Coventry University

DOCTOR OF PHILOSOPHY

Bayes-optimal linear discriminant analysis under heteroscedasticity

Gyamfi, Sarfo

Award date:
2018

Awarding institution:
Coventry University

Bayes-optimal linear discriminant analysis under heteroscedasticity

Kojo Sarfo Gyamfi

A thesis submitted in partial fulfilment
of the University's requirements for the Degree of
Doctor of Philosophy

January, 2018

Coventry University
Faculty of Engineering and Computing

To the memory of my father, the best there ever was, the kind I strive to be...

Abstract

Linear discriminant analysis (LDA) has been applied to many machine learning applications such as medical diagnosis, face and object detection, handwriting recognition, spam filtering and credit card fraud prediction. LDA is used either for supervised linear dimensionality reduction of high-dimensional datasets or for statistical classification. Under the assumptions of normally-distributed classes and equal covariance matrices among the classes, LDA is known to be optimal in terms of minimising the Bayes error—the minimum achievable error rate by a classifier whose predictions are based on the knowledge of the stochastic process generating the data. The widespread use of LDA in the application areas indicated above is not because the datasets necessarily satisfy the two assumptions, but mainly due to the robustness of LDA. Nonetheless, for many other applications, the performance of LDA can be unsatisfactory, if the assumptions of normally-distributed classes and equal covariance are not met.

This thesis primarily addresses the violation of the assumption of equal covariance, also known as homoscedasticity.

For statistical classification, accounting for homoscedasticity has led to a number of heteroscedastic extensions of LDA, the most natural extension being quadratic discriminant analysis (QDA). However, QDA tends to over-fit for many real-world datasets, especially if the normal distribution assumption is also violated. Thus, heteroscedastic LDA (HLDA) procedures have involved finding a linear approximation to the quadratic boundary in QDA. However, most of these HLDA procedures have no principled optimisation procedure, as they are obtained via trial and error. As a result, they tend to be computationally intractable for high-dimensional datasets. Other HLDA approaches constrain the domain of the search space in an attempt to reduce the computational complexity; this, however, leads to poor performance in terms of the classification accuracy and the area under the receiver operating characteristics curve (AUC) under class imbalance. Using first and second-order optimality conditions for the minimisation of the Bayes error, a dynamic Bayes-optimal linear classifier for heteroscedastic LDA that is robust against class imbalance, and is optimised via a computationally efficient iterative procedure, is derived. The proposed model, referred to as Gaussian linear discriminant (GLD), is also formulated as a kernel classifier, in order to learn non-linear decision boundaries.

For the purpose of linear dimensionality reduction (LDR), existing heteroscedastic LDA approaches involve the minimisation of some upper bounds of the Bayes error, or the maximisation of some measures of class separation. These procedures are often reformulated as eigenvalue decomposition or singular value decomposition (SVD) problems, after which a desired dimensionality q is chosen by taking the first q independent vectors after the decomposition. However, these procedures provide no optimal dimensionality to which to reduce the data, and consequently, they do not preserve the classification information in the original data after the dimensionality reduction. This thesis presents a novel LDR technique to reduce the dimensionality of the original data to $K-1$ for a K-class problem, such that the linearly-reduced data is well-primed for Bayesian classification. This technique is referred to as multi-class Gaussian linear discriminant (M-GLD), and it involves sequentially constructing GLD classifiers that minimise the Bayes error via a gradient descent procedure, under an assumption of within-class normality.

Experimental validation carried out on several artificial and real-world datasets from the University of California, Irvine (UCI) machine learning repository, highlight the scenarios under which the proposed algorithms achieve superior performance to the original LDA and existing HLDA approaches.

Finally, the utility of the proposed algorithms is demonstrated by applying them to flow meter fault diagnosis. Using data from 4 liquid ultrasonic flow meters, the proposed M-GLD dimensionality reduction procedure and GLD classifier are used to achieve diagnostic accuracies of between 97.2% and 100%; this far exceeds the performance of existing LDA procedures, as well as that of support vector machine (SVM). High diagnostic accuracies promise significant cost benefits in oil and gas operations.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my Director of Studies, Dr. James Brusey, for the continuous support throughout my doctoral research. James has been there to motivate me every step of the way. He has provided sage advice and timely, insightful feedback which have helped define the course of my research.

Next, I would like to thank my second supervisor, Professor Andrew Hunt, for the opportunity he gave me to view my research from different perspectives from the ones I was used to. Andy has been the one to always get me thinking about the real-world applications of my ideas and research outputs, particularly for flowmeter diagnostics.

I would especially like to thank my third supervisor, Professor Elena Gaura, for the faith she showed in me to complete my thesis in an unusually short time. Elena's critical feedback have been particularly useful in solidifying my research, and helping me see the bigger picture.

I am grateful also to the Cogent Labs team: Dr. Ross Wilkins, Alexandra Petre, James Wescott, Gene Palencia, Gaobo Chen, Nicolas Merlinge and Ross Drury, for providing a fun and conducive environment for learning and research.

Finally, I would like to express my profound gratitude to my family and friends for the immense love, care and encouragement given me during my studies, and for having to put up with my frustrations and sharing in my joy along this journey.

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

AGC	Automatic Gain Control
AUC	Area Under Curve
BER	Balanced Error Rate
CBM	Condition Based Management
CDF	Cumulative Distribution Function
C-HLD	Constrained Heteroscedastic Linear Discriminant
C-LDR	Chernoff criterion Linear Dimensionality Reduction
DDM	Directed Distance Matrix
D-GLD	Dynamic Gaussian Linear Discriminant
ER	Error Rate
FLD	Fisher's Linear Discriminant
F-LDR	Fisher's criterion Linear Dimensionality Reduction
FN	False Negatives
FP	False Positives
FPR	False Positive Rate
GLD	Gaussian Linear Discriminant
G-GLD	Gradient descent Gaussian Linear Discriminant
GVF	Gas Volume Fraction
HLD	Heteroscedastic Linear Discriminant
HLDA	Heteroscedastic Linear Discriminant Analysis
K-CHLD	Kernel Constrained Heteroscedastic Linear Discriminant
KFD	Kernel Fisher's Discriminant
K-GLD	Kernel Gaussian Linear Discriminant
K-PCA	Kernel Principal Component Analysis

K-RHLD	Kernel Random Heteroscedastic Linear Discriminant
K-SVM	Kernel Support Vector Machine
LDA	Linear Discriminant Analysis
LDR	Linear Dimensionality Reduction
LNS	Local Neighbourhood Search
MAP	Maximum *a posteriori*
M-GLD	Multiclass Gaussian Linear Discriminant
ML	Maximum Likelihood
M-LDR	Mahalanobis distance Linear Dimensionality Reduction
NEL	National Engineering Laboratory
OvA	One vs All
OvO	One vs One
PCA	Principal Component Analysis
PDF	Probability Density Function
QDA	Quadratic Discriminant Analysis
R-GLD	Recursive Gaussian Linear Discriminant
R-HLD	Random Heteroscedastic Linear Discriminant
ROC	Receiver Operating Characteristics
SMOTE	Synthetic Minority Over Sampling Technique
SNR	Signal to Noise Ratio
SoS	Speed of Sound
SSS	Small Sample Size
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TN	True Negatives
TP	True Positives
TPR	True Positive Rate
USM	Ultrasonic Flow Meter

Chapter 1

Introduction

A common theme in machine learning is how to correctly classify an object under one of a given number of categories or classes, based on the features of the object and existing data about the process; this is termed classification. A typical example is the task of having to classify an incoming email as spam or legitimate, or a breast tumour as benign or cancerous. One of the widely used machine learning techniques for classification is linear discriminant analysis (LDA), due to its simplicity and robustness [2]. Prior to classification, LDA can also be used to reduce the dimensionality of the existing data, which is given by the number of different features of the object to be classified.

Reducing the dimensionality of a dataset is an important preprocessing step in machine learning for a number of reasons. On the one hand, dimensionality reduction enables easy visualisation of data when the data is reduced to two or three dimensions. On the other hand, reducing the dimensionality often reduces the complexity of learning algorithms in many applications. For example, in face recognition, where the face images live in a high-dimensional space (often the dimensionality is equal to the number of pixels in the face image [3]), a large dimensionality drastically increases the complexity of the learning algorithm. To put this into perspective, an image with $2,000,000$ pixels (2MP) would require the inversion of a matrix of size $2,000,000$ in order to apply a classification algorithm such as quadratic discriminant analysis (QDA); this is computationally intractable for face recognition in most modern machines (including, most certainly, the iPhone X), without any dimensionality reduction. More importantly, however, dimensionality reduction often improves the accuracy of classification in the low-dimensional manifold in which the data is reduced to lie [4, 5]. This is usually due to the fact that the original high-dimensional data often contain noisy and redundant features, so that reducing the dimensionality results in useful feature extraction from the data, which tends to reduce over-fitting [6, 7].

LDA has been applied in several applications such as medical diagnosis, handwriting recognition, face and object detection and spam filtering, either for classification, or to reduce the dimensionality of high-dimensional datasets.

At its core, however, LDA assumes that the data in each class is normally distributed, and that the

covariance matrices are equal among the classes; the equal covariance assumption is referred to as homoscedasticity. When these assumptions are met, LDA minimises the Bayes error, which is the minimum achievable error rate by a classifier that makes predictions using knowledge of the true distribution of the data [7, 8]). Nonetheless, in many other applications, the assumptions of normally-distributed classes and equal covariance are not met, and therefore the Bayes error is not minimised; consequently, the performance of LDA is unsatisfactory in terms of the accuracy of classification. One such application in which homoscedasticity is not satisfied is flowmeter fault diagnosis.

1.1 Motivation

The work described in this thesis is motivated by the application of LDA to flowmeter fault diagnosis.

Flowmeters are devices used to measure the volumetric or mass flow rate of a fluid. They come in different forms, and the physical principles on which they operate include ultrasound Doppler shift, Coriolis effect, and capacitance and inductance tomography. In the oil and gas industry, these meters are often subject to several problems such as transducer failure and wax deposit, as well as harsh conditions including extremes in temperature and pressure. These problems affect the performance of the meter and, with time, cause the flow rate readings to be erroneous. The problem of incorrect measurement is of great concern in the industry, since, for example, an incorrect measurement indicating a high flow rate may attract high tax liabilities.

It is understood that after a period of systematic use of the meter, the errors associated with the flow measurement may become significant and fall outside an allowable range. Thus, it is the current practice that flowmeters are taken to accredited flow facilities to be recalibrated typically after one year in operation.

Nevertheless, this time-based recalibration system has two main drawbacks. First, a given flowmeter may encounter a problem, such as a transducer failure, even before the one year schedule, and continuously provide incorrect measurements until the recalibration period is up. Second, a flowmeter under consideration may be operating perfectly at the end of the one year period and still be taken in for recalibration, in line with regulatory requirements. However, recalibration of a flowmeter can be expensive. In the United Kingdom, for instance, it costs in the region of £30,000 for the recalibration of an ultrasonic flowmeter [1].

Thus, the trade-off between having accurate measurements and reducing costs incurred from frequent recalibration of a flowmeter calls for the adoption of a condition-based flowmeter management system. In

such a system, the condition of the flowmeter is continuously monitored so that if the monitored values indicate an unhealthy meter, flowmeter operators in the field may act to mitigate the problem, and restore the integrity of the measurement. Similarly, if the monitored values indicate a healthy meter, such that the measurement integrity of the meter is not compromised even at the point of its recalibration schedule, recalibration can be extended [1], thus resulting in significant cost savings.

Condition-based management is made possible with the advent of new flowmeters that provide secondary diagnostic information in addition to the primary flow measurement. Unfortunately, the volume of diagnostic variables available makes it particularly difficult for flowmeter operators to interpret the data to know the condition or health state of a meter. For example, for an 8-path ultrasonic flowmeter, the different diagnostic variables available can number be anywhere between 20 and 100 [1]; most of these variables happen to be noisy and redundant. Because of their number, only a handful of the diagnostic variables are utilised for diagnostics. Consequently, the full diagnostic capability of a flowmeter is underexploited. Specifically, flow computers are often only able to display the colours: "Red", "Amber" and "Green" to indicate an unhealthy meter, a warning, and a healthy meter respectively [9]. It does not suffice to simply know that a meter is unhealthy, without knowing the nature of the problem, as this can lead to long periods of downtime in order to isolate the problem, mitigate it and restore the measurement integrity. It is more appropriate to have an expert system that provides more specific diagnostics such as, a "wax deposit in Port A" or a "vertical misalignment of flowmeter" [9].

Thus, the aim of flowmeter diagnostics is twofold: first, to reduce the wealth of diagnostic information available for a given flowmeter to a few useful diagnostic variables that can be easily analysed by meter operators (dimensionality reduction); secondly, to design an expert system to correctly diagnose a given flowmeter under a number of known health states of the meter (classification).

Like many physical data, such as those involving measurement errors [10], flowmeter diagnostics data tend to be nearly-normally distributed in each class or health state (see section 5.3), thus satisfying the normality assumption in LDA. Yet, the peculiarities of the diagnostics problem do not allow a straightforward application of LDA. In particular,

1. The covariance matrices of the classes or health states of a given flowmeter are not necessarily equal (see section 5.3). This is known as heteroscedasticity.

2. There is the issue of class imbalance, a term used to describe the scenario where the cardinality of the data in one class far exceeds those in the other classes; this leads to one class being far more probable than the other classes in the classification task. In flowmeter diagnostics, class imbalance

is pertinent because for a given flowmeter in operation, there is a much higher probability that the meter is healthy than it is in a particular unhealthy state. It has been claimed that, under heteroscedasticity, class imbalance has a negative effect on LDA [11, 12].

3. There are more than two health states for a given flowmeter (see section 5.3). Given that the individual health states tend to be nearly normally distributed, optimum classification and dimensionality reduction can be achieved by minimising the Bayes error [8]. However, unless there are only two classes, LDA does not guarantee the minimisation of the Bayes error, even when the assumptions of homoscedasticity and normality are satisfied.

1.2 Research questions

In light of the above problems with the application of LDA to flowmeter diagnostics, this thesis attempts to answer the following research questions:

1. How can heteroscedasticity be accounted for in LDA while minimising the Bayes error for classification?

2. What is the effect of class imbalance on LDA when heteroscedasticity has been accounted for?

3. How can heteroscedasticity be accounted for in LDA while minimising the Bayes error for dimensionality reduction?

4. Does accounting for heteroscedasticity in LDA improve the accuracy of diagnosis for a given flowmeter?

1.3 Contributions to knowledge

The work described in this thesis has led to the following contributions to knowledge:

1. A computationally efficient heteroscedastic LDA procedure, termed the Gaussian Linear Discriminant (GLD), that minimises the Bayes error in the two-class scenario. This procedure is described in Chapter 3.

2. A local neighbourhood search procedure that accounts for non-normality in the data in each class. This procedure is described in Chapter 3.

3. An optimal design of a linear classifier for heteroscedastic LDA under class imbalance. This optimal design is described in Chapter 3.

4. A scheme for the generalisation of GLD to multiple classes for dimensionality reduction via a sequential minimisation of the Bayes error. Chapter 4 describes this generalisation.

1.4 Publications

The following are the publications that have resulted from the work on which this thesis is based:

Journal articles

- **K. S. Gyamfi**, J. Brusey, A. Hunt and E. Gaura. 'Linear classifier design for heteroscedastic LDA under class imbalance'. Accepted in Neurocomputing (to appear 2018).

- **K. S. Gyamfi**, J. Brusey, A. Hunt and E. Gaura. 'Linear dimensionality reduction for classification via a sequential Bayes error minimisation with an application to flowmeter diagnostics'. In: Expert Systems with Applications (2017), vol. 91, Sep. 2017, pp. 252-262. `https://doi.org/10.1016/j.eswa.2017.02.039`

- **K. S. Gyamfi**, J. Brusey, A. Hunt and E. Gaura. 'Linear classifier design under heteroscedasticity in Linear Discriminant Analysis'. In: Expert Systems with Applications, vol. 79, Aug. 2017, pp. 44-52. `https://doi.org/10.1016/j.eswa.2017.02.039`

Conference and workshop proceedings

- **K. S. Gyamfi**, J. Brusey, A. Hunt and E. Gaura. 'Linear classifier design for heteroscedastic LDA under class imbalance'. Proceedings of the Workshop on Learning in the Presence of Class Imbalance and Concept Drift, Melbourne, IJCAI 2017, pp. 8-15 `https://arxiv.org/ftp/arxiv/papers/1707/1707.09425.pdf`

- **K. S. Gyamfi**, J. Brusey, A. Hunt and E. Gaura. 'K-Means clustering using Tabu Search with quantized means'. Proceedings of the World Congress on Engineering and Computer Science. Vol. 1. 2016, pp. 426-432. `http://www.iaeng.org/publication/WCECS2016/WCECS2016_pp426-432.pdf`

1.5 Thesis structure

This chapter presents an introduction to the thesis, including the motivation for the work. The chapter has also presented the research questions the thesis aims to answer, as well as the contributions to knowledge that have resulted from the work described in the thesis.

The rest of the thesis is organised as follows:

Chapter 2 provides a survey and discussion of the literature in the area of linear dimensionality reduction and statistical classification, in line with the two objectives. A special emphasis is given to Linear Discriminant Analysis (LDA), which is both a linear dimensionality reduction procedure and a classification technique, and is the approach upon which this thesis builds.

Chapter 3 presents a computationally efficient heteroscedastic LDA procedure to minimise the Bayes error in the two-class scenario. The resulting algorithm is known as the Gaussian Linear Discriminant (GLD). The kernel formulation of the GLD is also provided. This chapter also presents an optimal design for a linear classifier in heteroscedastic LDA under class imbalance.

Chapter 4 presents a scheme for the generalisation of the GLD for multiple classes via a sequential minimisation of the Bayes error.

Chapter 5 discusses the application of the algorithms from Chapters 3 and 4 to ultrasonic flowmeter diagnostics.

Chapter 6 then concludes the thesis and discusses possible directions for future work.

1.6 Acknowledgement of contributed work

This section details the contribution made by other researchers which have aided the work presented in this thesis:

- The diagnostics data for the four liquid ultrasonic flowmeters were provided by Christopher Mills and Craig Marshall of NEL.
- Dr. Stefan Berres of the Department of Mathematical and Physical Science, Temuco Catholic University, Temuco, Chile, provided useful feedback regarding the mathematics involved in the GLD procedure.

Chapter 2

Linear discriminant analysis (LDA)[1]

This chapter provides the background for the work presented in subsequent chapters of this thesis. The chapter provides a detailed treatment of the machine learning technique of linear discriminant analysis (LDA), first as a classification technique and then as a linear dimensionality reduction procedure. In these two treatments, this chapter surveys the relevant literature, and in doing so, the chapter reveals the gaps in existing knowledge.

2.1 Statistical classification

In many applications, one encounters the need to classify a given object under one of a number of distinct groups or classes based on a set of features known as the feature vector, which is a numerical representation of the object. A typical example is the task of classifying a flowmeter under one of a number of health states. Other applications that involve classification include face detection [13, 14, 15], object recognition [16, 17, 18], medical diagnosis [19, 20, 21], credit card fraud prediction [22, 23, 24, 25] and machine fault diagnosis [26, 27, 28].

A common treatment of such classification problems is to model the conditional density functions of the feature vector [29]. This allows the construction of a Bayes classifier, which is the best possible classifier if the underlying distribution of the data is known [30]. The Bayes classifier assigns a given object to a class based on the *a posteriori* probability of the object. This is known as the maximum *a posteriori* (MAP) decision rule.

Consider a training dataset $\mathcal{X}$ made up of n feature vectors,each of dimensionality d, i.e., $\mathcal{X} = [\mathbf{x}_1, ..., \mathbf{x}_n]$ where $\mathbf{x}_i \in \mathbb{R}^d$, $\forall i \in \{1, ..., n\}$. Suppose that the data is labelled and can be divided into K classes thus: $\mathcal{X} = [\mathcal{D}_1, ..., \mathcal{D}_K]$, where $\mathcal{D}_k$ are the training samples belonging to the kth class ($k \in$

[1]Most of the work presented in this chapter first appeared in: K. S. Gyamfi, J. Brusey, A. Hunt and E. Gaura, "Linear classifier design under heteroscedasticity in Linear Discriminant Analysis," *Expert Systems with Applications*, vol. 79, Aug. 2017, pp. 44-52; in: K. S. Gyamfi, J. Brusey, A. Hunt and E. Gaura, "Linear classifier design for heteroscedastic LDA under class imbalance," *Proceedings of the Workshop on Learning in the Presence of Class Imbalance and Concept Drift*, Melbourne, IJCAI 2017, pp. 8-15; and in: K. S. Gyamfi, J. Brusey, A. Hunt and E. Gaura, "Linear dimensionality reduction for classification via a sequential Bayes error minimisation with an application to flowmeter diagnostics," *Expert Systems with Applications* (2017), vol. 91, Sep. 2017, pp. 252-262.

$\{1, 2, ..., K\}$). Let $\mathcal{C}_k$ be the class label for the kth class. Then for a given feature vector $\mathbf{x}$, the MAP decision rule for the classification task is to choose the most likely class of $\mathbf{x}$, $\mathcal{C}^*(\mathbf{x})$ given as:

$$\mathcal{C}^*(\mathbf{x}) = \arg\max_{\mathcal{C}_k} p(\mathcal{C}_k|\mathbf{x}), \quad k \in \{1, 2, ..., K\} \tag{2.1}$$

For the moment, it is assumed that there are only $K = 2$ classes, i.e. binary classification (see section 2.1.4 for multi-class classification). Then the decision rule of (2.1) may be expressed as:

$$\frac{p(\mathcal{C}_1|\mathbf{x})}{p(\mathcal{C}_2|\mathbf{x})} \underset{\mathcal{C}_2}{\overset{\mathcal{C}_1}{\gtrless}} 1 \tag{2.2}$$

Using Bayes rule, however, the two posterior probabilities can be expressed as:

$$p(\mathcal{C}_1|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_1) \times p(\mathcal{C}_1)}{p(\mathbf{x})} \quad \text{and} \quad p(\mathcal{C}_2|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_2) \times p(\mathcal{C}_2)}{p(\mathbf{x})} \tag{2.3}$$

It is sometimes the case that the prior probabilities $p(\mathcal{C}_1)$ and $p(\mathcal{C}_2)$ are known, or else they may be estimable from the relative frequencies of $\mathcal{D}_1$ and $\mathcal{D}_2$ in $\mathcal{X}$; let these priors be given by π_1 and π_2 respectively for class $\mathcal{C}_1$ and $\mathcal{C}_2$, i.e.,

$$\pi_1 = \frac{n_1}{n}, \quad \pi_2 = \frac{n_2}{n} \tag{2.4}$$

where n_1 and n_2 are the cardinalities of $\mathcal{D}_1$ and $\mathcal{D}_2$ respectively.

Thus, the decision rule of (2.2) may again be rewritten as:

$$\frac{p(\mathbf{x}|\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)} \underset{\mathcal{C}_2}{\overset{\mathcal{C}_1}{\gtrless}} \frac{\pi_2}{\pi_1}. \tag{2.5}$$

so that one decides on class $\mathcal{C}_1$ if $\lambda(\mathbf{x}) \geq \tau$ and class $\mathcal{C}_2$ otherwise, where $\lambda(\mathbf{x}) = p(\mathbf{x}|\mathcal{C}_1)/p(\mathbf{x}|\mathcal{C}_2)$ is the likelihood ratio and $\tau = \pi_2/\pi_1$ is a threshold. If $\mathcal{C}_1$ and $\mathcal{C}_2$ are equally probable, i.e., $\pi_1 = \pi_2$, then the MAP decision rule of (2.5) becomes a maximum likelihood (ML) decision rule.

A major limitation of the MAP decision rule is the difficulty in estimating the conditional distributions $p(\mathbf{x}|\mathcal{C}_1)$ and $p(\mathbf{x}|\mathcal{C}_1)$. For this reason, LDA proceeds from (2.5) with two basic assumptions [31, Chapter 8]:

1. The conditional probabilities $p(\mathbf{x}|\mathcal{C}_1)$ and $p(\mathbf{x}|\mathcal{C}_2)$ have multivariate normal distributions.

2. The two classes have equal covariance matrices, an assumption known as homoscedasticity.

Let $\bar{\mathbf{x}}_1$, $\mathbf{\Sigma}_1$ be the mean and covariance matrix of $\mathcal{D}_1$ and $\bar{\mathbf{x}}_2$, $\mathbf{\Sigma}_2$ be the mean and covariance of $\mathcal{D}_2$ respectively. The normality assumption allows the conditional probabilities $p(\mathbf{x}|\mathcal{C}_1)$ and $p(\mathbf{x}|\mathcal{C}_2)$ to be

expressed as:

$$p(\mathbf{x}|\mathcal{C}_1) = \frac{1}{\sqrt{(2\pi)^d \det(\mathbf{\Sigma}_1)}} \exp\left[-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}}_1)^\top \mathbf{\Sigma}_1^{-1}(\mathbf{x} - \bar{\mathbf{x}}_1)\right] \tag{2.6}$$

and

$$p(\mathbf{x}|\mathcal{C}_2) = \frac{1}{\sqrt{(2\pi)^d \det(\mathbf{\Sigma}_2)}} \exp\left[-\frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}}_2)^\top \mathbf{\Sigma}_2^{-1}(\mathbf{x} - \bar{\mathbf{x}}_2)\right]. \tag{2.7}$$

Given the above definitions of the conditional probabilities, one may evaluate the natural logarithm of (2.5), yielding the log-likelihood ratio given as:

$$\ln \lambda(\mathbf{x}) = \frac{1}{2} \ln \frac{\det \mathbf{\Sigma}_2}{\det \mathbf{\Sigma}_1} + \frac{1}{2}\left[(\mathbf{x} - \bar{\mathbf{x}}_2)^\top \mathbf{\Sigma}_2^{-1}(\mathbf{x} - \bar{\mathbf{x}}_2) - (\mathbf{x} - \bar{\mathbf{x}}_1)^\top \mathbf{\Sigma}_1^{-1}(\mathbf{x} - \bar{\mathbf{x}}_1)\right] \tag{2.8}$$

which is then compared against $\ln \tau$ so that $\mathcal{C}_1$ is chosen if $\ln \lambda(\mathbf{x}) \geq \ln \tau$, and $\mathcal{C}_2$ otherwise. Therefore, the decision rule for classifying a vector $\mathbf{x}$ under class $\mathcal{C}_1$ becomes:

$$(\mathbf{x} - \bar{\mathbf{x}}_2)^\top \mathbf{\Sigma}_2^{-1}(\mathbf{x} - \bar{\mathbf{x}}_2) - (\mathbf{x} - \bar{\mathbf{x}}_1)^\top \mathbf{\Sigma}_1^{-1}(\mathbf{x} - \bar{\mathbf{x}}_1) \geq \ln \frac{\tau^2 \det \mathbf{\Sigma}_1}{\det \mathbf{\Sigma}_2} \tag{2.9}$$

which can be expressed in the quadratic form:

$$\mathbf{x}^\top \mathbf{A}\mathbf{x} + \mathbf{b}^\top \mathbf{x} + c \geq \ln \frac{\tau^2 \det \mathbf{\Sigma}_1}{\det \mathbf{\Sigma}_2} \tag{2.10}$$

where

$$\mathbf{A} = (\mathbf{\Sigma}_2^{-1} - \mathbf{\Sigma}_1^{-1}) \quad \mathbf{b} = -2(\mathbf{\Sigma}_2^{-1}\bar{\mathbf{x}}_2 - \mathbf{\Sigma}_1^{-1}\bar{\mathbf{x}}_1) \quad c = \bar{\mathbf{x}}_2^\top \mathbf{\Sigma}_2^{-1}\bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1^\top \mathbf{\Sigma}_1^{-1}\bar{\mathbf{x}}_1 \tag{2.11}$$

In general, this result is a quadratic discriminant. However, a linear classifier is often desired for the following reasons:

1. A linear classifier is robust against noise since it tends not to overfit [2].

2. A linear classifier has relatively shorter training and testing times [32].

3. Many linear classifiers allow for a transformation of the original feature space into a higher dimensional feature space using the kernel trick for better classification in the case of a non-linear decision boundary [33, Chapter 6].

