma^2}\right)^{-1}(\widehat{\mu}_2 - \widehat{\mu}_1)} \\ &= \frac{\left(\widehat{\mu}_2^2 - \widehat{\mu}_1^2\right)}{2\left(\widehat{\mu}_2 - \widehat{\mu}_1\right)} \\ &= \widehat{\lambda} \end{aligned}$$

Appendix B

Quadratic discriminant analysis (QDA)

B.1 Obtaining the quadratic discriminant function

$$
\begin{aligned}
\log \frac{P\left(G=k|X=\underline{x}\right)}{P\left(G=l|X=\underline{x}\right)} &= \log \frac{f_k(\underline{x})\pi_k}{f_l(\underline{x})\pi_l} \\
&= \log\left(\frac{\pi_k}{\pi_l}\right) + \log\left(\left(\frac{(2\pi)^{p/2}\left|\mathbf{\Sigma}_l\right|^{1/2}}{(2\pi)^{p/2}\left|\mathbf{\Sigma}_k\right|^{1/2}}\right)\left(\frac{e^{-\frac{1}{2}\left(\underline{x}-\underline{\mu}_k\right)^T\mathbf{\Sigma}_k^{-1}\left(\underline{x}-\underline{\mu}_k\right)}}{e^{-\frac{1}{2}\left(\underline{x}-\underline{\mu}_l\right)^T\mathbf{\Sigma}_l^{-1}\left(\underline{x}-\underline{\mu}_l\right)}}\right)\right) \\
&= \log\left(\frac{\pi_k}{\pi_l}\right) + \log\left(\frac{(2\pi)^{p/2}\left|\mathbf{\Sigma}_l\right|^{1/2}}{(2\pi)^{p/2}\left|\mathbf{\Sigma}_k\right|^{1/2}}\right) + \log\left(\frac{e^{-\frac{1}{2}\left(\underline{x}-\underline{\mu}_k\right)^T\mathbf{\Sigma}_k^{-1}\left(\underline{x}-\underline{\mu}_k\right)}}{e^{-\frac{1}{2}\left(\underline{x}-\underline{\mu}_l\right)^T\mathbf{\Sigma}_l^{-1}\left(\underline{x}-\underline{\mu}_l\right)}}\right) \\
&= \log\pi_k - \log\pi_l + \log\left(\frac{\left|\mathbf{\Sigma}_l\right|^{1/2}}{\left|\mathbf{\Sigma}_k\right|^{1/2}}\right) + \log\left(e^{-\frac{1}{2}\left(\underline{x}-\underline{\mu}_k\right)^T\mathbf{\Sigma}_k^{-1}\left(\underline{x}-\underline{\mu}_k\right)}\right) \\
&\quad - \log\left(e^{-\frac{1}{2}\left(\underline{x}-\underline{\mu}_l\right)^T\mathbf{\Sigma}_l^{-1}\left(\underline{x}-\underline{\mu}_l\right)}\right) \\
&= \log\pi_k - \log\pi_l + \frac{1}{2}\log\left|\mathbf{\Sigma}_l\right| - \frac{1}{2}\log\left|\mathbf{\Sigma}_k\right| - \\
&\quad \frac{1}{2}\left(\underline{x}-\underline{\mu}_k\right)^T\mathbf{\Sigma}_k^{-1}\left(\underline{x}-\underline{\mu}_k\right) + \frac{1}{2}\left(\underline{x}-\underline{\mu}_l\right)^T\mathbf{\Sigma}_l^{-1}\left(\underline{x}-\underline{\mu}_l\right) \\
&= \left(\log\pi_k - \frac{1}{2}\log\left|\mathbf{\Sigma}_k\right| - \frac{1}{2}\left(\underline{x}-\underline{\mu}_k\right)^T\mathbf{\Sigma}_k^{-1}\left(\underline{x}-\underline{\mu}_k\right)\right) \\
&\quad - \left(\log\pi_l - \frac{1}{2}\log\left|\mathbf{\Sigma}_l\right| - \frac{1}{2}\left(\underline{x}-\underline{\mu}_l\right)^T\mathbf{\Sigma}_l^{-1}\left(\underline{x}-\underline{\mu}_l\right)\right)
\end{aligned}
$$

leading to the quadratic discriminant functions:

$$\delta_k\left(\underline{x}\right) = \log\pi_k - \frac{1}{2}\log\left|\mathbf{\Sigma}_k\right| - \frac{1}{2}\left(\underline{x}-\underline{\mu}_k\right)^T\mathbf{\Sigma}_k^{-1}\left(\underline{x}-\underline{\mu}_k\right)$$

If the number of dimensions p in R^p is large the number of parameters can increase substantially. If the necessary parameters are not known, sample estimates for π_k, $\underline{\mu}_k$ and $\mathbf{\Sigma}_k$ can again be used.

B.2 Determining the decision boundary between class 1 and 2 using QDA

QDA does not have the assumption that $\mathbf{\Sigma}_k = \mathbf{\Sigma}\ \forall k$, as LDA does. Assume there are two classes, denoted 1 and 2.

The linear discriminant function for class 1 is given by:

$$\delta_1(\underline{x}) = \log \pi_1 - \frac{1}{2}\log|\mathbf{\Sigma}_1| - \frac{1}{2}\left(\underline{x} - \underline{\mu}_1\right)^T \mathbf{\Sigma}_1^{-1}\left(\underline{x} - \underline{\mu}_1\right)$$

where $X \sim N(\underline{\mu}_1, \mathbf{\Sigma}_1^2)$ and π_1 is the probability of selecting a random variable from class 1.

Similar for class 2 the linear discriminant function for class 2 is given by :

$$\delta_2(\underline{x}) = \log \pi_2 - \frac{1}{2}\log|\mathbf{\Sigma}_2| - \frac{1}{2}\left(\underline{x} - \underline{\mu}_2\right)^T \mathbf{\Sigma}_2^{-1}\left(\underline{x} - \underline{\mu}_2\right)$$

where $X \sim N(\underline{\mu}_2, \mathbf{\Sigma}_2^2)$ and π_2 is the probability of selecting a random variable from class 2.

For both class 1 and 2, assume the multidimensional space - R^p.
Setting $\delta_1(\underline{x}) = \delta_2(\underline{x})$ the following holds:

$$\delta_1(x) = \delta_2(x)$$

$$\log \pi_1 - \frac{1}{2}\log|\mathbf{\Sigma}_1| - \frac{1}{2}\left(\underline{x} - \underline{\mu}_1\right)^T \mathbf{\Sigma}_1^{-1}\left(\underline{x} - \underline{\mu}_1\right) = \log \pi_2 - \frac{1}{2}\log|\mathbf{\Sigma}_2| - \frac{1}{2}\left(\underline{x} - \underline{\mu}_2\right)^T \mathbf{\Sigma}_2^{-1}\left(\underline{x} - \underline{\mu}_2\right)$$

$$\frac{1}{2}\left[\left(\underline{x} - \underline{\mu}_2\right)^T \mathbf{\Sigma}_2^{-1}\left(\underline{x} - \underline{\mu}_2\right) - \left(\underline{x} - \underline{\mu}_1\right)^T \mathbf{\Sigma}_1^{-1}\left(\underline{x} - \underline{\mu}_1\right)\right] = \log \pi_2 - \frac{1}{2}\log|\mathbf{\Sigma}_2| - \log \pi_1 + \frac{1}{2}\log|\mathbf{\Sigma}_1|$$

where an observation is classified as coming from class 1 if

$$\frac{1}{2}\left[\left(\underline{x} - \underline{\mu}_2\right)^T \mathbf{\Sigma}_2^{-1}\left(\underline{x} - \underline{\mu}_2\right) - \left(\underline{x} - \underline{\mu}_1\right)^T \mathbf{\Sigma}_1^{-1}\left(\underline{x} - \underline{\mu}_1\right)\right] > \log \pi_2 - \frac{1}{2}\log|\mathbf{\Sigma}_2| - \log \pi_1 + \frac{1}{2}\log|\mathbf{\Sigma}_1|$$

and from class 2 otherwise.

Alternatively, an observation is classified as coming from class 2 if

$$\frac{1}{2}\left[\left(\underline{x} - \underline{\mu}_1\right)^T \mathbf{\Sigma}_1^{-1}\left(\underline{x} - \underline{\mu}_1\right) - \left(\underline{x} - \underline{\mu}_2\right)^T \mathbf{\Sigma}_2^{-1}\left(\underline{x} - \underline{\mu}_2\right)\right] \geq \log \frac{\pi_1}{\pi_2} - \frac{1}{2}\log|\mathbf{\Sigma}_1| + \frac{1}{2}\log|\mathbf{\Sigma}_2|$$

, and from class 1 otherwise.

π_1 can be estimated by $\widehat{p}_1 = \frac{n_1}{n_1+n_2}$. Similarly π_2 can be estimated by $\widehat{p}_2 = \frac{n_2}{n_1+n_2}$. $\underline{\mu}_1$ and $\underline{\mu}_2$ can be estimated from the sample means for class 1 and class 2 respectively, namely $\widehat{\underline{\mu}}_1$ and $\widehat{\underline{\mu}}_2$ respectively. Similarly $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ can be estimated from the sample covariance matrices for class 1 and class 2 respectively, namely $\widehat{\mathbf{\Sigma}}_1$ and $\widehat{\mathbf{\Sigma}}_2$ respectively.

This leads to the following:

Classify an observation as coming from class 2 if:

$$\frac{1}{2}\left[\left(\underline{x}-\widehat{\underline{\mu}}_1\right)^T\widehat{\mathbf{\Sigma}}_1^{-1}\left(\underline{x}-\widehat{\underline{\mu}}_1\right)-\left(\underline{x}-\widehat{\underline{\mu}}_2\right)^T\widehat{\mathbf{\Sigma}}_2^{-1}\left(\underline{x}-\widehat{\underline{\mu}}_2\right)\right] \geq \log\frac{\widehat{p}_1}{\widehat{p}_2}-\frac{1}{2}\log\left|\widehat{\mathbf{\Sigma}}_1\right|+\frac{1}{2}\log\left|\widehat{\mathbf{\Sigma}}_2\right|$$

and coming from class 1 otherwise.

In the case where the prior probabilities are known:

Let $\widehat{p}_1 = \widehat{p}_2 = 0.5$. Classify an observation as coming from class 2 if:

$$\begin{aligned}\frac{1}{2}\left[\left(\underline{x}-\widehat{\underline{\mu}}_1\right)^T\widehat{\mathbf{\Sigma}}_1^{-1}\left(\underline{x}-\widehat{\underline{\mu}}_1\right)-\left(\underline{x}-\widehat{\underline{\mu}}_2\right)^T\widehat{\mathbf{\Sigma}}_2^{-1}\left(\underline{x}-\widehat{\underline{\mu}}_1\right)\right] &\geq \log\frac{\widehat{p}_1}{\widehat{p}_2}-\frac{1}{2}\log\left|\widehat{\mathbf{\Sigma}}_1\right|+\frac{1}{2}\log\left|\widehat{\mathbf{\Sigma}}_2\right| \\ &= \log(1)-\frac{1}{2}\log\left|\widehat{\mathbf{\Sigma}}_1\right|+\frac{1}{2}\log\left|\widehat{\mathbf{\Sigma}}_2\right| \\ &= \frac{1}{2}\left(\log\left|\widehat{\mathbf{\Sigma}}_2\right|-\log\left|\widehat{\mathbf{\Sigma}}_1\right|\right)\end{aligned}$$

Consider now the case where $p = 1$. For ease of use $\widehat{\Sigma}_1$ can be written as s_1^2, and $\widehat{\Sigma}_2$ can be written as s_2^2.

Classify an observation as coming from class 2 if:

$$\frac{1}{2}\left[\left(\underline{x}-\widehat{\underline{\mu}}_1\right)^T\left(s_1^2\right)^{-1}\left(\underline{x}-\widehat{\underline{\mu}}_1\right)-\left(\underline{x}-\widehat{\underline{\mu}}_2\right)^T\left(s_2^2\right)^{-1}\left(\underline{x}-\widehat{\underline{\mu}}_2\right)\right] \geq \frac{1}{2}\left(\log\left|s_2^2\right|-\log\left|s_1^2\right|\right)$$

Since the underlying distributions are Gaussian and the parameters are known, the optimal classifier for use in the Bayes error calculation can be determined using the equations derived above, and the optimal classifier in the single dimension case with different sample variances is then given by:

$$\frac{1}{2}\left[\left(\lambda-\mu_1\right)^T\sigma_1^{-2}\left(\lambda-\mu_1\right)-\left(\lambda-\mu_2\right)\sigma_2^{-2}\left(\lambda-\mu_2\right)\right] = \log\frac{\pi_1}{\pi_2}-\frac{1}{2}\log\left(\sigma_1^2\right)+\frac{1}{2}\log\left(\sigma_2^2\right)$$

therefore

$$\left(\lambda^2-2\lambda\mu_1+\mu_1^2\right)\left(\sigma_1^2\right)^{-1}-\left(\lambda^2-2\lambda\mu_2+\mu_2^2\right)\left(\sigma_2^2\right)^{-1}-2\log\left(\frac{\pi_1}{\pi_2}\right)+\log\left(\frac{\sigma_1^2}{\sigma_2^2}\right)=0$$

$$\left(\frac{1}{\sigma_1^2}-\frac{1}{\sigma_2^2}\right)\lambda^2-2\left(\frac{\mu_1}{\sigma_1^2}-\frac{\mu_2}{\sigma_2^2}\right)\lambda+\left(-2\log\left(\frac{\pi_1}{\pi_2}\right)+\log\left(\frac{\sigma_1^2}{\sigma_2^2}\right)+\frac{\mu_1^2}{\sigma_1^2}-\frac{\mu_2^2}{\sigma_2^2}\right)=0$$

which is a quadratic function of λ. Lambda can be solved as follows:

$$\lambda = \frac{2\left(\frac{\mu_1}{\sigma_1^2} - \frac{\mu_2}{\sigma_2^2}\right) \pm \sqrt[2]{4\left(\frac{\mu_1}{\sigma_1^2} - \frac{\mu_2}{\sigma_2^2}\right)^2 - 4\left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}\right)\left(-2\log\left(\frac{\pi_1}{\pi_2}\right) + \log\left(\frac{\sigma_1^2}{\sigma_2^2}\right) + \frac{\mu_1^2}{\sigma_1^2} - \frac{\mu_2^2}{\sigma_2^2}\right)}}{2\left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}\right)}$$

resulting in two different roots for λ.

Define

$$L(\lambda) = \left(\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}\right)\lambda^2 - 2\left(\frac{\mu_1}{\sigma_1^2} - \frac{\mu_2}{\sigma_2^2}\right)\lambda + \left(-2\log\left(\frac{\pi_1}{\pi_2}\right) + \log\left(\frac{\sigma_1^2}{\sigma_2^2}\right) + \frac{\mu_1^2}{\sigma_1^2} - \frac{\mu_2^2}{\sigma_2^2}\right)$$

The first derivative of $L(\lambda)$ with regards to λ in the respective roots will show whether a particular root is a local minimum or maximum. By choosing the root that maximises $\frac{\partial L}{\partial \lambda}$ the optimal classifier is chosen.