By calling on the assumption of homoscedasticity, i.e. $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \mathbf{\Sigma}_x$, the original quadratic discriminant given by (2.9) for classifying a given vector $\mathbf{x}$ decomposes into the following linear decision

rule:

$$\mathbf{x}^\top \boldsymbol{\Sigma}_x^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \underset{\mathcal{C}_2}{\overset{\mathcal{C}_1}{\gtrless}} \ln \tau + \frac{1}{2}(\bar{\mathbf{x}}_1^\top \boldsymbol{\Sigma}_x^{-1}\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2^\top \boldsymbol{\Sigma}_x^{-1}\bar{\mathbf{x}}_2) \tag{2.12}$$

which can be expressed as:

$$\mathbf{w}^\top \mathbf{x} \underset{\mathcal{C}_2}{\overset{\mathcal{C}_1}{\gtrless}} w_0 \tag{2.13}$$

where

$$\mathbf{w} = \boldsymbol{\Sigma}_x^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2), \quad w_0 = \ln \tau + \frac{1}{2}(\bar{\mathbf{x}}_1^\top \boldsymbol{\Sigma}_x^{-1}\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2^\top \boldsymbol{\Sigma}_x^{-1}\bar{\mathbf{x}}_2) \tag{2.14}$$

and $\boldsymbol{\Sigma}_x$ is the pooled within-class covariance matrix [34] given by:

$$\boldsymbol{\Sigma}_x = \pi_1 \bar{\boldsymbol{\Sigma}}_1 + \pi_2 \bar{\boldsymbol{\Sigma}}_2, \tag{2.15}$$

where $\bar{\boldsymbol{\Sigma}}_1$ and $\bar{\boldsymbol{\Sigma}}_1$ are the sample covariance estimates of classes $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively.

This linear classifier is also known as Fisher's Linear Discriminant (FLD) [35]. Hamsici *et al.* [36] and Izenman [31] show that FLD is the optimal Bayes classifier for binary classification if the normality and homoscedasticity assumptions hold.

LDA demands only the computation of the dot product between $\mathbf{w}$ and $\mathbf{x}$, which is a relatively computationally inexpensive operation. LDA has been used for statistical classification in several application areas. For example, Sharma and Paliwal [37], Coomans *et al.* [38], Sengur [39] and Polat *et al.* [40] have used LDA for medical diagnosis; Song *et al.* [41], Chen *et al.* [42], Liu *et al.* [43] and Yu and Yang [44] have employed LDA for face and object recognition; Jin *et al.* [45], Ayhan *et al.* [46] and Moosavian *et al.* [47] have used LDA for machine fault diagnosis; Khan *et al.* [48], Yan *et al.* [49] and Iosifidis *et al.* [50] have also employed LDA for human activity recognition. The widespread use of LDA in these areas is not because the datasets necessarily satisfy the normality and homoscedasticity assumptions, but mainly due to the fact that being a linear model, LDA is robust against noise [2]. Since the linear Support Vector Machine (SVM) [51] can be quite expensive to train, especially for large values of K or n, LDA is often relied upon [52].

Yet, practical implementation of LDA is not without problems. Of note is the small sample size (SSS) problem that LDA faces with high-dimensional data and much smaller training data [53, 54, 42, 55, 56]. When $d \gg n$, the scatter matrix $\boldsymbol{\Sigma}_x$ is not invertible, as it tends to be of reduced rank. Since the decision rule as given by (2.12) requires the computation of the inverse of $\boldsymbol{\Sigma}_x$, the singularity of Σ_x makes the solution infeasible. In works by, for example, Liu *et al.* [43] and Paliwal and Sharma [57], this problem is overcome by taking the Moore-Penrose pseudo-inverse of the scatter matrix given by $\boldsymbol{\Sigma}_x^\dagger = (\boldsymbol{\Sigma}_x^\top \boldsymbol{\Sigma}_x)^{-1}\boldsymbol{\Sigma}_x^\top$,

rather than the ordinary matrix inverse. Another approach to solving the SSS problem, employed by Friedman [58], Lu *et al.* [54] and Wu *et al.* [59], involves adding a scalar multiple of the identity matrix to the scatter matrix to make the resulting matrix non-singular, a method known as regularised discriminant analysis.

However, for a given dataset that does not satisfy the homoscedasticity or normality assumption, one would expect that modifications to the original LDA procedure accounting for these violations would yield an improved performance. One such modification, in the case of a non-normal distribution, is explored by Hastie and Tibshirani [60], McLachlan [61] and Ju *et al.* [62] in a procedure known as mixture discriminant analysis, in which a non-normal distribution is modelled as a mixture of Gaussians. However, the parameters of the mixture components or even the number of mixture components, are usually not known *a priori*. Cai *et al.* [63], Fukunaga [64] and Li *et al.* [65] propose other non-parametric approaches to LDA that remove the normality assumption through the use of a similarity matrix based on local neighbourhood structures in the data instead of the scatter matrix Σ_x used in LDA.

2.1.1 Kernel Fisher's discriminant analysis

Another modification, in the case of a non-linear decision boundary between $\mathcal{D}_1$ and $\mathcal{D}_2$, is the Kernel Fisher Discriminant (KFD), proposed by Mika *et al.* [2], and explored by Zhao *et al.* [66] and Polat *et al.* [40]. KFD maps the original feature space $\mathcal{X}$ into some other space $\mathcal{Y}$ (usually higher dimensional) via some transformation $\phi(\mathbf{x})$, i.e.,

$$\phi(\mathbf{x}) : \mathcal{X} \to \mathcal{Y} \tag{2.16}$$

where $\mathcal{X} = \mathbb{R}^d$, with d being the size of $\mathbf{x}$. The transformation $\phi(\mathbf{x})$ is such that the transformed space $\mathcal{Y}$ guarantees linear or near-linear separability between the classes.

Once the mapping is done, the decision rule of (2.13) for deciding on the class of a given vector $\mathbf{x}$ then becomes:

$$\mathbf{w}^\top \phi(\mathbf{x}) \underset{\mathcal{C}_2}{\overset{\mathcal{C}_1}{\gtrless}} w_0 \tag{2.17}$$

so that the aim is to find the vector of weights $\mathbf{w}$ and the threshold w_0.

Without any loss of generality, the vector $\mathbf{w}$ can be expressed as:

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i \phi(\mathbf{x}_i) \tag{2.18}$$

where n is the number of points in the training dataset $\mathcal{X}$, and α_i for $i \in \{1, ..., n\}$ are yet to be

determined. Then, substituting (2.18) into (2.17) results in:

$$\left(\sum_{i=1}^{n}\alpha_i\phi(\mathbf{x}_i)\right)^{\top}\phi(\mathbf{x})\underset{\mathcal{C}_2}{\overset{\mathcal{C}_1}{\gtrless}}w_0, \tag{2.19}$$

i.e.,

$$\sum_{i=1}^{n}\alpha_i\phi(\mathbf{x}_i)^{\top}\phi(\mathbf{x})\underset{\mathcal{C}_2}{\overset{\mathcal{C}_1}{\gtrless}}w_0, \tag{2.20}$$

which can be expressed as:

$$\sum_{i=1}^{n}\alpha_i\mathcal{K}(\mathbf{x}_i,\mathbf{x})\underset{\mathcal{C}_2}{\overset{\mathcal{C}_1}{\gtrless}}w_0 \tag{2.21}$$

where $\mathcal{K}$ is a kernel, which is an inner product between any two vectors in the space $\mathcal{Y}$, satisfying Mercer's conditions [67]. It will be noted that (2.21) avoids the explicit transformation $\phi(\mathbf{x})$, and hence avoids the computational difficulty involved in transforming the data to a higher dimensional feature space. This is known as the kernel trick [2, 68, 69]. Though there is no readily obvious choice of a kernel function for a given dataset, popular kernels used in practice include the polynomial and Gaussian kernels, the latter allowing transformation to an infinite dimensional space [67].

Note that (2.21) can be rewritten as:

$$\boldsymbol{\alpha}^{\top}\mathbf{k}\underset{\mathcal{C}_2}{\overset{\mathcal{C}_1}{\gtrless}}w_0 \tag{2.22}$$

where $\boldsymbol{\alpha}=[\alpha_1,...,\alpha_n]^{\top}$ and $\mathbf{k}=[\mathcal{K}(\mathbf{x}_1,\mathbf{x}),...,\mathcal{K}(\mathbf{x}_n,\mathbf{x})]^{\top}$. Then, by noting the similarity with (2.13), which has its LDA solution given by (2.14), KFD has the following solution for $\boldsymbol{\alpha}$ [2]:

$$\boldsymbol{\alpha}=\mathbf{N}^{-1}(\bar{\mathbf{M}}_1-\bar{\mathbf{M}}_2) \tag{2.23}$$

where

$$(\bar{\mathbf{M}}_k)_i=\frac{1}{n_k}\sum_{j=1}^{n_k}\mathcal{K}(\mathbf{x}_i,\mathbf{x}_j),\quad\text{for}\quad i=1,...,n\quad\text{and}\quad k=1,2 \tag{2.24}$$

and

$$\mathbf{N}=\mathbf{N}_1+\mathbf{N}_2. \tag{2.25}$$

Here,

$$\mathbf{N}_k=\mathbf{K}_k(\mathbf{I}-\mathbf{1}_{n_k})\mathbf{K}_k^{\top},\quad k=1,2 \tag{2.26}$$

with $\mathbf{K}_k(i,j)=\mathcal{K}(\mathbf{x}_i,\mathbf{x}_j)$ for all $\mathbf{x}_i\in\mathcal{X}$ and all $\mathbf{x}_j\in\mathcal{D}_k$, $\mathbf{1}_{n_k}$ being an n_k-sized square matrix with all entries being $1/n_k$, and $\mathbf{I}$ the n_k-sized identity matrix.

The model for the Kernel Fisher's discriminant is provided by (2.22); most other linear models in the form of (2.13) can similarly be kernelised. Mika *et al.* [2] show that KFD performs well on datasets with non-linear decision boundaries, and is even comparable to the non-linear SVM. However, since KFD requires the inversion of the matrix $\mathbf{N}$ which is of size n, the algorithm is ill-suited for large datasets (as they thus have large n), since inverting $\mathbf{N}$ may be computationally intractable.

2.1.2 Heteroscedastic LDA

Accounting for the differences in covariance matrices in LDA has led to several heteroscedastic extensions of LDA, the most natural extension being Quadratic Discriminant Analysis (QDA), which makes use of the quadratic discriminant given in (2.9) for classification. However, for the reasons of robustness, shorter training and testing times, as well as the fact that linear classifiers can be kernelised, a linear approximation to the quadratic boundary in QDA is often preferred. An example of the quadratic boundary for heteroscedastic data and the possible linear approximations are indicated in Figure 2.1.

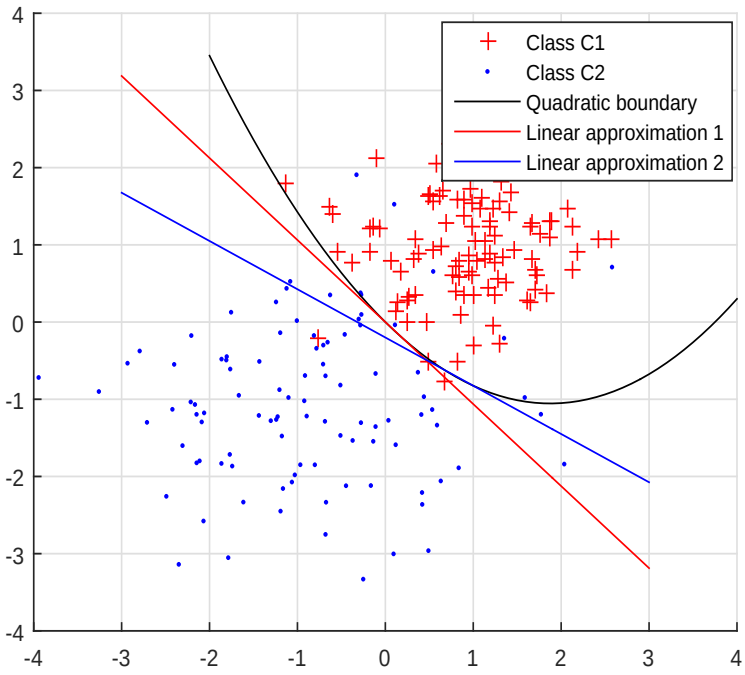

Figure 2.1: Quadratic boundary in heteroscedastic LDA and linear approximations

As there is a multitude of such linear approximations that can be made, the works by Marks and Dunn [70], Anderson and Bahadur [71], Peterson and Mattson [72] and Fukunaga [8] describe several

heteroscedastic LDA procedures aimed at minimising the probability of misclassification p_e as given by:

$$p_e = \pi_1 p(y < w_0|\mathcal{C}_1) + \pi_2 p(y \geq w_0|\mathcal{C}_2) \tag{2.27}$$

where $y = \mathbf{w}^\top \mathbf{x}$. In the heteroscedastic case, the Bayes optimal weight vector and threshold required for the linear decision rule in (2.13) are no longer as given in (2.14), as they do not minimise the Bayes error. Unfortunately, there is no closed-form solution to the minimisation of (2.27) [71] under heteroscedasticity, even though the optimal solution of $\mathbf{w}$ is known to be of the form:

$$\mathbf{w} = \left[s_1\boldsymbol{\Sigma}_1 + s_2\boldsymbol{\Sigma}_2\right]^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{2.28}$$

where s_1 and s_2 are unknown parameters [8, 70].

Marks and Dunn [70] describe an iterative procedure that involves solving for the optimal $\mathbf{w}$ as given by (2.28) and w_0 as given by:

$$w_0 = \mu_1 - s_1\sigma_1^2 = \mu_2 + s_2\sigma_2^2, \tag{2.29}$$

where

$$\mu_1 = \mathbf{w}^\top \bar{\mathbf{x}}_1 \quad \mu_2 = \mathbf{w}^\top \bar{\mathbf{x}}_2 \quad \sigma_1^2 = \mathbf{w}^\top \boldsymbol{\Sigma}_1 \mathbf{w} \quad \sigma_2^2 = \mathbf{w}^\top \boldsymbol{\Sigma}_2 \mathbf{w}. \tag{2.30}$$

Here, one obtains the optimal values of s_1 and s_2 via systematic trial and error. This heteroscedastic LDA procedure is denoted by R-HLD-2, for the reason that two parameters s_1 and s_2 are chosen at random.

Anderson and Bahadur [71] make the observation that if the weight vector $\mathbf{w}$ and the threshold w_0 are both multiplied by the same positive scalar, the decision boundary remains unchanged. Therefore, by multiplying (2.28) and (2.29) through by the scalar $s_1 + s_2$, $\mathbf{w}$ and w_0 can be put in the form of:

$$\begin{aligned} \mathbf{w} &= \left[s\boldsymbol{\Sigma}_2 + (1-s)\boldsymbol{\Sigma}_1\right]^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \\ w_0 &= \mu_1 - (1-s)\sigma_1^2 = \mu_2 + s\sigma_2^2 \end{aligned} \tag{2.31}$$

Still, the optimal value of s has to be chosen by systematic trial and error. This heteroscedastic LDA approach is denoted by R-HLD-1, for the reason that only one parameter s is chosen at random. As is shown Chapter 4, s is unbounded. Therefore, the difficulty faced by this approach is that s has to be chosen from the interval $(-\infty, \infty)$, so that the probability of finding the optimal s for a given dataset

is low, without extensive trial and error to limit the choice of s to some finite interval $[a, b]$. Thus, the approaches taken by Marks and Dunn [70] and Anderson and Bahadur [71] present no principled computational procedure for optimum parameter selection.

To avoid the unguided trial and error procedure by Marks and Dunn [70] and Anderson and Bahadur [71], Peterson and Mattson [72] and Fukunaga [8] propose a theoretical approach described below:

1. Change s from 0 to 1 with small step increments Δs.

2. Evaluate $\mathbf{w}$ as given by:
$$\mathbf{w} = \left[s\boldsymbol{\Sigma}_1 + (1-s)\boldsymbol{\Sigma}_2\right]^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{2.32}$$

3. Evaluate w_0 as given by:
$$w_0 = \frac{s\mu_2\sigma_1^2 + (1-s)\mu_1\sigma_2^2}{s\sigma_1^2 + (1-s)\sigma_2^2} \tag{2.33}$$

4. Compute the probability of misclassification p_e.

5. Choose $\mathbf{w}$ and w_0 that minimise p_e.

This procedure is referred to as C-HLD, for the reason that the optimal s is constrained in the interval $[0, 1]$.

However, two main problems with the above C-HLD procedure are highlighted:

1. There is no obvious choice of the step rate Δs. Too small a value of Δs will demand too many matrix inversions in Step 2, as there will be too many s values, thus increasing the computational complexity especially for very-high dimensional datasets. On the other hand, if Δs is too large, the optimal s may not be refined enough, and the vector $\mathbf{w}$ obtained may not be optimal. Specifically, the change in $\mathbf{w}$ that results from a small change in s is given as:
$$\mathrm{d}\mathbf{w} = \big(s\boldsymbol{\Sigma}_2 + (1-s)\boldsymbol{\Sigma}_1\big)^{-1}(\boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_2)\big(s\boldsymbol{\Sigma}_2 + (1-s)\boldsymbol{\Sigma}_\mathbf{1}\big)^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)\mathrm{d}s \tag{2.34}$$
Such a change in $\mathbf{w}$ may significantly affect the classification performance of the linear model especially if the two classes are not well-separated.

2. The solution obtained this way is only locally optimal as s is constrained in the interval $[0, 1]$. When there is a class imbalance [12, 73, 74, 75], the optimal s may be found outside the interval $[0, 1]$ so that the vector $\mathbf{w}$ found by this approach leads to poor classification accuracy.

A recurring procedure among all three heteroscedastic LDA approaches is the matrix inversion as given in (2.28) for R-HLD-2, (2.31) for R-HLD-1 and (2.32) for C-HLD. However, if these heteroscedastic LDA procedures were to be kernalised in the manner shown in Section 2.1.1, by replacing $\mathbf{x}_1, \mathbf{x}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2$ with $\bar{\mathbf{M}}_1, \bar{\mathbf{M}}_1, \mathbf{N}_1, \mathbf{N}_2$ respectively, there would be too many inversions of size-n matrices, which can be computationally infeasible for large datasets.

2.1.3 Class imbalance

The class imbalance problem arises when the number of objects in one class far exceeds the cardinality of the other classes. Such datasets are often found in anomaly detection applications like falls detection in remote health monitoring [76, 77, 78], customer churn prediction in telecommunication systems [79, 80, 81], or machine health monitoring [82, 83]. In these applications, a "fault" state is not as probable as the "normal" state of the system.

As the prior probabilities are often estimated from the cardinality of each class, data imbalance in binary classification leads to the case where $\pi_1 \gg \pi_2$ or $\pi_1 \ll \pi_2$. If the means are not well separated, class imbalance has the effect of shifting the decision threshold w_0 well beyond the closed interval $[\mu_1, \mu_2]$, where μ_1 and μ_2 are the projected class means for classes $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively.

To put things in perspective, consider the linear discrimination rule in LDA as given by (2.13) and (2.14). In the event of class imbalance, $\tau = \pi_2/\pi_1$ is affected. If $\pi_1 \gg \pi_2$, then τ approaches 0, and the decision threshold w_0 approaches $-\infty$. In such a case, the decision rule tends to favour class $\mathcal{C}_1$. On the other hand, if $\pi_1 \ll \pi_2$, then τ approaches ∞, and the decision threshold w_0 approaches ∞ as well, in which case the decision rule tends to favour class $\mathcal{C}_2$. This then tends to skew the classification accuracy in favour of the majority class. This can be problematic in several applications, for instance, in medical diagnosis. In many Sub-Saharan African countries, for example, since the number of Malaria cases usually far outweighs the number of cases for many other diseases, patients tend to be wrongly diagnosed with Malaria.

In binary classification, the classification accuracy is obtained by evaluating the classifier on a test dataset, and is given by:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2.35}$$

where TP, the true positives, is the number of positive samples that are correctly classified; TN, the true negatives, is the number of negative samples that are correctly classified; FP, the false positives, is the number of negative samples that are wrongly classified as positive; and FN, the false negatives, is the

number of positive samples that are wrongly classified as negative. Let the discriminating hyperplane be given by:

$$\mathbf{w}^\top \mathbf{x} - w_0 = 0. \tag{2.36}$$

To avoid any ambiguity, this thesis considers all samples $\mathbf{x}$ such that $\mathbf{w}^\top \mathbf{x} - w_0 \geq 0$ as positive samples, and negative examples as all samples $\mathbf{x}$ such that $\mathbf{w}^\top \mathbf{x} - w_0 < 0$.

Due to the bias of the classification accuracy as an evaluation metric, other evaluation metrics such as Precision and Recall [84, 85] and the F-Measure [86, 87] are favoured over the classification accuracy [88, 89, 90] in the presence of class imbalance. One other such metric is the area under the Receiver Operating Characteristics (ROC) curve, often referred to simply as area under curve (AUC), [91, 92], which is obtained by plotting the false positive rate (FPR) against the true positive rate (TPR), as the decision threshold w_0 is varied. An example ROC plot is shown in Fig. 2.2.

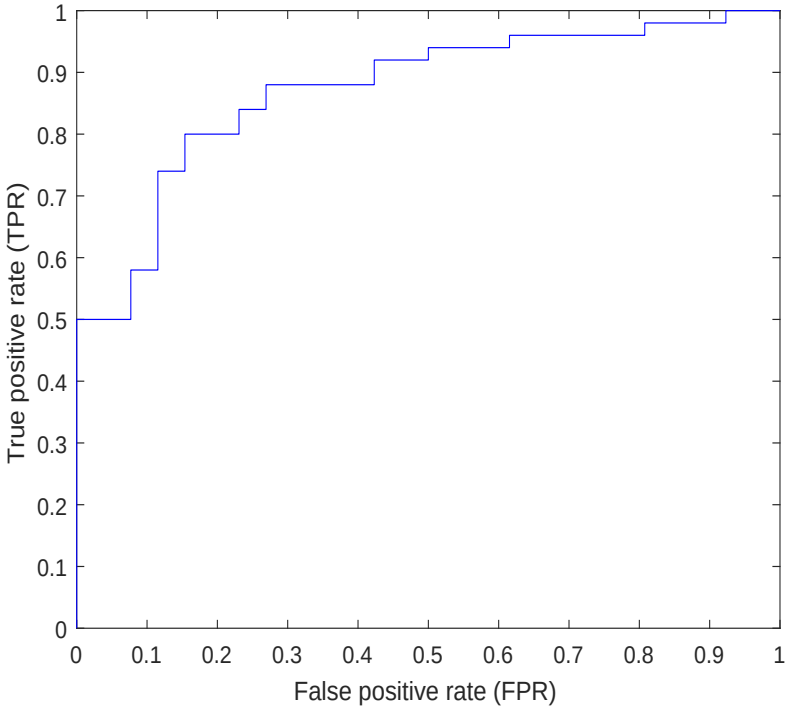

Figure 2.2: Receiver Operating Characteristics

TPR and FPR are given by:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{2.37}$$

Xue and Titterington [12] and Xie and Zhengding [11] have studied class imbalance in the context of

LDA. Xie [11] has claimed that class imbalance has a negative effect on LDA in terms of the AUC when the covariance matrices of the classes are unequal, based on experimental results. This claim has since been disputed by Xue [12] because, in the event of class imbalance, the only parameter that is affected in the LDA discrimination rule in (2.12) is $\tau = \pi_2/\pi_1$, which only changes the decision threshold w_0; the pooled within-class covariance matrix is still as given by (2.15). However, the AUC is not affected by the decision threshold, as w_0 is not kept constant, but is varied to obtain different false positive and true positive rates.

On the other hand, if one accounted for heteroscedasticity in LDA, as in the C-HLD, R-HLD-1 and R-HLD-2 procedures, the Bayes optimal classifier is such that the weight vector $\mathbf{w}$ is no longer as given in (2.14), but given by (2.28), as has already been mentioned. Moreover, it can be shown that the unknown parameters s_1 and s_2 in (2.28) do themselves depend on the threshold w_0 (see Section 3.1). The implication of this dependence of s_1 and s_2 on w_0 is that varying w_0 to obtain different values of FPR and TPR for the ROC necessarily causes $\mathbf{w}$ to vary. Therefore, if $\mathbf{w}$ is kept unchanged while varying w_0 to obtain the ROC, the performance of heteroscedastic LDA is negatively affected in terms of the AUC. Since all three heteroscedastic LDA approaches described above (R-HLD-1, R-HLD-2 and C-HLD) leave no room to express s_1 and s_2 in terms of the threshold w_0, $\mathbf{w}$ is kept constant while varying w_0 to obtain the ROC, and hence their AUC performance are suboptimal.

Even so, the suboptimal performance of the existing heteroscedastic LDA approaches is not particular to class imbalance, except in the case of C-HLD. In (2.32) of the C-HLD procedure, s may be thought of as s_1 and $1 - s$ as s_2, thus satisfying the general solution of (2.28). However, s is constrained to vary in the closed interval $[0, 1]$. While this may be valid under nearly equal priors, i.e., $\pi_1 \approx \pi_2$, in the presence of class imbalance, s tends to fall outside the interval $[0, 1]$ (see section 3.4), so that the weight vector $\mathbf{w}$ obtained in the C-HLD procedure is suboptimal. Moreover, if $s \in [0, 1]$, then the threshold w_0 as given by (2.35) of the C-HLD procedure is a convex combination of the two projected means μ_1 and μ_2, such that w_0 is bounded in the interval $[\mu_2, \mu_1]$. This tends to negatively impact on the classification accuracy of C-HLD since the optimal threshold w_0 should approach $+\infty$ or $-\infty$ in the limit of τ as can be seen from (2.14).

2.1.4 Multiclass classification

Suppose now that there are $K > 2$ classes in the dataset $\mathcal{D}$, then the multiclass classification problem may be reduced to a number of binary classification problems. The two main approaches usually taken for this reduction are the One-vs-All (OvA) and One-vs-One (OvO) strategies [33, 93].

One-vs-All (OvA)

In OvA, one trains a classifier to discriminate between one class and the remaining $K-1$ classes. Thus, there are K different classifiers that can be constructed. Since the classifiers are linear, this method creates decision regions that are partitioned by hyperplanes. For the case $K = 3$, class $\mathcal{C}_1$ is trained against $\mathcal{C}_2$ and $\mathcal{C}_3$, class $\mathcal{C}_2$ is trained against $\mathcal{C}_1$ and $\mathcal{C}_3$, and class $\mathcal{C}_3$ is trained against $\mathcal{C}_1$ and $\mathcal{C}_2$. This is shown in Figure 2.3. The decision region for each of the three classes is also shown in the figure. It will be noted that there are ambiguous regions outside the three decision regions. An unknown test vector falling in these ambiguous regions could be wrongly classified. However, OvA is able to get around this problem in such a way that an unknown vector $\mathbf{x}$ is tested on all K classifiers so that the class corresponding to the classifier with the highest discriminant score is chosen. The discriminant score is given by $\mathbf{w}^\top \mathbf{x} - w_0$. This method of testing removes the ambiguity.