Appendix C

Linear regression of an indicator matrix (LRIM)

Consider the case where $\underline{\beta} = \{\beta_0,\ \beta_1,\ ...,\ \beta_p\}$, and let $\mathbf{X}$ be a $n \times (p+1)$ matrix with the first column populated with 1's, and the remaining columns populated with the p-dimensions of the x values. For row i, the following holds:

$$e_i = y_i - \underline{x_i}\underline{\beta}$$

or in matrix notation

$$\underline{e} = \underline{y} - \mathbf{X}\underline{\beta}$$

Define $O = \Sigma_{i=1}^{n} e_i^2 = \Sigma_{i=1}^{n} \left(y_i - \beta_0 - x_{i1}\beta_1 - ... - x_{ip}\beta_p\right)^2 = \left(\underline{y} - \mathbf{X}\underline{\beta}\right)^T \left(\underline{y} - \mathbf{X}\underline{\beta}\right)$, the objective function that needs to be minimised. Taking the derivative with respect to β, setting the resulting equation equal to 0 and solving the equation, will yield the parameter estimate. This is done by using the product-rule, as follows:

$$\frac{\partial O}{\partial \underline{\beta}} = -\mathbf{X}^T \left(\underline{y} - \mathbf{X}\underline{\beta}\right) - \left(\underline{y} - \mathbf{X}\underline{\beta}\right)^T \mathbf{X}$$

Therefore

$$\begin{aligned} 0 &= -\mathbf{X}^T \left(\underline{y} - \mathbf{X}\underline{\widehat{\beta}}\right) - \left(\underline{y} - \mathbf{X}\underline{\widehat{\beta}}\right)^T \mathbf{X} \\ &= -2\mathbf{X}^T \left(\underline{y} - \mathbf{X}\underline{\widehat{\beta}}\right) \\ &= \mathbf{X}^T\underline{y} - \mathbf{X}^T\mathbf{X}\underline{\widehat{\beta}} \\ \mathbf{X}^T\mathbf{X}\underline{\widehat{\beta}} &= \mathbf{X}^T\underline{y} \\ \underline{\widehat{\beta}} &= \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\underline{y} \end{aligned}$$

provided $\mathbf{X}^T\mathbf{X}$ is invertible. This is the case as there can be no linear dependence in the regressors, i.e. $rank(\mathbf{X}) = n$.

Appendix D

Sequential procedure results and critical values

D.1 LDA results

TABLE D.1.1 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated from a* $N(\Delta,2)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 90 *(i.e.* $M = 90$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32771	Error (Std)	0.34213 (0.018359)	0.34146 (0.017922)	0.34237 (0.020012)	0.34359 (0.023878)
		Min; Max	90; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	90; 0	88.065; 12.0943	79.852; 26.6529	69.485; 34.5361
1.3	0.28633	Error (Std)	0.29478 (0.012652)	0.29545 (0.013126)	0.2968 (0.017072)	0.29883 (0.022177)
		Min; Max	25; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.935; 2.0555	84.691; 19.415	75.823; 30.1001	58.663; 37.9973
1.5	0.25982	Error (Std)	0.26723 (0.011468)	0.26682 (0.011181)	0.2675 (0.013486)	0.27079 (0.01965)
		Min; Max	22; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.67; 4.6599	83.559; 21.2761	71.541; 32.6515	52.911; 37.982
2	0.19885	Error (Std)	0.20239 (0.00682)	0.20339 (0.009471)	0.20579 (0.011638)	0.20703 (0.014133)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	88.825; 8.6931	74.512; 30.34	53.219; 37.756	34.685; 33.5103
2.3	0.16656	Error (Std)	0.16973 (0.005759)	0.17095 (0.008537)	0.17382 (0.012125)	0.17512 (0.013123)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	86.523; 14.6285	65.512; 34.6576	43.868; 35.9818	25.023; 26.3481
2.5	0.14697	Error (Std)	0.14973 (0.005744)	0.15168 (0.008366)	0.15395 (0.011929)	0.15577 (0.017156)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	84.669; 17.7922	60.158; 35.4467	36.04; 33.052	19.885; 20.9906
3	0.10485	Error (Std)	0.10818 (0.006095)	0.1112 (0.010245)	0.11307 (0.011942)	0.11391 (0.013291)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	77.831; 24.7271	46.582; 34.4316	24; 22.423	13.482; 10.1931
4	0.04789	Error (Std)	0.05335 (0.006692)	0.05432 (0.0084)	0.05543 (0.010144)	0.05622 (0.0104)
		Min; Max	21; 90	11; 90	8; 69	6; 33
		$\bar{n}$; Sd	54.778; 28.6004	25.465; 15.8907	15.564; 6.133	11.724; 3.5686

TABLE D.1.2 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated from a* $N(\Delta,3)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 90 *(i.e.* $M = 90$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32919	Error (Std)	0.3575 (0.031626)	0.35978 (0.034427)	0.35908 (0.03513)	0.35816 (0.032694)
		Min; Max	90; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	90; 0	88.376; 11.0963	81.886; 24.1202	73.101; 32.1264
1.3	0.29473	Error (Std)	0.31238 (0.022352)	0.3133 (0.022889)	0.31356 (0.026254)	0.31514 (0.030256)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.931; 2.182	86.873; 15.1419	78.263; 28.0213	65.061; 36.33
1.5	0.27214	Error (Std)	0.28638 (0.018654)	0.28628 (0.018269)	0.28731 (0.02077)	0.28934 (0.024393)
		Min; Max	22; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.804; 3.5789	84.235; 20.2695	74.882; 31.0399	57.445; 37.9089
2	0.21843	Error (Std)	0.22441 (0.009906)	0.22574 (0.011788)	0.22694 (0.015744)	0.22883 (0.017745)
		Min; Max	22; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.532; 5.5771	77.057; 28.5359	59.896; 36.852	40.789; 36.0522
2.3	0.18885	Error (Std)	0.19281 (0.00739)	0.19435 (0.009445)	0.19709 (0.013318)	0.19923 (0.020051)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	88.89; 8.4582	73.926; 30.5681	49.805; 37.529	30.576; 30.9091
2.5	0.17042	Error (Std)	0.17433 (0.006829)	0.17553 (0.010075)	0.17857 (0.015888)	0.18002 (0.015846)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	86.659; 14.3207	69.083; 33.1024	44.7; 36.3731	26.969; 28.3265
3	0.12928	Error (Std)	0.13314 (0.006612)	0.13567 (0.010111)	0.13703 (0.011958)	0.13918 (0.013587)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	82.949; 20.0131	57.069; 35.7983	31.257; 29.5961	17.554; 17.9315
4	0.06833	Error (Std)	0.0767 (0.009122)	0.07919 (0.012792)	0.08035 (0.013991)	0.08032 (0.014431)
		Min; Max	21; 90	11; 90	8; 90	6; 56
		$\bar{n}$; Sd	64.857; 29.661	30.366; 24.0243	16.694; 11.2796	12.117; 5.8521

TABLE D.1.3 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated from a* $N(\Delta,4)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 90 *(i.e.* $M = 90$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32743	Error (Std)	0.36906 (0.044901)	0.36699 (0.040876)	0.36692 (0.043322)	0.36877 (0.047113)
		Min; Max	90; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	90; 0	88.612; 10.262	82.951; 22.5522	71.266; 33.7254
1.3	0.29754	Error (Std)	0.32529 (0.032975)	0.32646 (0.034348)	0.32706 (0.035409)	0.324 (0.033601)
		Min; Max	22; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.932; 2.1503	86.374; 16.3413	78.833; 27.5635	66.075; 35.9166
1.5	0.27772	Error (Std)	0.29935 (0.027887)	0.29891 (0.026571)	0.29959 (0.028251)	0.30143 (0.033257)
		Min; Max	23; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.815; 3.3853	84.331; 20.0784	76.066; 29.853	60.788; 37.3334
2	0.22973	Error (Std)	0.24053 (0.017303)	0.24005 (0.015219)	0.2423 (0.01997)	0.24436 (0.02434)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.673; 4.6184	79.781; 25.7797	65.799; 35.3784	46.477; 37.3938
2.3	0.20265	Error (Std)	0.20917 (0.010605)	0.21092 (0.015785)	0.21206 (0.016624)	0.21454 (0.020225)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	88.721; 9.0039	76.887; 28.4162	56.575; 37.4452	37.992; 34.6864
2.5	0.18551	Error (Std)	0.19031 (0.008284)	0.19219 (0.010959)	0.19466 (0.014394)	0.1976 (0.021479)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	88.467; 10.0035	71.239; 32.3081	52.018; 37.6832	31.561; 31.6372
3	0.14628	Error (Std)	0.15051 (0.006941)	0.15321 (0.011234)	0.15525 (0.013058)	0.15703 (0.015483)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	85.513; 16.5186	62.028; 35.2244	35.803; 32.8645	21.802; 22.2585
4	0.08473	Error (Std)	0.09431 (0.011011)	0.09689 (0.013444)	0.09875 (0.016726)	0.09912 (0.015896)
		Min; Max	21; 90	11; 90	8; 90	6; 68
		$\bar{n}$; Sd	72.975; 27.7846	38.084; 30.4163	19.609; 16.7255	11.863; 6.2081

TABLE D.1.4 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated from a* $N(\Delta,2)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 40 *(i.e.* $M = 40$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32771	Error (Std)	0.34668 (0.027005)	0.35077 (0.033684)	0.3492 (0.031919)	0.35045 (0.034353)
		Min; Max	40; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	40; 0	39.186; 4.641	36.04; 10.2068	32.841; 13.1466
1.3	0.28633	Error (Std)	0.29933 (0.019455)	0.30037 (0.019765)	0.30102 (0.023275)	0.30178 (0.027489)
		Min; Max	22; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.982; 0.5692	38.563; 6.0639	34.311; 11.7361	29.316; 14.6251
1.5	0.25982	Error (Std)	0.27008 (0.015047)	0.2706 (0.017171)	0.27307 (0.019948)	0.27189 (0.019654)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.919; 1.1721	37.25; 8.1275	33.354; 12.3946	27.261; 15.0468
2	0.19885	Error (Std)	0.20522 (0.011071)	0.20636 (0.012408)	0.20729 (0.016768)	0.20814 (0.014681)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.708; 2.1595	34.726; 10.5284	28.298; 14.1609	21.607; 14.5378
2.3	0.16656	Error (Std)	0.17228 (0.009381)	0.17356 (0.011724)	0.17482 (0.013791)	0.17557 (0.014464)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.231; 3.492	32.783; 11.5275	24.962; 13.9632	18.184; 13.2302
2.5	0.14697	Error (Std)	0.15185 (0.00769)	0.1527 (0.009323)	0.15455 (0.012507)	0.15628 (0.013643)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	38.891; 3.9561	31.212; 11.983	23.078; 13.5826	16.391; 11.7048
3	0.10485	Error (Std)	0.10945 (0.007478)	0.1113 (0.00974)	0.11323 (0.01263)	0.11367 (0.012913)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	37.43; 5.7397	27.163; 12.1542	18.011; 10.868	13.961; 8.8593
4	0.04789	Error (Std)	0.05316 (0.006441)	0.05492 (0.008822)	0.05571 (0.009841)	0.05556 (0.009773)
		Min; Max	21; 40	11; 40	8; 40	6; 39
		$\bar{n}$; Sd	34.237; 7.1215	22.638; 8.8796	15.803; 5.9511	11.886; 3.9054

TABLE D.1.5 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated from a* $N(\Delta,3)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 40 *(i.e.* $M = 40$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32919	Error (Std)	0.36347 (0.042796)	0.36595 (0.04442)	0.36462 (0.043172)	0.36615 (0.045298)
		Min; Max	40; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	40; 0	39.265; 4.4266	37.095; 8.9149	33.274; 13.026
1.3	0.29473	Error (Std)	0.32022 (0.036081)	0.3199 (0.036628)	0.32355 (0.039007)	0.32116 (0.039056)
		Min; Max	22; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.982; 0.5692	38.449; 6.2792	35.805; 10.3589	31.699; 13.8307
1.5	0.27214	Error (Std)	0.29092 (0.028227)	0.29279 (0.031916)	0.29236 (0.031114)	0.29068 (0.027591)
		Min; Max	40; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	40; 0	38.164; 6.791	33.523; 12.298	29.064; 14.7981
2	0.21843	Error (Std)	0.23049 (0.019357)	0.22993 (0.01717)	0.23017 (0.018992)	0.23103 (0.019863)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.632; 2.4377	36.118; 9.4156	29.814; 13.7881	24.411; 15.1074
2.3	0.18885	Error (Std)	0.19685 (0.014729)	0.19655 (0.011935)	0.19861 (0.014529)	0.20049 (0.019473)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.641; 2.3937	34.542; 10.5863	27.333; 14.2596	21.007; 14.3842
2.5	0.17042	Error (Std)	0.17667 (0.011154)	0.17812 (0.012498)	0.17974 (0.015986)	0.18051 (0.015711)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.287; 3.296	33.543; 11.1652	25.882; 14.2651	18.933; 13.4643
3	0.12928	Error (Std)	0.135 (0.008405)	0.13672 (0.01133)	0.13888 (0.013959)	0.13987 (0.015023)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	38.622; 4.4286	29.693; 12.0907	20.715; 12.7546	14.629; 10.0176
4	0.06833	Error (Std)	0.07707 (0.010142)	0.0787 (0.011643)	0.07958 (0.013122)	0.08039 (0.014611)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	34.977; 7.0508	23.532; 10.5689	15.872; 7.9147	11.853; 5.0538

TABLE D.1.6 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated from a* $N(\Delta,4)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 40 *(i.e.* $M = 40$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32743	Error (Std)	0.37961 (0.055843)	0.37962 (0.055753)	0.37982 (0.055878)	0.37635 (0.05814)
		Min; Max	40; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	40; 0	39.331; 4.2189	37.564; 8.2501	33.642; 12.6193
1.3	0.29754	Error (Std)	0.33395 (0.047255)	0.33545 (0.047369)	0.3343 (0.047524)	0.33577 (0.04843)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.966; 0.7651	38.833; 5.4861	35.864; 10.2969	31.716; 13.8854
1.5	0.27772	Error (Std)	0.30925 (0.043605)	0.30712 (0.040461)	0.30917 (0.044479)	0.30666 (0.04258)
		Min; Max	40; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	40; 0	38.737; 5.6712	34.278; 11.7532	30.519; 14.235
2	0.22973	Error (Std)	0.24539 (0.025884)	0.2473 (0.031235)	0.24705 (0.027879)	0.24782 (0.029647)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.838; 1.6341	36.621; 8.8369	31.893; 13.0557	25.837; 15.2271
2.3	0.20265	Error (Std)	0.21385 (0.020064)	0.21488 (0.019533)	0.2158 (0.022856)	0.21744 (0.022402)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.787; 1.7666	35.114; 10.1687	28.508; 14.1268	22.066; 14.7935
2.5	0.18551	Error (Std)	0.19453 (0.016391)	0.19464 (0.015508)	0.1974 (0.019603)	0.19828 (0.021302)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.712; 2.21	34.485; 10.7583	28.002; 14.1218	20.542; 14.3107
3	0.14628	Error (Std)	0.15296 (0.009778)	0.15514 (0.013816)	0.15706 (0.016693)	0.15724 (0.016253)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	38.96; 3.9958	31.415; 11.9876	23.304; 13.7932	16.33; 11.8058
4	0.08473	Error (Std)	0.09429 (0.010871)	0.09719 (0.013907)	0.09805 (0.015412)	0.0991 (0.016281)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	36.634; 6.3081	25.539; 11.7311	16.641; 9.4992	12.537; 6.8928