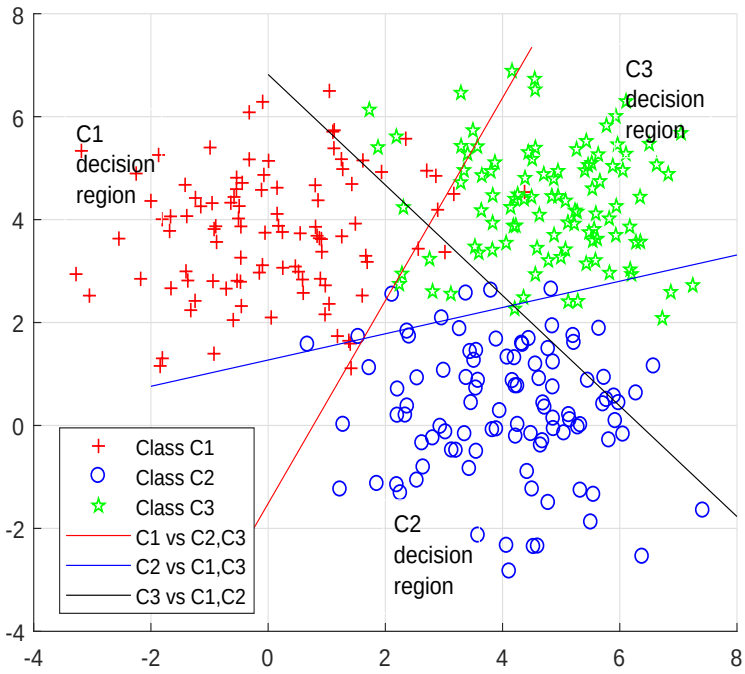

Figure 2.3: Ambiguous region in one vs all multiclass classification

Nevertheless, with respect to LDA, this may be an ill-suited approach. This is because the collection of all other classes on one side of the discriminating hyperplane will not necessarily have a normal distribution as is required in LDA, and it could in fact be multimodal, if the means are well-separated. If the normality assumption holds true for each class, then the distribution of $K-1$ classes on one side

of the discriminant is actually a mixture of $K-1$ Gaussians. Thus, LDA is expected to perform poorly in OvA multiclass classification.

One-vs-One (OvO)

In OvO, a classifier is trained to discriminate between every pair of classes in the dataset, ignoring the other $K-2$ classes. Thus, there are $K(K-1)/2$ unique classifiers that may be constructed. For the case $K=3$, class $\mathcal{C}_1$ is trained against $\mathcal{C}_2$, class $\mathcal{C}_1$ is trained against $\mathcal{C}_3$, and class $\mathcal{C}_2$ is trained against $\mathcal{C}_3$. This is indicated in Figure 2.4. Here too, there is an ambiguous region in the middle, outside the three decision regions, where the three classifiers intersect. Even though the area of this ambiguous region is

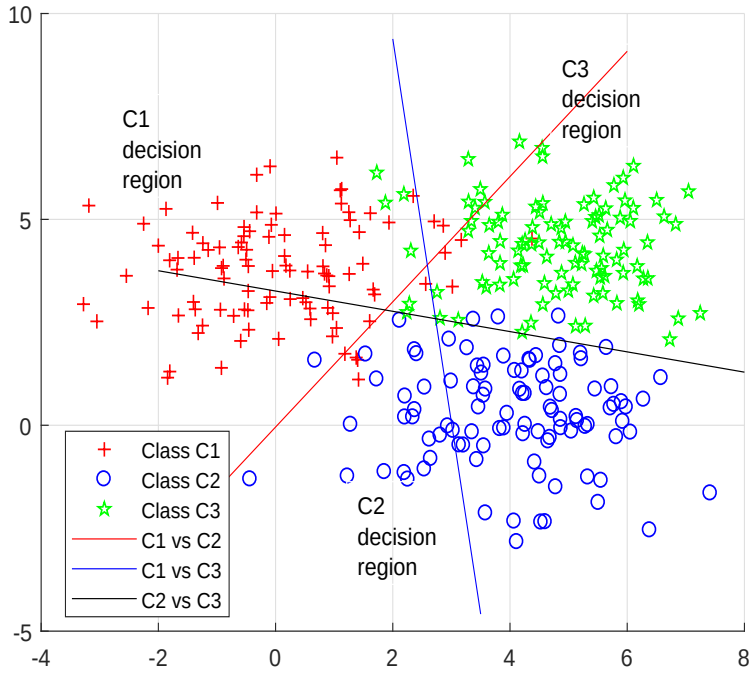

Figure 2.4: Ambiguous region in one vs one multiclass classification

often much smaller than in the OvA approach, the particular method of testing an unknown vector in this multiclass strategy does not remove the ambiguity, unlike in the OvA scenario. Here, an unknown vector $\mathbf{x}$ is tested on all $K(K-1)/2$ classifiers. The predicted classes for all the classifiers are then tallied so that the class that occurs most frequently is chosen. This is equivalent to a majority vote decision. In a lot of cases, however, there is no clear-cut winner, as more than one class may have the highest number of votes. In such a case, the most likely class is often chosen randomly between those most frequently occurring classes.

However, unlike in the OvA strategy, the data on each side of each of the $K(K-1)/2$ classifiers remains normally distributed, if the normality assumption holds true in each class. Thus, LDA is expected to perform well on each of the classifiers in the OvO multiclass strategy.

2.2 Linear dimensionality reduction (LDR)

In applications such as face recognition where the images live in a high dimensional space (often d is equal to the number of pixels in the facial image [3]), the large dimensionality d increases model complexity. For example, an image with $2,000,000$ pixels would require the inversion of a size-$2,000,000$ scatter matrix $\mathbf{\Sigma}_x$ in order to apply LDA. This makes LDR a critical preprocessing step in order to reduce model complexity. As an added advantage, classification accuracy is often improved after LDR [4, 5].

Consider a dataset $\mathcal{X} = [\mathbf{x}_1, ..., \mathbf{x}_n]$ with n examples and a dimensionality of d, i.e. $\mathbf{x}_i \in \mathbb{R}^d$, $\forall i \in \{1, ..., n\}$. The aim of LDR is to find a linear transformation $\mathbf{M}$ such that $\tilde{\mathcal{X}} = \mathbf{M}^\top \mathcal{X}$ has a dimensionality of q, i.e., $\mathbf{M} \in \mathbb{R}^{d \times q}$, where $\mathbf{M}$ is of rank q and $q < d$.

Cunningham and Ghahramani [94] give the general optimisation framework for this task as:

$$\begin{aligned} &\min f_{\mathcal{X}}(\mathbf{M}) \\ &\text{subject to} \quad \mathbf{M} \in \mathcal{M} \end{aligned} \tag{2.38}$$

where $\mathcal{M}$ is the matrix manifold over which the optimisation is done. For example, $\mathcal{M}$ could be the set of all rank q matrices in $\mathbb{R}^{d \times q}$ or the set of all orthogonal matrices $\mathcal{O}^{d \times q} \in \mathbb{R}^{d \times q}$. The objective function $f_{\mathcal{X}}(\mathbf{M})$ is chosen to yield a good representation of the original dataset $\mathcal{X}$ in the low-dimensional manifold. There is no obvious choice of this objective function, and the differences in this choice bring about the different LDR procedures.

2.2.1 Principal component analysis

One of the most well-known LDR procedures is Principal Component Analysis (PCA) [95]. Mika *et al.* [96], Scholkopf *et al.* [97] and Hoffman [98] have also extended PCA for non-linear dimensionality reduction via the kernel trick (section 2.1.1) in a procedure referred to as kernel PCA (KPCA).

PCA aims to maximise the variance of the projected data in each dimension, while minimising the covariance between any two variables. Suppose that the dataset $\mathcal{X}$ has a mean of $\bar{\mathbf{x}}$ and a covariance of

$\mathbf{\Sigma}_{\mathcal{X}}$ given by the following sample estimates:

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i \quad \text{and} \quad \mathbf{\Sigma}_{\mathcal{X}} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top. \tag{2.39}$$

Then, the projected data $\tilde{\mathcal{X}}$ has a sample covariance of $\mathbf{\Sigma}_{\tilde{\mathcal{X}}}$ given by:

$$\mathbf{\Sigma}_{\tilde{\mathcal{X}}} = \frac{1}{n-1}\sum_{i=1}^{n}\mathbf{M}^\top(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top\mathbf{M} = \mathbf{M}^\top\mathbf{\Sigma}_{\mathcal{X}}\mathbf{M} \tag{2.40}$$

Since $\mathbf{\Sigma}_{\mathcal{X}}$ is square and real symmetric, it is orthogonally diagonalisable [99]. Thus, $\mathbf{\Sigma}_{\mathcal{X}}$ can be expressed as:

$$\mathbf{\Sigma}_{\mathcal{X}} = \mathbf{V}\mathbf{D}\mathbf{V}^\top \tag{2.41}$$

via an eigenvalue decomposition, where $\mathbf{V}$ is an orthonormal matrix whose columns are the eigenvectors of $\mathbf{\Sigma}_{\mathcal{X}}$ and $\mathbf{D}$ is a diagonal matrix whose diagonal entries are the eigenvalues corresponding to the eigenvectors in $\mathbf{V}$. Therefore, the covariance $\mathbf{\Sigma}_{\tilde{\mathcal{X}}}$ of the projected data is given by:

$$\mathbf{\Sigma}_{\tilde{\mathcal{X}}} = \mathbf{M}^\top\mathbf{\Sigma}_{\mathcal{X}}\mathbf{M} = \mathbf{M}^\top\mathbf{V}\mathbf{D}\mathbf{V}^\top\mathbf{M} \tag{2.42}$$

Notice that by setting $\mathbf{M}$ as $\mathbf{M} = \mathbf{V}$, the covariance of $\mathbf{\Sigma}_{\tilde{\mathcal{X}}}$ becomes:

$$\mathbf{\Sigma}_{\tilde{\mathcal{X}}} = \mathbf{V}^\top\mathbf{V}\mathbf{D}\mathbf{V}^\top\mathbf{V} = \mathbf{D} \tag{2.43}$$

(since $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$), which maximises the variance (the diagonal entries) and minimises the covariance (the off-diagonal entries).

Still, $\mathbf{M}$ is not necessarily of rank q, since its rank is equal to the number of eigenvectors in $\mathbf{V}$. Thus, if $\mathbf{\Sigma}_{\mathcal{X}}$ has a full rank, then $\mathbf{M}$ also has a rank of d, and no dimensionality reduction is achieved from the linear transformation $\tilde{\mathcal{X}} = \mathbf{M}^\top\mathcal{X}$. In order to achieve dimensionality reduction, the transformation matrix $\mathbf{M}$ is assigned instead to the q eigenvectors corresponding to the q largest eigenvalues in $\mathbf{D}$. In this manner, the columns of the matrix $\mathbf{M}$ represents the q directions along which the variability of the original data is most pronounced; the variability of the data in the remaining $d-q$ directions are assumed to be insignificant or due to noise in the data.

However, when statistical classification is desired after dimensionality reduction, PCA may lose the class-discriminatory information in the data, as the directions of maximum variance do not always coincide with the most class-discriminative directions. In order to maximise the class-discriminatory information

while linearly reducing the dimensionality, LDR that is aimed for classification involves the use of class labels to inform the choice of the transformation matrix $\mathbf{M}$. In this case, the optimum objective function to minimise is the Bayes error in the linearly reduced space [8, 100].

Nevertheless, an analytic expression for the Bayes error is hard to obtain for the general K-class problem, except for special cases such as $K = 2$ normally distributed classes with equal covariance matrices. As such, several approximations have been made, for example, by Fukunaga [8], Duda *et al.* [101] Buturovic [100], leading to several supervised dimension reduction techniques [94, 5, 4, 102, 103, 104]. These techniques have been classified into two by Fukunaga [8] and Buturovic [100]. The first involves the use of a family of functions of scatter matrices that attempt to maximise the separability of the K classes, but are not directly related to the Bayes error. The second approach involves the minimisation of some upper bounds on the Bayes error.

2.2.2 Fisher's linear discriminant (FLD)

One popular method for such a supervised LDR is Fisher's linear discriminant (FLD) [35], which follows the approach of maximising the class separability of the K classes using functions of the scatter matrix. Mika *et al.* [2] Aitchison and Aitken [68] and Jaakkola *et al.* [69] have also extended FLD for non-linear dimensionality reduction via the kernel trick (section 2.1.1).

Suppose that the dataset is labelled such that it can be divided into K classes thus: $\mathcal{X} = [\mathcal{D}_1, ..., \mathcal{D}_K]$, where $\mathcal{D}_k$ are the training samples belonging to the kth class ($k \in \{1, 2, ..., K\}$). Let $\bar{\mathbf{x}}_k$, $\mathbf{S}_k$ and $\pi_k = p(\mathcal{C}_k)$ be the sample mean, sample covariance and empirical prior probability of the kth class respectively, for $k \in \{1, ..., K\}$, as given by:

$$\bar{\mathbf{x}}_k = \frac{1}{n_k} \sum_{\mathbf{x} \in \mathcal{D}_k} \mathbf{x}, \quad \mathbf{\Sigma}_k = \frac{1}{n_k - 1} \sum_{\mathbf{x} \in \mathcal{D}_k} (\mathbf{x} - \bar{\mathbf{x}}_k)(\mathbf{x} - \bar{\mathbf{x}}_k)^\top \quad \text{and} \quad \pi_k = \frac{n_k}{n} \tag{2.44}$$

where n_k is the number of samples in the kth class. Also, let $\bar{\mathbf{x}}$ be the mean of the overall dataset $\mathcal{X}$, i.e.,

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{\mathbf{x} \in \mathcal{X}} \mathbf{x} = \sum_{k=1}^{K} \pi_k \mathbf{x}_k \tag{2.45}$$

The particular measure of class separability used in the FLD is Fisher's criterion J_f, which is the ratio of the between-class scatter to the within-class scatter in the linearly reduced space, and is given by:

$$J_f = \text{trace}[(\mathbf{M}^\top \mathbf{S}_w \mathbf{M})^{-1} (\mathbf{M}^\top \mathbf{S}_b \mathbf{M})] \tag{2.46}$$

where $\mathbf{S}_w$, the within-class scatter matrix and $\mathbf{S}_b$, the between-class scatter matrix are both given as:

$$\mathbf{S}_w = \sum_{k=1}^{K} \pi_k \mathbf{\Sigma}_k \quad \text{and} \quad \mathbf{S}_b = \sum_{k=1}^{K} \pi_k (\bar{\mathbf{x}}_k - \bar{\mathbf{x}})(\bar{\mathbf{x}}_k - \bar{\mathbf{x}})^\top. \tag{2.47}$$

Fisher's criterion can be considered as a form of the generalised Rayleigh quotient [102, 105, 106]. For $K > 2$, i.e., for reduction to more than one dimension, FLD is also referred to as canonical variates [31, 107, 108, 102].

In order to maximise J_f, (2.46) is differentiated and equated to zero. From the Matrix Cookbook [109], the derivative of the trace in the form of (2.46) is given as:

$$\frac{\partial J_f}{\partial \mathbf{M}} = -2\mathbf{S}_w\mathbf{M}(\mathbf{M}^\top\mathbf{S}_w\mathbf{M})^{-1}\mathbf{M}^\top\mathbf{S}_b\mathbf{M}(\mathbf{M}^\top\mathbf{S}_w\mathbf{M})^{-1} + 2\mathbf{S}_b\mathbf{M}(\mathbf{M}^\top\mathbf{S}_w\mathbf{M})^{-1} \tag{2.48}$$

By equating (2.48) to zero, the following is obtained:

$$\mathbf{S}_b\mathbf{M}(\mathbf{M}^\top\mathbf{S}_w\mathbf{M})^{-1} = \mathbf{S}_w\mathbf{M}(\mathbf{M}^\top\mathbf{S}_w\mathbf{M})^{-1}\mathbf{M}^\top\mathbf{S}_b\mathbf{M}(\mathbf{M}^\top\mathbf{S}_w\mathbf{M})^{-1} \tag{2.49}$$

Assuming that $\mathbf{S}_w$ is invertible, (2.49) can be rewritten as:

$$\mathbf{S}_w^{-1}\mathbf{S}_b\mathbf{M} = \mathbf{M}(\mathbf{M}^\top\mathbf{S}_w\mathbf{M})^{-1}\mathbf{M}^\top\mathbf{S}_b\mathbf{M} \tag{2.50}$$

Suppose that $\mathbf{S}_w^{-1}\mathbf{S}_b$ is not defective and can be orthogonally diagonalised via an eigenvalue decomposition of the form:

$$\mathbf{S}_w^{-1}\mathbf{S}_b = \mathbf{T}\mathbf{\Lambda}\mathbf{T}^\top, \tag{2.51}$$

i.e.

$$\mathbf{S}_w^{-1}\mathbf{S}_b\mathbf{T} = \mathbf{T}\mathbf{\Lambda}\mathbf{T}^\top\mathbf{T} \tag{2.52}$$

where $\mathbf{T}$ is an orthonormal matrix whose columns are the eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_b$ and $\mathbf{\Lambda}$ is a diagonal matrix whose diagonal entries are the eigenvalues corresponding to the eigenvectors in $\mathbf{T}$. Since $\mathbf{T}$ is orthonormal, $\mathbf{T}^\top\mathbf{T} = \mathbf{I}$, so that

$$\mathbf{S}_w^{-1}\mathbf{S}_b\mathbf{T} = \mathbf{T}\mathbf{\Lambda} \tag{2.53}$$

Comparing (2.53) to (2.50), it will be noted that $\mathbf{M}$ can be set equal to $\mathbf{T}$. It can be verified, using the relation $\mathbf{S}_b = \mathbf{S}_w\mathbf{T}\mathbf{\Lambda}\mathbf{T}^\top$ obtainable from (2.51), that with this choice of $\mathbf{M}$ the right-hand side of (2.50) becomes equal to $\mathbf{T}\mathbf{\Lambda}$ as in (2.53).

However, there are at most $K-1$ non-zero eigenvalues in $\mathbf{\Lambda}$ due to the fact that the rank of $\mathbf{S}_w^{-1}\mathbf{S}_b$ is at most $K-1$ (see proof in section 4.A.1), and therefore there are at most $K-1$ useful eigenvectors in $\mathbf{T}$ required to satisfy (2.53). This implies that the transformation matrix $\mathbf{M}$ has to be set equal to the eigenvectors corresponding to the first $K-1$ largest (or non-zero) eigenvalues, if the trace of (2.46) is to be maximised. For this reason, FLD projects the original d-dimensional data onto a $K-1$ dimensional subspace within which the separation between the classes is maximised in terms of Fisher's criterion.

It will be recalled that this analysis is made possible by assuming that $\mathbf{S}_w \in \mathbb{R}^{d\times d}$ is invertible. For this to be true, $\mathbf{S}_w$ has to have a rank of d. It can be shown, however, that the rank of $\mathbf{S}_w$ is at most $n-K$ (see proof in Section 4.A.1). Therefore, in the scenario a dataset has a very high dimensionality d and a rather small number of samples n, such that $d > n-K$, $\mathbf{S}_w$ becomes singular, and FLD fails. This is often referred to as the small sample size (SSS) problem [42, 56, 3], which is the same drawback that has already been mentioned in the context of statistical classification. Belhumeur *et al.* [3] describes a way to avoid the SSS problem, by applying PCA to reduce the dimensionality of the original dataset to $q = n-K$, and then perform FLD on that reduced dataset. Other approaches, for example, by Sharma and Paliwal [37], have avoided the computation of the inverse altogether, and instead have applied the gradient descent procedure [110] wherein an iterative algorithm starts from some initial choice of $\mathbf{M}$ and takes small steps in the negative direction of the gradient of J_f.

FLD ($K = 2$)

Now, for the two-class case, where reduction to only one dimension is possible, the transformation matrix $\mathbf{M}$ contains only one column vector $\mathbf{v}$, and Fisher's criterion as given by (2.46) reduces to:

$$J_f = \frac{\mathbf{v}^\top \mathbf{S}_b \mathbf{v}}{\mathbf{v}^\top \mathbf{S}_w \mathbf{v}} \tag{2.54}$$

where $\mathbf{S}_w$ is now given by:

$$\mathbf{S}_w = \pi_1 \mathbf{S}_1 + \pi_2 \mathbf{S}_2, \tag{2.55}$$

and $\mathbf{S}_b$ by:

$$\mathbf{S}_b = \pi_1(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}})(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}})^\top + \pi_2(\bar{\mathbf{x}}_2 - \bar{\mathbf{x}})(\bar{\mathbf{x}}_2 - \bar{\mathbf{x}})^\top \tag{2.56}$$

which can be simplified thus:

$$\mathbf{S}_b = \pi_1\pi_2(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^\top \quad . \tag{2.57}$$

when the following substitution is made in (2.56):

$$\bar{\mathbf{x}} = \pi_1 \bar{\mathbf{x}}_1 + \pi_2 \bar{\mathbf{x}}_2 \tag{2.58}$$

The optimal $\mathbf{v}$ can be obtained by differentiating J_f with respect to $\mathbf{v}$ as shown below

$$\frac{\partial J_f}{\partial \mathbf{v}} = \frac{\partial}{\partial \mathbf{v}} \frac{\mathbf{v}^\top \mathbf{S}_b \mathbf{v}}{\mathbf{v}^\top \mathbf{S}_w \mathbf{v}} = 2 \frac{(\mathbf{v}^\top \mathbf{S}_w \mathbf{v}) \mathbf{S}_b \mathbf{v} - 2(\mathbf{v}^\top \mathbf{S}_b \mathbf{v}) \mathbf{S}_w \mathbf{v}}{(\mathbf{v}^\top \mathbf{S}_w \mathbf{v})^2} \tag{2.59}$$

and setting the partial derivative to zero, resulting in the following:

$$(\mathbf{v}^\top \mathbf{S}_w \mathbf{v}) \mathbf{S}_b \mathbf{v} = (\mathbf{v}^\top \mathbf{S}_b \mathbf{v}) \mathbf{S}_w \mathbf{v} \tag{2.60}$$

Thus, (2.60) can be rewritten as:

$$\mathbf{S}_w \mathbf{v} = \frac{\mathbf{v}^\top \mathbf{S}_w \mathbf{v}}{\mathbf{v}^\top \mathbf{S}_b \mathbf{v}} \mathbf{S}_b \mathbf{v} \tag{2.61}$$

By substituting the expression for $\mathbf{S}_b$ into (2.61), the following equation is obtained:

$$\mathbf{S}_w \mathbf{v} = \frac{\mathbf{v}^\top \mathbf{S}_w \mathbf{v}}{\mathbf{v}^\top \mathbf{S}_b \mathbf{v}} \pi_1 \pi_2 (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^\top \mathbf{v} \tag{2.62}$$

Notice that the factors $(\mathbf{v}^\top \mathbf{S}_w \mathbf{v})/(\mathbf{v}^\top \mathbf{S}_b \mathbf{v})$, $\pi_1 \pi_2$ and $(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^\top \mathbf{v}$ are all scalars. Thus, (2.62) can be rewritten in the form:

$$\mathbf{v} = k \mathbf{S}_w^{-1} (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{2.63}$$

The constant k can be dropped off entirely as Fisher's criterion as given by (2.54) is not affected by any scale factor. Thus k can be set to 1, in which case it will be noted that the optimal $\mathbf{v}$ as given by (2.63) is the same as the weight vector $\mathbf{w}$ as given by (2.14) and (2.15) obtained for the Bayes optimal linear classifier in section 2.1. It is for this reason that the FLD is more commonly referred to simply as linear discriminant analysis (LDA).

Since $\mathbf{w}$ as given by (2.14) minimises the Bayes error for two classes when the normality and homoscedasticity assumptions are satisfied, maximising Fisher's criterion for LDR in the two-class case also minimises the Bayes error in the linearly reduced space for $K = 2$ normally distributed classes with equal covariance matrices. For more than two classes, however, maximisation of Fisher's criterion does not guarantee the minimisation of the Bayes error, even when the assumptions of homoscedasticity and normality are satisfied.

2.2.3 Mahalanobis distance criterion

As mentioned previously, one other approach to supervised LDR involves minimising certain bounds on the Bayes error. One of such bounds is based on the Mahalanobis distance [4, 111, 112, 113]. The Mahalanobis distance measures the separation between two populations of equal covariance. The square of the Mahalanobis distance is defined for two classes with common covariance $\mathbf{\Sigma}_x$ as:

$$\Delta_{12} = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)\mathbf{\Sigma}_x^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{2.64}$$

Using this, Brunzell and Eriksson [4] give an upper bound on the Bayes error as:

$$\epsilon_m \leq \frac{2\pi_1\pi_2}{1 + \pi_1\pi_2\Delta_{12}} \tag{2.65}$$

For the K-class problem, the Mahalanobis distance is generalised as:

$$J_m = \prod_{1\leq i<j\leq K} (\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j)^\top (\pi_i\mathbf{\Sigma}_i + \pi_j\mathbf{\Sigma}_j)^{-1}(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j) \tag{2.66}$$

Subsequently, the task of the Mahalanobis-based LDR is to find a linear transformation that preserves the separation given by J_m in the linearly reduced space. The optimal linear transformation $\mathbf{M}$ is found by first defining a matrix $\mathbf{U}$ as:

$$\begin{aligned}\mathbf{U} = \Big[&(\pi_1\mathbf{\Sigma}_1 + \pi_2\mathbf{\Sigma}_2)^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2), ..., \\ &(\pi_i\mathbf{\Sigma}_i + \pi_j\mathbf{\Sigma}_j)^{-1}(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j), ..., \\ &(\pi_{K-1}\mathbf{\Sigma}_{K-1} + \pi_K\mathbf{\Sigma}_K)^{-1}(\bar{\mathbf{x}}_{K-1} - \bar{\mathbf{x}}_K)\Big]\end{aligned} \tag{2.67}$$

for $1 \leq i < j \leq K$. Then, an eigenvalue decomposition of $\mathbf{U}$ is found, after which $\mathbf{M}$ is set to the q eigenvectors corresponding to the q largest eigenvalues of $\mathbf{U}$.

It will be observed (by verifying with several values of K) that $\mathbf{U}$ has $K(K-1)/2$ columns, and it can, in fact, equivalently be obtained by performing One-vs-One multi-class classification using LDA, and forming a matrix out of the $K(K-1)/2$ weight vectors $\mathbf{w}$ given by (2.14). However, just like with LDA, the Mahalanobis-based LDR assumes a common covariance matrix among the classes [5].

2.2.4 Chernoff criterion

To account for the differences in covariance matrices among the classes, Decell and Marani [114] propose a heteroscedastic extension of the Mahalanobis distance based on the Bhattacharya distance [101, 115] for LDR. The Bhattacharya distance, which is a measure of class separability, also gives an upper bound on the Bayes error for normally distributed data [8]. Following this, there has been the use of a wider class of Bregman divergences [116, 117], notably, the Kullback-Leibler (KL) divergence for heteroscedastic LDR by Decell and Mayekar [118], even though the KL divergence is known to have a weaker link to the Bayes error [8]. Yet, while the Bhattacharya distance provides a good enough bound on the Bayes error, Qin *et al.* [119] and Das and Nenadic [120] have shown that the Chernoff criterion J_c provides a slightly tighter bound than the Bhattacharya distance, which implies that the Chernoff criterion approximates the Bayes error better.