D.2 QDA results

TABLE D.2.1 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the QDA classifier is given below. 5 initial observations were generated from a $N(0,1)$ distribution and 5 additional observations were generated a $N(\Delta,2)$ distribution, $\alpha = 0.05$ and the maximum number of sequential steps is 90 (i.e. $M = 90$). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max, $\bar{n}$ and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32771	Error (Std)	0.33453 (0.010855)	0.33455 (0.01186)	0.33681 (0.01559)	0.33862 (0.019479)
		Min; Max	23; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.933; 2.1187	88.304; 11.3182	82.844; 22.6474	71.278; 33.1351
1.3	0.28633	Error (Std)	0.29188 (0.009462)	0.29342 (0.012437)	0.29442 (0.015079)	0.29806 (0.022251)
		Min; Max	23; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.801; 3.63	86.067; 16.8015	75.378; 30.4173	62.159; 37.1646
1.5	0.25982	Error (Std)	0.26488 (0.008434)	0.26491 (0.008664)	0.26679 (0.013398)	0.27122 (0.023493)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.598; 5.1781	84.505; 19.7587	73.873; 31.1175	57.044; 37.3739
2	0.19885	Error (Std)	0.20203 (0.005463)	0.20344 (0.010275)	0.20689 (0.01701)	0.21075 (0.021042)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	88.699; 9.1262	75.652; 29.3991	57.469; 36.8005	35.941; 33.7879
2.3	0.16656	Error (Std)	0.16959 (0.005793)	0.17142 (0.011234)	0.17592 (0.017874)	0.17802 (0.019448)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	86.683; 14.2895	69.512; 32.9141	45.823; 36.2616	27.626; 28.0359
2.5	0.14697	Error (Std)	0.14999 (0.005355)	0.15227 (0.011003)	0.15657 (0.018465)	0.15899 (0.020766)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	85.16; 16.9747	63.892; 34.664	37.951; 33.4661	21; 22.041
3	0.10485	Error (Std)	0.10736 (0.00493)	0.11119 (0.011602)	0.11377 (0.015354)	0.11788 (0.019991)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	79.473; 23.4577	47.392; 34.3341	24.845; 22.9883	14.533; 12.6255
4	0.04789	Error (Std)	0.05163 (0.006898)	0.05419 (0.011979)	0.05747 (0.017973)	0.05945 (0.020641)
		Min; Max	21; 90	11; 90	8; 90	6; 44
		$\bar{n}$; Sd	55.933; 28.5638	25.996; 16.4854	16.351; 8.2681	11.91; 4.1595

TABLE D.2.2 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the QDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated a* $N(\Delta,3)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 90 *(i.e.* $M = 90$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32919	Error (Std)	0.32764 (0.009462)	0.32816 (0.010577)	0.32921 (0.01703)	0.33253 (0.019323)
		Min; Max	24; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.934; 2.0871	87.219; 14.4033	82.343; 23.4365	69.098; 34.5981
1.3	0.29473	Error (Std)	0.29676 (0.007636)	0.29746 (0.009432)	0.29941 (0.013281)	0.30245 (0.017837)
		Min; Max	27; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.874; 2.816	86.76; 15.4976	76.916; 29.3799	64.179; 36.5078
1.5	0.27214	Error (Std)	0.27585 (0.008488)	0.27585 (0.009083)	0.27855 (0.016207)	0.28206 (0.019422)
		Min; Max	22; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.932; 2.1503	84.977; 18.9702	74.636; 30.9374	59.878; 37.0008
2	0.21843	Error (Std)	0.22193 (0.007074)	0.224 (0.011136)	0.22675 (0.017317)	0.22963 (0.021516)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	88.669; 9.3272	79.322; 26.3339	64.513; 35.4401	44.142; 36.5686
2.3	0.18885	Error (Std)	0.1923 (0.005892)	0.19418 (0.010661)	0.19699 (0.017032)	0.20051 (0.021067)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	88.063; 11.0399	74.885; 29.8532	53.278; 37.3805	35.911; 33.016
2.5	0.17042	Error (Std)	0.17332 (0.005243)	0.17554 (0.010344)	0.1795 (0.018371)	0.18286 (0.01963)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	87.535; 12.4624	70.781; 32.1873	46.983; 36.3242	27.771; 27.8828
3	0.12928	Error (Std)	0.13181 (0.005755)	0.13636 (0.015396)	0.13883 (0.016785)	0.14269 (0.023183)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	83.751; 18.8746	57.975; 34.9321	32.89; 30.3712	18.943; 18.9039
4	0.06833	Error (Std)	0.07154 (0.005797)	0.07574 (0.014191)	0.07833 (0.0168)	0.08116 (0.023903)
		Min; Max	21; 90	11; 90	8; 90	6; 61
		$\bar{n}$; Sd	67.111; 29.1485	31.531; 24.5383	17.177; 11.2866	12.464; 6.6285

TABLE D.2.3 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the QDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated a* $N(\Delta,4)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 90 *(i.e.* $M = 90$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32743	Error (Std)	0.31194 (0.009211)	0.31351 (0.012472)	0.31472 (0.013347)	0.31751 (0.018714)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.931; 2.182	87.213; 14.4318	80.795; 25.2825	67.589; 34.9202
1.3	0.29754	Error (Std)	0.29097 (0.008797)	0.29172 (0.010829)	0.29352 (0.013153)	0.29601 (0.019326)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.798; 3.6849	86.525; 16.0217	77.346; 28.7818	66.091; 35.3665
1.5	0.27772	Error (Std)	0.27437 (0.007577)	0.27525 (0.009885)	0.27769 (0.013128)	0.28134 (0.020095)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.865; 3.0179	85.202; 18.5662	75.042; 30.4093	58.975; 37.6196
2	0.22973	Error (Std)	0.23128 (0.006429)	0.23307 (0.013839)	0.23592 (0.016343)	0.24065 (0.021995)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.484; 5.7615	81.416; 23.9031	65.031; 35.8512	46.577; 37.1275
2.3	0.20265	Error (Std)	0.20506 (0.006029)	0.20694 (0.011691)	0.2101 (0.015902)	0.21485 (0.021891)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.28; 6.857	76.736; 28.5393	57.56; 37.2625	37.639; 34.6961
2.5	0.18551	Error (Std)	0.18879 (0.006833)	0.19065 (0.011716)	0.19476 (0.01737)	0.19906 (0.024365)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	88.304; 10.4046	74.005; 30.7121	51.44; 37.0648	32.878; 31.7466
3	0.14628	Error (Std)	0.14897 (0.00507)	0.15211 (0.012537)	0.15642 (0.018796)	0.15984 (0.021086)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	86.046; 15.4518	64.316; 34.8455	38.914; 33.5584	21.24; 22.2746
4	0.08473	Error (Std)	0.08819 (0.007174)	0.09268 (0.015164)	0.09662 (0.021407)	0.09879 (0.021733)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	70.009; 29.1724	37.571; 29.7439	19.75; 16.3559	12.986; 9.0941

TABLE D.2.4 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the QDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated a* $N(\Delta,2)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 40 *(i.e.* $M = 40$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32771	Error (Std)	0.34402 (0.024489)	0.34335 (0.023477)	0.34481 (0.027123)	0.34619 (0.027567)
		Min; Max	22; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.965; 0.7826	39.484; 3.6572	37.088; 9.0111	33.896; 12.4706
1.3	0.28633	Error (Std)	0.29798 (0.01879)	0.29788 (0.018208)	0.30063 (0.022539)	0.30206 (0.025902)
		Min; Max	22; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.966; 0.7612	38.711; 5.7165	35.446; 10.715	31.266; 13.9251
1.5	0.25982	Error (Std)	0.2691 (0.013152)	0.27011 (0.01624)	0.27246 (0.022644)	0.27541 (0.025552)
		Min; Max	25; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.985; 0.4743	37.895; 7.1985	33.882; 11.9993	28.955; 14.716
2	0.19885	Error (Std)	0.20551 (0.010035)	0.20696 (0.014277)	0.2084 (0.016667)	0.2111 (0.01978)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.703; 2.1807	35.39; 9.9803	28.892; 14.1344	22.829; 14.7719
2.3	0.16656	Error (Std)	0.17185 (0.008289)	0.17352 (0.012181)	0.17627 (0.015844)	0.17884 (0.019759)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.361; 3.0955	32.856; 11.3877	27.017; 14.0226	19.274; 13.4279
2.5	0.14697	Error (Std)	0.15206 (0.008458)	0.15377 (0.011507)	0.15615 (0.016615)	0.15899 (0.020624)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.115; 3.6638	32.078; 11.6256	23.892; 13.639	17.387; 12.2986
3	0.10485	Error (Std)	0.10908 (0.007011)	0.11152 (0.011836)	0.11432 (0.015096)	0.11756 (0.020324)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	37.575; 5.6198	28.033; 12.0646	19.585; 11.5574	13.126; 8.0256
4	0.04789	Error (Std)	0.05225 (0.006968)	0.05454 (0.013103)	0.05711 (0.016361)	0.05952 (0.01899)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	34.409; 7.1468	22.902; 9.6333	15.759; 6.2103	11.706; 3.9287

TABLE D.2.5 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the QDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated a* $N(\Delta,3)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 40 *(i.e.* $M = 40$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32919	Error (Std)	0.33538 (0.023581)	0.33543 (0.022999)	0.33566 (0.021343)	0.33842 (0.024084)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.981; 0.6008	39.276; 4.3645	37.091; 8.9518	33.019; 13.1015
1.3	0.29473	Error (Std)	0.30355 (0.01708)	0.30394 (0.01652)	0.30501 (0.019857)	0.30708 (0.021311)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.965; 0.7851	39.05; 4.9374	36.239; 9.9186	30.788; 14.2028
1.5	0.27214	Error (Std)	0.2819 (0.017886)	0.28342 (0.019388)	0.28439 (0.020174)	0.28555 (0.022249)
		Min; Max	22; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.965; 0.7826	38.09; 6.9014	34.274; 11.7731	28.494; 15.048
2	0.21843	Error (Std)	0.22646 (0.012718)	0.22694 (0.012924)	0.23002 (0.01779)	0.23351 (0.023268)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.878; 1.4564	36.442; 8.9565	30.377; 13.5813	24.149; 15.0411
2.3	0.18885	Error (Std)	0.1964 (0.011725)	0.19719 (0.013034)	0.19871 (0.016607)	0.20334 (0.024273)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.597; 2.5681	35.118; 10.0921	28.726; 14.0559	21.506; 14.4718
2.5	0.17042	Error (Std)	0.17652 (0.008564)	0.17823 (0.012688)	0.18117 (0.016814)	0.1834 (0.019881)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.385; 3.0887	33.949; 10.8101	26.258; 14.1655	20.236; 13.9321
3	0.12928	Error (Std)	0.13454 (0.007787)	0.13646 (0.011883)	0.1398 (0.01715)	0.1434 (0.024023)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	38.847; 4.0289	30.24; 12.1003	22.794; 13.3659	15.879; 11.2038
4	0.06833	Error (Std)	0.07294 (0.008035)	0.07663 (0.014831)	0.07925 (0.020909)	0.08023 (0.019564)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	35.521; 6.7656	23.765; 10.7928	16.345; 8.7334	12.256; 5.7692

TABLE D.2.6 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the QDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated a* $N(\Delta,4)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 40 *(i.e.* $M = 40$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32743	Error (Std)	0.32001 (0.019228)	0.32056 (0.020111)	0.32124 (0.020435)	0.32316 (0.023011)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.981; 0.6008	38.941; 5.2735	35.852; 10.3742	32.677; 13.2864
1.3	0.29754	Error (Std)	0.29729 (0.015866)	0.29809 (0.016015)	0.29983 (0.020057)	0.30113 (0.021798)
		Min; Max	23; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.968; 0.7166	38.571; 6.0244	35.268; 10.8468	30.506; 14.3292
1.5	0.27772	Error (Std)	0.28097 (0.015925)	0.28139 (0.014609)	0.2812 (0.015791)	0.28544 (0.020793)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.875; 1.4925	38.492; 6.2252	35.101; 11.0691	29.244; 14.703
2	0.22973	Error (Std)	0.23642 (0.011852)	0.23643 (0.01321)	0.23891 (0.01649)	0.2404 (0.019565)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.877; 1.4056	36.822; 8.5655	31.948; 12.986	25.23; 15.1764
2.3	0.20265	Error (Std)	0.20924 (0.011097)	0.21074 (0.013255)	0.21249 (0.01795)	0.21752 (0.025535)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.698; 2.2144	35.632; 9.7616	29.268; 13.8959	22.533; 14.7411
2.5	0.18551	Error (Std)	0.19239 (0.01123)	0.19443 (0.014678)	0.19643 (0.017698)	0.2009 (0.024818)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.553; 2.6691	34.95; 10.2862	27.53; 14.1318	20.33; 14.2799
3	0.14628	Error (Std)	0.15208 (0.008508)	0.15372 (0.011267)	0.15867 (0.02362)	0.16087 (0.022908)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	38.918; 4.0241	32.025; 11.82	24.056; 13.8729	17.429; 12.5729
4	0.08473	Error (Std)	0.08961 (0.008093)	0.09234 (0.012821)	0.09673 (0.021061)	0.09817 (0.022029)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	36.579; 6.3049	25.509; 11.6262	17.781; 10.1072	12.794; 7.1701

D.3 5-nn results

TABLE D.3.1 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the 5-nn classifier is given below. 5 initial observations were generated from a $N(0,1)$ distribution and 5 additional observations were generated a $N(\Delta,2)$ distribution, $\alpha = 0.05$ and the maximum number of sequential steps is 90 (i.e. $M = 90$). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max, $\bar{n}$ and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32771	Error (Std)	0.38008 (0.025224)	0.37782 (0.023774)	0.3775 (0.025864)	0.37529 (0.031878)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.931; 2.182	88.851; 9.3293	82.897; 22.6492	72.182; 33.1842
1.3	0.28633	Error (Std)	0.33257 (0.024872)	0.33011 (0.025554)	0.32924 (0.026277)	0.32685 (0.030739)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.862; 3.0842	86.021; 17.0056	78.332; 27.8111	63.626; 37.0076
1.5	0.25982	Error (Std)	0.29941 (0.023229)	0.2992 (0.025167)	0.29817 (0.029154)	0.29498 (0.030096)
		Min; Max	22; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.805; 3.562	84.344; 19.9144	73.039; 32.1075	59.407; 37.7755
2	0.19885	Error (Std)	0.22692 (0.020414)	0.22641 (0.020916)	0.22293 (0.022254)	0.22535 (0.027977)
		Min; Max	22; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.016; 7.9808	76.429; 28.8554	57.455; 37.6951	43.159; 36.8851
2.3	0.16656	Error (Std)	0.18932 (0.018129)	0.18923 (0.020346)	0.18821 (0.02428)	0.18671 (0.024048)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	87.115; 13.5024	70.683; 32.3577	49.054; 37.2711	29.153; 30.5044
2.5	0.14697	Error (Std)	0.16607 (0.017434)	0.16534 (0.019292)	0.16676 (0.022461)	0.16557 (0.024204)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	85.545; 16.42	64.066; 34.6543	41.983; 35.286	23.275; 24.9212
3	0.10485	Error (Std)	0.11877 (0.014383)	0.11894 (0.017221)	0.11921 (0.018859)	0.12051 (0.020939)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	78.822; 24.1662	47.099; 34.7159	26.649; 25.962	15.509; 14.692
4	0.04789	Error (Std)	0.05409 (0.008925)	0.0557 (0.010886)	0.05709 (0.012795)	0.058 (0.013084)
		Min; Max	21; 90	11; 90	8; 90	6; 50
		$\bar{n}$; Sd	54.246; 28.3958	26.587; 18.3184	15.796; 8.0502	11.831; 4.7725