For the two-class case, the criterion is defined by Loog [121] as:

$$J_c = \text{trace}\big[(\mathbf{M}^\top \mathbf{S}_w \mathbf{M})^{-1}(\pi_1\pi_2 \mathbf{M}^\top(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^\top \mathbf{M} - \mathbf{M}^\top \mathbf{S}_w^{-\frac{1}{2}}(\pi_1 \log(\mathbf{S}_w^{-\frac{1}{2}}\boldsymbol{\Sigma}_1\mathbf{S}_w^{-\frac{1}{2}}) + \pi_2 \log(\mathbf{S}_w^{-\frac{1}{2}}\boldsymbol{\Sigma}_2\mathbf{S}_w^{-\frac{1}{2}}))\mathbf{S}_w^{\frac{1}{2}}\mathbf{M})\big] \tag{2.68}$$

For the multi-class setting, the criterion is generalised by [5] as:

$$J_c = \sum_{i=1}^{K-1}\sum_{j=i+1}^{K} \pi_i\pi_j \text{trace}\big[(\mathbf{M}^\top \mathbf{S}_w \mathbf{M})^{-1}\mathbf{M}^\top \mathbf{S}_w^{\frac{1}{2}}\big((\mathbf{S}_w^{-\frac{1}{2}}\mathbf{S}_{ij}\mathbf{S}_w^{-\frac{1}{2}})^{-\frac{1}{2}}\mathbf{S}_w^{-\frac{1}{2}}(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j)(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j)^\top \mathbf{S}_w^{-\frac{1}{2}}(\mathbf{S}_w^{-\frac{1}{2}}\mathbf{S}_{ij}\mathbf{S}_w^{-\frac{1}{2}})^{-\frac{1}{2}} + \frac{1}{\tau_i\tau_j}(\log(\mathbf{S}_w^{-\frac{1}{2}}\mathbf{S}_{ij}\mathbf{S}_w^{-\frac{1}{2}}) - \tau_i \log(\mathbf{S}_w^{-\frac{1}{2}}\mathbf{S}_i\mathbf{S}_w^{-\frac{1}{2}}) - \tau_j \log(\mathbf{S}_w^{-\frac{1}{2}}\mathbf{S}_j\mathbf{S}_w^{-\frac{1}{2}}))\big)\mathbf{S}_w^{\frac{1}{2}}\mathbf{M}\big], \tag{2.69}$$

with

$$\tau_i = \frac{\pi_i}{\pi_i + \pi_j}, \quad \tau_j = \frac{\pi_j}{\pi_i + \pi_j} \quad \text{and} \quad \mathbf{S}_{ij} = \pi_i\boldsymbol{\Sigma}_i + \pi_j\boldsymbol{\Sigma}_j \tag{2.70}$$

The maximisation of this criterion is similar to that of FLD and the Mahalanobis distance-based LDR. First, an eigenvalue decomposition of the following directed-distance matrix (DDM)[122] is found:

$$\text{DDM} = \sum_{i=1}^{K-1}\sum_{j=i+1}^{K} \pi_i\pi_j \mathbf{S}_w^{-1}\mathbf{S}_w^{\frac{1}{2}}\big((\mathbf{S}_w^{-\frac{1}{2}}\mathbf{S}_{ij}\mathbf{S}_w^{-\frac{1}{2}})^{-\frac{1}{2}}\mathbf{S}_w^{-\frac{1}{2}}(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j)(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j)^\top \mathbf{S}_w^{-\frac{1}{2}}(\mathbf{S}_w^{-\frac{1}{2}}\mathbf{S}_{ij}\mathbf{S}_w^{-\frac{1}{2}})^{-\frac{1}{2}} + \frac{1}{\tau_i\tau_j}(\log(\mathbf{S}_w^{-\frac{1}{2}}\mathbf{S}_{ij}\mathbf{S}_w^{-\frac{1}{2}}) - \tau_i \log(\mathbf{S}_w^{-\frac{1}{2}}\mathbf{S}_i\mathbf{S}_w^{-\frac{1}{2}}) - \tau_j \log(\mathbf{S}_w^{-\frac{1}{2}}\mathbf{S}_j\mathbf{S}_w^{-\frac{1}{2}}))\big)\mathbf{S}_w^{\frac{1}{2}}, \tag{2.71}$$

Then, $\mathbf{M}$ is set to the q eigenvectors corresponding to the q largest eigenvalues of the DDM.

However, while the FLD procedure provides reduction to at most $K-1$ dimensions, the remaining LDR approaches described do not provide the optimal dimensionality to which to the data is to be reduced. Existing procedures are often reformulated as eigenvalue decomposition or singular value decomposition (SVD) problems [94], after which a desired dimensionality q is chosen by taking the first q independent vectors after the decomposition. Yet, Fukunaga [8] has shown that if a Bayes classifier [30, 123] (which is the best possible classifier if the underlying distribution of the data is known [30]), such as quadratic discriminant analysis (QDA), is to be applied after LDR, the smallest set of independent features required is $K-1$. This corresponds to a reduction of the original dataset to a $(K-1)$-dimensional space . The value of $K-1$ is due to the fact that the optimal Bayes classifier evaluates K posterior probabilities, among which the highest is chosen. Since the K probabilities must sum up to one, only $K-1$ of these K probability functions suffices, and are independent. Thus, reduction to a $(K-1)$-dimensional subspace is necessary and sufficient to preserve the classification information in the original feature space [8].

In the absence of an optimal dimensionality q in the existing LDR procedures described, if q is set to $K-1$, there is no guarantee that the first $K-1$ independent vectors alone preserve the class-discriminatory information in the original space. As a result, classification information can be lost in the $(K-1)$-dimensional subspace, formed from the first $K-1$ singular vectors or eigenvectors following an SVD or eigenvalue decomposition, leading to a reduced classification accuracy using a Bayes classifier. On the other hand, if q is chosen to be much greater than $K-1$, such as the number of non-zero eigenvectors after the decomposition, the classification accuracy is not improved much, as will be shown in Chapter 4, and the problem of model complexity faced in applications such as face recognition may still persist.

2.3 Chapter summary

This chapter has provided a detailed background of linear discriminant analysis (LDA), thus setting the stage for the rest of the work described in the thesis.

First, LDA is treated as a statistical classification technique. Here, assumptions of normality and equal covariance (homoscedasticity) among the various classes in the data are required to make the procedure optimal in terms of minimising the Bayes error. In many applications, such as flowmeter diagnostics, however, the homoscedasticity assumption does not hold, leading to the development of heteroscedastic LDA (HLDA) procedures. The chapter has highlighted the main problems with the existing heteroscedastic LDA approaches. Firstly, most have no principled optimisation procedure, as

they are obtained via trial and error. As a result, they tend to be computationally intractable for high-dimensional datasets. Moreover, other HLDA approaches constrain the domain of the search space (by constraining the parameter s to the interval $[0, 1]$) in an attempt to reduce the computational complexity; this, however, leads to poor performance in terms of the classification accuracy and the area under the receiver operating characteristics curve (AUC) under class imbalance.

Two main approaches to multi-class classification using LDA are also discussed.

Subsequent to that, LDA is also treated as a linear dimensionality reduction (LDR) technique, involving the maximisation of Fisher's criterion to obtain Fisher Linear Discriminant (FLD). Several other LDR procedures are surveyed. While LDA is able to minimise the Bayes error for the two-class case with normally distributed and homoscedastic data, the overarching issue among all the LDR methods is their inability to provide a dimension-reducing transform that minimises the Bayes error for the general K-class problem.

Moreover, the LDR methods, with the exception of FLD, provide no optimal dimensionality q to which to linearly reduce a given dataset; the existing procedures are often reformulated as eigenvalue or singular-value decomposition problems, after which the desired dimensionality q is chosen by taking the first q eigenvectors or singular vectors. In the absence of the optimal value of q in the existing LDR procedures, if q is set to $K-1$, such as is necessary and sufficient for Bayesian classification, unsatisfactory classification accuracies may result.

Chapter 3

Heteroscedastic LDA for linear classification[1]

This chapter introduces a novel approach to linear discriminant analysis (LDA) that accounts for heteroscedasticity by directly minimising the Bayes error. This approach, unlike the trial and error procedure by Marks [70] and Anderson [71], has a principled optimisation procedure, and unlike the method by Fukunaga [8] and Peterson [72], does not encounter the problem of choosing an inappropriate step rate Δs, nor restricts s to the interval $[0, 1]$. Consequently, the proposed algorithm achieves an improved classification accuracy for roughly the same computational effort. Moreover, under class imbalance, the proposed heteroscedatic LDA procedure is optimal in terms of the misclassification rate and the area under the receiver operating characteristics curve, unlike the existing procedures.

This chapter focuses on binary classification; the procedures described here are then extended to multiclass classification in Chapter 4.

3.1 Gaussian linear discriminant (GLD)

Consider a training dataset $\mathcal{D}$ that is labelled and made up two classes $\mathcal{C}_1$ and $\mathcal{C}_2$ so that $\mathcal{D}$ can be partitioned as $\mathcal{D} = [\mathcal{D}_1, \mathcal{D}_2]$. Let $\bar{\mathbf{x}}_1$, $\mathbf{\Sigma}_1$ be the mean and covariance matrix of $\mathcal{D}_1$ and $\bar{\mathbf{x}}_2$, $\mathbf{\Sigma}_2$ be the mean and covariance of $\mathcal{D}_2$ respectively. Also, let π_1 and π_2 respectively be the prior probabilities of $\mathcal{C}_1$ and $\mathcal{C}_2$.

[1] Most of the work presented in this chapter first appeared in: K. S. Gyamfi, J. Brusey, A. Hunt and E. Gaura, "Linear classifier design under heteroscedasticity in Linear Discriminant Analysis," *Expert Systems with Applications*, vol. 79, Aug. 2017, pp. 44-52. and in: K. S. Gyamfi, J. Brusey, A. Hunt and E. Gaura, "Linear classifier design for heteroscedastic LDA under class imbalance," *Proceedings of the Workshop on Learning in the Presence of Class Imbalance and Concept Drift*, Melbourne, IJCAI 2017, pp. 8-15

Let $\mathbf{w} \in \mathbb{R}^d$ be a vector of weights and $w_0 \in \mathbb{R}$ a threshold such that for a given test vector $\mathbf{x}$:

$$\mathcal{C}^*(\mathbf{x}) = \begin{cases} \mathcal{C}_1 & \text{if} \quad y = \mathbf{w}^\top \mathbf{x} \geq w_0 \\ \mathcal{C}_2 & \text{if} \quad y = \mathbf{w}^\top \mathbf{x} < w_0 \end{cases} \tag{3.1}$$

The test vector $\mathbf{x}$ is assumed to have a multivariate normal distribution in classes $\mathcal{C}_1$ and $\mathcal{C}_2$, and therefore, y has a mean of μ_1 and a variance of σ_1^2 for class $\mathcal{C}_1$ and a mean of μ_2 and a variance of σ_2^2 for class $\mathcal{C}_2$; these are given as:

$$\mu_1 = \mathbf{w}^\top \bar{\mathbf{x}}_1 \quad \mu_2 = \mathbf{w}^\top \bar{\mathbf{x}}_2 \quad \sigma_1^2 = \mathbf{w}^\top \boldsymbol{\Sigma}_1 \mathbf{w} \quad \sigma_2^2 = \mathbf{w}^\top \boldsymbol{\Sigma}_2 \mathbf{w} \tag{3.2}$$

It will be recalled from Chapter 2 that the Bayes error is given as.

$$p_e = \pi_1 p(y < w_0 | \mathcal{C}_1) + \pi_2 p(y \geq w_0 | \mathcal{C}_2), \tag{3.3}$$

The individual misclassification probabilities can be expressed as:

$$p(y < w_0 | \mathcal{C}_1) = \int_{-\infty}^{w_0} \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(\zeta - \mu_1)^2}{2\sigma_1^2} \right] d\zeta \tag{3.4}$$

and

$$p(y \geq w_0 | \mathcal{C}_2) = \int_{w_0}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left[-\frac{(\zeta - \mu_2)^2}{2\sigma_2^2} \right] d\zeta \tag{3.5}$$

By letting $z = \frac{\zeta - \mu_1}{\sigma_1}$, (3.4) can be rewritten as:

$$p(y < w_0 | \mathcal{C}_1) = \int_{-\infty}^{\frac{w_0 - \mu_1}{\sigma_1}} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz = 1 - Q\left(\frac{w_0 - \mu_1}{\sigma_1} \right) \tag{3.6}$$

Similarly, by letting $z = \frac{\zeta - \mu_2}{\sigma_2}$, (3.5) can be expressed as:

$$p(y \geq w_0 | \mathcal{C}_2) = \int_{\frac{w_0 - \mu_2}{\sigma_2}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} dz = Q\left(\frac{w_0 - \mu_2}{\sigma_2} \right) \tag{3.7}$$

where $Q(\cdot)$ is the Q-function. Therefore, the Bayes error to be minimised may be rewritten as:

$$p_e = \pi_1 \big[1 - Q(z_1) \big] + \pi_2 \big[Q(z_2) \big] \tag{3.8}$$

where

$$z_1 = \frac{w_0 - \mu_1}{\sigma_1} \quad \text{and} \quad z_2 = \frac{w_0 - \mu_2}{\sigma_2} \tag{3.9}$$

3.1.1 Optimality conditions

Our aim is to find a local minimum of p_e. A necessary condition is for the gradient of p_e to be zero, i.e.,

$$\nabla p_e(\mathbf{w}, w_0) = \left[\frac{\partial p_e}{\partial \mathbf{w}^\top}, \frac{\partial p_e}{\partial w_0}\right]^\top = \mathbf{0}. \tag{3.10}$$

Note that:

$$Q(x) = 1 - \Phi(x) \tag{3.11}$$

where $\Phi(x)$ is the cumulative distribution function (CDF) of the standard normal random variable x. Therefore,

$$\frac{dQ(x)}{dx} = -\Phi'(x) \tag{3.12}$$

But, by definition,

$$\Phi'(x) = f(x) \tag{3.13}$$

where $f(x)$ is the probability density function (PDF) of x.

Using the above relations, it can be shown that:

$$\frac{\partial p_e}{\partial \mathbf{w}} = \pi_1\left(\frac{1}{\sqrt{2\pi}}e^{-\frac{z_1^2}{2}}\frac{\partial z_1}{\partial \mathbf{w}}\right) - \pi_2\left(\frac{1}{\sqrt{2\pi}}e^{-\frac{z_2^2}{2}}\frac{\partial z_2}{\partial \mathbf{w}}\right) \tag{3.14}$$

From (3.9) and (3.2), it can be shown that

$$\frac{\partial z_1}{\partial \mathbf{w}} = \frac{\sigma_1(-\bar{\mathbf{x}}_1) - (w_0 - \mu_1)\left(\frac{1}{2}(\mathbf{w}^\top \boldsymbol{\Sigma}_1 \mathbf{w})^{-\frac{1}{2}}\right)2\boldsymbol{\Sigma}_1\mathbf{w}}{\sigma_1^2} \tag{3.15}$$

$$\frac{\partial z_1}{\partial \mathbf{w}} = \frac{-\sigma_1\bar{\mathbf{x}}_1 - z_1\boldsymbol{\Sigma}_1\mathbf{w}}{\sigma_1^2} \tag{3.16}$$

In a similar way, it can be shown from (3.9) and (3.2) that

$$\frac{\partial z_2}{\partial \mathbf{w}} = \frac{-\sigma_2\bar{\mathbf{x}}_2 - z_2\boldsymbol{\Sigma}_2\mathbf{w}}{\sigma_2^2} \tag{3.17}$$

Therefore,

$$\frac{\partial p_e}{\partial \mathbf{w}} = \frac{\pi_1}{\sqrt{2\pi}}\left[-e^{-\frac{z_1^2}{2}}\left(\frac{\sigma_1\bar{\mathbf{x}}_1 + z_1\boldsymbol{\Sigma}_1\mathbf{w}}{\sigma_1^2}\right)\right] + \frac{\pi_2}{\sqrt{2\pi}}\left[e^{-\frac{z_2^2}{2}}\left(\frac{\sigma_2\bar{\mathbf{x}}_2 + z_2\boldsymbol{\Sigma}_2\mathbf{w}}{\sigma_2^2}\right)\right] \tag{3.18}$$

Also, from (3.8),

$$\frac{\partial p_e}{\partial w_0} = \pi_1\left(\frac{1}{\sqrt{2\pi}}e^{-\frac{z_1^2}{2}}\frac{\partial z_1}{\partial w_0}\right) - \pi_2\left(\frac{1}{\sqrt{2\pi}}e^{-\frac{z_2^2}{2}}\frac{\partial z_2}{\partial w_0}\right) \tag{3.19}$$

Recalling (3.9) and (3.2), it can be shown that

$$\frac{\partial z_1}{\partial w_0} = \frac{1}{\sigma_1} \quad \text{and} \quad \frac{\partial z_2}{\partial w_0} = \frac{1}{\sigma_2} \tag{3.20}$$

Therefore,

$$\frac{\partial p_e}{\partial w_0} = \frac{\pi_1}{\sqrt{2\pi}}\left(\frac{1}{\sigma_1}e^{-\frac{z_1^2}{2}}\right) - \frac{\pi_2}{\sqrt{2\pi}}\left(\frac{1}{\sigma_2}e^{-\frac{z_2^2}{2}}\right) \tag{3.21}$$

Now, equating the gradient $\nabla p_e(\mathbf{w}, w_0)$ to zero, the following set of equations are obtained:

$$\left(\frac{\pi_2 z_2}{\sigma_2^2}e^{-z_2^2/2}\boldsymbol{\Sigma}_2 - \frac{\pi_1 z_1}{\sigma_1^2}e^{-z_1^2/2}\boldsymbol{\Sigma}_1\right)\mathbf{w} = \left(\frac{\pi_1}{\sigma_1}e^{-z_1^2/2}\right)\bar{\mathbf{x}}_1 - \left(\frac{\pi_2}{\sigma_2}e^{-z_2^2/2}\right)\bar{\mathbf{x}}_2 \tag{3.22}$$

$$\frac{\pi_1}{\sigma_1}e^{-z_1^2/2} = \frac{\pi_2}{\sigma_2}e^{-z_2^2/2} \tag{3.23}$$

Substituting (3.23) into (3.22) yields:

$$\left(\frac{z_2}{\sigma_2}\boldsymbol{\Sigma}_2 - \frac{z_1}{\sigma_1}\boldsymbol{\Sigma}_1\right)\mathbf{w} = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{3.24}$$

Then the vector $\mathbf{w}$ can be given by:

$$\mathbf{w} = \left(\frac{z_2}{\sigma_2}\boldsymbol{\Sigma}_2 - \frac{z_1}{\sigma_1}\boldsymbol{\Sigma}_1\right)^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{3.25}$$

But for the fact that z_1 and z_2 are functions of w_0 as can be seen from (3.9), $\mathbf{w}$ could have been solved for iteratively from (3.25) starting from some initial solution. To overcome this problem, an explicit representation of w_0 in terms of $\mathbf{w}$ is needed from (3.23) to substitute in z_1 and z_2 in (3.25) to allow for the iterative solution of $\mathbf{w}$ from (3.25). Solving for w_0 from (3.23) results in the following quadratic:

$$\frac{z_2^2}{2} - \frac{z_1^2}{2} - \ln\left(\frac{\tau\sigma_1}{\sigma_2}\right) = 0 \tag{3.26}$$

which can be expanded to:

$$\left(\frac{w_0 - \mu_2}{\sigma_2}\right)^2 - \left(\frac{w_0 - \mu_1}{\sigma_1}\right)^2 - 2\ln\frac{\tau\sigma_1}{\sigma_2} = 0 \tag{3.27}$$

and subsequently as:

$$\left(\frac{1}{\sigma_2^2} - \frac{1}{\sigma_1^2}\right)w_0^2 + 2\left(\frac{\mu_1}{\sigma_1^2} - \frac{\mu_2}{\sigma_2^2}\right)w_0 + \frac{\mu_2^2}{\sigma_2^2} - \frac{\mu_1^2}{\sigma_1^2} - 2\log\frac{\tau\sigma_1}{\sigma_2} = 0, \tag{3.28}$$

where τ is given as before as $\tau = \pi_2/\pi_1$. If τ is defined and not equal to zero, and $\sigma_1^2 \neq \sigma_2^2$ (since $\mathbf{\Sigma}_1 \neq \mathbf{\Sigma}_2$ for heteroscedastic LDA), (3.27) can be shown to have the following solutions:

$$w_0 = \frac{\mu_2\sigma_1^2 - \mu_1\sigma_2^2 \pm \sigma_1\sigma_2\sqrt{(\mu_1-\mu_2)^2 + 2(\sigma_1^2-\sigma_2^2)\ln\left(\frac{\tau\sigma_1}{\sigma_2}\right)}}{\sigma_1^2-\sigma_2^2}, \tag{3.29}$$

i.e.,

$$w_0^+ = \frac{\mu_2\sigma_1^2 - \mu_1\sigma_2^2 + \sigma_1\sigma_2\beta}{\sigma_1^2-\sigma_2^2} \tag{3.30}$$

and

$$w_0^- = \frac{\mu_2\sigma_1^2 - \mu_1\sigma_2^2 - \sigma_1\sigma_2\beta}{\sigma_1^2-\sigma_2^2}, \tag{3.31}$$

where

$$\beta = \sqrt{(\mu_1-\mu_2)^2 + 2(\sigma_1^2-\sigma_2^2)\ln\left(\frac{\tau\sigma_1}{\sigma_2}\right)} \tag{3.32}$$

Nevertheless, since there are two solutions, a choice has to be made as to which of them is substituted into (3.25). To eliminate one of the solutions, the second-order partial derivative of p_e with respect to w_0 is considered, evaluated at w_0 as given by (3.29), to determine under what condition the second-order derivative is greater than or equal to zero. This is a second-order necessary condition for p_e to be a local minimum. From (3.21), it can be shown that:

$$\frac{\partial^2 p_e}{\partial w_0^2} = \frac{\pi_1}{\sqrt{2\pi}}\left(-\frac{z_1}{\sigma_1^2}e^{-z_1^2/2}\right) + \frac{\pi_2}{\sqrt{2\pi}}\left(\frac{z_2}{\sigma_2^2}e^{-z_2^2/2}\right) \tag{3.33}$$

This second-order derivative is denoted by h. Then all possibilities of z_1 and z_2 (which are the variables in (3.33) that depend on w_0) are considered under three cases, and the sign of h is analysed in each.

1. Case 1: $z_2 \leq 0$ and $z_1 \geq 0$: then h is trivially non-positive.

2. Case 2: $z_2 \geq 0$ and $z_1 \leq 0$: then h is trivially non-negative.

3. Case 3: $z_2 > 0$ and $z_1 > 0$ or $z_2 < 0$ and $z_1 < 0$: then h is non-negative if and only if

$$\ln\left(\frac{\pi_2 z_2}{\sigma_2^2}\right) - \frac{z_2^2}{2} \geq \ln\left(\frac{\pi_1 z_1}{\sigma_1^2}\right) - \frac{z_1^2}{2} \tag{3.34}$$

i.e.,

$$\ln\left(\frac{z_2}{\sigma_2}\Big/\frac{z_1}{\sigma_1}\right) \geq \frac{z_2^2}{2} - \frac{z_1^2}{2} - \ln\left(\frac{\tau\sigma_1}{\sigma_2}\right) \tag{3.35}$$

It will be noted that the right-hand side of the inequality (3.35) is identically zero, as can be seen

from (3.26). Therefore, the condition under which h is greater than or equal to zero is when:

$$\frac{z_2}{\sigma_2} \geq \frac{z_1}{\sigma_1} \tag{3.36}$$

Note also that Case 2 necessarily satisfies (3.36) so that (3.36) is considered as the general inequality for the non-negativity of h for all cases, and thus for w_0 to be a local minimum.

Now, when one considers the two solutions of w_0 in (3.29), only w_0^+ satisfies the inequality of (3.36), i.e., only this choice of w_0 corresponds to a local minimum. The proof of this is shown below.

Theorem 1. *Let w_0^+ and w_0^- be the two distinct solutions of* (3.29)*, then w_0^+ and w_0^- cannot both satisfy* (3.36) *given that $\sigma_1 \neq \sigma_2$; only w_0^+ satisfies* (3.36)*.*

Proof. Let β be a positive scalar given by:

$$\beta = \sqrt{(\mu_1 - \mu_2)^2 + 2(\sigma_1^2 - \sigma_2^2)\ln\left(\frac{\tau\sigma_1}{\sigma_2}\right)} \tag{3.37}$$

Note that when $\beta = 0$, (3.29) has a repeated root so that $w_0^+ = w_0^-$, which are not distinct. Also, let

$$w_0^+ = \frac{\mu_2\sigma_1^2 - \mu_1\sigma_2^2 + \sigma_1\sigma_2\beta}{\sigma_1^2 - \sigma_2^2} \tag{3.38}$$

Then

$$z_2 = \frac{(\mu_2 - \mu_1)\sigma_2 + \beta\sigma_1}{\sigma_1^2 - \sigma_2^2}, \quad \text{and} \quad z_1 = \frac{(\mu_2 - \mu_1)\sigma_1 + \beta\sigma_2}{\sigma_1^2 - \sigma_2^2} \tag{3.39}$$

so that

$$\frac{z_2}{\sigma_2} = \frac{(\mu_2 - \mu_1)\sigma_2 + \beta\sigma_1}{\sigma_2(\sigma_1^2 - \sigma_2^2)}, \quad \frac{z_1}{\sigma_1} = \frac{(\mu_2 - \mu_1)\sigma_1 + \beta\sigma_2}{\sigma_1(\sigma_1^2 - \sigma_2^2)} \tag{3.40}$$

Suppose that w_0^+ satisfies (3.36), then

$$\frac{(\mu_2 - \mu_1)\sigma_2 + \beta\sigma_1}{\sigma_2(\sigma_1^2 - \sigma_2^2)} \geq \frac{(\mu_2 - \mu_1)\sigma_1 + \beta\sigma_2}{\sigma_1(\sigma_1^2 - \sigma_2^2)} \tag{3.41}$$

i.e.,

$$\frac{\beta\sigma_1^2}{\sigma_1^2 - \sigma_2^2} \geq \frac{\beta\sigma_2^2}{\sigma_1^2 - \sigma_2^2} \implies \beta\frac{\sigma_1^2 - \sigma_2^2}{\sigma_1^2 - \sigma_2^2} \geq 0, \tag{3.42}$$

Therefore $\beta \geq 0$.

Consider now w_0^- given as:

$$w_0^- = \frac{\mu_2\sigma_1^2 - \mu_1\sigma_2^2 - \sigma_1\sigma_2\beta}{\sigma_1^2 - \sigma_2^2} \tag{3.43}$$

Then

$$z_2 = \frac{(\mu_2 - \mu_1)\sigma_2 - \beta\sigma_1}{\sigma_1^2 - \sigma_2^2}, \quad \text{and} \quad z_1 = \frac{(\mu_2 - \mu_1)\sigma_1 - \beta\sigma_2}{\sigma_1^2 - \sigma_2^2} \tag{3.44}$$

such that

$$\frac{z_2}{\sigma_2} = \frac{(\mu_2 - \mu_1)\sigma_2 - \beta\sigma_1}{\sigma_2(\sigma_1^2 - \sigma_2^2)}, \quad \frac{z_1}{\sigma_1} = \frac{(\mu_2 - \mu_1)\sigma_1 - \beta\sigma_2}{\sigma_1(\sigma_1^2 - \sigma_2^2)} \tag{3.45}$$

In order for (3.36) to be satisfied,

$$\frac{(\mu_2 - \mu_1)\sigma_2 - \beta\sigma_1}{\sigma_2(\sigma_1^2 - \sigma_2^2)} \geq \frac{(\mu_2 - \mu_1)\sigma_1 - \beta\sigma_2}{\sigma_1(\sigma_1^2 - \sigma_2^2)}, \tag{3.46}$$

i.e.,

$$\frac{-\beta\sigma_1^2}{\sigma_1^2 - \sigma_2^2} \geq \frac{-\beta\sigma_2^2}{\sigma_1^2 - \sigma_2^2} \implies \beta\frac{\sigma_1^2 - \sigma_2^2}{\sigma_1^2 - \sigma_2^2} \leq 0 \tag{3.47}$$

(3.47) implies that $\beta \leq 0$. However, as given in the preamble, $\beta > 0$. Thus, w_0^- does not satisfy (3.36), and only w_0^+ does. □

This expression of w_0^+ may then be substituted into (3.25) so that (3.25) is in terms of $\mathbf{w}$ only. Even so, $\mathbf{w}$ has to be solved for iteratively. This is because (3.25) has no closed-form solution since $\mu_1, \mu_2, \sigma_1, \sigma_2$ are themselves functions of $\mathbf{w}$. The iterative procedure, denoted as the Recursive Gaussian Linear Discriminant (R-GLD), is described in detail in Algorithm 1.

Algorithm 1 Recursive GLD (R-GLD)

1: Input: $\mathcal{D}_1$ and $\mathcal{D}_2$
2: Obtain $\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \mathbf{\Sigma}_1, \mathbf{\Sigma}_2$
3: Initialise $\mathbf{w}$: $\mathbf{w} \leftarrow (\beta_2\mathbf{\Sigma}_2 - \beta_1\mathbf{\Sigma}_1)^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)$
4: Evaluate $\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, z_1, z_2$.
5: **while** Stopping criteria are not satisfied **do**
6: Solve for $w_0 = w_0^+$ from (3.30)
7: Evaluate z_1, z_2
8: Evaluate the Bayes error $p_e = \pi_1\big[1 - Q(z_1)\big] + \pi_2\big[Q(z_2)\big]$
9: Update $\mathbf{w}$ as $\mathbf{w} \leftarrow \left(\frac{z_2}{\sigma_2}\mathbf{\Sigma}_2 - \frac{z_1}{\sigma_1}\mathbf{\Sigma}_1\right)^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)$
10: Evaluate $\mu_1, \mu_2, \sigma_1, \sigma_2$
11: **end while**

In Step 2 of Algorithm 1, β_1 and β_2 are chosen randomly such that $\beta_2 > \beta_1$ in order to satisfy (3.36).

3.1.2 Stopping criteria

The R-GLD algorithm may be terminated under any of the following conditions:

1. When the change in p_e, $\Delta p_e < \epsilon_1$, where ϵ_1 is some predefined tolerance.
2. When the change in the norm of $\mathbf{w}$, $\Delta\|\mathbf{w}\|^2 < \epsilon_2$, where ϵ_2 is some predefined tolerance.
3. When the gradient of p_e as given by (3.18) is less than a certain tolerance ϵ_3.
4. After a fixed number of iterations I, if convergence is slow.

After termination, the final solution may be chosen either as the solution to which the iterations converge, or the solution corresponding to the minimum p_e found in the iterative updates.

However, the procedure described in Algorithm 1 is not guaranteed to converge to a minimum of p_e. This is because the optimal $\mathbf{w}$ as given by (3.25) is obtained using first-order optimality conditions which are equally satisfied for other critical points such as local maxima or saddle points. For this reason, Algorithm 1 may have to be run a number of times in order to increase the chances of convergence to a local minimum. An alternative approach is to minimise the Bayes error using a gradient descent procedure.

3.1.3 Gradient descent GLD (G-GLD)

Gradient descent is a first-order optimisation procedure (i.e., it requires the computation of the gradient), frequently employed in machine learning to minimise a differentiable function. Since the gradient represents the direction of the greatest rate of increase of a function, the negative direction of the gradient represents a descent direction. Thus, taking steps in the negative direction of the gradient consequently minimises the Bayes error, provided the steps taken are small enough to satisfy Wolfe's conditions [124]. Then, for $i = 0$ until some stopping criteria are satisfied, $\mathbf{w}$ and w_0 are updated as follows:

$$\mathbf{w}^{i+1} = \mathbf{w}^i - \gamma \frac{\partial p_e}{\partial \mathbf{w}^i} \tag{3.48}$$

$$w_0^{i+1} = w_0^i - \gamma \frac{\partial p_e}{\partial w_0^i} \tag{3.49}$$

This is shown in Algorithm 2 in its proper context.

The algorithm may be terminated using the same stopping criteria outlined above for the R-GLD. However, the gradient descent procedure, unlike the R-GLD algorithm, is guaranteed to converge to a local minimum, if the step rate γ is small enough. Still, gradient descent can be rather slow to converge,

Algorithm 2 Gradient descent GLD (G-GLD)

1: Input: $\mathcal{D}_1$ and $\mathcal{D}_2$

2: Obtain $\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \mathbf{\Sigma}_1, \mathbf{\Sigma}_2$

3: Initialise $\mathbf{w}$: $\mathbf{w} \leftarrow (\beta_2\mathbf{\Sigma}_2 - \beta_1\mathbf{\Sigma}_1)^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)$

4: Evaluate $\mu_1, \mu_2, \sigma_1^2, \sigma_2^2$.

5: Solve for $w_0 = w_0^+$ from (3.30)

6: $i \leftarrow 0$

7: **while** Stopping criteria are not satisfied **do**

8: Evaluate z_1, z_2

9: Evaluate the gradient of p_e w.r.t. $\mathbf{w}$ as given by:

$$\frac{\partial p_e}{\partial \mathbf{w}} = \frac{\pi_1}{\sqrt{2\pi}}\left[- e^{-\frac{z_1^2}{2}}\left(\frac{\sigma_1\bar{\mathbf{x}}_1 + z_1\mathbf{\Sigma}_1\mathbf{w}}{\sigma_1^2}\right)\right] + \frac{\pi_2}{\sqrt{2\pi}}\left[e^{-\frac{z_2^2}{2}}\left(\frac{\sigma_2\bar{\mathbf{x}}_2 + z_2\mathbf{\Sigma}_2\mathbf{w}}{\sigma_2^2}\right)\right] \tag{3.50}$$

10: Evaluate the gradient of p_e w.r.t. w_0 as given by:

$$\frac{\partial p_e}{\partial w_0} = \frac{\pi_1}{\sqrt{2\pi}}\left(\frac{1}{\sigma_1}e^{-\frac{z_1^2}{2}}\right) - \frac{\pi_2}{\sqrt{2\pi}}\left(\frac{1}{\sigma_2}e^{-\frac{z_2^2}{2}}\right) \tag{3.51}$$

11: Evaluate the Bayes error $p_e = \pi_1\big[1 - Q(z_1)\big] + \pi_2\big[Q(z_2)\big]$

12: Update $\mathbf{w}$ as $\mathbf{w}^{i+1} \leftarrow \mathbf{w}^i - \gamma\frac{\partial p_e}{\partial \mathbf{w}^i}$

13: Update w_0 as $w_0^{i+1} \leftarrow w_0^i - \gamma\frac{\partial p_e}{\partial w_0^i}$

14: Evaluate $\mu_1, \mu_2, \sigma_1, \sigma_2$

15: **end while**

as its rate of convergence is linear. In this regard, an alternative approach to minimising the Bayes error is Newton's method.

3.1.4 Newton's method

Newton's method is a second-order optimisation procedure derived from the Taylor's series expansion of a twice-differentiable function as follows:

Consider the twice differentiable function $f(\mathbf{x})$. Suppose it is desired to find a descent direction $\mathbf{d}_i$ at $\mathbf{x} = \mathbf{x}^i$ in the ith iteration along which the objective function may be minimised. The Taylor's series approximation of $f(\mathbf{x} + \Delta\mathbf{x})$ at $\mathbf{x} = \mathbf{x}^i$, ignoring all third and higher-order terms, is given by:

$$f(\mathbf{x} + \Delta\mathbf{x}) \approx f(\mathbf{x}^i) + \Delta\mathbf{x}^\top \nabla f(\mathbf{x}^i) + \frac{\Delta\mathbf{x}^\top \nabla^2 f(\mathbf{x}^i) \Delta\mathbf{x}}{2} \tag{3.52}$$

By differentiating (3.52) w.r.t. $\Delta\mathbf{x}$ and equating to zero, the optimal $\Delta\mathbf{x}$ can be found, i.e.,

$$\frac{df(\mathbf{x}^i + \Delta\mathbf{x})}{d\Delta\mathbf{x}} = \nabla f(\mathbf{x}^i) + \nabla^2 f(\mathbf{x}^i)\Delta\mathbf{x} = 0, \tag{3.53}$$

so that the optimal $\Delta\mathbf{x}$ is given as:

$$\Delta\mathbf{x} = -[\nabla^2 f(\mathbf{x}^i)]^{-1} \nabla f(\mathbf{x}^i) \tag{3.54}$$

If the Hessian $[\nabla^2 f(\mathbf{x}^i)]$ is positive definite, then the descent direction can be given as $\mathbf{d}_i = \Delta\mathbf{x}$. This descent direction $\mathbf{d}_i$ may replace the negative gradient used in gradient descent as a descent direction, so that with regard to the minimisation of the Bayes error, the following iterative procedure is obtained:

$$\hat{\mathbf{w}}^{i+1} = \hat{\mathbf{w}}^i - \gamma[\nabla^2 p_e(\hat{\mathbf{w}}^i)]^{-1} \nabla p_e(\hat{\mathbf{w}}^i), \tag{3.55}$$

where $\hat{\mathbf{w}}^i = [\mathbf{w}^{i\top}, w_0]^\top$.

While the above procedure, known as Newton's method, converges much faster than gradient descent, there are often some computational issues. In particular, the Hessian $\nabla^2 p_e(\hat{\mathbf{w}}^i)$ may not be invertible or it may be close to singular, so that in practical implementations the matrix is often preconditioned [124] before the above procedure. Also, for very high-dimensional data, computing the Hessian or its inverse in every iteration may be computationally expensive. This has led to the use of quasi-Newton methods, such as the BFGS algorithm [124], that approximate the Hessian and avoid an explicit matrix

inversion. Furthermore, the algorithm tends to be numerically unstable as it approaches the minima since the gradient tends to $\mathbf{0}$.

3.1.5 Non-convexity of p_e

In each of the algorithms described so far, the focus has been on finding a local minimum of the Bayes error p_e. Whether a local minimum found is the global minimum depends on the convexity of the objective function. For p_e to be convex, the Hessian matrix $\mathcal{H} = \nabla^2 p_e(\mathbf{w}, w_0)$ has to be positive semi-definite for all $\mathbf{w} \in \mathbb{R}^d$ and $w_0 \in \mathbb{R}$, i.e.,

$$\mathbf{v}^\top \mathcal{H} \mathbf{v} \geq 0 \tag{3.56}$$

for every non-zero $\mathbf{v} \in \mathbb{R}^{d+1}$. Observe that:

$$\nabla^2 p_e(\mathbf{w}, w_0) = \begin{bmatrix} \dfrac{\partial p_e^2}{\partial \mathbf{w} \partial \mathbf{w}^\top} & \dfrac{\partial p_e^2}{\partial \mathbf{w} \partial w_0} \\ \dfrac{\partial p_e^2}{\partial w_0 \partial \mathbf{w}^\top} & \dfrac{\partial p_e^2}{\partial w_0^2} \end{bmatrix} \tag{3.57}$$

Then, suppose that $\mathbf{v} = [\mathbf{0}_d^\top, x]^\top$, where $x \in \mathbb{R}$ and $\mathbf{0}_d$ is a d-dimensional vector of all zeros. Then,

$$\mathbf{v}^\top \mathcal{H} \mathbf{v} = x^2 \frac{\partial p_e^2}{\partial w_0^2} \tag{3.58}$$

It may be recalled from (3.33) that:

$$\frac{\partial^2 p_e}{\partial w_0^2} = \frac{\pi_1}{\sqrt{2\pi}} \left(-\frac{z_1}{\sigma_1^2} e^{-z_1^2/2} \right) + \frac{\pi_2}{\sqrt{2\pi}} \left(\frac{z_2}{\sigma_2^2} e^{-z_2^2/2} \right)$$

Thus, the positive semi-definiteness of $\mathcal{H}$ requires that $\frac{\partial^2 p_e}{\partial w_0^2} \geq 0$.

However, recall that even under the first-order optimality condition of p_e w.r.t. w_0, $\frac{\partial^2 p_e}{\partial w_0^2} \geq 0$ if and only if $\frac{z_2}{\sigma_2} \geq \frac{z_1}{\sigma_1}$. Thus, as has been shown in Theorem 1, for w_0 given by:

$$w_0 = w_0^- = \frac{\mu_2 \sigma_1^2 - \mu_1 \sigma_2^2 - \sigma_1 \sigma_2 \beta}{\sigma_1^2 - \sigma_2^2}, \tag{3.59}$$

this condition is not satisfied.

Therefore $\frac{\partial^2 p_e}{\partial w_0^2}$ is not greater than or equal to zero for every $\mathbf{w} \in \mathbb{R}^d$ and $w_0 \in \mathbb{R}$. Hence, $\mathcal{H}$ is not positive-semidefinite. This in turn implies that p_e is non-convex.

For this reason, the Bayes error is characterised by multiple local minima, so that the algorithms

described have to be run several times, from different initial solutions, to improve the quality of the overall solution.

3.2 Non-normal distributions

So far, the fundamental assumption that has been used to derive the proposed algorithms is that the data in each class has a normal distribution. Thus, for a non-normal distribution, the linear classifier that has been obtained does not minimise the Bayes error for that distribution. It can be argued, however, that if this unknown distribution is nearly-normal [125], then a more robust linear classifier may be found in some neighbourhood of the GLD classifier. For this reason, a local neighbourhood search algorithm can be employed to explore the region in $\mathbb{R}^{d+1}$ around the GLD to obtain the classifier that minimises the number of misclassifications on the training dataset. This can be done by perturbing each of the $d+1$ vector elements in the optimal $\tilde{\mathbf{w}} = [w_0, \mathbf{w}^\top]^\top$ obtained from the GLD procedure (i.e., R-GLD or G-GLD) by a small amount $\delta\tilde{w}_i$. After every perturbation, the resulting classifier is evaluated on the training dataset. This procedure is repeated until the stopping criterion is satisfied, as described in Algorithm 3.

Algorithm 3 Local Neighbourhood Search (LNS)

Input: Optimal $\tilde{\mathbf{w}} = [w_0, \mathbf{w}^\top]^\top$ obtained from the GLD.
while Stopping criterion is not satisfied **do**
    Let $\tilde{\mathbf{w}}$ be the current solution.
    for $i \leftarrow 1$ to d **do**
        $\mathbf{v}^+ \leftarrow \tilde{\mathbf{w}}$, $\mathbf{v}^- \leftarrow \tilde{\mathbf{w}}$.
        $\mathbf{v}^+ \leftarrow v_i^+ + \delta v_i^+$
        Evaluate the misclassifications on the training set using $\mathbf{v}^+$
        $\mathbf{v}^- \leftarrow v_i^- - \delta v_i^-$
        Evaluate the misclassifications on the training set using $\mathbf{v}^-$
    end for
    Set the classifier with the minimum number of misclassifications as the current solution $\tilde{\mathbf{w}}$.
end while
Choose the classifier with the smallest number of misclassifications.

The stopping criterion is such that the algorithm is terminated after a certain maximum number of iterations R_{max} is reached, or after a predefined number of iterations r_{min} when there is no improvement

in the minimum number of misclassifications that has been found during the search.

3.3 Fisher's Linear Discriminant

It will be recalled from Chapter 2 that if a common covariance matrix is assumed between the two classes, Linear Discriminant Analysis (LDA) results in the following choice of the weight vector $\mathbf{w}$ and the threshold w_0:

$$\mathbf{w} = \mathbf{\Sigma}_x^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2), \quad w_0 = \ln \tau + \frac{1}{2}(\bar{\mathbf{x}}_1^\top \mathbf{\Sigma}_x^{-1} \bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2^\top \mathbf{\Sigma}_x^{-1} \bar{\mathbf{x}}_2), \tag{3.60}$$

where $\mathbf{\Sigma}_x$ is the common covariance matrix defined as:

$$\mathbf{\Sigma}_x = \pi_1 \bar{\mathbf{\Sigma}}_1 + \pi_2 \bar{\mathbf{\Sigma}}_2, \tag{3.61}$$

and $\bar{\mathbf{\Sigma}}_1, \bar{\mathbf{\Sigma}}_2$ are the sample covariance estimates of the data in classes $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively. π_1 and π_2 retain their usual definitions as the prior probabilities of classes $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively.

It is straightforward to show that on the assumption of homoscedasticity, the optimal solution derived in (3.25) readily yields the optimal Fisher's Linear Discriminant given by (3.60). If $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \mathbf{\Sigma}_x$, then $\mathbf{w}^\top \mathbf{\Sigma}_1 \mathbf{w} = \mathbf{w}^\top \mathbf{\Sigma}_2 \mathbf{w}$, and $\sigma_1 = \sigma_2 = \sigma_x$. Then the optimal weight vector $\mathbf{w}$ as given by (3.25) becomes:

$$\mathbf{w} = \left(\frac{z_2}{\sigma_2} - \frac{z_1}{\sigma_1}\right)^{-1} \mathbf{\Sigma}_x^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) = \frac{\sigma_x^2}{\mu_1 - \mu_2} \mathbf{\Sigma}_x^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{3.62}$$

Also, (3.28) decomposes into the following linear equation in w_0:

$$2\left(\frac{\mu_1 - \mu_2}{\sigma_x^2}\right) w_0 + \frac{\mu_2^2 - \mu_1^2}{\sigma_x^2} - 2 \ln \tau = 0, \tag{3.63}$$

which has the following solution:

$$w_0 = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma_x^2}{\mu_1 - \mu_2} \ln \tau \tag{3.64}$$

Notice that by definition (3.1), $\mu_1 = \mathbf{w}^\top \bar{\mathbf{x}}_1 > \mu_2 = \mathbf{w}^\top \bar{\mathbf{x}}_2$. Therefore, the factor $\frac{\sigma_x^2}{\mu_1 - \mu_2}$ in (3.62) can only be positive; thus, $\mathbf{w}$ and w_0 given by (3.62) and (3.64) may be proportionally scaled down by $\frac{\sigma_x^2}{\mu_1 - \mu_2}$ without changing the discrimination criterion as given by (3.1).

After scaling down, $\mathbf{w}$ becomes:

$$\mathbf{w} = \mathbf{\Sigma}_x^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{3.65}$$

and w_0 becomes:

$$w_0 = \ln \tau + \frac{\mu_1^2 - \mu_2^2}{2\sigma_x^2} \tag{3.66}$$

Still, the term $\frac{\mu_1^2-\mu_2^2}{2\sigma_x^2}$ can be expressed in terms of $\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2$ and $\mathbf{\Sigma}_x$ as follows.

It will be noted that:

$$\frac{\mu_1^2 - \mu_2^2}{2\sigma_x^2} = \frac{(\mu_1 - \mu_2)(\mu_1 + \mu_2)}{2\sigma_x^2}, \tag{3.67}$$

while

$$\begin{aligned}
\mu_1 - \mu_2 &= \mathbf{w}^\top(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^\top \mathbf{\Sigma}_x^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \\
\mu_1 + \mu_2 &= \mathbf{w}^\top(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2) = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^\top \mathbf{\Sigma}_x^{-1}(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2) \\
\sigma_x^2 &= \mathbf{w}^\top \mathbf{\Sigma}_x \mathbf{w} = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^\top \mathbf{\Sigma}_x^{-1} \mathbf{\Sigma}_x \mathbf{\Sigma}_x^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^\top \mathbf{\Sigma}_x^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)
\end{aligned} \tag{3.68}$$

Substituting (3.68) into (3.67) results in the following:

$$\frac{\mu_1^2 - \mu_2^2}{2\sigma_x^2} = \frac{1}{2}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^\top \mathbf{\Sigma}_x^{-1}(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2) = \frac{1}{2}\big(\bar{\mathbf{x}}_1^\top \mathbf{\Sigma}_x^{-1} \bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2^\top \mathbf{\Sigma}_x^{-1} \bar{\mathbf{x}}_2\big) \tag{3.69}$$

Therefore, (3.66) becomes:

$$w_0 = \ln \tau + \frac{1}{2}\big(\bar{\mathbf{x}}_1^\top \mathbf{\Sigma}_x^{-1} \bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2^\top \mathbf{\Sigma}_x^{-1} \bar{\mathbf{x}}_2\big) \tag{3.70}$$

By comparing (3.65) and (3.70) to (3.60), it will be noted that the optimal solutions derived in the proposed procedure is equivalent to Fisher's Linear Discriminant if homoscedasticity is assumed.

3.4 Class imbalance

This subsection investigates the effect of class imbalance on the proposed and existing heteroscedastic LDA algorithms. As has already been indicated in Chapter 2, class imbalance is the scenario where the data in one class far exceeds the data in the other classes. For the two-class case, this implies that $\pi_1 \gg \pi_2$ or $\pi_1 \ll \pi_2$, since the prior probabilities π_1 and π_2 are often estimated empirically from the cardinality of the data in each class. By defining τ as

$$\tau = \frac{\pi_2}{\pi_1}, \tag{3.71}$$

the limiting behaviour of LDA and the existing heteroscedastic LDA procedures can studied as τ tends towards 0 or ∞.

3.4.1 LDA

From (2.13) and (2.14), as $\tau \to \infty$, the discriminating threshold w_0 approaches ∞. Similarly, as $\tau \to 0$, the discriminating threshold approaches $-\infty$. This tends to skew the decision rule in favour of the majority class.

3.4.2 R-HLD-2

First, it will be recalled that the existing R-HLD-2 heteroscedastic LDA procedure introduced in Chapter 2 involves solving for the optimal $\mathbf{w}$ as given by:

$$\mathbf{w} = \left[s_1\boldsymbol{\Sigma}_1 + s_2\boldsymbol{\Sigma}_2\right]^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{3.72}$$

by obtaining the optimal values of s_1 and s_2 via systematic trial and error.

If this solution in (3.72) is compared to the optimal solution derived in (3.25), it will be noted that:

$$s_2 = \frac{z_2}{\sigma_2} \quad \text{and} \quad s_1 = -\frac{z_1}{\sigma_1} \tag{3.73}$$

1. Case 1: $\pi_1 \ll \pi_2$. Then, from the definition of τ in (3.71), $\tau \to \infty$. As $\tau \to \infty$, notice that in (3.30), $w_0 = w_0^+ \to \infty$, in which case both z_2 and z_1 approach ∞ as can be seen from (3.9). Therefore, if the Bayes error is to be minimised in the event of a class imbalance such that $\pi_2 \gg \pi_1$, s_2 approaches ∞ and s_1 tends toward $-\infty$.

2. Case 2: $\pi_1 \gg \pi_2$. Then, from the definition of τ in (3.71), $\tau \to 0$. As $\tau \to 0$, $2(\sigma_1^2 - \sigma_2^2)\ln(\tau\sigma_1/\sigma_2)$ approaches $-\infty$ as can be seen from (3.32). However, since the Bayes optimal threshold w_0 is supposed to be real, as is defined in the problem description in (3.1), β has to be non-negative in (3.32). Therefore, as $\tau \to 0$, $\beta \to 0$, and $w_0 = w_0^+ \to \frac{\mu_2\sigma_1^2 - \mu_1\sigma_2^2}{\sigma_1^2 - \sigma_2^2}$. By substituting this value of w_0 into z_2 and z_1 as defined in (3.9), s_2 can be shown to approach:

$$s_2 = \frac{z_2}{\sigma_2} = \frac{w_0 - \mu_2}{\sigma_2^2} = \frac{\mu_2\sigma_1^2 - \mu_1\sigma_2^2 - \mu_2(\sigma_1^2 - \sigma_2^2)}{\sigma_2^2(\sigma_1^2 - \sigma_2^2)} = \frac{\mu_2 - \mu_1}{\sigma_1^2 - \sigma_2^2} \tag{3.74}$$

while s_1 can also be shown to approach:

$$s_1 = \frac{z_1}{\sigma_1} = \frac{w_0 - \mu_1}{\sigma_1^2} = \frac{\mu_2\sigma_1^2 - \mu_1\sigma_2^2 - \mu_1(\sigma_1^2 - \sigma_2^2)}{\sigma_1^2(\sigma_1^2 - \sigma_2^2)} = \frac{\mu_2 - \mu_1}{\sigma_1^2 - \sigma_2^2} \tag{3.75}$$

By considering the two cases, it will be noted that for any given dataset, s_1 is constrained in the interval $\left(-\infty, \frac{\mu_2-\mu_1}{\sigma_1^2-\sigma_2^2}\right)$ while s_2 is constrained in the interval $\left(\frac{\mu_2-\mu_1}{\sigma_1^2-\sigma_2^2}, \infty\right)$. This limiting behaviour makes it difficult to find the optimal values of s_2 and s_1 in the unbounded interval $(-\infty, \infty)$ by trial and error for an arbitrary dataset with class imbalance in the R-HLD-2 heteroscedastic LDA procedure, unless a very large number of trials are performed.

3.4.3 R-HLD-1 and C-HLD

Moreover, if the optimal solution obtained in (3.25) as well as the optimal threshold given in (3.30) are multiplied proportionally by:

$$c = \frac{\sigma_1 z_2 - \sigma_2 z_1}{\sigma_1\sigma_2}, \tag{3.76}$$

(3.25) may then be expressed as:

$$\mathbf{w} = \left[s\boldsymbol{\Sigma}_2 + (1-s)\boldsymbol{\Sigma}_1\right]^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{3.77}$$

where

$$s = -\frac{\sigma_2 z_1}{\sigma_1 z_2 - \sigma_2 z_1} = \frac{1}{1 - \frac{\sigma_1 z_2}{\sigma_2 z_1}} = \frac{1}{1 - \frac{\sigma_1^2}{\sigma_2^2}\frac{w_0 - \mu_2}{w_0 - \mu_1}} \tag{3.78}$$

Due to (3.36) c is non-negative and hence the discrimination criterion given by (3.1) is not changed.

Notice that (3.77) is then in the form of the R-HLD-1 and C-HLD heteroscedastic LDA solutions described in Chapter 2. However, the R-HLD-1 procedure obtains s by systematic trial and error, unlike in (3.78), while s is varied between 0 and 1 to obtain the optimal value in the C-HLD procedure.

By analysing the behaviour of the optimal s as given by (3.78) under class imbalance, the performance of C-HLD and R-HLD-1 under class imbalance can be understood.

1. Case 1: $\pi_1 \ll \pi_2$. Then, from the definition of τ in (3.71), $\tau \to \infty$. As $\tau \to \infty$, notice that in (3.30), $w_0 = w_0^+ \to \infty$. Then,

$$\lim_{\tau\to\infty} s = \frac{1}{1 - \frac{\sigma_1^2}{\sigma_2^2}} = \frac{\sigma_2^2}{\sigma_2^2 - \sigma_1^2} \tag{3.79}$$

A consequence of the inequality of (3.36) which is necessary for the minimisation of the Bayes error is that, as $\tau \to \infty$, $\sigma_2 \leq \sigma_1$. Thus, the limit shown in (3.79) tends to be negative.

2. Case 2: $\pi_1 \gg \pi_2$. Then, from the definition of τ in (3.71), $\tau \to 0$. As $\tau \to 0$, $2(\sigma_1^2 - \sigma_2^2)\ln(\tau\sigma_1/\sigma_2)$ approaches $-\infty$ as can be seen from (3.32). However, since the Bayes optimal threshold w_0 is supposed to be real, as is defined in the problem description in (3.1), β has to be non-negative in (3.32). Therefore, as $\tau \to 0$, $\beta \to 0$, and $w_0 = w_0^+ \to \frac{\mu_2\sigma_1^2 - \mu_1\sigma_2^2}{\sigma_1^2 - \sigma_2^2}$. Once again, by substituting this value of w_0 in (3.78), the limit of s as $\tau \to \infty$ can be shown to approach ∞.

Since s tends to be negative in Case 1 as well as approaching ∞ in Case 2, clearly the C-HLD procedure that constrains s to the interval $[0, 1]$ yields solutions that are only locally optimal in the interval $[0, 1]$. Moreover, since s is constrained to this interval, the threshold w_0 computed in step 3 of the C-HLD procedure in (2.33) (page 15), is a convex combination of the two projected means μ_1 and μ_2. Therefore, the discriminating threshold is always bounded between μ_1 and μ_2, even when it ought to approach $\pm\infty$ under class imbalance. Thus, the C-HLD heteroscedastic LDA procedure tends to be suboptimal in terms of the error rate under class imbalance, when s falls outside the interval $[0, 1]$.

Also, due to the fact that s is unbounded in the limit of τ, it is tedious to obtain the optimal value of s for an arbitrary dataset in the R-HLD-1 procedure, except when one runs a very large number of trials.

3.4.4 A dynamic linear model

Often, under class imbalance, the misclassification rate (which weights the false positive and false negative rates equally) may not be a preferable evaluation metric. For instance, in credit card fraud prediction, it is less costly to wrongly flag genuine transactions as fraudulent than to incorrectly classify fraudulent transactions as genuine. Thus, if it is assumed that genuine transactions are positive examples while fraudulent transactions are negative examples, minimising the false positive rate is considered more important than minimising the false negative rate in this scenario.