TABLE D.3.2 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the* 5*-nn classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated a* $N(\Delta,3)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 90 *(i.e.* $M = 90$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32919	Error (Std)	0.47587 (0.01054)	0.47513 (0.01494)	0.47062 (0.02676)	0.46195 (0.03812)
		Min; Max	90; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	90; 0	88.091; 11.9448	83.647; 21.7113	73.297; 32.5036
1.3	0.29473	Error (Std)	0.47972 (0.00999)	0.47592 (0.01956)	0.46784 (0.03513)	0.45932 (0.04452)
		Min; Max	22; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.866; 2.9952	86.687; 15.5768	79.254; 26.9635	69.231; 34.3071
1.5	0.27214	Error (Std)	0.47992 (0.01139)	0.47696 (0.02061)	0.46777 (0.04024)	0.45094 (0.05541)
		Min; Max	22; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.865; 3.0173	85.419; 18.1154	76.477; 29.7417	60.786; 37.6795
2	0.21843	Error (Std)	0.48434 (0.01254)	0.47435 (0.03545)	0.45828 (0.05652)	0.4392 (0.0738)
		Min; Max	23; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.606; 5.0755	79.444; 26.1593	66.452; 35.2402	48.224; 38.0001
2.3	0.18885	Error (Std)	0.48528 (0.0141)	0.47143 (0.04425)	0.44422 (0.07311)	0.42356 (0.08473)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	88.364; 10.2347	75.016; 29.7408	55.085; 37.4359	35.673; 34.1691
2.5	0.17042	Error (Std)	0.48412 (0.01743)	0.46722 (0.05426)	0.44301 (0.08157)	0.41683 (0.08968)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	86.437; 14.8029	71.31; 32.039	51.163; 37.037	33.553; 32.7304
3	0.12928	Error (Std)	0.48497 (0.02165)	0.45434 (0.06958)	0.41211 (0.09612)	0.38423 (0.10764)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	83.998; 18.7398	58.425; 35.4486	35.113; 32.5414	19.104; 20.3698
4	0.06833	Error (Std)	0.46801 (0.04596)	0.4081 (0.09942)	0.36813 (0.11527)	0.34372 (0.12761)
		Min; Max	21; 90	11; 90	8; 90	6; 66
		$\bar{n}$; Sd	65.277; 29.7882	32.148; 25.5308	17.353; 12.5453	12.093; 6.095

TABLE D.3.3 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the* 5*-nn classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated a* $N(\Delta,4)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 90 *(i.e.* $M = 90$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32743	Error (Std)	0.46677 (0.01336)	0.46384 (0.01958)	0.45815 (0.03408)	0.4502 (0.04189)
		Min; Max	90; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	90; 0	88.295; 11.3756	83.282; 22.2223	75.013; 30.679
1.3	0.29754	Error (Std)	0.46895 (0.01362)	0.46594 (0.02476)	0.45617 (0.04483)	0.44428 (0.05355)
		Min; Max	22; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.932; 2.1503	87.013; 14.9119	79.906; 26.2451	70.299; 33.992
1.5	0.27772	Error (Std)	0.46993 (0.01429)	0.46644 (0.02439)	0.45356 (0.04642)	0.4389 (0.06025)
		Min; Max	23; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.809; 3.4907	86.082; 16.812	75.642; 30.2631	64.773; 36.3862
2	0.22973	Error (Std)	0.47528 (0.01395)	0.46588 (0.03776)	0.44306 (0.06716)	0.42257 (0.0754)
		Min; Max	22; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.47; 5.9064	82.296; 22.912	68.042; 34.8597	50.959; 38.0277
2.3	0.20265	Error (Std)	0.47643 (0.01755)	0.45974 (0.05047)	0.43481 (0.07291)	0.40605 (0.08976)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.151; 7.411	77.475; 28.1124	60.241; 37.0223	42.559; 36.6259
2.5	0.18551	Error (Std)	0.47694 (0.01953)	0.4587 (0.05668)	0.42913 (0.08143)	0.39585 (0.09029)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	88.105; 10.9917	74.328; 30.3255	56.29; 37.323	35.026; 34.1627
3	0.14628	Error (Std)	0.47805 (0.0223)	0.44268 (0.07541)	0.40243 (0.09674)	0.36727 (0.10308)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	85.761; 16.039	64.17; 34.8395	42.109; 35.2254	25.141; 27.0931
4	0.08473	Error (Std)	0.46472 (0.05164)	0.39132 (0.10606)	0.35029 (0.1123)	0.31949 (0.12178)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	73.297; 27.426	39.006; 31.2692	22.003; 20.4566	13.955; 10.9875

TABLE D.3.4 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the* 5*-nn classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated a* $N(\Delta,2)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 40 *(i.e.* $M = 40$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32771	Error (Std)	0.37937 (0.033388)	0.38009 (0.035261)	0.37747 (0.03465)	0.37769 (0.037315)
		Min; Max	22; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.982; 0.5692	39.305; 4.273	36.999; 9.0736	34.121; 12.2671
1.3	0.28633	Error (Std)	0.3331 (0.034691)	0.33426 (0.034905)	0.33273 (0.037773)	0.33041 (0.037796)
		Min; Max	23; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.936; 1.0114	38.892; 5.3223	35.167; 11.012	31.211; 13.9482
1.5	0.25982	Error (Std)	0.29947 (0.033142)	0.30006 (0.033847)	0.2973 (0.034383)	0.30011 (0.037935)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.894; 1.2745	38.413; 6.2464	34.239; 11.7336	29.71; 14.4815
2	0.19885	Error (Std)	0.22881 (0.028436)	0.22996 (0.032241)	0.22858 (0.032494)	0.22709 (0.032091)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.765; 1.927	35.43; 9.9185	29.545; 13.92	23.909; 15.0414
2.3	0.16656	Error (Std)	0.18955 (0.025204)	0.18946 (0.025547)	0.18865 (0.026054)	0.18918 (0.027498)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.335; 3.2045	33.928; 10.8236	25.813; 14.1121	19.703; 14.0017
2.5	0.14697	Error (Std)	0.16784 (0.022451)	0.16671 (0.022231)	0.1672 (0.025731)	0.16683 (0.025506)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.043; 3.7123	31.529; 11.9305	24.184; 13.8285	18.496; 13.0681
3	0.10485	Error (Std)	0.11877 (0.018617)	0.11882 (0.018709)	0.12072 (0.02103)	0.11992 (0.020296)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	37.882; 5.2509	27.719; 12.1494	19.444; 11.74	13.782; 9.1602
4	0.04789	Error (Std)	0.05562 (0.011459)	0.05636 (0.013687)	0.05636 (0.011894)	0.05801 (0.014381)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	34.214; 7.0592	22.49; 9.0293	15.627; 6.3773	11.861; 4.1772

TABLE D.3.5 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the 5-nn classifier is given below. 5 initial observations were generated from a $N(0,1)$ distribution and 5 additional observations were generated a $N(\Delta,3)$ distribution, $\alpha = 0.05$ and the maximum number of sequential steps is 40 (i.e. $M = 40$). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max, $\bar{n}$ and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32919	Error (Std)	0.37644 (0.032149)	0.37585 (0.032117)	0.37614 (0.033723)	0.37456 (0.036759)
		Min; Max	23; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.983; 0.5376	39.457; 3.8117	37.255; 8.727	34.329; 12.1091
1.3	0.29473	Error (Std)	0.34094 (0.032699)	0.34243 (0.032815)	0.34167 (0.035179)	0.33937 (0.036404)
		Min; Max	22; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.966; 0.7612	38.873; 5.4119	36.278; 9.8639	31.736; 13.8738
1.5	0.27214	Error (Std)	0.31754 (0.033909)	0.31711 (0.035132)	0.31837 (0.036265)	0.31349 (0.039845)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.962; 0.8493	38.692; 5.788	35.116; 11.0317	30.463; 14.339
2	0.21843	Error (Std)	0.25273 (0.030605)	0.25228 (0.031475)	0.25142 (0.032935)	0.25082 (0.034571)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.882; 1.4114	36.67; 8.8435	31.019; 13.4416	25.723; 15.1465
2.3	0.18885	Error (Std)	0.21691 (0.028628)	0.21703 (0.028561)	0.21627 (0.029775)	0.21587 (0.033217)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.671; 2.3057	35.198; 10.0886	29.093; 13.7923	22.178; 14.5761
2.5	0.17042	Error (Std)	0.19614 (0.026441)	0.19179 (0.025208)	0.19589 (0.03094)	0.19456 (0.029315)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.414; 2.9886	33.763; 10.9945	27.49; 14.0977	19.776; 13.8594
3	0.12928	Error (Std)	0.1467 (0.022029)	0.14643 (0.021026)	0.14742 (0.024216)	0.14763 (0.025942)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	38.745; 4.2023	30.257; 12.1217	22.267; 13.4266	15.459; 11.3411
4	0.06833	Error (Std)	0.07777 (0.013909)	0.07991 (0.016911)	0.08 (0.016163)	0.08121 (0.017996)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	35.826; 6.677	24.268; 10.9084	16.39; 8.6732	12.318; 6.1575

TABLE D.3.6 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the* 5*-nn classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated a* $N(\Delta,4)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 40 *(i.e.* $M = 40$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32743	Error (Std)	0.43061 (0.02548)	0.4287 (0.02762)	0.42395 (0.0321)	0.42021 (0.03881)
		Min; Max	22; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.982; 0.5692	39.178; 4.6832	37.306; 8.6756	34.631; 11.8441
1.3	0.29754	Error (Std)	0.43251 (0.02921)	0.43013 (0.0326)	0.42692 (0.03658)	0.41576 (0.04795)
		Min; Max	26; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.986; 0.4427	38.671; 5.8315	36.393; 9.7504	32.002; 13.582
1.5	0.27772	Error (Std)	0.43316 (0.03236)	0.43418 (0.03322)	0.42523 (0.04502)	0.41317 (0.05415)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.966; 0.7651	38.583; 5.9881	35.235; 11.0008	31.031; 14.0708
2	0.22973	Error (Std)	0.44114 (0.03536)	0.43581 (0.04063)	0.42276 (0.05574)	0.39974 (0.07036)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.899; 1.3132	37.365; 7.95	33.076; 12.3897	26.443; 15.159
2.3	0.20265	Error (Std)	0.44547 (0.03575)	0.43284 (0.05082)	0.41326 (0.069)	0.39206 (0.07984)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.718; 2.1576	35.764; 9.6803	29.79; 13.8388	24.042; 15.0507
2.5	0.18551	Error (Std)	0.4445 (0.03635)	0.43124 (0.0547)	0.41066 (0.07183)	0.38618 (0.08564)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.517; 2.817	35.169; 10.1494	28.785; 13.9364	22.239; 14.7034
3	0.14628	Error (Std)	0.44587 (0.04192)	0.42105 (0.06971)	0.38981 (0.08847)	0.35973 (0.10269)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	38.772; 4.2734	32.333; 11.6052	24.683; 13.9553	17.695; 12.8894
4	0.08473	Error (Std)	0.44185 (0.05049)	0.38467 (0.09713)	0.34131 (0.11291)	0.30578 (0.11667)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	36.536; 6.4154	26.051; 11.8011	17.571; 10.3112	12.605; 7.2595

D.4 3-nn results

TABLE D.4.1 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the 3-nn classifier is given below. 5 initial observations were generated from a $N(0,1)$ distribution and 5 additional observations were generated a $N(\Delta,2)$ distribution, $\alpha = 0.05$ and the maximum number of sequential steps is 90 (i.e. $M = 90$). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max, $\bar{n}$ and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32771	Error (Std)	0.39172 (0.021777)	0.39169 (0.024267)	0.38762 (0.025446)	0.38796 (0.028179)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.931; 2.182	87.6; 13.4286	82.728; 22.9411	76.368; 29.9515
1.3	0.28633	Error (Std)	0.34796 (0.024607)	0.34575 (0.025236)	0.34443 (0.027725)	0.34231 (0.029582)
		Min; Max	24; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.934; 2.0871	86.918; 15.1119	79.217; 27.135	69.703; 34.441
1.5	0.25982	Error (Std)	0.31676 (0.023472)	0.31505 (0.026178)	0.31335 (0.029644)	0.31021 (0.033922)
		Min; Max	22; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.864; 3.0395	84.498; 19.7321	74.399; 31.3084	60.673; 37.9134
2	0.19885	Error (Std)	0.24221 (0.022653)	0.23806 (0.02362)	0.23559 (0.028605)	0.23479 (0.033729)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.398; 6.3215	77.97; 27.7126	59.113; 37.4405	42.66; 37.1164
2.3	0.16656	Error (Std)	0.20237 (0.02125)	0.19944 (0.023581)	0.19665 (0.026826)	0.19744 (0.031488)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	87.966; 11.392	71.578; 32.0532	51.285; 37.6364	32.203; 33.0427
2.5	0.14697	Error (Std)	0.17892 (0.020614)	0.17421 (0.022943)	0.17474 (0.027917)	0.17512 (0.03249)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	86.164; 15.3737	64.973; 34.8347	45.633; 36.5035	25.717; 28.0994
3	0.10485	Error (Std)	0.12603 (0.017224)	0.12541 (0.020215)	0.1253 (0.025664)	0.12615 (0.029248)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	79.008; 24.0248	52.368; 35.4262	27.356; 27.3635	16.819; 16.5841
4	0.04789	Error (Std)	0.05692 (0.011568)	0.05799 (0.014494)	0.05796 (0.013016)	0.05927 (0.017202)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	56.826; 28.6241	26.455; 18.4834	16.274; 8.6034	11.959; 5.7582

TABLE D.4.2 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the* 3*-nn classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated a* $N(\Delta,3)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 90 *(i.e.* $M = 90$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32919	Error (Std)	0.48528 (0.007337)	0.48362 (0.010941)	0.48044 (0.019645)	0.47661 (0.026224)
		Min; Max	23; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.933; 2.1187	88.386; 11.0293	83.952; 21.1633	76.894; 29.5135
1.3	0.29473	Error (Std)	0.48639 (0.00822)	0.48538 (0.012964)	0.48107 (0.021532)	0.47229 (0.032846)
		Min; Max	25; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.935; 2.0555	87.079; 14.7172	80.53; 25.7445	68.481; 35.4979
1.5	0.27214	Error (Std)	0.48747 (0.008276)	0.48572 (0.013954)	0.48219 (0.02224)	0.46952 (0.038827)
		Min; Max	22; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.803; 3.5949	85.404; 18.216	79.694; 26.3809	62.52; 37.3572
2	0.21843	Error (Std)	0.49035 (0.008122)	0.48592 (0.021276)	0.47284 (0.039867)	0.46383 (0.046541)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.206; 7.2129	80.889; 24.5866	65.505; 35.9409	51.704; 38.417
2.3	0.18885	Error (Std)	0.49054 (0.008725)	0.4844 (0.025918)	0.47296 (0.041601)	0.45503 (0.057508)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	88.394; 10.086	75.594; 29.5917	59.964; 36.8411	39.528; 36.1869
2.5	0.17042	Error (Std)	0.49101 (0.010228)	0.48299 (0.029293)	0.47083 (0.045376)	0.45375 (0.060451)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	88.106; 10.9909	71.809; 31.9027	54.078; 37.3575	34.558; 33.551
3	0.12928	Error (Std)	0.49088 (0.014025)	0.47427 (0.042024)	0.46031 (0.059006)	0.43874 (0.070531)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	83.388; 19.4522	59.041; 35.9602	36.726; 33.3188	22.431; 24.0352
4	0.06833	Error (Std)	0.48428 (0.027055)	0.45844 (0.05882)	0.43228 (0.073295)	0.40747 (0.084775)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	66.744; 29.5869	33.402; 27.5377	20.097; 17.7052	12.572; 8.4013