Most classifiers tend to perform poorly under class imbalance in terms of detecting the minority class, and LDA is no exception. A common approach to dealing with unbalanced data involves rebalancing the dataset by procedures such as random oversampling, random undersampling and SMOTE [126, 127]. However, it is known that rebalancing the data does not guarantee a better performance in LDA [12]. This is due to the fact that LDA, unlike support vector machine (SVM) or logistic regression, is a generative classifier that attempts to learn the model that generates the data. Specifically, LDA relies on knowledge

of the true class prior probabilities; if these probabilities are not known *a priori*, then they are best estimated from the empirical distribution of the classes in the dataset. Rebalancing the dataset therefore removes this information about the prior probabilities.

Another common approach to handling class imbalance is to bias the discriminating threshold so that more minority samples are detected [126]. Certainly, if the minority class is considered as the positive class, then shifting the discriminating threshold in such a way as to improve the correct classification of more positive samples, i.e., the true positive rate (TPR) will necessarily increase the majority negative samples that are wrongly classified as positive, i.e., increase the false positive rate (FPR). In this case, a measure of the robustness of the classifier is the level of FPR–TPR trade-off it is able to provide, such as is measured by the area under the receiver operating characteristics (ROC) curve, or AUC. Thus, a large AUC generally indicates a robust classifier performance under class imbalance.

For a given binary classifier, varying the discriminating threshold w_0 has the effect of putting more emphasis on either the false positive rate or the false negative rate, depending on the specific application of the classifier. For a lot of classifiers such as the SVM or LDA under equal covariance (3.60), the discriminating threshold w_0 is independent of the weight vector $\mathbf{w}$, and therefore varying w_0 does not change $\mathbf{w}$. However, under unequal covariance, the Bayes optimal weight vector is a function of w_0 as can be noted in (3.22). Therefore, if a different threshold w_0 is to be used, other than the optimal threshold w_0, then (3.25) is no longer optimal, as it is obtained using the optimal threshold in (3.23). In such a case, for any arbitrary w_0 chosen in such a way as to emphasise the false positive rate or false negative rate, the optimal $\mathbf{w}$ can be obtained from (3.22) as follows:

$$\mathbf{w} = \left(\frac{\pi_2 z_2}{\sigma_2^2} e^{-\frac{z_2^2}{2}} \boldsymbol{\Sigma}_2 - \frac{\pi_1 z_1}{\sigma_1^2} e^{-\frac{z_1^2}{2}} \boldsymbol{\Sigma}_1\right)^{-1} \left(\frac{\pi_1}{\sigma_1} e^{-\frac{z_1^2}{2}} \bar{\mathbf{x}}_1 - \frac{\pi_2}{\sigma_2} e^{-\frac{z_2^2}{2}} \bar{\mathbf{x}}_2\right) \tag{3.80}$$

If both sides of (3.22) were projected onto $\mathbf{w}$, the following is obtained:

$$\mathbf{w}^\top \left(\frac{\pi_2 z_2}{\sigma_2^2} e^{-z_2^2/2} \boldsymbol{\Sigma}_2 - \frac{\pi_1 z_1}{\sigma_1^2} e^{-z_1^2/2} \boldsymbol{\Sigma}_1\right) \mathbf{w} = \mathbf{w}^\top \left(\frac{\pi_1}{\sigma_1} e^{-z_1^2/2}\right) \bar{\mathbf{x}}_1 - \mathbf{w}^\top \left(\frac{\pi_2}{\sigma_2} e^{-z_2^2/2}\right) \bar{\mathbf{x}}_2. \tag{3.81}$$

The above result can be simplified as follows:

$$\pi_2 z_2 e^{-\frac{z_2^2}{2}} - \pi_1 z_1 e^{-\frac{z_1^2}{2}} = \frac{\pi_1 \mu_1}{\sigma_1} e^{-\frac{z_1^2}{2}} - \frac{\pi_2 \mu_2}{\sigma_2} e^{-\frac{z_2^2}{2}}. \tag{3.82}$$

$$\pi_2 e^{-\frac{z_2^2}{2}} \left(z_2 + \frac{\mu_2}{\sigma_2}\right) = \pi_1 e^{-\frac{z_1^2}{2}} \left(z_1 + \frac{\mu_1}{\sigma_1}\right) \tag{3.83}$$

$$\frac{\pi_2 w_0}{\sigma_2} e^{-\frac{z_2^2}{2}} = \frac{\pi_1 w_0}{\sigma_1} e^{-\frac{z_1^2}{2}} \tag{3.84}$$

Substituting (3.84) into (3.80) then results in:

$$\mathbf{w} = \left(\frac{z_2}{\sigma_2} \boldsymbol{\Sigma}_2 - \frac{z_1}{\sigma_1} \boldsymbol{\Sigma}_1 \right)^{-1} (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{3.85}$$

Still, (3.80) has no closed form solution (since $\sigma_1, \sigma_2, z_1, z_2$ are themselves functions of $\mathbf{w}$), and it must be solved iteratively, for example using gradient descent.

By optimising (3.85) via the gradient descent procedure described in Algorithm 2, a dynamic model of the weight vector may be obtained as a function of a given threshold value w_0 as:

$$\mathbf{w}(w_0) = \left(\frac{z_2'}{\sigma_2^*} \boldsymbol{\Sigma}_2 - \frac{z_1'}{\sigma_1^{*2}} \boldsymbol{\Sigma}_1 \right)^{-1} (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{3.86}$$

where

$$\sigma_1^{*2} = \mathbf{w}^{*\top} \boldsymbol{\Sigma}_1 \mathbf{w}^*, \quad \sigma_2^{*2} = \mathbf{w}^{*\top} \boldsymbol{\Sigma}_2 \mathbf{w}^*, \quad z_1' = \frac{w_0 - \mathbf{w}^{*\top} \bar{\mathbf{x}}_1^*}{\sigma_1^*}, \quad z_2' = \frac{w_0 - \mathbf{w}^{*\top} \bar{\mathbf{x}}_2^*}{\sigma_2^*} \tag{3.87}$$

and $\mathbf{w}^*$ is the optimal weight vector that the gradient descent procedure in Algorithm 2 yields. This model is referred to as dynamic GLD (D-GLD).

The procedure described in (3.86) is dynamic in the sense that it optimises the weight vector $\mathbf{w}$ depending on the specified discriminating threshold w_0. Therefore, it is more generalised than the existing heteroscedastic LDA procedures, as it can easily be employed in different applications with different goals in terms of minimising the probability of false alarm or the probability of missed detection.

In terms of the area under the receiver operating characteristics curve therefore, since $\mathbf{w}$ is optimised for any given w_0 in the dynamic linear model, the true positive rate and false positive rate obtained at every threshold setting are equally optimal. This is unlike the existing heteroscedastic LDA procedures where a single weight vector $\mathbf{w}$ (as given in (3.72) or (3.77) which implicitly employs the optimal value of w_0) is used for all other threshold settings. Employing a constant $\mathbf{w}$ for any value of w_0 violates the result in (3.80) which indicates that under unequal covariance, the Bayes-optimal $\mathbf{w}$ is a non-linear function of w_0.

3.5 The kernel formulation

If the classes $\mathcal{C}_1$ and $\mathcal{C}_2$ are not linearly separable in the space $\mathcal{X}$, which is the original feature space of the feature vector $\mathbf{x}$, accounting for heteroscedasticity may still yield unsatisfactory results. It would thus be

appropriate to first map the training data $\mathcal{X}$ via a transformation $\phi(\mathbf{x})$ into a higher dimensional feature space where linear separability would be guaranteed.

After this transformation, the aim, as before, is to find a linear discriminant of the form $\mathbf{w}^\top\phi(\mathbf{x})-w_0 = 0$, i.e. a vector of weights $\mathbf{w}$ and a threshold w_0 such that for a given vector $\mathbf{x}$:

$$\mathcal{C}^*(\mathbf{x}) = \begin{cases} \mathcal{C}_1 & \text{if} \quad y = \mathbf{w}^\top\phi(\mathbf{x}) \geq w_0 \\ \mathcal{C}_2 & \text{if} \quad y = \mathbf{w}^\top\phi(\mathbf{x}) < w_0 \end{cases} \tag{3.88}$$

Let the weight $\mathbf{w}$ be given by the relation:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i \phi(\mathbf{x}_i) \tag{3.89}$$

where n is the cardinality of the training dataset $\mathcal{X}$, and α_i for $i \in \{1, ..., N\}$ are unknown. Then, the random variable y is given by:

$$y = \mathbf{w}^\top\phi(\mathbf{x}) = \left(\sum_{i=1}^{N} \alpha_i \phi(\mathbf{x}_i)\right)^\top \phi(\mathbf{x}) \tag{3.90}$$

$$y = \sum_{i=1}^{N} \alpha_i \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}) \tag{3.91}$$

$$y = \sum_{i=1}^{N} \alpha_i \mathcal{K}(\mathbf{x}_i, \mathbf{x}) \tag{3.92}$$

where $\mathcal{K}(\mathbf{x}_i, \mathbf{x}) = \phi(\mathbf{x}_i)^\top\phi(\mathbf{x})$, as introduced in Section 2.1.1.

Since the minimisation of the error probability p_e as given in (3.8) assumes a normal distribution in the two classes, it is assumed in this formulation also that the data in the transformed space $\mathcal{Y}$ is normally distributed, so that the variable y in turn is normally distributed as follows:

$$\begin{aligned} \mathcal{C}_1 &: y \sim \mathcal{N}(\mu_1, \sigma_1^2) \\ \mathcal{C}_1 &: y \sim \mathcal{N}(\mu_2, \sigma_2^2) \end{aligned} \tag{3.93}$$

where

$$\mu_1 = \mathbf{w}^\top\bar{\mathbf{m}}_1 \quad \mu_2 = \mathbf{w}^\top\bar{\mathbf{m}}_2 \quad \sigma_1^2 = \mathbf{w}^\top\mathbf{K}_1\mathbf{w} \quad \sigma_2^2 = \mathbf{w}^\top\mathbf{K}_2\mathbf{w}. \tag{3.94}$$

Here, $\bar{\mathbf{m}}_1, \bar{\mathbf{m}}_2$ are the sample means for classes $\mathcal{C}_1$ and $\mathcal{C}_2$, and $\mathbf{K}_1, \mathbf{K}_2$ are the sample covariance

matrices for classes $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively, all given as:

$$\bar{\mathbf{m}}_1 = \frac{1}{n_1}\sum_{i=1}^{n_1}\phi(\mathbf{x}_i), \quad \bar{\mathbf{m}}_2 = \frac{1}{n_2}\sum_{i=1}^{n_2}\phi(\mathbf{x}_i) \tag{3.95}$$

$$\mathbf{K}_1 = \frac{1}{n_1-1}\sum_{i=1}^{n_1}\left(\phi(\mathbf{x}_i) - \frac{1}{n_1}\sum_{j=1}^{n_1}\phi(\mathbf{x}_j)\right)\left(\phi(\mathbf{x}_i) - \frac{1}{n_1}\sum_{k=1}^{n_1}\phi(\mathbf{x}_k)\right)^\top \tag{3.96}$$

$$\mathbf{K}_2 = \frac{1}{n_2-1}\sum_{i=1}^{n_2}\left(\phi(\mathbf{x}_i) - \frac{1}{n_2}\sum_{j=1}^{n_2}\phi(\mathbf{x}_j)\right)\left(\phi(\mathbf{x}_i) - \frac{1}{n_2}\sum_{k=1}^{n_2}\phi(\mathbf{x}_k)\right)^\top \tag{3.97}$$

with n_1 and n_2 being the number of training points in classes $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively.

But for the unknown transformation $\phi(\mathbf{x})$, it would have been straightforward in this kernel formulation to minimise the Bayes error using Algorithm 2, by replacing $\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ with $\bar{\mathbf{m}}_1, \bar{\mathbf{m}}_2, \mathbf{K}_1$ and $\mathbf{K}_2$ respectively. However, by employing the kernel trick, an explicit representation of $\phi(\mathbf{x})$ is rendered unnecessary, since the formulation is reduced to an inner product between two vectors in $\mathcal{Y}$, which is the kernel $\mathcal{K}$. Thus, μ_1, μ_2, σ_1 and σ_2 are derived in terms of the kernel $\mathcal{K}$.

$$\mu_1 = \mathbf{w}^\top\bar{\mathbf{m}}_1 = \left(\sum_{i=1}^{n}\alpha_i\phi(\mathbf{x}_i)\right)^\top\left(\frac{1}{n_1}\sum_{j=1}^{n_1}\phi(\mathbf{x}_j)\right) \tag{3.98}$$

$$\mu_1 = \frac{1}{n_1}\sum_{i=1}^{n}\sum_{j=1}^{n_1}\alpha_i\phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j) \tag{3.99}$$

$$\mu_1 = \frac{1}{n_1}\sum_{i=1}^{n}\sum_{j=1}^{n_1}\alpha_i\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) \tag{3.100}$$

$$\mu_1 = \boldsymbol{\alpha}^\top\bar{\mathbf{b}}_1 \tag{3.101}$$

where $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_n]$ and

$$\bar{\mathbf{b}}_1 = \frac{1}{n_1}\mathbf{M}_1\mathbf{1}_{n_1} \tag{3.102}$$

with $\mathbf{1}_{n_1}$ being an n_1-dimensional vector with all entries being 1, and $\mathbf{M}_1$ an $n \times n_1$-dimensional matrix with $\mathbf{M}_1(i,j) = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_j)$ for all $\mathbf{x}_i \in \mathcal{X}$ and all $\mathbf{x}_j \in \mathcal{D}_1$. Similarly,

$$\mu_2 = \frac{1}{n_2}\sum_{i=1}^{n}\sum_{j=1}^{n_2}\alpha_i\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) \tag{3.103}$$

$$\mu_2 = \boldsymbol{\alpha}^\top\bar{\mathbf{b}}_2 \tag{3.104}$$

where

$$\bar{\mathbf{b}}_2 = \frac{1}{n_2}\mathbf{M}_2\mathbf{1}_{n_2} \tag{3.105}$$

with $\mathbf{1}_{n_2}$ being an n_2-dimensional vector with all entries being 1, and $\mathbf{M}_2$, an $n \times n_2$-dimensional matrix with $\mathbf{M}_2(i,j) = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$ for all $\mathbf{x}_i \in \mathcal{X}$ and all $\mathbf{x}_j \in \mathcal{D}_2$.

Note that $\mathbf{K}_1$ as given by (3.96) can be expanded as follows:

$$\mathbf{K}_1 = \frac{1}{n_1 - 1}\sum_{i=1}^{n_1}\left[\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^\top - \frac{1}{n_1}\phi(\mathbf{x}_i)\sum_{k=1}^{n_1}\phi(\mathbf{x}_k)^\top - \frac{1}{n_1}\sum_{j=1}^{n_1}\phi(\mathbf{x}_j)\phi(\mathbf{x}_i)^\top + \frac{1}{n_1^2}\sum_{j=1}^{n_1}\phi(\mathbf{x}_j)\sum_{k=1}^{n_1}\phi(\mathbf{x}_k)^\top\right] \tag{3.106}$$

$$\mathbf{K}_1 = \frac{1}{n_1 - 1}\Bigg[\sum_{i=1}^{n_1}\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^\top - \frac{1}{n_1}\sum_{i=1}^{n_1}\phi(\mathbf{x}_i)\sum_{k=1}^{n_1}\phi(\mathbf{x}_k)^\top - \frac{1}{n_1}\sum_{j=1}^{n_1}\phi(\mathbf{x}_j)\sum_{i=1}^{n_1}\phi(\mathbf{x}_i)^\top + \frac{n_1}{n_1^2}\sum_{j=1}^{n_1}\phi(\mathbf{x}_j)\sum_{k=1}^{n_1}\phi(\mathbf{x}_k)^\top\Bigg] \tag{3.107}$$

$$\mathbf{K}_1 = \frac{1}{n_1 - 1}\left[\sum_{i=1}^{n_1}\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^\top - \frac{1}{n_1}\sum_{i=1}^{n_1}\phi(\mathbf{x}_i)\sum_{k=1}^{n_1}\phi(\mathbf{x}_k)^\top\right] \tag{3.108}$$

so that

$$\sigma_1^2 = \mathbf{w}^\top\mathbf{K}_1\mathbf{w} = \frac{1}{n_1 - 1}\left(\sum_{m=1}^{n}\alpha_m\phi(\mathbf{x}_m)\right)^\top\left[\sum_{i=1}^{n_1}\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^\top - \frac{1}{n_1}\sum_{i=1}^{n_1}\phi(\mathbf{x}_i)\sum_{k=1}^{n_1}\phi(\mathbf{x}_k)^\top\right]\left(\sum_{t=1}^{n}\alpha_t\phi(\mathbf{x}_t)\right) \tag{3.109}$$

$$\sigma_1^2 = \frac{1}{n_1 - 1}\Bigg[\sum_{m=1}^{n}\alpha_m\phi(\mathbf{x}_m)^\top\sum_{i=1}^{n_1}\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^\top\sum_{t=1}^{n}\alpha_t\phi(\mathbf{x}_t) - \frac{1}{n_1}\sum_{m=1}^{n}\alpha_m\phi(\mathbf{x}_m)^\top\sum_{i=1}^{n_1}\phi(\mathbf{x}_i)\sum_{k=1}^{n_1}\phi(\mathbf{x}_k)^\top\sum_{t=1}^{n}\alpha_n\phi(\mathbf{x}_t)\Bigg] \tag{3.110}$$

$$\sigma_1^2 = \frac{1}{n_1 - 1}\Bigg[\sum_{m=1}^{n}\sum_{i=1}^{n_1}\alpha_m\phi(\mathbf{x}_m)^\top\phi(\mathbf{x}_i)\sum_{t=1}^{n}\alpha_t\phi(\mathbf{x}_i)^\top\phi(\mathbf{x}_t) - \frac{1}{n_1}\sum_{m=1}^{n}\sum_{i=1}^{n_1}\alpha_m\phi(\mathbf{x}_m)^\top\phi(\mathbf{x}_i)\sum_{k=1}^{n_1}\sum_{t=1}^{n}\alpha_t\phi(\mathbf{x}_k)^\top\phi(\mathbf{x}_t)\Bigg] \tag{3.111}$$

$$\sigma_1^2 = \frac{1}{n_1 - 1}\left(\boldsymbol{\alpha}^\top\mathbf{M}_1\mathbf{M}_1^\top\boldsymbol{\alpha} - \frac{1}{n_1}\boldsymbol{\alpha}\mathbf{M}_1\mathbf{1}_n\mathbf{1}_n^\top\mathbf{M}_1^\top\boldsymbol{\alpha}\right) \tag{3.112}$$

$$\sigma_1^2 = \boldsymbol{\alpha}^\top\mathbf{C}_1\boldsymbol{\alpha} \tag{3.113}$$

where

$$\mathbf{C}_1 = \frac{1}{n_1 - 1}\left(\mathbf{M}_1\mathbf{I}_{n_1}\mathbf{M}_1^\top - \frac{1}{n_1}\mathbf{M}_1\mathbf{1}_{n_1\times n_1}\mathbf{M}_1^\top\right) \tag{3.114}$$

with $\mathbf{I}_{n_1}$ being an n_1-sized identity matrix, and $\mathbf{1}_{n_1\times n_1}$ being an n_1-sized square matrix with all entries being 1.

Similarly, it can be shown that:

$$\sigma_2^2 = \boldsymbol{\alpha}^\top\mathbf{C}_2\boldsymbol{\alpha} \tag{3.115}$$

where

$$\mathbf{C}_2 = \frac{1}{n_2 - 1}\left(\mathbf{M}_2\mathbf{I}_{n_2}\mathbf{M}_2^\top - \frac{1}{n_2}\mathbf{M}_2\mathbf{1}_{n_2\times n_2}\mathbf{M}_2^\top\right) \tag{3.116}$$

with $\mathbf{I}_{n_2}$ being an n_2-sized identity matrix, and $\mathbf{1}_{n_2\times n_2}$ being an n_2-sized square matrix with all entries being 1.

Having obtained μ_1, μ_2, σ_1 and σ_2 in terms of the kernel function $\mathcal{K}$, the gradient descent procedure in Algorithm 2 is then readily applicable to minimise the Bayes error in the transformed space $\mathcal{Y}$, where the gradient of p_e w.r.t. $\boldsymbol{\alpha}$ is given by:

$$\frac{\partial p_e}{\partial \boldsymbol{\alpha}} = \frac{\pi_1}{\sqrt{2\pi}}\left[-e^{-\frac{z_1^2}{2}}\left(\frac{\sigma_1\bar{\mathbf{b}}_1 + z_1\mathbf{C}_1\boldsymbol{\alpha}}{\sigma_1^2}\right)\right] + \frac{\pi_2}{\sqrt{2\pi}}\left[e^{-\frac{z_2^2}{2}}\left(\frac{\sigma_2\bar{\mathbf{b}}_2 + z_2\mathbf{C}_2\mathbf{w}}{\sigma_2^2}\right)\right]. \tag{3.117}$$

This is done by replacing $\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2$ and $\mathbf{w}$ with $\bar{\mathbf{b}}_1, \bar{\mathbf{b}}_2, \mathbf{C}_1, \mathbf{C}_2$ and $\boldsymbol{\alpha}$ respectively as indicated in Algorithm 4.

The output of the gradient descent procedure would be the vector $\boldsymbol{\alpha}$ and w_0 so that for a given vector $\mathbf{x}$, one predicts class $\mathcal{C}^*(\mathbf{x})$ based on the following:

$$\mathcal{C}^*(\mathbf{x}) = \begin{cases} \mathcal{C}_1 & \text{if} \quad y = \sum_{i=1}^n \alpha_i\mathcal{K}(\mathbf{x}_i, \mathbf{x}) \geq w_0 \\ \mathcal{C}_2 & \text{if} \quad y = \sum_{i=1}^n \alpha_i\mathcal{K}(\mathbf{x}_i, \mathbf{x}) < w_0 \end{cases} \tag{3.121}$$

Moreover, the computation associated in each iteration of the gradient descent procedure is cheap, and no matrix inversion is need for the minimisation of the Bayes error, unlike in the Kernel Fisher's discriminant or the kernel adaptations of the existing heteroscedastic LDA approaches.

While the main utility of the kernel function is to guarantee linear separability of the two classes in the transformed space $\mathcal{Y}$, the kernel may also be chosen in such a way that it transforms non-normal data into one that is nearly-normal, so that the assumption of normality used in this formulation may be satisfied.

Algorithm 4 Kernel GLD (K-GLD)

1: Input: $\mathcal{D}_1$ and $\mathcal{D}_2$

2: Choose a kernel function $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$

3: Evaluate $\mathbf{M}_1, \mathbf{M}_2$

4: Evaluate $\bar{\mathbf{b}}_1, \bar{\mathbf{b}}_2, \mathbf{C}_1, \mathbf{C}_2$

5: Initialise $\boldsymbol{\alpha}$: $\boldsymbol{\alpha} \leftarrow (\beta_2\mathbf{C}_2 - \beta_1\mathbf{C}_1)^{-1}(\bar{\mathbf{b}}_1 - \bar{\mathbf{b}}_2)$,

6: Evaluate $\mu_1, \mu_2, \sigma_1^2, \sigma_2^2$

7: Solve for $w_0 = w_0^+$ from

$$w_0^+ = \frac{\mu_2\sigma_1^2 - \mu_1\sigma_2^2 + \sigma_1\sigma_2\sqrt{(\mu_1-\mu_2)^2 + 2(\sigma_1^2-\sigma_2^2)\ln\left(\frac{\tau\sigma_1}{\sigma_2}\right)}}{\sigma_1^2 - \sigma_2^2} \tag{3.118}$$

8: **while** Stopping criteria are not satisfied **do**

9: Evaluate z_1, z_2

10: Evaluate the gradient of p_e w.r.t. $\boldsymbol{\alpha}$ as given by:

$$\frac{\partial p_e}{\partial \boldsymbol{\alpha}} = \frac{\pi_1}{\sqrt{2\pi}}\left[-e^{-\frac{z_1^2}{2}}\left(\frac{\sigma_1\bar{\mathbf{b}}_1 + z_1\mathbf{C}_1\boldsymbol{\alpha}}{\sigma_1^2}\right)\right] + \frac{\pi_2}{\sqrt{2\pi}}\left[e^{-\frac{z_2^2}{2}}\left(\frac{\sigma_2\bar{\mathbf{b}}_2 + z_2\mathbf{C}_2\boldsymbol{\alpha}}{\sigma_2^2}\right)\right]. \tag{3.119}$$

11: Evaluate the gradient of p_e w.r.t. w_0 as given by:

$$\frac{\partial p_e}{\partial w_0} = \frac{\pi_1}{\sqrt{2\pi}}\left(\frac{1}{\sigma_1}e^{-\frac{z_1^2}{2}}\right) - \frac{\pi_2}{\sqrt{2\pi}}\left(\frac{1}{\sigma_2}e^{-\frac{z_2^2}{2}}\right) \tag{3.120}$$

12: Update $\boldsymbol{\alpha}$ as $\boldsymbol{\alpha}^{i+1} \leftarrow \boldsymbol{\alpha}^i - \gamma\frac{\partial p_e}{\partial \boldsymbol{\alpha}^i}$

13: Update w_0 as $w_0^{i+1} \leftarrow w_0^i - \gamma\frac{\partial p_e}{\partial w_0^i}$

14: Evaluate $\mu_1, \mu_2, \sigma_1, \sigma_2$

15: **end while**

3.6 Experimental validation

Three different experiments are described in this section. The first experiment is aimed at investigating the performance of the proposed GLD algorithm for balanced datasets, while the second experiment examines the effect of class imbalance on the proposed algorithm as well as the existing heteroscedastic LDA procedures. The third experiment looks at the performance of the GLD procedure when kernalised to learn non-linear decision boundaries.

3.6.1 Balanced datasets

GLD is validated on two artificial datasets denoted by D1 and D2, and on ten real-world datasets taken from the University of California, Irvine (UCI) Machine Learning Repository. These datasets are shown in Table 3.1 and cut across a wide range of applications including handwriting recognition, medical diagnosis, remote sensing and spam filtering. D1 and D2 are normally distributed with different covariance matrices. For D1, 1000 samples are generated each for class $\mathcal{C}_1$ and class $\mathcal{C}_2$ using the following Gaussian parameters:

$$\begin{aligned}
\bar{\mathbf{x}}_2 &= [3.86, 3.10, 0.84, 0.84, 1.64, 1.08, 0.26, 0.01]^\top, \\
\mathbf{\Sigma}_2 &= \text{diag}(8.41, 12.06, 0.12, 0.22, 1.49, 1.77, 0.35, 2.73) \\
\bar{\mathbf{x}}_1 &= \bar{\mathbf{x}}_2 - 0.3, \quad \mathbf{\Sigma}_1 = \mathbf{I}
\end{aligned} \tag{3.122}$$

Similarly, for D2, 1000 samples are generated each for class $\mathcal{C}_1$ and class $\mathcal{C}_2$ using the following Gaussian parameters:

$$\begin{aligned}
\bar{\mathbf{x}}_2 &= [-1.5, -0.75, 0.75, 1.5]^\top, \\
\mathbf{\Sigma}_2 &= \text{diag}(0.25, 0.75, 1.25, 1.75) \\
\bar{\mathbf{x}}_1 &= \bar{\mathbf{x}}_2 - 0.75, \quad \mathbf{\Sigma}_1 = \mathbf{I}
\end{aligned} \tag{3.123}$$

The above Gaussian parameters are slightly modified from the two class data used by Fukunaga [8] and Xue [12] in order to make the sample means less separated.

For each dataset in Table 3.1, 10-fold cross validation is repeated for 20 different trials. On each training dataset, the average minimum Bayes error achievable by the proposed GLD algorithm is obtained. The recursive GLD (R-GLD) optimisation method is used in these experiments. If there are more than two classes, the OvO multiclass strategy is used, and then the mean Bayes error over all $K(K-1)/2$

Table 3.1: List and characteristics of datasets
K is the number of classes, d is the dimensionality of the dataset, and n is the number of data points (or feature vectors) in the dataset.

Dataset	Label	n	d	K
D1	(a)	2000	8	2
D2	(b)	2000	4	2
Liver	(c)	345	6	2
Shuttle	(d)	58000	9	7
Vowels	(e)	990	10	11
Zernike Moments	(f)	2000	47	10
Image Segmentation (Statlog)	(g)	2310	19	7
Spambase	(h)	4601	37	2
Wine Quality (White)	(i)	4898	11	7
Optical Digits	(j)	5620	64	10
Satellite (Statlog)	(k)	6435	36	6
Letters	(l)	20000	16	26

discriminants is calculated. The performance of the R-GLD is compared with the original LDA as well as the heteroscedastic LDA procedures by Fukunaga [8], Anderson [71] and Marks [70] in terms of the Bayes error 3.8. For the sake of brevity, these three heteroscedastic LDA algorithms are denoted by the annotations earlier introduced: C-HLD, R-HLD-1 and R-HLD-2 respectively. These results are shown in Table 3.2. Quadratic discriminant analysis (QDA) is not included in this comparison because the Bayes error in (3.8) is only defined for linear classification.

Moreover, for each of the test datasets, the average classification accuracy for each of LDA, C-HLD, R-HLD-1, R-HLD-2, GLD and GLD with local neighbourhood search (LNS) (Algorithm 3) are also evaluated. The performance of these LDA approaches are also compared to support vector machines (SVMs). These results are shown in Table 3.3, while the average training times of the algorithms are shown in Table 3.4 and pictorially in Figure 3.1.

The prior probabilities are estimated based on the relative frequencies of the data in each class in the dataset, and the stopping criterion for the R-GLD is thus: the algorithm is stopped if the Bayes error p_e is less than or equal to 10^{-6}, or else it is terminated after 20 iterations, if p_e is not within the 10^{-6} tolerance; the solution corresponding to the minimum p_e is then chosen. Also, for the LNS procedure, each vector element is perturbed by 10% of its absolute value, i.e. $\delta\tilde{w}_i = 0.1|\tilde{w}_i|$, and the procedure is run for R=1000 iterations, terminating prematurely if $r_{max} = 0.1R_{max}$ (page 42). A step size of $\Delta s = 0.001$ is used for the C-HLD algorithm, and 1000 trials are run for R-HLD-1 and R-HLD-2. All the parameter

settings used in the experiments are optimised via cross-validation. Note that if the sample covariance matrix is singular, the Moore-Penrose pseudo-inverse is used.

Results and Discussion

Table 3.2: Average Bayes error (%)
Best values are in bold. The values with asterisk (*) are those that are statistically indiscernible from the best values based on the Wilcoxon's signed rank test at a significance level of 0.01.

Dataset	LDA	C-HLD	R-HLD-1	R-HLD-2	R-GLD
(a)	0.0397	0.0382	0.0383	0.0361*	**0.0360**
(b)	0.0774	0.0749	0.0749	0.0740*	**0.0739**
(c)	0.9981	**0.9838**	**0.9838**	**0.9838**	**0.9838**
(d)	**0.0001**	**0.0001**	**0.0001**	**0.0001**	**0.0001**
(e)	0.0339	**0.0326**	**0.0326**	**0.0326**	**0.0326**
(f)	0.0054	0.0051	**0.0048**	**0.0048**	0.0050*
(g)	0.0037	**0.0029**	**0.0029**	**0.0029**	**0.0029**
(h)	0.0253	**0.022**8	**0.0228**	**0.0228**	**0.0228**
(i)	0.0162	0.0201	0.0156*	0.0155*	**0.0154**
(j)	**0.0002**	**0.0002**	**0.0002**	**0.0002**	**0.0002**
(k)	0.0046	**0.0039**	**0.0039**	**0.0039**	**0.0039**
(l)	**0.0007**	**0.0007**	**0.0007**	**0.0007**	**0.0007**

Table 3.3: Average classification accuracy (%)
In bold for each dataset is the best values among the six LDA procedures. The values with asterisk (*) are those that are statistically indiscernible from the best values based on the Wilcoxon's signed rank test at a significance level of 0.01.

Dataset	LDA	C-HLD	R-HLD-1	R-HLD-2	R-GLD	GLD+LNS	SVM
(a)	76.00	77.18	77.00	78.48	**78.65**	78.57	77.47
(b)	76.87	77.93	77.93	78.17*	**78.37**	78.00	77.70
(c)	67.83	63.19	62.32	62.03	63.77	**68.12**	68.70
(d)	94.10	96.60	96.74	96.73	96.59	**97.91**	84.39
(e)	73.64	74.14	74.44	74.44	74.14	**75.66**	76.77
(f)	84.00	83.90	84.10	84.15	**84.80**	84.00	81.90
(g)	94.33	94.59	94.59	94.63*	94.59	**94.89**	96.15
(h)	88.76	88.29	88.26	88.15	88.26	**90.28**	85.68
(i)	53.41	46.59	53.37	53.33	53.55	**54.1**4	51.88
(j)	96.74	96.99	96.97	96.98	97.01	**97.41**	97.84
(k)	85.69	86.06	86.06	86.03	86.08	**86.65**	86.85*
(l)	81.67	81.87	81.83	81.78	81.88	**82.25**	85.39

Table 3.4: Average training time (s)
Best values are in bold.

Dataset	LDA	C-HLD	R-HLD-1	R-HLD-2	R-GLD	R-GLD+LNS	SVM
(a)	**0.001**	0.161	0.140	0.139	0.002	0.181	23.192
(b)	**0.001**	0.142	0.121	0.121	0.002	0.060	0.721
(c)	**0.001**	0.155	0.142	0.134	0.003	0.028	2.673
(d)	**0.037**	3.531	3.023	3.012	0.089	43.32	4623.138
(e)	**0.036**	11.099	9.409	9.751	0.167	2.075	1.173
(f)	**0.387**	123.662	123.649	121.906	1.955	110.694	23.126
(g)	**0.128**	37.320	37.876	37.875	0.488	2.143	21.775
(h)	**0.101**	10.437	7.729	7.474	0.753	36.83	804.574
(i)	**0.017**	4.257	3.691	3.750	0.080	5.928	914.257
(j)	**0.638**	10.099	9.358	9.171	0.915	168.190	409.380
(k)	**0.304**	18.067	17.842	17.912	0.858	13.919	311.202
(l)	**0.835**	73.050	64.022	65.414	3.245	37.202	109.232

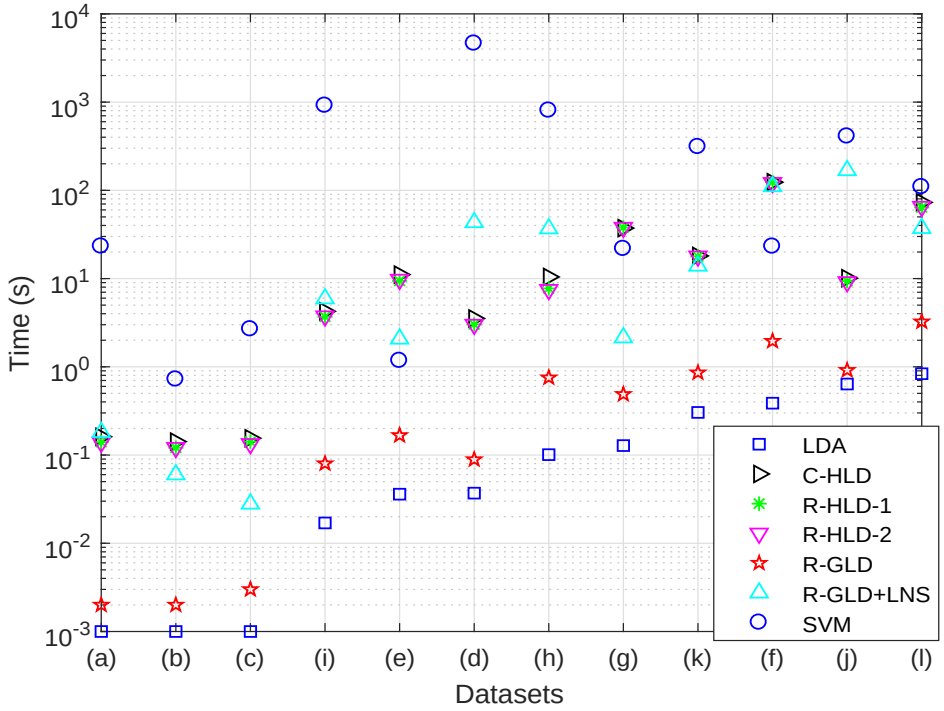

Figure 3.1: Average training time (s)

For real-world datasets, the covariance matrices of the classes are rarely equal, therefore the homoscedasticity assumption in LDA does not hold. The results in Table 3.2 confirm that LDA does not minimise the Bayes error under heteroscedasticity, as none of the datasets used has equal covariance matrices. With the exception of datasets (d), (j) and (l), where LDA achieves an equal Bayes error as the other heteroscedastic LDA approaches, LDA is outperformed by R-GLD on all remaining datasets in terms of minimising the Bayes error. The differences in the Bayes error are due to small changes in the weight vector $\mathbf{w}$ as given in (2.34) (page 15). Though this difference in Bayes error is insignificant on most of the datasets, the weight vector $\mathbf{w}$ is sensitive enough to small changes that it results in significant differences in the classification accuracies, as is seen on datasets (a), (b), (c), (d) and (f). It will be noted that the other three heteroscedastic LDA approaches algorithms achieve a performance comparable to R-GLD on all the datasets in terms of the Bayes error. However, R-HLD-1 and R-HLD-2 require a lot of trials (1000 in our experiments) in order to obtain the optimal parameters s and s_1, s_2 respectively, while C-HLD requires a step size of $\Delta s = 0.001$ which translates to 1001 trials. Consequently, the training time for these algorithms far exceed that of R-GLD, as can be seen in Table 3.4. For example, the gain in training time of R-GLD over C-HLD, R-HLD-1 and R-HLD-2 is over 62 fold for dataset (g), and about 20 fold for dataset (l). Moreover, since C-HLD, R-HLD-1 and R-HLD-2 all require matrix inversions, performing a matrix inversion for each of the 1000 trials can be a computationally demanding task especially for high-dimensional data, which have large covariance matrices. Instead, since R-GLD follows a principled optimisation procedure, the number of matrix inversions required is far lower. For example, on dataset (f), which has a dimensionality of 47, R-GLD requires over 60 times less time to train than the other heteroscedastic LDA approaches.

It is conceivable that the minimisation of the Bayes error would translate into a good performance in terms of the classification accuracy, if the normality assumption of LDA holds. For this reason, it can be seen in Table 3.3 that R-GLD achieves the best classification accuracy on datasets (a) and (b), which are generated from known normal distributions. Thus, the proposed R-GLD algorithm is particularly suited for applications with datasets that tend to be normally distributed in each class e.g., in machine fault diagnosis, or accelerometer-based human activity recognition [76], as it also requires far less training time than the existing heteroscedastic LDA approaches.

However, for datasets (c) through to (l), the classes do not have any known normal distribution. Therefore, minimising the Bayes error under the normality assumption would not necessarily result in a classifier that has the best classification accuracy, even if the difference in covariance matrices has been accounted for. For this reason, it is not surprising that LDA achieves a superior classification accuracy

than C-HLD, R-HLD-1, R-HLD-2 and R-GLD on datasets (c) and (h) as can be seen in Table 3.3. However, by searching around the neighbourhood of the R-GLD classifier, the LNS procedure is able to account for the non-normality and obtain a more robust classifier. Thus, R-GLD, together with the LNS procedure, achieves a higher classification accuracy than all the LDA approaches on all the real-world datasets (i.e. (c) to (l)) with the exception of dataset (f) which has R-GLD showing superior classification accuracy.

While the SVM outperforms the LDA approaches on half of the datasets, its training time can be rather long for large datasets. For instance, for dataset (d) which has 58000 elements, the SVM takes about 1.3 hours to train whereas R-GLD with LNS, which achieves the best classification accuracy on this dataset, takes 43 seconds to train, representing over 100 fold savings in computational time over the SVM. Similar patterns can be seen in other datasets like dataset (i), where R-GLD with LNS achieves a superior classification accuracy with over 150 times shorter training time than SVM. This suggests that for such large datasets, R-GLD with Local Neighbourhood Search is a low-complexity alternative to SVM, as it requires far less computational time than SVM.

However, since the LNS procedure involves evaluating the misclassification rate on the training set for every perturbation, the procedure does not scale well with large amounts of training data. Because of this, it is important to have a good initial solution like R-GLD, so that an early termination may be performed if there is no improvement after some number of iterations.

3.6.2 Imbalanced datasets

The effects of class imbalance is investigated experimentally in this section. While the classification accuracy may be skewed toward the majority class under class imbalance, the datasets used here are not rebalanced, since rebalancing the data results in poor estimates of the true class prior probabilities employed in LDA (see Section 1). Instead, the discriminating threshold is varied to allow for the detection of more minority samples. Thus, the AUC has been provided as the evaluation metric. The AUC provides a measure of the trade-off between the false positive rate and the true positive rate, as the discriminating threshold is varied. The proposed D-HLD model is evaluated on an artificial dataset $\mathcal{D}_3$ with the following

Gaussian parameters:

$$\begin{aligned}\bar{\mathbf{x}}_1 &= [-1.5, -0.75, 0.75, 1.5]^\top, \\ \boldsymbol{\Sigma}_1 &= \text{diag}(0.25, 0.75, 1.25, 1.75) \\ \bar{\mathbf{x}}_2 &= \bar{\mathbf{x}}_1 - \omega, \quad \boldsymbol{\Sigma}_2 = \mathbf{I}\end{aligned} \tag{3.124}$$

Here, ω controls the degree of class overlap, and is set to $\omega = 0.5$ in the experiments. $100,000$ points are generated in class $\mathcal{C}_1$ and $100,000f$ points are generated in class $\mathcal{C}_2$ to simulate an unbalanced data. Two values of f are used, i.e., $f = 2$ and $f = 10$, representing an imbalance ratio 2 and 10 respectively. This is followed by 10 trials of 10-fold cross validation. The same parameters for all algorithms used for the first experiment in Section 3.5.1 are used here. Note that the heteroscedastic LDA procedure, R-HLD-1, is not simulated here, as it has been shown to have roughly the same performance as that of R-HLD-2 in Section 3.5.1, the only difference being the number of random parameters that are controlled.

The experiment is repeated for 8 real-world datasets for which the fraction of the minority class range between 0.77% and 42.56%. The characteristics of the datasets are shown in Table 3.5.

Table 3.5: Characteristics of artificial and UCI datasets
The dimensionality of the dataset is denoted by d, while n is the number of samples in the dataset. f represents the ratio of the majority class to the minority class. Indices appended to a dataset represents the minority class, while all remaining classes form the majority class.

Dataset	d	n	Minority (%)
$\mathcal{D}_3(f = 10)$	4	$1,100,000$	9.09
$\mathcal{D}_3(f = 2)$	4	$300,000$	33.33
E-Coli-1	7	1484	42.56
Liver	6	345	42.03
Diabetes	8	768	34.90
WpBC	33	194	23.71
USPS-1	256	1484	16.70
Yeast-1	8	1484	16.44
Yeast-6	8	1484	3.44
Abalone-19	7	4177	0.77

The average AUC and training time over all 10 folds for each of the artificial and real-world datasets for our algorithm compared with LDA, QDA, R-HLD and C-HLD are then computed. These results are shown in Tables 3.6 to 3.15. Other metrics of interest provided in the results include the error rate (ER), and the balanced error rate (BER), which is defined as half the sum of the false positive and false

negative rates, i.e., BER = 0.5(FPR + FNR). While the F-measure is another common evaluation metric under class imbalance scenarios, it is not included here because it only considers the precision and recall values which do not take into account the true negatives, so that the true negatives can be allowed to vary freely without significantly changing the F-measure [88]. Additionally, the existing LDA approaches are compared in the tables with the linear SVM [93] without any enhancement by rebalancing procedures such as SMOTE.

Note that in all the tables, the best average values are in bold, while the values in asterisk are those that do not differ statistically from the best values based on Wilcoxon's signed rank test at a significance level of 0.01.

Results and discussions

Table 3.6: Artificial dataset $\mathcal{D}_3$ (f=10): AUC, Error Rate (ER), Balanced Error Rate (BER), Time

Algorithm	AUC	ER	BER	Time (s)
LDA	0.748	0.090*	0.490	**0.038**
C-HLD	0.747	0.195	**0.334**	0.174
R-HLD	0.712	**0.089**	0.483	0.154
QDA	**0.817**	**0.089**	0.463	0.046
SVM	0.740	0.091*	0.500	7667.820
D-GLD	0.788	**0.089**	0.483	0.038

Table 3.7: Artificial dataset $\mathcal{D}_3$ (f=2): AUC, Error Rate (ER), Balanced Error Rate (BER), Time

Algorithm	AUC	ER	BER	Time (s)
LDA	0.674	0.330	0.443	**0.008**
C-HLD	0.667	0.334	0.378	0.153
R-HLD	0.654	0.306	0.422	0.133
QDA	**0.760**	**0.272**	**0.332**	0.012
SVM	0.719	0.323	0.481	1347.635
D-GLD	0.745	0.305	0.422	0.013

The results in Table 3.6 and 3.7 show QDA having the largest AUC and error rate among all the classifiers compared. This is consistent with the fact that the artificial dataset is normally distributed in each class with unequal covariances. Therefore the Bayes-optimal classifier is obtained from quadratic discriminant analysis. The SVM shows a competitive performance on this dataset to QDA (below the performance of D-GLD). However, since the SVM does not make any assumptions on the distribution of the data, maximising the margin between the positive and negative examples does not necessarily

Table 3.8: E-Coli-1 dataset: AUC, Error Rate (ER), Balanced Error Rate (BER), Time

Algorithm	AUC	ER	BER	Time (s)
LDA	0.980	0.032*	0.032*	**0.022**
C-HLD	0.980	0.032*	0.032*	0.191
R-HLD	0.980	0.032*	0.032*	0.217
QDA	0.971	0.441	0.500	0.045
SVM	0.979	**0.031**	**0.030**	1.242
D-GLD	**0.995**	0.032*	0.032*	0.069

Table 3.9: Liver disorders dataset: AUC, Error Rate (ER), Balanced Error Rate (BER), Time

Algorithm	AUC	ER	BER	Time (s)
LDA	0.703	**0.309**	**0.333**	**0.001**
C-HLD	0.700	0.358	0.366	0.155
R-HLD	0.699	0.359	0.367	0.132
QDA	0.692	0.401	0.386	0.001*
SVM	0.728	0.403	0.472	0.014
D-GLD	**0.757**	0.359	0.367	0.005

Table 3.10: Diabetes dataset: AUC, Error Rate (ER), Balanced Error Rate (BER), Time

Algorithm	AUC	ER	BER	Time (s)
LDA	0.828	0.226	**0.274**	0.004
C-HLD	0.827	0.229	0.283	0.167
R-HLD	0.827	0.229	0.282	0.144
QDA	0.805	0.258	0.300	**0.001**
SVM	0.836	**0.223**	0.275*	0.031
D-GLD	**0.845**	0.229	0.283	0.005

Table 3.11: WpBC dataset: AUC, Error Rate (ER), Balanced Error Rate (BER), Time

Algorithm	AUC	ER	BER	Time (s)
LDA	0.788	**0.202**	**0.300**	0.043
C-HLD	0.785	0.212	0.307	0.581
R-HLD	0.785	0.212	0.307	0.522
QDA	0.641	0.230	0.480	**0.001**
SVM	0.720	**0.202**	0.411	0.031
D-GLD	**0.923**	0.211	0.306	0.057

Table 3.12: USPS-1: AUC, Error Rate (ER), Balanced Error Rate (BER), Time

Algorithm	AUC	ER	BER	Time (s)
LDA	0.997*	0.017	0.032	0.091
C-HLD	0.997*	**0.015**	**0.022**	19.722
R-HLD	0.997*	**0.015**	**0.022**	19.732
QDA	0.984	0.167	0.500	**0.035**
SVM	**0.999**	0.128	0.384	12.491
D-GLD	0.990	**0.015**	**0.022**	0.396

Table 3.13: Yeast-1 dataset: AUC, Error Rate (ER), Balanced Error Rate (BER), Time

Algorithm	AUC	ER	BER	Time (s)
LDA	0.832	0.131	0.313	0.002*
C-HLD	0.825	0.131	**0.273**	0.167
R-HLD	0.820	0.131	0.313	0.143
QDA	0.817	0.164	0.500	**0.001**
SVM	0.854*	**0.117**	0.324	0.039
D-GLD	**0.856**	0.131	0.312	0.005

Table 3.14: Yeast-6 dataset: AUC, Error Rate (ER), Balanced Error Rate (BER), Time

Algorithm	AUC	ER	BER	Time (s)
LDA	0.878	0.037*	0.390	0.001*
C-HLD	0.878	0.054	**0.243**	0.165
R-HLD	0.877	0.037*	0.390	0.143
QDA	0.845	**0.034**	0.500	**0.001**
SVM	0.845	**0.034**	0.500	0.028
D-GLD	**0.911**	0.037*	0.390	0.005

Table 3.15: Abalone-19 dataset: AUC, Error Rate (ER), Balanced Error Rate (BER), Time

Algorithm	AUC	ER	BER	Time (s)
LDA	0.847	0.015	0.504	**0.001**
C-HLD	0.848	0.085	**0.308**	0.163
R-HLD	0.724	0.014	0.503	0.140
QDA	0.737	0.016	0.504	0.001*
SVM	0.662	**0.008**	0.500	0.083
D-GLD	**0.862**	0.014	0.503	0.007

yield the Bayes-optimal discrimination for this dataset with a known normal distribution. Moreover, the training time of the SVM is large, taking as much as 2.1 hours in Table 3.6, due to the fact that as a kernel classifier, it doesn't scale well with a lot of training data; the training time can be prohibitive on larger datasets.

For arbitrary non-normal distributions, however, QDA may be prone to overfitting, and may not perform satisfactorily, due, in part, to the fact that it is a quadratic classifier. Linear classifiers, on the other hand, tend to be more robust to non-normality than quadratic classifiers [128]. Thus, LDA, as well as R-HLD and D-HLD outperform QDA in terms of the error rate on most of the real-world datasets. However, C-HLD, while also being a linear model, constrains the parameter s to $[0, 1]$ in (2.32) and (2.33) on page 15. This has been shown analytically to affect the classification accuracy in Section 3.3.2 (page 46), since s tends to fall outside the interval $[0, 1]$ under class imbalance. This accounts for why C-HLD shows the largest error rate in both Tables 3.7 and 3.6. It is for this same reason that the C-HLD achieves the best BER in both Tables 3.7 and 3.6, since by constraining s to the interval $[0, 1]$, the discriminating threshold is always bounded between the projected class means, and hence the error rate tends to be more balanced.

It will also be noted that the error rate (ER) of the LDA procedure is significantly worse than that of QDA in Table 3.7, but only marginally in Table 3.6. This is because as the degree of class imbalance increases, the majority class becomes far more probable than the minority class. Therefore, the decision rule depends less on the differences in covariance matrices, but depends more on the discriminating threshold w_0. Since the threshold obtained by LDA as given by (2.14) is unbounded and depends on the ratio of the prior probabilities (or equivalently the degree of class imbalance), LDA is able to track the optimal w_0 under high degrees of class imbalance and yields a satisfactory performance in terms of the error rate. This result confirms the conclusions by Xue [12] that unbalanced data have no negative effect on LDA in terms of the error rate.

Unlike LDA however, R-HLD and the D-HLD account for heteroscedasticity by obtaining a linear approximation to the quadratic boundary in QDA that minimises the Bayes error. Due to this, their error rate performance is closest to QDA on the toy dataset under any degree of class imbalance as can be seen from Tables 3.6 and 3.7. Since the criterion that is minimised in the R-HLD and D-HLD procedures is the Bayes error (or the probability of misclassification), which makes use of the empirical prior probabilities, the BER is not necessarily minimised for these procedures. However, regarding the AUC, D-HLD dynamically optimises the weight vector $\mathbf{w}$ to minimise the Bayes error for any given threshold w_0, so that for the FPR corresponding to that threshold, the TPR is maximised. Therefore,

D-HLD results in an improved AUC over R-HLD.

For the real-world datasets, due to the fact that they are not drawn from a normal distribution, QDA is no longer superior in terms of the error rate. For these datasets, the best error rate performance is dominated by SVM, which is a non-parametric classifier. The original LDA and heteroscedastic LDA procedures compare closely to the SVM in terms of the error rate, and consistently outperform QDA due to the fact the linear models provide robustness over QDA, even if the normal distribution assumption is not satisfied.

Still, the fact that the BER happens to be significantly larger than the ER values on most of the real-world datasets suggests that the classification is skewed toward the majority class. This is particularly so for the SVM and QDA classifiers on the USPS, Liver, WpBC, Yeast-1 and Yeast-6 datasets. The AUC is then a preferred evaluation criterion. For the same reason as indicated for the artificial datasets, the proposed D-HLD procedure yields the best AUC values over all the real-world datasets, with the exception of the USPS dataset. Moreover, D-HLD is superior to the other heteroscedastic LDA procedures (R-HLD and C-HLD) in terms of the training time, since D-HLD follows a principled optimisation procedure for minimising the Bayes error, unlike in R-HLD and C-HLD. This computational gain increases with the dimensionality d of the dataset, and is most profound on the USPS dataset, since the bulk of the computation required in the heteroscedastic LDA procedures is for the inversion of a d-sized scatter matrix.

3.6.3 Kernel classification

Apart from robustness, one other advantage of LDA and heteroscedastic LDA over QDA is the ability of the linear models to be kernelised to learn general non-linear decision boundaries for non-linear classification. To do this, the data is implicitly transformed via a non-linear kernel function into a higher-dimensional feature space where linear separability is guaranteed (See Section 3.5 on page 49). While there is no obvious choice of a kernel function for a given data, some popular kernels such as the Gaussian kernel and polynomial kernel have been successfully applied in many applications. In this experiment, both the Gaussian kernel and the polynomial kernel of degree 2, i.e., the quadratic kernel, are employed. The quadratic kernel $\mathcal{K}$ is defined for any two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ as:

$$\mathcal{K}_q(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^\top \mathbf{x}_j)^2, \tag{3.125}$$

while the Gaussian kernel is given as:

$$\mathcal{K}_g(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(- \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{c} \right), \tag{3.126}$$

where c is set to 1 in this experiment.

The proposed kernel GLD algorithm (K-GLD) in Algorithm 4 is used for this evaluation. For comparison, the kernel SVM (K-SVM) and Kernel Fisher Discriminant (KFD) in Section 2.1.1 (page 11) are also simulated. Moreover, the existing heteroscedastic LDA procedures of C-HLD and R-HLD-2 are kernelised in the manner as is described on page 15 and simulated. The kernelised forms of C-HLD and R-HLD are denoted as K-CHLD and K-RHLD respectively.

The KFD solution is used as the initial solution to the K-GLD algorithm; the algorithm is stopped if the Bayes error p_e is less than or equal to 10^{-6}, or else it is terminated after $2,000$ iterations. Due to the time complexity of the heteroscedastic LDA procedures, a step size of $\Delta s = 0.001$ is used for C-HLD, and 1000 trials are run for R-HLD-2. The datasets employed for this experiment, and their corresponding kernel transformations, are shown in Table 3.16.

Table 3.16: List and characteristics of datasets
K is the number of classes, d is the dimensionality of the dataset, and n is the number of data points in the dataset.