TABLE D.4.3 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the 3-nn classifier is given below. 5 initial observations were generated from a $N(0,1)$ distribution and 5 additional observations were generated a $N(\Delta,4)$ distribution, $\alpha = 0.05$ and the maximum number of sequential steps is 90 (i.e. $M = 90$). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max, $\bar{n}$ and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32743	Error (Std)	0.37104 (0.023112)	0.37128 (0.022905)	0.36882 (0.025041)	0.36838 (0.0282)
		Min; Max	29; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.939; 1.929	88.627; 10.1606	82.244; 23.5661	72.579; 32.7042
1.3	0.29754	Error (Std)	0.34917 (0.022847)	0.34747 (0.023812)	0.34667 (0.027186)	0.34416 (0.028594)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.931; 2.182	87.995; 12.2813	79.485; 26.8133	71.16; 33.5598
1.5	0.27772	Error (Std)	0.3318 (0.023383)	0.33036 (0.025457)	0.32886 (0.027332)	0.32872 (0.032012)
		Min; Max	23; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.933; 2.1187	86.116; 16.7781	77.527; 28.866	66.362; 35.9851
2	0.22973	Error (Std)	0.28136 (0.023891)	0.28021 (0.026451)	0.27769 (0.029602)	0.27483 (0.033094)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.33; 6.6728	82.726; 22.3923	70.154; 33.7939	54.911; 38.3179
2.3	0.20265	Error (Std)	0.25019 (0.023997)	0.24798 (0.026197)	0.24489 (0.029991)	0.24242 (0.03448)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	88.998; 8.1249	79.548; 26.0293	62.698; 36.6844	45.874; 37.8914
2.5	0.18551	Error (Std)	0.22905 (0.023946)	0.22632 (0.026271)	0.22568 (0.029665)	0.22535 (0.036741)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	88.442; 9.9594	75.041; 29.9259	59.3; 37.0329	41.091; 36.7081
3	0.14628	Error (Std)	0.17764 (0.021018)	0.17732 (0.02539)	0.17298 (0.026092)	0.17516 (0.032837)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	85.705; 16.1086	66.665; 34.1264	42.895; 35.9381	27.317; 29.0824
4	0.08473	Error (Std)	0.1002 (0.015097)	0.10134 (0.018631)	0.10226 (0.024309)	0.10397 (0.02713)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	73.88; 26.8382	41.797; 32.7118	22.025; 21.3859	14.032; 11.2883

TABLE D.4.4 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the 3-nn classifier is given below. 5 initial observations were generated from a $N(0,1)$ distribution and 5 additional observations were generated a $N(\Delta,2)$ distribution, $\alpha = 0.05$ and the maximum number of sequential steps is 40 (i.e. $M = 40$). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max, $\bar{n}$ and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32771	Error (Std)	0.39357 (0.030982)	0.39229 (0.033702)	0.39001 (0.034)	0.38943 (0.035035)
		Min; Max	23; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.967; 0.7379	39.376; 4.0283	37.407; 8.5079	35.084; 11.5109
1.3	0.28633	Error (Std)	0.34834 (0.033819)	0.34452 (0.034354)	0.34499 (0.03632)	0.34323 (0.037422)
		Min; Max	22; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.965; 0.7826	39.039; 4.9906	36.17; 10.0255	32.261; 13.4693
1.5	0.25982	Error (Std)	0.31688 (0.034872)	0.31684 (0.034608)	0.31374 (0.036979)	0.31362 (0.038973)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.956; 0.8347	38.282; 6.5621	34.306; 11.6334	30.181; 14.4227
2	0.19885	Error (Std)	0.24196 (0.03162)	0.2431 (0.033311)	0.23916 (0.034535)	0.23905 (0.038948)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.659; 2.3884	36.286; 9.1102	30.285; 13.7033	23.736; 15.2432
2.3	0.16656	Error (Std)	0.2028 (0.02945)	0.20044 (0.02798)	0.19877 (0.031281)	0.19884 (0.035)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.353; 3.1674	34.106; 10.8102	26.472; 14.176	20.795; 14.271
2.5	0.14697	Error (Std)	0.17946 (0.026675)	0.17703 (0.029052)	0.17668 (0.030434)	0.17494 (0.033741)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.15; 3.602	32.204; 11.6193	25.105; 13.9992	18.158; 13.1286
3	0.10485	Error (Std)	0.12847 (0.023725)	0.12477 (0.022838)	0.12534 (0.024392)	0.12495 (0.02676)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	37.868; 5.2353	27.57; 12.1852	20.376; 12.4737	14.251; 9.9921
4	0.04789	Error (Std)	0.05772 (0.013146)	0.05817 (0.014663)	0.05918 (0.017306)	0.05959 (0.017662)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	34.511; 7.0599	22.615; 9.3544	15.898; 6.4988	12.002; 4.4748

TABLE D.4.5 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the 3-nn classifier is given below. 5 initial observations were generated from a $N(0,1)$ distribution and 5 additional observations were generated a $N(\Delta,3)$ distribution, $\alpha = 0.05$ and the maximum number of sequential steps is 40 (i.e. $M = 40$). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max, $\bar{n}$ and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32919	Error (Std)	0.3856 (0.03063)	0.38742 (0.031068)	0.38814 (0.032929)	0.38625 (0.033353)
		Min; Max	22; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.982; 0.5692	39.482; 3.7285	37.573; 8.2635	34.482; 11.9649
1.3	0.29473	Error (Std)	0.35426 (0.032155)	0.35381 (0.032967)	0.35383 (0.0335)	0.35228 (0.03661)
		Min; Max	25; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.985; 0.4743	38.851; 5.5088	36.37; 9.7653	32.744; 13.1969
1.5	0.27214	Error (Std)	0.33157 (0.034664)	0.33036 (0.03324)	0.32887 (0.034763)	0.32719 (0.038121)
		Min; Max	28; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.988; 0.3795	38.625; 5.9109	35.41; 10.7379	31.604; 13.8305
2	0.21843	Error (Std)	0.26974 (0.032535)	0.26896 (0.035273)	0.2649 (0.035132)	0.26448 (0.038265)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.863; 1.4575	37.12; 8.2175	31.875; 13.1544	26.512; 15.1299
2.3	0.18885	Error (Std)	0.23353 (0.033054)	0.23046 (0.033772)	0.22794 (0.033376)	0.22628 (0.035137)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.646; 2.4513	34.968; 10.2919	29.273; 13.9224	23.802; 14.9461
2.5	0.17042	Error (Std)	0.2078 (0.029908)	0.2064 (0.031027)	0.20676 (0.032965)	0.20371 (0.034122)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.326; 3.1862	34.319; 10.6794	27.703; 14.2385	21.691; 14.6322
3	0.12928	Error (Std)	0.15767 (0.026345)	0.1553 (0.025536)	0.15605 (0.030945)	0.15538 (0.030674)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	38.678; 4.3191	30.529; 12.2089	22.919; 13.4354	16.831; 12.1695
4	0.06833	Error (Std)	0.08346 (0.018727)	0.08136 (0.016796)	0.08394 (0.022692)	0.08251 (0.020032)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	35.748; 6.752	24.919; 11.0562	17.074; 9.2116	12.474; 6.1762

TABLE D.4.6 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the* 3*-nn classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated a* $N(\Delta,4)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 40 *(i.e.* $M = 40$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32743	Error (Std)	0.4552 (0.02009)	0.45499 (0.021269)	0.45036 (0.025856)	0.44514 (0.032981)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.981; 0.6008	39.504; 3.6171	37.306; 8.6513	33.968; 12.317
1.3	0.29754	Error (Std)	0.45876 (0.0206)	0.45734 (0.023308)	0.45348 (0.027979)	0.44465 (0.037658)
		Min; Max	24; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.984; 0.506	39.081; 4.8565	37.012; 8.998	32.378; 13.552
1.5	0.27772	Error (Std)	0.46003 (0.021726)	0.45849 (0.024087)	0.4532 (0.031151)	0.44398 (0.039247)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.948; 0.953	38.919; 5.3121	36.149; 10.0247	31.415; 14.0089
2	0.22973	Error (Std)	0.46513 (0.023061)	0.46055 (0.029709)	0.45166 (0.043546)	0.43854 (0.053676)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.885; 1.3826	37.35; 7.9882	32.264; 12.9896	27.46; 15.0951
2.3	0.20265	Error (Std)	0.46842 (0.023388)	0.46261 (0.031914)	0.45155 (0.045113)	0.4353 (0.058335)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.649; 2.4265	36.214; 9.2082	30.723; 13.5376	25.297; 15.1314
2.5	0.18551	Error (Std)	0.46842 (0.024383)	0.46326 (0.034579)	0.44478 (0.053689)	0.43143 (0.061907)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.573; 2.643	35.982; 9.3953	28.896; 14.1434	23.317; 14.8676
3	0.14628	Error (Std)	0.4715 (0.023959)	0.45867 (0.044452)	0.43695 (0.063977)	0.4187 (0.071335)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.135; 3.5586	32.694; 11.3888	25.324; 13.8616	18.742; 13.3721
4	0.08473	Error (Std)	0.47105 (0.030817)	0.44462 (0.061241)	0.41695 (0.079434)	0.38996 (0.086296)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	36.855; 6.2197	26.324; 11.5999	19.081; 11.0799	13.219; 8.1392

D.5 LRIM results

TABLE D.5.1 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LRIM classifier is given below. 5 initial observations were generated from a $N(0,1)$ distribution and 5 additional observations were generated a $N(\Delta,2)$ distribution, $\alpha = 0.05$ and the maximum number of sequential steps is 90 (i.e. $M = 90$). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max, $\bar{n}$ and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32771	Error (Std)	0.34321 (0.022092)	0.34223 (0.020212)	0.34343 (0.022548)	0.34312 (0.022382)
		Min; Max	23; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.933; 2.1187	87.448; 13.8267	81.771; 24.1033	71.967; 32.626
1.3	0.28633	Error (Std)	0.29573 (0.013159)	0.29548 (0.013211)	0.29678 (0.013767)	0.29688 (0.017492)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.662; 4.7714	84.825; 19.1871	74.848; 30.8957	60.863; 37.6043
1.5	0.25982	Error (Std)	0.26723 (0.011193)	0.2675 (0.011418)	0.26852 (0.014951)	0.27016 (0.01764)
		Min; Max	22; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.676; 4.5863	82.879; 22.1823	71.839; 32.8119	54.38; 37.9256
2	0.19885	Error (Std)	0.20222 (0.006478)	0.2037 (0.008973)	0.2051 (0.01155)	0.20814 (0.015034)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	88.797; 8.707	75.59; 29.418	54.502; 37.4299	33.652; 32.6839
2.3	0.16656	Error (Std)	0.16987 (0.006159)	0.17127 (0.010333)	0.17347 (0.012298)	0.17531 (0.014156)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	86.37; 14.9604	66.835; 33.9901	42.949; 35.6598	25.233; 27.0529
2.5	0.14697	Error (Std)	0.1498 (0.005169)	0.15172 (0.009265)	0.15429 (0.01389)	0.15598 (0.013704)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	85.36; 16.7214	58.711; 35.7186	35.966; 33.0031	19.354; 20.1823
3	0.10485	Error (Std)	0.10817 (0.006161)	0.11106 (0.010497)	0.11279 (0.012403)	0.11357 (0.0129)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	76.273; 26.1044	43.937; 33.4581	23.224; 21.5337	13.882; 10.9684
4	0.04789	Error (Std)	0.05298 (0.006028)	0.05453 (0.008534)	0.05532 (0.009749)	0.05627 (0.010761)
		Min; Max	21; 90	11; 90	8; 82	6; 55
		$\bar{n}$; Sd	54.574; 28.6094	24.486; 15.3112	15.28; 5.9858	11.683; 3.8193

TABLE D.5.2 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LRIM classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated a* $N(\Delta,3)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 90 *(i.e.* $M = 90$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32919	Error (Std)	0.35661 (0.030651)	0.35656 (0.031817)	0.35683 (0.032831)	0.35921 (0.034343)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.931; 2.182	87.905; 12.5848	83.609; 21.6526	73.933; 31.811
1.3	0.29473	Error (Std)	0.31405 (0.023897)	0.31532 (0.024884)	0.31332 (0.024811)	0.31461 (0.028482)
		Min; Max	23; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.87; 2.9068	87.556; 13.4558	77.992; 28.0492	65.171; 35.9034
1.5	0.27214	Error (Std)	0.28594 (0.019476)	0.28673 (0.020435)	0.28742 (0.023892)	0.28861 (0.025745)
		Min; Max	22; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.932; 2.1503	84.294; 20.137	74.373; 31.1169	57.543; 38.3332
2	0.21843	Error (Std)	0.22492 (0.011295)	0.22623 (0.013054)	0.22799 (0.014911)	0.22929 (0.018452)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.35; 6.4818	77.99; 27.5135	60.576; 36.7273	41.619; 36.5687
2.3	0.18885	Error (Std)	0.19343 (0.00794)	0.19505 (0.011899)	0.19659 (0.014635)	0.199 (0.016005)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	88.211; 10.7548	71.905; 32.006	52.084; 37.2906	33.635; 32.7116
2.5	0.17042	Error (Std)	0.17438 (0.00704)	0.17647 (0.011585)	0.1782 (0.013331)	0.18146 (0.020007)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	86.931; 13.7569	68.128; 33.6102	44.84; 36.412	26.713; 28.6331
3	0.12928	Error (Std)	0.13312 (0.006529)	0.13523 (0.010408)	0.13774 (0.012981)	0.13961 (0.014768)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	82.257; 20.9007	55.383; 35.5615	32.69; 30.1293	17.48; 18.3451
4	0.06833	Error (Std)	0.07607 (0.008425)	0.07861 (0.01159)	0.08006 (0.014317)	0.08009 (0.013937)
		Min; Max	21; 90	11; 90	8; 90	6; 83
		$\bar{n}$; Sd	66.061; 29.8858	31.88; 25.8305	16.782; 10.7704	12.004; 5.7697

TABLE D.5.3 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LRIM classifier is given below. 5 initial observations were generated from a $N(0,1)$ distribution and 5 additional observations were generated a $N(\Delta,4)$ distribution, $\alpha = 0.05$ and the maximum number of sequential steps is 90 (i.e. $M = 90$). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max, $\bar{n}$ and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32743	Error (Std)	0.37024 (0.044895)	0.36938 (0.043749)	0.36871 (0.045063)	0.37281 (0.049199)
		Min; Max	90; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	90; 0	88.382; 11.0571	82.384; 23.3289	74.246; 31.5387
1.3	0.29754	Error (Std)	0.32656 (0.034877)	0.32566 (0.03293)	0.32463 (0.033979)	0.3248 (0.03642)
		Min; Max	24; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.803; 3.5932	87.5; 13.6963	78.952; 27.3065	65.851; 36.073
1.5	0.27772	Error (Std)	0.29889 (0.025596)	0.29903 (0.028859)	0.29869 (0.028373)	0.30133 (0.032515)
		Min; Max	22; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.867; 2.9732	85.849; 17.3942	74.992; 30.7914	64.231; 36.3779
2	0.22973	Error (Std)	0.23999 (0.015894)	0.24013 (0.016679)	0.24261 (0.019197)	0.24385 (0.027122)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.388; 6.4263	80.116; 25.4999	66.034; 35.6971	46.182; 38.0856
2.3	0.20265	Error (Std)	0.2096 (0.012568)	0.21081 (0.014377)	0.21258 (0.015641)	0.21536 (0.02424)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.145; 7.4573	75.566; 29.3639	55.684; 37.5297	39.249; 35.1915
2.5	0.18551	Error (Std)	0.19023 (0.008723)	0.19226 (0.011123)	0.19512 (0.015605)	0.19693 (0.018229)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	88.2; 10.6457	71.959; 31.9092	49.833; 37.5697	31.577; 32.4725
3	0.14628	Error (Std)	0.15015 (0.006662)	0.15326 (0.011835)	0.15528 (0.01336)	0.15821 (0.018936)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	85.434; 16.6738	62.348; 35.2392	36.581; 32.5813	20.405; 22.0441
4	0.08473	Error (Std)	0.09352 (0.009206)	0.09682 (0.013424)	0.09807 (0.01515)	0.09903 (0.016667)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	73.71; 27.2572	39.688; 31.0294	19.908; 17.3107	12.483; 8.0551

TABLE D.5.4 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LRIM classifier is given below. 5 initial observations were generated from a $N(0,1)$ distribution and 5 additional observations were generated a $N(\Delta,2)$ distribution, $\alpha = 0.05$ and the maximum number of sequential steps is 40 (i.e. $M = 40$). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max, $\bar{n}$ and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32771	Error (Std)	0.34926 (0.031343)	0.34995 (0.03346)	0.34864 (0.030902)	0.35134 (0.037352)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.981; 0.6008	39.315; 4.294	36.799; 9.3066	33.275; 12.9652
1.3	0.28633	Error (Std)	0.30048 (0.021323)	0.30008 (0.023404)	0.30229 (0.025664)	0.30179 (0.024801)
		Min; Max	23; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.968; 0.7166	38.282; 6.5579	35.204; 10.9594	31.122; 14.0018
1.5	0.25982	Error (Std)	0.26974 (0.01678)	0.27184 (0.019493)	0.27175 (0.019917)	0.27299 (0.021182)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.946; 0.988	38.168; 6.7469	33.612; 12.0751	28.231; 14.9074
2	0.19885	Error (Std)	0.2054 (0.011534)	0.2062 (0.013043)	0.20711 (0.013785)	0.20922 (0.0177)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.699; 2.1788	34.172; 10.8557	28.005; 14.1075	21.27; 14.6065
2.3	0.16656	Error (Std)	0.17233 (0.009645)	0.1729 (0.010467)	0.17462 (0.012467)	0.17587 (0.01415)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.188; 3.4838	32.768; 11.4942	25.281; 14.1195	18.613; 13.2493
2.5	0.14697	Error (Std)	0.15189 (0.007591)	0.15368 (0.009851)	0.15464 (0.010937)	0.15593 (0.015927)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	38.872; 4.1159	30.632; 12.2539	22.367; 13.4611	16.473; 11.5894
3	0.10485	Error (Std)	0.10996 (0.00776)	0.11176 (0.010843)	0.11289 (0.012427)	0.11422 (0.014432)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	37.51; 5.6498	26.758; 12.2308	18.532; 11.1349	13.495; 8.4692
4	0.04789	Error (Std)	0.05363 (0.007139)	0.05461 (0.008654)	0.05591 (0.010262)	0.05632 (0.011065)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	34; 7.2119	22.768; 9.3724	15.386; 5.9315	11.882; 3.9045

TABLE D.5.5 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LRIM classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated a* $N(\Delta,3)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 40 *(i.e.* $M = 40$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32919	Error (Std)	0.36683 (0.047542)	0.36676 (0.04473)	0.36648 (0.044871)	0.37004 (0.051158)
		Min; Max	22; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.982; 0.5692	39.063; 5.003	37.187; 8.7945	34.336; 12.0658
1.3	0.29473	Error (Std)	0.31878 (0.033682)	0.31923 (0.034966)	0.32145 (0.036532)	0.32056 (0.036299)
		Min; Max	23; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.966; 0.7599	39.011; 5.0749	35.92; 10.3288	31.495; 14.0346
1.5	0.27214	Error (Std)	0.29198 (0.029549)	0.29198 (0.029839)	0.29228 (0.032058)	0.29378 (0.034066)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.963; 0.8272	37.966; 7.1369	33.98; 11.9482	28.825; 14.8945
2	0.21843	Error (Std)	0.22826 (0.016839)	0.22962 (0.01913)	0.23188 (0.02394)	0.23149 (0.021645)
		Min; Max	23; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.911; 1.1526	36.262; 9.2036	30.955; 13.4049	24.329; 15.1144
2.3	0.18885	Error (Std)	0.19624 (0.013342)	0.19763 (0.01436)	0.19961 (0.018911)	0.20138 (0.021371)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.494; 2.7389	34.401; 10.6692	27.404; 14.1794	21.352; 14.4776
2.5	0.17042	Error (Std)	0.17732 (0.011333)	0.17932 (0.015112)	0.17985 (0.014651)	0.17981 (0.014612)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.29; 3.3537	32.707; 11.6783	25.826; 14.2547	19.037; 13.5243
3	0.12928	Error (Std)	0.13528 (0.008498)	0.13693 (0.011078)	0.13807 (0.013833)	0.14033 (0.015943)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	38.368; 4.8231	29.463; 12.3366	21.095; 12.9532	14.702; 10.3468
4	0.06833	Error (Std)	0.07664 (0.009478)	0.07855 (0.012151)	0.07976 (0.013665)	0.07989 (0.013984)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	35.534; 6.9528	24.048; 10.7124	16.153; 7.853	11.746; 5.0888

TABLE D.5.6 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LRIM classifier is given below. 5 initial observations were generated from a $N(0,1)$ distribution and 5 additional observations were generated a $N(\Delta,4)$ distribution, $\alpha = 0.05$ and the maximum number of sequential steps is 40 (i.e. $M = 40$). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max, $\bar{n}$ and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.32743	Error (Std)	0.38395 (0.060374)	0.38187 (0.059518)	0.37992 (0.058269)	0.38172 (0.060556)
		Min; Max	27; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.987; 0.4111	39.274; 4.385	37.513; 8.2841	34.975; 11.5586
1.3	0.29754	Error (Std)	0.33832 (0.052738)	0.33416 (0.048372)	0.33413 (0.04926)	0.33375 (0.050707)
		Min; Max	22; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.947; 0.967	39.027; 5.0529	35.563; 10.6616	31.621; 13.7554
1.5	0.27772	Error (Std)	0.30961 (0.042489)	0.30894 (0.044759)	0.30816 (0.044169)	0.30793 (0.044131)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.936; 1.0349	38.247; 6.6437	35.085; 11.0313	30.391; 14.3168
2	0.22973	Error (Std)	0.24422 (0.025062)	0.24554 (0.026011)	0.24643 (0.027813)	0.24794 (0.0289)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.777; 1.895	37.101; 8.2092	31.542; 13.2497	25.166; 15.1485
2.3	0.20265	Error (Std)	0.21418 (0.021255)	0.21625 (0.024131)	0.21723 (0.025172)	0.21893 (0.031311)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.687; 2.3204	34.825; 10.4703	29.418; 13.8575	23.051; 14.8569
2.5	0.18551	Error (Std)	0.1955 (0.017943)	0.19607 (0.018171)	0.19785 (0.020131)	0.19799 (0.019091)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	39.478; 2.8649	34.792; 10.3935	27.458; 14.1972	20.901; 14.3016
3	0.14628	Error (Std)	0.15359 (0.013717)	0.15492 (0.01322)	0.15661 (0.016054)	0.15783 (0.016823)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	38.738; 4.2689	30.814; 12.2029	22.413; 13.5995	16.735; 11.9037
4	0.08473	Error (Std)	0.09469 (0.010962)	0.09657 (0.013524)	0.09893 (0.017288)	0.09866 (0.015547)
		Min; Max	21; 40	11; 40	8; 40	6; 40
		$\bar{n}$; Sd	36.484; 6.4957	25.826; 11.6396	17.75; 10.1283	12.431; 7.0953

D.6 Changes to sampling probabilities

TABLE D.6.1 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LDA classifier is given below.* 9 *initial observations were generated from a* $N(0,1)$ *distribution and* 1 *observation was generated from a* $N(\Delta,1)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 90 *(i.e.* $M = 90$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.098664	Error (Std)	0.10225 (0.026447)	0.10498 (0.044611)	0.1116 (0.0762)	0.11494 (0.084567)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	77.817; 24.843	50.492; 33.7206	30.509; 27.1625	20.11; 18.897
1.3	0.093761	Error (Std)	0.0967 (0.005075)	0.10131 (0.045158)	0.10192 (0.038002)	0.10877 (0.072855)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	78.008; 24.7133	46.024; 33.2122	25.781; 22.798	16.554; 14.7071
1.5	0.088312	Error (Std)	0.09194 (0.004915)	0.09506 (0.027302)	0.09823 (0.04567)	0.10089 (0.052934)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	73.131; 27.6366	41.933; 31.8929	24.459; 22.2731	14.842; 12.4204
2	0.070061	Error (Std)	0.07461 (0.007512)	0.07742 (0.010173)	0.08094 (0.038682)	0.08338 (0.040373)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	66.587; 29.5584	33.837; 26.648	18.163; 14.0808	13.117; 8.5015
2.3	0.058151	Error (Std)	0.06263 (0.007725)	0.06622 (0.011103)	0.06844 (0.029516)	0.06997 (0.030014)
		Min; Max	21; 90	11; 90	8; 90	6; 66
		$\bar{n}$; Sd	58.837; 29.5645	27.295; 20.5604	16.491; 9.7223	12.308; 5.8753
2.5	0.050496	Error (Std)	0.05473 (0.007316)	0.05878 (0.028859)	0.05979 (0.012412)	0.06143 (0.029758)
		Min; Max	21; 90	11; 90	8; 77	6; 59
		$\bar{n}$; Sd	56.338; 29.2241	26.389; 17.4138	16.319; 8.1052	11.828; 4.7258
3	0.033651	Error (Std)	0.03774 (0.007368)	0.03963 (0.0091)	0.04057 (0.010352)	0.04133 (0.011524)
		Min; Max	21; 90	11; 90	8; 52	6; 43
		$\bar{n}$; Sd	48.116; 23.9813	23.523; 11.2685	16.096; 5.2633	12.258; 3.4154
4	0.0122	Error (Std)	0.01401 (0.003331)	0.01557 (0.005611)	0.01611 (0.005976)	0.01687 (0.007458)
		Min; Max	21; 90	11; 53	8; 29	6; 22
		$\bar{n}$; Sd	45.553; 16.197	24.749; 6.5831	17.061; 3.6073	12.993; 2.3118

TABLE D.6.2 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LDA classifier is given below.* 8 *initial observations were generated from a* $N(0,1)$ *distribution and* 2 *observations were generated from a* $N(\Delta,1)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 90 *(i.e.* $M = 90$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.18616	Error (Std)	0.19046 (0.005939)	0.19257 (0.028325)	0.19392 (0.022398)	0.19765 (0.031246)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	87.171; 13.3666	75.465; 29.5437	54.239; 37.0799	36.393; 33.727
1.3	0.16672	Error (Std)	0.17098 (0.006678)	0.17344 (0.011299)	0.17502 (0.012073)	0.18088 (0.042394)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	87.121; 13.3035	70.53; 32.3272	47.176; 36.4995	27.27; 28.5264
1.5	0.15147	Error (Std)	0.15578 (0.007262)	0.15833 (0.011653)	0.16137 (0.02448)	0.16308 (0.015486)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	85.861; 15.7832	62.164; 35.1934	39.594; 34.6786	22.963; 23.8766
2	0.11207	Error (Std)	0.11584 (0.007707)	0.11839 (0.01179)	0.12085 (0.013284)	0.1229 (0.016186)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	80.838; 22.2787	48.292; 34.6782	26.975; 26.3049	15.261; 13.1234
2.3	0.09028	Error (Std)	0.09353 (0.006207)	0.0966 (0.01102)	0.09941 (0.013595)	0.1014 (0.016833)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	72.473; 27.9107	41.437; 31.8886	20.288; 18.0169	12.927; 8.7389
2.5	0.07714	Error (Std)	0.08055 (0.006219)	0.08315 (0.009691)	0.08484 (0.011095)	0.08725 (0.01588)
		Min; Max	21; 90	11; 90	8; 90	6; 61
		$\bar{n}$; Sd	67.642; 29.4206	35.615; 27.6538	17.884; 12.8587	12.42; 6.6786
3	0.04983	Error (Std)	0.05273 (0.005114)	0.05481 (0.008117)	0.05636 (0.009283)	0.05747 (0.01037)
		Min; Max	21; 90	11; 90	8; 74	6; 56
		$\bar{n}$; Sd	53.821; 27.9111	26.066; 17.2744	16.259; 8.002	12.067; 4.3932
4	0.0174	Error (Std)	0.01921 (0.003242)	0.02002 (0.004331)	0.02074 (0.005767)	0.02099 (0.005194)
		Min; Max	21; 90	11; 76	8; 29	6; 21
		$\bar{n}$; Sd	45.062; 17.9185	24.328; 7.5398	16.861; 3.7659	12.821; 2.4118

TABLE D.6.3 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LDA classifier is given below.* 6 *initial observations were generated from a* $N(0,1)$ *distribution and* 4 *observations were generated from a* $N(\Delta,1)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 90 *(i.e.* $M = 90$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.2945	Error (Std)	0.29926 (0.008016)	0.29977 (0.01028)	0.30086 (0.012626)	0.30301 (0.015594)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.865; 3.0179	86.391; 16.2662	77.098; 28.9675	61.648; 37.5361
1.3	0.24789	Error (Std)	0.2515 (0.006611)	0.25206 (0.007578)	0.2536 (0.011309)	0.25615 (0.016119)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.535; 5.545	81.815; 23.6122	69.696; 33.9644	49.839; 37.889
1.5	0.21856	Error (Std)	0.22171 (0.006204)	0.2228 (0.009954)	0.22447 (0.010955)	0.22697 (0.01647)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	89.129; 7.5979	77.902; 27.6509	60.662; 37.2291	40.918; 36.1438
2	0.15378	Error (Std)	0.15639 (0.005393)	0.15825 (0.007623)	0.16005 (0.011232)	0.16171 (0.012494)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	85.66; 16.2358	61.804; 35.3964	39.059; 33.7807	21.264; 22.7407
2.3	0.12146	Error (Std)	0.12398 (0.005067)	0.12579 (0.008092)	0.12841 (0.010793)	0.12993 (0.012452)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	81.153; 21.9087	52.55; 35.5923	27.428; 26.6925	16.293; 15.6695
2.5	0.1027	Error (Std)	0.1048 (0.004987)	0.10741 (0.00841)	0.10908 (0.010715)	0.11024 (0.01161)
		Min; Max	21; 90	11; 90	8; 90	6; 90
		$\bar{n}$; Sd	76.55; 25.5821	44.975; 33.4877	21.759; 20.0345	13.658; 10.3907
3	0.06507	Error (Std)	0.06744 (0.004433)	0.06926 (0.007111)	0.07065 (0.009354)	0.07173 (0.009869)
		Min; Max	21; 90	11; 90	8; 90	6; 72
		$\bar{n}$; Sd	60.75; 29.9426	29.427; 22.5406	16.495; 9.8049	11.795; 5.0299
4	0.02221	Error (Std)	0.02375 (0.002975)	0.02448 (0.003917)	0.02497 (0.004302)	0.02511 (0.004673)
		Min; Max	21; 90	11; 65	8; 35	6; 33
		$\bar{n}$; Sd	44.116; 18.3193	23.78; 7.5385	16.347; 4.0987	12.624; 2.6829

D.7 Changing maximum number of sequential steps

TABLE D.7.1 - *The average and standard deviation of the misclassification rate (denoted Error and Std respectively) in training the LDA classifier is given below.* 5 *initial observations were generated from a* $N(0,1)$ *distribution and 5 additional observations were generated from a* $N(\Delta,1)$ *distribution,* $\alpha = 0.05$ *and the maximum number of sequential steps is* 300 *(i.e.* $M = 300$*). The minimum, maximum, mean and standard deviation of the number of sequential steps required to train the classifier is also provided (denoted Min, Max,* $\bar{n}$ *and Sd respectively).*

Δ	Bayes error		$\epsilon = 0.05$	$\epsilon = 0.10$	$\epsilon = 0.15$	$\epsilon = 0.20$
1	0.30854	Error (Std)	0.3116 (0.006455508)	0.31243 (0.010963)	0.31551 (0.019036)	0.31926 (0.025056)
		Min; Max	23; 300	11; 300	8; 300	6; 300
		$\bar{n}$; Sd	299.723; 8.75951	290.47; 51.196	247.571; 107.621	183.643; 128.818
1.3	0.25785	Error (Std)	0.261 (0.006524)	0.26293 (0.013892)	0.26774 (0.02549)	0.27162 (0.027994)
		Min; Max	21; 300	11; 300	8; 300	6; 300
		$\bar{n}$; Sd	298.342; 21.352	276.46; 77.858	214.121; 122.996	128.188; 119.637
1.5	0.22663	Error (Std)	0.23009 (0.00736)	0.2329 (0.01807)	0.24028 (0.029012)	0.24709 (0.036581)
		Min; Max	21; 300	11; 300	8; 300	6; 300
		$\bar{n}$; Sd	297.238; 27.496	263.568; 93.408	180.131; 128.659	90.52; 103.66
2	0.15866	Error (Std)	0.16332 (0.011152)	0.17098 (0.029493)	0.1807 (0.040026)	0.18882 (0.045265)
		Min; Max	21; 300	11; 300	8; 300	6; 300
		$\bar{n}$; Sd	288.726; 54.556	209.605; 122.798	84.253; 96.374	31.797; 43.806
2.3	0.12507	Error (Std)	0.13109 (0.01504)	0.144 (0.039547)	0.15594 (0.050247)	0.16581 (0.061292)
		Min; Max	21; 300	11; 300	8; 300	6; 210
		$\bar{n}$; Sd	278.697; 72.449	151.871; 125.587	47.81; 65.011	20.547; 25.992
2.5	0.10565	Error (Std)	0.11282 (0.016814)	0.12694 (0.043258)	0.13572 (0.048943)	0.14534 (0.059173)
		Min; Max	21; 300	11; 300	8; 300	6; 234
		$\bar{n}$; Sd	263.073; 92.03	111.339; 112.812	35.646; 45.936	16.752; 17.778
3	0.06681	Error (Std)	0.0775 (0.021215)	0.09704 (0.049834)	0.10324 (0.06012)	0.11316 (0.066498)
		Min; Max	21; 300	11; 300	8; 228	6; 87
		$\bar{n}$; Sd	210.269; 120.207	50.234; 62.692	20.858; 22.037	11.982; 7.006
4	0.02275	Error (Std)	0.04087 (0.03075)	0.05803 (0.060197)	0.06842 (0.074307)	0.06586 (0.071539)
		Min; Max	21; 300	11; 300	8; 98	6; 62
		$\bar{n}$; Sd	72.33; 69.521	24.957; 18.653	15.298; 7.001	11.773; 3.702

D.8 Critical values for prescribed coverage probability

TABLE D.8.1 - *The parameter values for a and γ as determined by Frey (2010), used to determine a fixed-width interval for a proportion p. The half-width of the interval is denoted by h.*

	$\alpha = 0.1$		$\alpha = 0.05$		$\alpha = 0.01$	
Half-width h	α	γ	α	γ	α	γ
0.1	4	0.0754	4	0.0356	6	0.0068
0.05	4	0.0859	6	0.0433	8	0.0083
0.01	8	0.0972	10	0.0487	14	0.0097

Appendix E

SAS Code

```
options compress = Y minoperator cpucount = actual threads;

    %macro determine_sample_size(StopRule,sample_sz1,sample_sz2,mu1,mu2,sig1,sig2,
                                 eps,alpha,h,repititions,max_sample,lambda_method,
                                 Nearest_Neighbours,two_normal_out_m);

    %if &StopRule. = 'NEW' %then %do;
        %let eps_loop = 0.2;
        %let epsloop = 20;
    %end;

    proc iml;

    start LDA_two_normal_distributions(StopRule,sample_sz1,sample_sz2,mu1,mu2,sig1,
                                       sig2,eps,alpha,h,repititions,max_sample,lambda_method,
                                       KNN_Neighbours,two_normal_out_m);

    /*StopRule - NEW indicates proposed method, OLD indicates current methodology*/
    /*sample_sz1 = the number of observations to generate from sample1;*/
    /*sample_sz2 = the number of observations to generate from sample2;*/
    /*mu1 = mean value for generating sample1 observations;*/
    /*mu2 = mean value for generating sample2 observations;*/
    /*sig1 = variance for generating sample1 observations;*/
    /*sig2 = variance for generating sample2 observations;*/
    /*eps = epsilon;*/
    /*a = alpha significance level;*/
    /*h = halfwidth of confidence interval for proposed approach*/
    /*repititions = number of repititions that the recursive method should be iterated;*/
    /*max_sample = maximum number of trails allowed;*/
    /*lambda_method = LDA, QDA, KNN, REG variable to select whether lambda is
    calculated using K-Nearest, LDA, QDA or Regression of Indicator Matrix*/
    /*KNN_Neighbours = K-Nearest Neighbours if LAMBDA_METHOD = KNN*/
```

```
/*two_normal_out_m = Output matrix for two normal distribution inputs*/

************************ Parameters ************************;
sampsize1 = sample_sz1;
sampsize2 = sample_sz2;
epsilon = eps;
mean1 = mu1;
mean2 = mu2;      ***** changes with delta;
var1 = sig1;
var2 = sig2;
sample_ratio = sample_sz1/(sample_sz1 + sample_sz2);
if KNN_Neighbours > 0 then KNN_Number = KNN_Neighbours;
if strip(upcase(StopRule)) = 'NEW' then max_sample = 50000;

************************ Recursive ************************;
N_Zero = log(alpha)/(log(1-epsilon));
do rep1 = 1 to repititions;

    ************************ Generating data ************************;
    sample = (randnormal(sampsize1,mean1,var1)||J(sampsize1,1,0))//
    (randnormal(sampsize2,mean2,var2)||J(sampsize2,1,1));

    *********** Determining the initial Classifier - LDA ***********;
    if upcase(lambda_method) = 'LDA' then do;
        lambda_use_in = sample;
        call lambda_calc_LDA(lambda_use_in,lambda_use);
    end;
    if upcase(lambda_method) = 'QDA' then do;
        lambda_use_in = sample;
        call lambda_calc_QDA(lambda_use_in,lambda_use,qda_mean_0,
                                qda_mean_1,qda_var_0,qda_var_1);
    end;
    if upcase(lambda_method) = 'REG' then do;
        lambda_use_in = sample;
        call lambda_calc_REG(lambda_use_in,lambda_use);
    end;

    *********** Set up initial variables ***********;
    Perfect_Class_Counter = 0;
    Missclass_Matrix = J(1,1,0);
```

```
********** Recursive run to reach stopping critria **********;
do i3 = 1 to max_sample;

    ********** Randomly Sampling Record **********;
    u = ranuni(0);
    val_chosen = randnormal(1,mean1,var1)||0;
    if u >= sample_ratio then val_chosen = randnormal(1,mean2,var2)||1;
    sample = sample//val_chosen;

    ********** Reclassifying Lambda - KNN **********;
    if upcase(lambda_method) = 'KNN' then do;
        call lambda_calc_KNN(sample,KNN_Number,lambda_use);
    end;

    ********** Classify Sampled Record **********;
    if upcase(lambda_method) = 'LDA' then do;
        if (val_chosen[,2] = 0 & val_chosen[,1] < lambda_use) |
        (val_chosen[,2] = 1 & val_chosen[,1] >= lambda_use) then do;
            Missclass_Matrix = Missclass_Matrix//0;
            Perfect_Class_Counter = Perfect_Class_Counter + 1;
        end;
        else do;
            Missclass_Matrix = Missclass_Matrix//1;
            Perfect_Class_Counter = 0;
        end;
    end;
    if upcase(lambda_method) = 'QDA' then do;
        if (val_chosen[,2] = 0 & 0.5*((val_chosen[,1]-qda_mean_0)`*
        inv(qda_var_0)*(val_chosen[,1]-qda_mean_0) -
        (val_chosen[,1]-qda_mean_1)`*inv(qda_var_1)*
        (val_chosen[,1]-qda_mean_1)) < lambda_use) |
        (val_chosen[,2] = 1 & 0.5*((val_chosen[,1]-qda_mean_0)`*
        inv(qda_var_0)*(val_chosen[,1]-qda_mean_0) -
        (val_chosen[,1]-qda_mean_1)`*inv(qda_var_1)*
        (val_chosen[,1]-qda_mean_1)) >= lambda_use) then do;
            Perfect_Class_Counter = Perfect_Class_Counter + 1;
            Missclass_Matrix = Missclass_Matrix//0;
        end;
        else do;
            Missclass_Matrix = Missclass_Matrix//1;
```

```
            Perfect_Class_Counter = 0;
        end;
end;
if upcase(lambda_method) = 'KNN' then do;
        if val_chosen[,2] = lambda_use then do;
            Perfect_Class_Counter = Perfect_Class_Counter + 1;
            Missclass_Matrix = Missclass_Matrix//0;
        end;
        else do;
            Missclass_Matrix = Missclass_Matrix//1;
            Perfect_Class_Counter = 0;
        end;
end;
if upcase(lambda_method) = 'REG' then do;
        if (val_chosen[,2] = 0 & max((1||val_chosen[,1])*lambda_use) =
        ((1||val_chosen[,1])*lambda_use)[,1]) |
        (val_chosen[,2] = 1 & max((1||val_chosen[,1])*lambda_use) =
        ((1||val_chosen[,1])*lambda_use)[,2]) then do;
            Perfect_Class_Counter = Perfect_Class_Counter + 1;
            Missclass_Matrix = Missclass_Matrix//0;
        end;
        else do;
            Missclass_Matrix = Missclass_Matrix//1;
            Perfect_Class_Counter = 0;
        end;
end;
if i3 = 1 then Missclass_Matrix = Missclass_Matrix[2,];
Q = Missclass_Matrix[,1];

********** Stopping Criteria **********;
accept_ind = 0;

/******************** Current Rules ********************/
if strip(upcase(StopRule)) = 'OLD' then do;
        pie_i = J(i3,1,0);
        do i4 = 1 to i3;
            pie_i[i4,1] = (1/i4)*(Q[1:i4,])`*J(i4,1,1);
        end;
        Kappa_N = (1/i3)*((pie_i#(1-pie_i))`*J(nrow(pie_i),1,1));
        param1 = inv(i3)*(Q[1:i3]`*J(i3,1,1));
```

```
    if param1 < epsilon && Kappa_N > 0 && param1 > 0 then do;
        crit1 = ( (probit(1-alpha)*kappa_n)/(epsilon - param1) )**2;
        if i3 >= crit1 | perfect_class_Counter >= N_Zero
        then accept_ind = 1;
    end;
    else do;
        if perfect_class_Counter >= N_Zero then
        accept_ind = 1;
    end;
end;

/******************** New Rules ********************/
if strip(upcase(StopRule)) = 'NEW' then do;

    if alpha = 0.1 then do;
        if h = 0.1 then do;
            a = 4;      gamma = 0.0754;
        end;
        if h = 0.05 then do;
            a = 4;      gamma = 0.0859;
        end;
        if h = 0.03 then do;
            a = 6;      gamma = 0.0916;
        end;
        if h = 0.02 then do;
            a = 6;      gamma = 0.0945;
        end;
        if h = 0.01 then do;
            a = 6;      gamma = 0.0972;
        end;
    end;
    if alpha = 0.05 then do;
        if h = 0.1 then do;
            a = 4;      gamma = 0.0356;
        end;
        if h = 0.05 then do;
            a = 6;      gamma = 0.0433;
        end;
        if h = 0.03 then do;
```

```
            a = 6;      gamma = 0.0455;
        end;
        if h = 0.02 then do;
            a = 8;      gamma = 0.0472;
        end;
        if h = 0.01 then do;
            a = 10;     gamma = 0.0487;
        end;
    end;
    if alpha = 0.01 then do;
        if h = 0.1 then do;
            a = 6;      gamma = 0.0068;
        end;
        if h = 0.05 then do;
            a = 8;      gamma = 0.0083;
        end;
        if h = 0.03 then do;
            a = 12;     gamma = 0.0092;
        end;
        if h = 0.02 then do;
            a = 14;     gamma = 0.0095;
        end;
        if h = 0.01 then do;
            a = 14;     gamma = 0.0097;
        end;
    end;
    p_hat      = (Q[+,1])/(nrow(Q));
    p_hat_adj = (Q[+,1] + a)/(nrow(Q) + 2*a);
    interval_upper = p_hat + probit(1 - gamma/2)*
    sqrt(p_hat_adj*(1 - p_hat_adj)/nrow(Q));
    interval_lower = p_hat - probit(1 - gamma/2)*
    sqrt(p_hat_adj*(1 - p_hat_adj)/nrow(Q));
    if interval_lower >= (p_hat - h) & interval_lower <= (p_hat + h) &
     interval_upper >= (p_hat - h) & interval_upper <= (p_hat + h)
    then accept_ind = 1;
end;

/******* Calculate Misclassification Rate *******/
if accept_ind = 1 then do;
```

```
        if strip(upcase(lambda_method)) = 'LDA' then do;
            call Error_calc_lda(lambda_use,missclass_rate,mu1,mu2,sig1,sig2,
                                10000,sample_sz1,sample_sz2);
        end;
        if upcase(lambda_method) = 'QDA' then do;
            call Error_calc_qda(lambda_use,missclass_rate,mu1,mu2,sig1,sig2,
                                10000,sample_sz1,sample_sz2,qda_mean_0,
                                qda_mean_1,qda_var_0,qda_var_1);
        end;
        if upcase(lambda_method) = 'KNN' then do;
            call Error_calc_knn(sample,KNN_Number,missclass_rate,mu1,mu2,
                                sig1,sig2,10000,sample_sz1,sample_sz2);
        end;
        if upcase(lambda_method) = 'REG' then do;
            call Error_calc_reg(lambda_use,missclass_rate,mu1,mu2,sig1,sig2,
                                10000,sample_sz1,sample_sz2);
        end;
        call Error_calc_theoret(sample_sz1,sample_sz2,mu1,
                                mu2,sig1,sig2,Bayes_Error);
        Two_Normal_Out_M = Two_Normal_Out_M//
                                (mu1||mu2||eps||alpha||h||i3||
        missclass_rate||Bayes_Error);
        i3 = max_Sample;
    end;
    else if accept_ind = 0 & i3 = max_sample then do;
        if upcase(lambda_method) = 'LDA' then do;
            call Error_calc_lda(lambda_use,missclass_rate,mu1,mu2,sig1,
                                sig2,10000,sample_sz1,sample_sz2);
        end;
        if upcase(lambda_method) = 'QDA' then do;
            call Error_calc_qda(lambda_use,missclass_rate,mu1,mu2,sig1,
                                sig2,10000,sample_sz1,sample_sz2,
                                qda_mean_0,qda_mean_1,
                                qda_var_0,qda_var_1);
        end;
        if upcase(lambda_method) = 'KNN' then do;
            call Error_calc_knn(sample,KNN_Number,missclass_rate,mu1,mu2,
                                sig1,sig2,10000,sample_sz1,sample_sz2);
        end;
```

```
            if upcase(lambda_method) = 'REG' then do;
                call Error_calc_reg(lambda_use,missclass_rate,mu1,mu2,sig1,sig2,
                                    10000,sample_sz1,sample_sz2);
            end;
            call Error_calc_theoret(sample_sz1,sample_sz2,mu1,mu2,
                                    sig1,sig2,Bayes_Error);
            Two_Normal_Out_M = Two_Normal_Out_M//
                                    (mu1||mu2||eps||alpha||h||i3||
            missclass_rate||Bayes_Error);
        end;

        ********** Reclassifying Lambda - LDA **********;
        if accept_ind = 0 & i3 < max_sample then do;
            lambda_use_in = sample;
            if upcase(lambda_method) = 'LDA' then do;
                call lambda_calc_LDA(lambda_use_in,lambda_use);
            end;
            if upcase(lambda_method) = 'QDA' then do;
                call lambda_calc_QDA(lambda_use_in,lambda_use,qda_mean_0,
                                     qda_mean_1,qda_var_0,qda_var_1);
            end;
            if upcase(lambda_method) = 'REG' then do;
                call lambda_calc_REG(lambda_use_in,lambda_use);
            end;
        end;
    end;
end;
varnames = 'mu1'||'mu2'||'eps'||'alpha'||'h'||'sequential_steps'||
           'missclass_rate'||'Bayes_Error';
create Two_Normal_Out_D from Two_Normal_Out_M [colname = varnames];
append from Two_Normal_Out_M;
finish LDA_two_normal_distributions;

*************** Calculating lambda practically - LDA ***************;
start lambda_calc_LDA(input_matrix,output_matrix);
    mean_matrix = input_matrix[loc(input_matrix[,ncol(input_matrix)] = 0),1][:]||
                  input_matrix[loc(input_matrix[,ncol(input_matrix)] = 1),1][:];
    var_matrix = (input_matrix[loc(input_matrix[,ncol(input_matrix)] = 0),1] -
                  mean_matrix[,1])[##,] +
                  (input_matrix[loc(input_matrix[,ncol(input_matrix)] = 1),1]
```

```
                    - mean_matrix[,2])[##,];
        var_matrix = var_matrix/nrow(input_matrix - 2);
        var_matrix = var_matrix||var_matrix;
        output_matrix = ( log(input_matrix[+,2]/(nrow(input_matrix)-input_matrix[+,2]))
        + ((mean_matrix[1,1]**2)/(var_matrix[1,1]) - (mean_matrix[1,2]**2)/
        (var_matrix[1,2]))/2 ) /(mean_matrix[1,1]/
        var_matrix[1,1] - mean_matrix[1,2]/var_matrix[1,2]);
finish lambda_calc_LDA;

*************** Calculating lambda practically - QDA ***************;
start lambda_calc_QDA(input_matrix,output_matrix,mean_matrix_0,
                              mean_matrix_1,var_matrix_0,var_matrix_1);
        qda_train_0 = input_matrix[loc(input_matrix[,ncol(input_matrix)] = 0),];
        qda_train_1 = input_matrix[loc(input_matrix[,ncol(input_matrix)] = 1),];
        mean_matrix_0 = qda_train_0[:,1:(ncol(qda_train_0)-1)];
        mean_matrix_1 = qda_train_1[:,1:(ncol(qda_train_1)-1)];
        var_matrix_0 = (qda_train_0[,1:(ncol(qda_train_0)-1)]-mean_matrix_0)`*
                        (qda_train_0[,1:(ncol(qda_train_0)-1)]-mean_matrix_0)/
                        (nrow(qda_train_0)-1);
        var_matrix_1 = (qda_train_1[,1:(ncol(qda_train_1)-1)]-mean_matrix_1)`*
                        (qda_train_1[,1:(ncol(qda_train_1)-1)]-mean_matrix_1)/
                        (nrow(qda_train_1)-1);
        output_matrix = log(nrow(qda_train_0)/nrow(qda_train_1)) -
                            0.5*log(det(var_matrix_0)) +
                            0.5*log(det(var_matrix_1));
finish lambda_calc_QDA;

*************** Calculating lambda practically - Regression ***************;
start lambda_calc_REG(input_matrix,output_matrix);
        X_Matrix = J(nrow(input_matrix),1,1)||input_matrix[,1];
        Y_Matrix = (1-input_matrix[,2])||input_matrix[,2];
        output_matrix = (inv(X_Matrix`*X_Matrix))*X_Matrix`*Y_Matrix;

finish lambda_calc_REG;
*************** Calculating lambda practically - KNN ***************;
start lambda_calc_KNN(input_matrix,KNN,Output_value);
        KNN_sample = input_matrix[1:nrow(input_matrix)-1,]||
                        abs(input_matrix[1:nrow(input_matrix)-1,]-
                        input_matrix[nrow(input_matrix),1]);
        ndx = 0;
```

```
        call sortndx(ndx,KNN_sample,3,);
        new_index = ndx[1:KNN,];
        KNN_Sample_KNN = J(1,ncol(KNN_Sample),987);
        do knn_cnt = 1 to KNN;
            Knn_val_idx = new_index[knn_cnt,1];
            KNN_Sample_KNN = KNN_Sample_KNN//
                                KNN_Sample[Knn_val_idx,];
        end;
        KNN_Sample_KNN = KNN_Sample_KNN[2:nrow(KNN_Sample_KNN),];
        if (KNN_Sample_KNN[,2][+])/KNN < 0.5 then Output_Value = 0;
        else Output_Value = 1;
finish lambda_calc_KNN;

*************** Calculating error LDA ***************;
start Error_calc_lda(input_val,output_val,mean_val1,mean_val2,var_val1,
                     var_val2,holdout_sample_size,sample1,sample2);
    output_val = 0;
    rat = sample1/(sample1 + sample2);
    output_val = ncol(loc(randnormal(holdout_sample_size*
                     rat,mean_val1,var_val1) > input_val));
    output_val = output_val + ncol(loc(randnormal(holdout_sample_size*
                                (1-rat),mean_val2,var_val2) <= input_val)) ;
    output_val = output_val/holdout_sample_size;
finish Error_calc_lda;

*************** Calculating error KNN ***************;
start Error_calc_knn(training,KNN,output_val,mean_val1,mean_val2,var_val1,
                  var_val2,holdout_sample_size,sample1,sample2);
    output_val = 0;
    rat = sample1/(sample1 + sample2);
    input_matrix = (randnormal(holdout_sample_size*rat,mean_val1,var_val1)
                     ||J(holdout_sample_size*rat,1,0))//
                     (randnormal(holdout_sample_size*(1-rat),mean_val2,var_val2)
                     ||J(holdout_sample_size*(1-rat),1,1));
    do iii = 1 to nrow(input_matrix);
        value = ((training[,1:ncol(training)-1]-
                 input_matrix[iii,1:ncol(input_matrix)-1])##2)
                 *J(ncol(input_matrix)-1,1,1);
        KNN_sample = training||value;
        ndx = 0;
```

```
            call sortndx(ndx,KNN_sample,ncol(KNN_sample),);
            new_index = ndx[1:KNN,];
            KNN_Sample_KNN = J(1,ncol(KNN_Sample),9876);
            do knn_cnt = 1 to KNN;
                Knn_val_idx = new_index[knn_cnt,1];
                KNN_Sample_KNN = KNN_Sample_KNN//
                                        KNN_Sample[Knn_val_idx,];
            end;
            KNN_Sample_KNN = KNN_Sample_KNN[2:nrow(KNN_Sample_KNN),];
            if (KNN_Sample_KNN[,ncol(KNN_Sample_KNN)-1][+])/KNN < 0.5
            then knn_class = 0;
            else knn_class = 1;
            if knn_class ^= input_matrix[iii,ncol(training)] then
            output_val = output_val + 1;
        end;
        output_val = output_val/nrow(input_matrix);
finish Error_calc_knn;

*************** Calculating error QDA ***************;
start Error_calc_qda(input_val,output_val,mean_val1,mean_val2,var_val1,var_val2,
                        holdout_sample_size,sample1,sample2,mean_matrix_0,
                        mean_matrix_1,var_matrix_0,var_matrix_1);
        rat = sample1/(sample1 + sample2);
        x0 = randnormal(holdout_sample_size*rat,mean_val1,var_val1);
        x1 = randnormal(holdout_sample_size*(1-rat),mean_val2,var_val2);
        x0 = 0.5*((x0-mean_matrix_0)#inv(var_matrix_0)#(x0-mean_matrix_0) -
            (x0-mean_matrix_1)#inv(var_matrix_1)#(x0-mean_matrix_1));
        x1 = 0.5*((x1-mean_matrix_0)#inv(var_matrix_0)#(x1-mean_matrix_0) -
            (x1-mean_matrix_1)#inv(var_matrix_1)#(x1-mean_matrix_1));
        output_val = (ncol(loc(x0 > input_val)) + ncol(loc(x1 <= input_val)))
                        /holdout_sample_size;
finish Error_calc_qda;

*************** Calculating error Regression ***************;
start Error_calc_reg(input_val,output_val,mean_val1,mean_val2,var_val1,var_val2,
                        holdout_sample_size,sample1,sample2);
        rat = sample1/(sample1 + sample2);
        x0 = randnormal(holdout_sample_size*rat,mean_val1,var_val1);
        x1 = randnormal(holdout_sample_size*(1-rat),mean_val2,var_val2);
        output_val = 0;
```

```
    do ii1 = 1 to nrow(x0);
        if max((1||x0[ii1,])*input_val) = ((1||x0[ii1,])*input_val)[,2] then
                                        output_val = output_val + 1;
    end;
    do ii2 = 1 to nrow(x1);
        if max((1||x1[ii2,])*input_val) = ((1||x1[ii2,])*input_val)[,1] then
                                        output_val = output_val + 1;
    end;
    output_val = output_val/holdout_sample_size;
finish Error_calc_reg;

*************** Calculating error theoretically i.e. Bayes error ***************;
start Error_calc_theoret(sampsize1,sampsize2,mean1,mean2,sigma1,sigma2,output_val);
    rat = sampsize1/(sampsize1 + sampsize2);
    if sigma1 = sigma2 then do;
        sigma_overall = ( sigma1*(sampsize1 - 1) + sigma2*(sampsize2 - 1) )/
                        (sampsize1 + sampsize2 - 2);
        lambda_bayes = (log(sampsize1/sampsize2) + 0.5*(mean2*inv(sigma_overall)*
                        mean2 - mean1*inv(sigma_overall)*mean1))/
                        (inv(sigma_overall)*(mean2 - mean1));
        output_val = rat*(1 - probnorm( (lambda_bayes - mean1)/sqrt(sigma1) ))
                    +(1 - rat)*probnorm( (lambda_bayes - mean2)/
                    sqrt(sigma2) );
    end;
    else do;
        lambda_bayes1 = (2*(mean1/sigma1 - mean2/sigma2) +
                        sqrt(4*(mean1/sigma1 - mean2/sigma2)**2
                        - 4*(1/sigma1 - 1/sigma2)*( -2*log(sampsize1/sampsize2)
                        + log(sigma1/sigma2) +mean1*mean1/sigma1 -
                        mean2*mean2/sigma2 ))) / (2*(1/sigma1 - 1/sigma2));
        lambda_bayes2 = (2*(mean1/sigma1 - mean2/sigma2) -
                        sqrt(4*(mean1/sigma1 - mean2/sigma2)**2
                        - 4*(1/sigma1 - 1/sigma2)*( -2*log(sampsize1/sampsize2)
                        + log(sigma1/sigma2) +mean1*mean1/sigma1 -
                        mean2*mean2/sigma2 ))) / (2*(1/sigma1 - 1/sigma2));
        if lambda_bayes1 > lambda_bayes2 then lambda_bayes = lambda_bayes1;
        else lambda_bayes = lambda_bayes2;
        output_val = rat*(1 - probnorm( (lambda_bayes - mean1)/sqrt(sigma1) ))
                    + (1 - rat)*probnorm( (lambda_bayes - mean2)/
```

```
                                    sqrt(sigma2) );
        end;
finish Error_calc_theoret;

call LDA_two_normal_distributions(&StopRule.,&sample_sz1.,&sample_sz2.,&mu1.,
                                                    &mu2.,&sig1.,&sig2.,&eps.,&alpha.,&h.,
                                                    &repititions.,&max_sample.,&lambda_method.,
                                                    &Nearest_Neighbours.,&two_normal_out_m.);

                                *** Parameter order ****;
/*StopRule - NEW indicates proposed method, OLD indicates current methodology*/
/*sample_sz1 = the number of observations to generate from sample1;*/
/*sample_sz2 = the number of observations to generate from sample2;*/
/*mu1 = mean value for generating sample1 observations;*/
/*mu2 = mean value for generating sample2 observations;*/
/*sig1 = variance for generating sample1 observations;*/
/*sig2 = variance for generating sample2 observations;*/
/*eps = epsilon;*/
/*a = alpha significance level;*/
/*h = halfwidth of confidence interval for proposed approach*/
/*repititions = number of repititions that the recursive method should be iterated;*/
/*max_sample = maximum number of trails allowed;*/
/*lambda_method = LDA, QDA, KNN, REG variable to select whether lambda
is calculated using K-Nearest, LDA, QDA or Regression of Indicator Matrix*/
/*KNN_Neighbours = K-Nearest Neighbours if LAMBDA_METHOD = KNN*/
/*two_normal_out_m = Output matrix for two normal distribution inputs*/
quit;

data _null_;
      delta = transtrn(input(&mu2. - &mu1.,$12.),'.',('_'));
      epsname = transtrn(input(&eps.,$12.),'.','_');
      call symput('delta',compress(delta));
      call symput('epsname',compress(epsname));
run;

proc means data = Two_normal_out_d noprint;
var sequential_steps missclass_rate Bayes_Error;
output out = output_&delta._&epsname._metric (keep = mean_missclass
std_missclass min_seq max_seq mean_seq std_seq mean_Bayes)
min = min_seq min_missclass min_Bayes max = max_seq max_missclass
```

```
max_Bayes mean = mean_seq mean_missclass mean_Bayes std = std_seq
                        std_missclass std_Bayes;
quit;

data output_&delta._&epsname._metric;
set output_&delta._&epsname._metric;
delta = transtrn("&delta.","_",".");
epsilon = transtrn("&epsname.","_",".");
run;

%if %sysfunc(EXIST(final_output)) ^= 1 %then %do;
data final_output;
    set Output_1_0_05_metric;
run;
%end;
%else %do;
    data final_output;
        set final_output output_&delta._&epsname._metric;
    run;
%end;
%mend determine_sample_size;

%macro all_delta;
    %do epsloop = %sysevalf(0.05*100) %to %sysevalf(0.2*100) %by %sysevalf(0.05*100);
        %let eps_loop = %sysevalf(&epsloop./100);
        %loop(%str(1, 1.3, 1.5, 2, 2.3, 2.5, 3, 4));
/*This is the list of all delta values you want to loop through*/
    %end;
%mend all_delta;

%macro loop(values);
/* Count the number of values in the string */
%let count=%sysfunc(countw(&values,','));

/* Loop through the total number of values */
%do i = 1 %to &count;
%let delta_loop=%qscan(&values,&i,%str(,));
    %determine_sample_size('OLD',5,5,0,&delta_loop.,1,1,&eps_loop.,0.05,0.1,1000
                                    ,90,'LDA',3,Output);
%end;
%mend;
```

******* Now running the program *******;
%all_delta;