Dataset	Label	n	d	K	Kernel transform
Vowels	(e)	990	10	11	Quadratic
Zernike Moments	(f)	2000	47	10	Quadratic
Image Segmentation (Statlog)	(g)	2310	19	7	Gaussian

Table 3.17: Average classification accuracy with kernel classifiers (%)

Algorithm	(e)	(f)	(g)
KFD	97.98	82.00	96.54
K-CHLD	93.94	75.00	88.31
K-RHLD	93.94	74.50	93.51
K-SVM	86.87	85.50	94.37
K-GLD	98.99	85.00	96.54

Results and discussions

In Table 3.17, it is seen that K-GLD achieves classification accuracies of 98.99% on dataset (e). It will be recalled, however, that using a linear decision rule, R-GLD achieves much lower classification accuracies

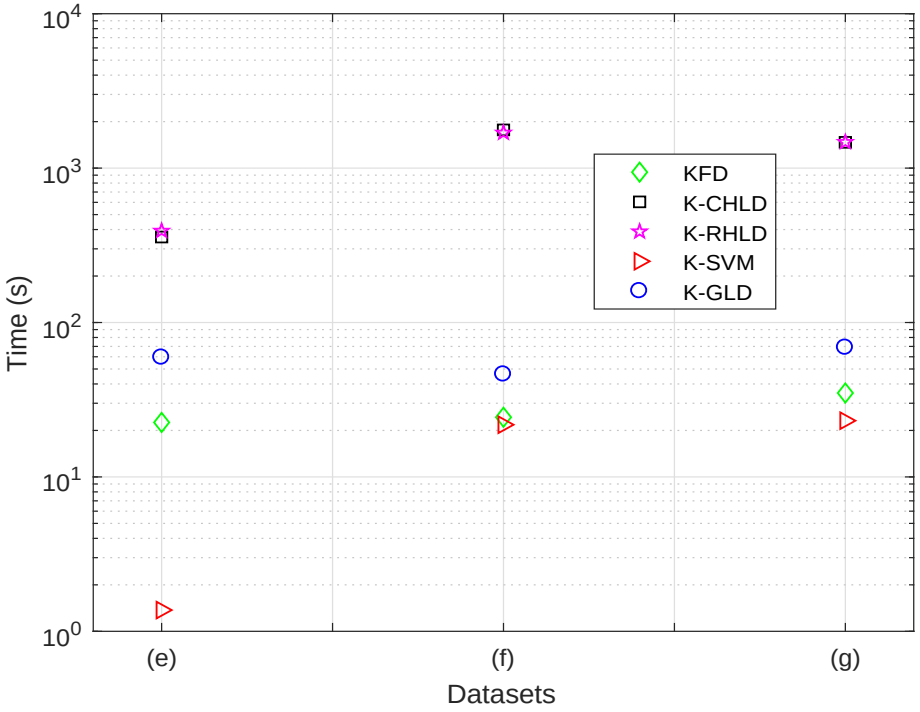

Figure 3.2: Average training time (s)

on these datasets as shown in Table 3.3 (page 57). This implies that a non-linear classifier is more suitable to discriminate the classes in this dataset. Therefore, by employing the Gaussian kernel transformation, R-GLD ensures better discrimination, and hence better classification. Similar performance improvement is seen on this dataset between the linear classifiers (LDA, R-HLD-2, C-HLD and SVM) in Table 3.3 and their respective kernelised versions (KFD, K-RHLD, K-CHLD, K-SVM) in Table 3.17, although K-GLD shows superior performance over the remaining kernel algorithms.

Similarly for datasets (f) and (g), K-SVM and K-GLD achieve improved classification accuracies over their linear versions, with K-SVM being superior on these datasets, as shown in Table 3.17. However, since the kernel LDA procedures all require the inversion of kernel matrices of size n, their time complexity can be prohibitive for very large datasets, especially for K-RHLD and K-CHLD which require several matrix inversions. Since it does not require any matrix inversions, K-SVM performs best in terms of the training time among the different algorithms on all the datasets, as shown in Figure 3.2. While the performance of K-RHLD and K-CHLD are suboptimal on datasets (f) and (g), they can be improved by increasing the number of random trials (page 14) or the step rate Δs (page 15) at the expense of an even more prohibitive computation time.

3.7 Chapter summary

This chapter has introduced a linear discriminant analysis (LDA) procedure, known as the Gaussian linear discriminant (GLD), that accounts for heteroscedasticity by finding a linear approximation to the quadratic boundary in quadratic discriminant analysis (QDA). This is done by minimising the Bayes error using two optimisation procedures.

The first optimisation procedure, known as recursive GLD (R-GLD), makes use of the first and second order optimality conditions for the minimisation of the Bayes error, involves recursive matrix inversions, and is suitable for low-dimensional data. The second optimisation procedure, known as gradient-descent GLD (G-GLD), uses only the first-order optimality conditions, requires no matrix inversions, and is thus suitable for very high-dimensional data for which matrix inversion can be computationally intractable. The Bayes error is shown to be non-convex, and thus, the GLD procedures require the use of multiple initial solutions to improve the quality of the local minimum found.

Since LDA assumes a normal distribution in each class of a dataset, the chapter also discusses modifications to the GLD, a local neighbourhood search (LNS) procedure, that allows it to be robust when dealing with non-normal distributions (nearly-normal distributions) to achieve a satisfactory classification accuracy.

Following this, the performance of LDA and existing heteroscedastic LDA approaches are discussed under the scenario of class imbalance, using the first and second order optimality conditions for the Bayes error minimisation. The existing heteroscedastic LDA models are shown to be suboptimal in terms of the area under the receiver operating characteristics curve (AUC). A dynamic GLD model (D-GLD) is then proposed to overcome the class imbalance problem.

The kernel formulation of the GLD is also provided in this chapter, that permits the construction of a non-linear decision boundary if a linear discriminant is inappropriate, using an appropriate kernel function.

Finally, the proposed R-GLD procedure is experimentally validated in three sets of experiments. The first experiment deals with balanced data where R-GLD, together with LNS, outperforms the existing heteroscedastic LDA procedures in terms of the classification accuracy, and is shown to be comparable with the classification performance of SVM, but with a much smaller training time. The second experiment on imbalanced data shows the proposed D-GLD procedure achieves superior AUC values over all the existing heteroscedastic LDA approaches as well as the SVM. The final experiment on datasets that are linearly non-separable indicates that the kernel version of GLD (K-GLD) is able to learn non-linear decision boundaries for such datasets, resulting in an improved classification performance.

Chapter 4

Heteroscedastic LDA for dimensionality reduction [1]

In the previous chapter, a heteroscedastic extension of linear discriminant analysis (LDA), known as Gaussian linear discriminant (GLD), which is optimal in terms of minimising the Bayes error, was proposed for linear classification in the two-class case. More commonly, however, LDA is employed for linear dimensionality reduction (LDR), which involves a linear transformation of the original dataset (see section 2.2.2), followed by classification. In the two-class case, the weight vector $\mathbf{w}$ used for linear classification in LDA is also the linear transformation used for LDR. In this chapter, GLD is extended to the multi-class scenario for supervised LDR, which involves linearly reducing the dimensionality of a dataset while maximising the class-discriminatory information. The proposed procedure, referred to as multi-class GLD (M-GLD), involves the sequential minimisation of the Bayes error, via the successive construction of $(K^2 + K - 4)/2$ GLD classifiers for the K-class problem. The M-GLD procedure reduces the dimensionality of the dataset to $K - 1$, such that it is well-primed for Bayesian classification.

4.1 Multi-class Gaussian linear discriminant (M-GLD)

As with the two-class case, it is assumed that the data in each of the K classes is normally distributed with a mean of $\bar{\mathbf{x}}_k$ and a covariance matrix of $\mathbf{S}_k$ for every $k \in \{1, ..., K\}$. The aim is to apply a Bayes classifier after linear dimensionality reduction (LDR). Bayesian classification involves the evaluation of K posterior probabilities, among which the highest is chosen. These K probability functions must however sum up to one, making only $K - 1$ of them independent. Therefore, the smallest set of features required for Bayesian classification is $K - 1$, corresponding to an optimum transformation to a $K - 1$-dimensional space.

[1] Most of the work presented in this chapter first appeared in: K. S. Gyamfi, J. Brusey, A. Hunt and E. Gaura, "Linear dimensionality reduction for classification via a sequential Bayes error minimisation with an application to flowmeter diagnostics," *Expert Systems with Applications* (2017), vol. 91, Sep. 2017, pp. 252-262.

Along this line, it is sought to project the dataset $\mathcal{D}$ onto a $(K-1)$-dimensional subspace. Therefore, the transformation matrix $\mathbf{M}$ is given as $\mathbf{M} = [\mathbf{v}_1, ..., \mathbf{v}_{K-1}]$, where $\mathbf{v}_i \in \mathbb{R}^d$ for $i \in \{1, ..., K-1\}$. The proposed algorithm is such that one column of $\mathbf{M}$ is found in each of $K-1$ steps.

Before fully detailing the proposed linear dimensionality reduction procedure for the general K-class problem, the special cases of $K = 2$ and $K = 3$ are first considered.

4.1.1 Two-class case

In the two-class case, $\mathbf{M} = \mathbf{v}_1$, and therefore the task of finding $\mathbf{v}_1$ that preserves the classification information in the original space is equivalent to obtaining a linear discriminant that best divides the two classes $\mathcal{C}_1$ and $\mathcal{C}_2$. That is, it is desired to obtain a linear classifier $\{\mathbf{w}_1, t_1\}$ such that, for every data sample $\mathbf{x} \in \mathcal{D}$, the true class of $\mathbf{x}$, $\mathcal{C}^*(\mathbf{x})$, is decided according to the following decision rule:

$$\mathcal{C}^*(\mathbf{x}) = \begin{cases} \mathcal{C}_1 & \text{if} \quad \mathbf{w}_1^\top \mathbf{x} \geq t_1 \\ \mathcal{C}_2 & \text{if} \quad \mathbf{w}_1^\top \mathbf{x} < t_1 \end{cases} \tag{4.1}$$

The optimal $\mathbf{w}_1$ minimises the Bayes error given by:

$$\epsilon_1 = \pi_1 p\big(y < t_1 | \mathcal{C}_1\big) + \pi_2 p\big(y \geq t_1 | \mathcal{C}_2\big) \tag{4.2}$$

where $y = \mathbf{w}_1^\top \mathbf{x}$.

Since $\mathbf{x}$ is assumed to have a normal distribution in classes $\mathcal{C}_1$ and $\mathcal{C}_2$, y is expected to be normally distributed with a mean of μ_1 and a variance of σ_1^2 for class $\mathcal{C}_1$, and a mean of μ_2 and a variance of σ_2^2 for class $\mathcal{C}_2$ given as:

$$\mu_1 = \mathbf{w}_1^\top \bar{\mathbf{x}}_1 \quad \mu_2 = \mathbf{w}_1^\top \bar{\mathbf{x}}_2 \quad \sigma_1^2 = \mathbf{w}_1^\top \mathbf{S}_1 \mathbf{w}_1 \quad \sigma_2^2 = \mathbf{w}_1^\top \mathbf{S}_2 \mathbf{w}_1 \tag{4.3}$$

The normality assumption allows the individual misclassification probabilities in (4.2) to be expressed as:

$$p(y < t_1 | \mathcal{C}_1) = \int_{-\infty}^{t_1} \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{(r-\mu_1)^2}{2\sigma_1^2} \right] dr = 1 - Q\left(\frac{t_1 - \mu_1}{\sigma_1} \right) \tag{4.4}$$

and

$$p(y \geq t_1 | \mathcal{C}_2) = \int_{t_1}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_2} \exp\left[-\frac{(r-\mu_2)^2}{2\sigma_2^2} \right] dr = Q\left(\frac{t_1 - \mu_2}{\sigma_2} \right) \tag{4.5}$$

where $Q(\cdot)$ is the Q-function, so that the Bayes error ϵ_1 may be rewritten as:

$$\epsilon_1 = \pi_1\big(1 - Q(z_1)\big) + \pi_2\big(Q(z_2)\big) \tag{4.6}$$

where

$$z_1 = \frac{t_1 - \mu_1}{\sigma_1} \quad \text{and} \quad z_2 = \frac{t_1 - \mu_2}{\sigma_2} \tag{4.7}$$

In Section 3.1.1, it was shown using first and second-order optimality conditions that the optimal $\mathbf{w}_1$ and t_1 that minimise ϵ_1 can be obtained by solving the following equations iteratively:

$$\mathbf{w}_1 = \left(\frac{z_2}{\sigma_2}\mathbf{S}_2 - \frac{z_1}{\sigma_1}\mathbf{S}_1\right)^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{4.8}$$

and

$$t_1 = \frac{\mu_2\sigma_1^2 - \mu_1\sigma_2^2 + \sigma_1\sigma_2\sqrt{(\mu_1 - \mu_2)^2 + 2(\sigma_1^2 - \sigma_2^2)\ln\left(\frac{\tau\sigma_1}{\sigma_2}\right)}}{\sigma_1^2 - \sigma_2^2} \tag{4.9}$$

where $\tau = \pi_2/\pi_1$ [129].

Though the aim is to find only the weight vector $\mathbf{w}_1$, it is noted that $\mathbf{w}_1$ is not independent of t_1, as it is related to t_1 through z_1 and z_2. Therefore, the optimal choice of $\mathbf{w}_1$ is obtained only by optimising $\mathbf{w}_1$ and t_1 simultaneously.

The resulting transform is then $\mathbf{M} = [\mathbf{v}_1]$, where $\mathbf{v}_1$ is set to the optimal $\mathbf{w}_1$ as above.

4.1.2 Three-class case

In the case where $K = 3$, the transformation matrix is $\mathbf{M} = [\mathbf{v}_1, \mathbf{v}_2]$.

Step 1

In the first step, it is sought to find the first column of $\mathbf{M}$, i.e., $\mathbf{v}_1$. To do this, a linear classifier is trained to separate one class from the remaining classes; since there are three classes, there are three different classifiers that could be constructed to this end. The idea is to choose $\mathbf{v}_1$ to correspond to the classifier among these three whose minimum Bayes error is smallest.

First, the possibility of training a linear classifier $\{\mathbf{w}_1, t_1\}$ to discriminate class $\mathcal{C}_1$ from classes $\mathcal{C}_2$ and

$\mathcal{C}_3$ is considered. Then, for every data sample $\mathbf{x} \in \mathcal{D}$, the following decision rule applies:

$$\mathcal{C}^*(\mathbf{x}) = \begin{cases} \mathcal{C}_1 & \text{if} \quad \mathbf{w}_1^\top \mathbf{x} \geq t_1 \\ \mathcal{C}_2, \mathcal{C}_3 & \text{if} \quad \mathbf{w}_1^\top \mathbf{x} < t_1 \end{cases} \tag{4.10}$$

Notice that, as with the case $K = 2$, the projected data in class $\mathcal{C}_1$ is normally distributed on one side of the linear discriminant with a mean of μ_1 and a variance of σ_1^2 as given by (4.3), while the projected data in classes $\mathcal{C}_2$ and $\mathcal{C}_3$, on the other side of the discriminant, form a mixture of two Gaussians $\mathcal{M}_1$ given by:

$$\begin{aligned} &\mathcal{M}_1 \sim \sum_{i=2}^{3} p_i \mathcal{N}(\mu_i, \sigma_i^2), \quad \text{where} \quad p_i = \frac{\pi_i}{1 - \pi_1}, \quad \text{such that} \quad \sum_{i=2}^{3} p_i = 1 \\ &\text{and} \quad \mu_i = \mathbf{w}_1^\top \bar{\mathbf{x}}_i \quad \sigma_i^2 = \mathbf{w}_1^\top \mathbf{S}_i \mathbf{w}_1 \end{aligned} \tag{4.11}$$

As before, the optimal $\mathbf{w}_1$ then minimises the Bayes error. However, the Bayes error is now given by:

$$\epsilon_1 = \pi_1 p\big(y < t_1 | \mathcal{C}_1\big) + (1 - \pi_1) p\big(y \geq t_1 | \mathcal{M}_1\big) \tag{4.12}$$

Here, $p\big(y < t_1 | \mathcal{C}_1\big)$ is as given before in (4.4), while $p\big(y \geq t_1 | \mathcal{M}_1\big)$ can be expressed as:

$$\begin{aligned} p(y \geq t_1 | \mathcal{M}_1) &= \sum_{i=2}^{3} p_i \int_{t_1}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_i} \exp\bigg[- \frac{(r - \mu_i)^2}{2\sigma_i^2} \bigg] dr \\ &= \sum_{i=2}^{3} p_i Q\bigg(\frac{t_1 - \mu_i}{\sigma_i} \bigg). \end{aligned} \tag{4.13}$$

Thus, the Bayes error ϵ_1 can be evaluated as:

$$\begin{aligned} &\epsilon_1 = \pi_1 \big(1 - Q(z_{1,1})\big) + (1 - \pi_1) \sum_{i=2}^{3} p_i Q(z_{1,i}), \\ &\quad \text{where} \quad z_{1,k} = \frac{t_1 - \mathbf{w}_1^\top \bar{\mathbf{x}}_k}{(\mathbf{w}_1^\top \mathbf{S}_k \mathbf{w}_1)^{1/2}}, \quad \text{for every} \quad k \in \{1, 2, 3\}. \end{aligned} \tag{4.14}$$

i.e.,

$$\epsilon_1 = \pi_1 \big(1 - Q(z_{1,1})\big) + \sum_{i=2}^{3} \pi_i Q(z_{1,i}) \tag{4.15}$$

Next, the second possibility of training a linear classifier $\{\mathbf{w}_2, t_2\}$ to discriminate class $\mathcal{C}_2$ from classes $\mathcal{C}_1$ and $\mathcal{C}_3$ is considered. The optimal $\{\mathbf{w}_2, t_2\}$ minimises the Bayes error which can be shown, similar to

the derivation of ϵ_1, to be given by:

$$\epsilon_2 = \pi_2\big(1 - Q(z_{2,2})\big) + \sum_{i\in\{2,3\}} \pi_i Q(z_{2,i}),$$
$$\text{where} \quad z_{2,k} = \frac{t_2 - \mathbf{w}_2^\top \bar{\mathbf{x}}_k}{(\mathbf{w}_2^\top \mathbf{S}_k \mathbf{w}_2)^{1/2}}, \quad \text{for every} \quad k \in \{1,2,3\}. \tag{4.16}$$

Finally, a classifier $\{\mathbf{w}_3, t_3\}$ is trained to linearly discriminate class $\mathcal{C}_3$ from classes $\mathcal{C}_1$ and $\mathcal{C}_2$. The optimal values of $\mathbf{w}_3$ and t_3 are obtained by minimising the Bayes error ϵ_3 which can be shown to be given by:

$$\epsilon_3 = \pi_3\big(1 - Q(z_{3,3})\big) + \sum_{i=1}^{2} \pi_i Q(z_{3,i}),$$
$$\text{where} \quad z_{3,k} = \frac{t_3 - \mathbf{w}_3^\top \bar{\mathbf{x}}_k}{(\mathbf{w}_3^\top \mathbf{S}_k \mathbf{w}_3)^{1/2}}, \quad \text{for every} \quad k \in \{1,2,3\} \tag{4.17}$$

It is assumed, without any loss of generality, that the first classifier $\{\mathbf{w}_1, t_1\}$ yields the smallest Bayes error, i.e., $\epsilon_1 < \epsilon_2, \epsilon_3$. Then, $\mathbf{v}_1$ is set to the optimal vector $\mathbf{w}_1$ corresponding to the minimisation of ϵ_1.

Step 2

Following Step 1, the next task is to find the second column of $\mathbf{M}$, i.e., $\mathbf{v}_2$. As a classifier has already been trained to separate class $\mathcal{C}_1$ from the two remaining classes in Step 1, $\mathcal{C}_1$ is now removed from the dataset $\mathcal{D}$. This permits the construction of a linear classifier $\{\mathbf{w}_2, t_2\}$ to linearly discriminate classes $\mathcal{C}_2$ and $\mathcal{C}_3$, in the fashion of the case $K = 2$,. This is done by minimising the Bayes error ϵ_2 given by:

$$\epsilon_2 = \pi_2'\big(1 - Q(z_2)\big) + \pi_3'\big(Q(z_3)\big) \tag{4.18}$$

It will be noted that by removing $\mathcal{C}_1$ from the dataset $\mathcal{D}$, the prior probabilities of the remaining classes change. Therefore, in (4.18), π_2' and π_3' are the prior probabilities of classes $\mathcal{C}_2$ and $\mathcal{C}_3$ respectively, conditional on class $\mathcal{C}_1$ being removed from the dataset $\mathcal{D}$, and they are given by:

$$\pi_2' = \frac{\pi_2}{1 - \pi_1} \quad \text{and} \quad \pi_3' = \frac{\pi_3}{1 - \pi_1} \tag{4.19}$$

The optimal $\mathbf{w}_2$ is then assigned to $\mathbf{v}_2$.

Note that often, the transformation matrix $\mathbf{M}$ is constrained to be orthogonal [94]. Thus, it would be necessary to have an orthogonality constraint in the form, $\mathbf{w}_1^\top \mathbf{w}_2 = 0$, while minimising the Bayes error

ϵ_1 in the second step. However, such an orthogonality constraint is not binding, if classification is desired after dimensionality reduction [8]; it is sufficient that the component vectors of $\mathbf{M}$ be independent.

4.1.3 Arbitrary number of classes

Having detailed the fundamentals of the proposed LDR procedure for the special cases of $K = 2$ and $K = 3$, the description of the proposed algorithm for a general value of K is now provided.

Let $\mathcal{L} = \{1, ..., K\}$, and l be an arbitrary element in $\mathcal{L}$. Define a conditional prior probability $\pi'_i = p(\mathcal{C}_i|\bar{\mathcal{C}}_l)$ to be the prior probability of Class $\mathcal{C}_i$ conditional on the data in Class $\mathcal{C}_l$ being removed from the dataset $\mathcal{D}$, for all $i \in \mathcal{L}$. Then for the $k = 1$st iteration, when no class has been removed yet, $\pi'_i = \pi_i$.

Step 1

A linear classifier $\{\mathbf{w}_i, t_i\}$ that discriminates class $\mathcal{C}_i$ from all other classes is constructed, for every $i \in \mathcal{L}$, by minimising the Bayes error ϵ_i given by:

$$\epsilon_i = \pi'_i\big(1 - Q(z_i)\big) + \sum_{j \in \mathcal{L}\setminus\{i\}} \pi'_j Q(z_j), \tag{4.20}$$

where

$$z_k = \frac{t_i - \mu_k}{\sigma_k}, \quad \mu_k = \mathbf{w}_i^\top \bar{\mathbf{x}}_k \quad \text{and} \quad \sigma_k^2 = \mathbf{w}_i^\top \mathbf{S}_k \mathbf{w}_i, \quad \text{for every} \quad k \in \mathcal{L}. \tag{4.21}$$

After the minimisation of ϵ_i, $\mathbf{v}_k$, i.e., the kth column of $\mathbf{M}$ is set to $\mathbf{w}_l$, where

$$l = \arg\min_i \{\epsilon_1, ..., \epsilon_{|\mathcal{L}|}\} \tag{4.22}$$

Step 2

As the optimal classifier $\{\mathbf{w}_l, t_l\}$ linearly separates class $\mathcal{C}_l$ from all other classes, class $\mathcal{C}_l$ can be excluded from the dataset $\mathcal{D}$ to allow for the construction of other classifiers to linearly discriminate the remaining classes. Correspondingly, we remove l from the set $\mathcal{L}$. The conditional prior probabilities of the remaining classes are then updated as:

$$\pi'_i := \frac{\pi'_i}{1 - \pi'_l}, \quad \text{for all} \quad i \in \mathcal{L}. \tag{4.23}$$

The index k is then incremented by one.

Step 3

Steps 1 and 2 are repeated until all $K-1$ columns of the transformation matrix $\mathbf{M}$ have been determined.

4.1.4 Optimisation of the Bayes error ϵ_i

Up until this point, it has only been indicated that the classifier $\{\mathbf{w}_i, t_i\}$ ought to minimise the Bayes error given by (4.20). However, explicit expressions for the optimality conditions under which the Bayes error is minimised are computationally difficult to solve (see section 4.A.2). Therefore, a gradient descent procedure is proposed:

For $j = 0$:

$$\mathbf{w}_i^{j+1} = \mathbf{w}_i^j - \alpha \frac{\partial \epsilon_i}{\partial \mathbf{w}_i^j} \tag{4.24}$$

$$t_i^{j+1} = t_i^j - \alpha \frac{\partial \epsilon_i}{\partial t_i^j} \tag{4.25}$$

starting from an initial choice of $\mathbf{w}_i$ and t_i, where α is the learning rate. Note that the partial derivatives of ϵ_i have been derived in (4.55) and (4.58). Since the Bayes error is known to be non-convex and is characterised by multiple local minima [71], the gradient descent algorithm may have to be performed using different initial solutions to improve the quality of the local minima to which the algorithm converges.

Though only the optimal $\mathbf{w}_i$ is required to form the columns of $\mathbf{M}$, it will be noted that the optimal $\mathbf{w}_i$ is tied to the optimal threshold t_i through z_i and z_j as can be seen from (4.21) and (4.63), requiring that they both be minimised in the gradient descent procedure.

In all, $(K^2 + K - 4)/2$ classifiers are constructed in the proposed algorithm for a K-class problem.

The proposed LDR procedure is detailed in Algorithm 5.

4.2 Experimental validation

In this section, the proposed LDR technique is validated experimentally, first on two artificial datasets, and then on 10 UCI datasets. The characteristics the UCI datasets are shown in Table 4.1, while the artificial datasets are generated as follows:

4.2.1 Artificial dataset 1 (DS1)

1. DS1 has a dimensionality of $d = 20$, and $K = 3$ classes with each of the classes being normally distributed.

Algorithm 5 Multiclass GLD (M-GLD)

Input: $\mathcal{X}$
$\mathcal{L} \leftarrow \{1, ..., K\}$
$\pi'_t = \pi_t$ for all $t \in \mathcal{L}$
for $k \leftarrow 1$ to $K-1$ **do**
    for $i \in \mathcal{L}$ **do**
        Initialise $\mathbf{w}_i$
        $j \leftarrow 0$
        while Gradient descent stopping criteria are not satisfied **do**
            for $t \in \mathcal{L}$ **do**
                Evaluate μ_t, σ_t, z_t as given by (4.21)
            end for
            Evaluate the gradient of ϵ_i w.r.t. $\mathbf{w}_i$ as given by (4.55)
            Evaluate the gradient of ϵ_i w.r.t. t_i as given by (4.58)
            Evaluate the Bayes error ϵ_i given by (4.20)
            Update $\mathbf{w}_i$ as $\mathbf{w}_i^{j+1} = \mathbf{w}_i^j - \gamma \frac{\partial \epsilon_i}{\partial \mathbf{w}_i^j}$
            Update t_i as $t_i^{j+1} = t_i^j - \gamma \frac{\partial \epsilon_i}{\partial t_i^j}$
        end while
    end for
    $l \leftarrow \arg\min_i \{\epsilon_1, ..., \epsilon_{|\mathcal{L}|}\}$
    $\mathbf{v}_k \leftarrow \mathbf{w}_l$
    $\mathcal{L} \leftarrow \{1, ..., l-1, l+1, ..., K\}$
    $\pi'_i := \frac{\pi'_i}{1-\pi'_l}$
end for
$\mathbf{M} = [\mathbf{v}_1, ..., \mathbf{v}_{K-1}]$

2. The means of the classes are randomly sampled uniformly from a Latin hypercube of side length 6 in 2 dimensions. The following instances of the mean parameters are used in the experiment:

$$\mathbf{x}_1 = \begin{bmatrix} 1.6638 \\ 3.6366 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 5.1489 \\ 1.2540 \end{bmatrix}, \quad \mathbf{x}_3 = \begin{bmatrix} 3.9970 \\ 5.6633 \end{bmatrix} \tag{4.26}$$

3. The covariance matrix of each class is determined as the 2-dimensional identity matrix $\mathbf{I}_2$.

4. Using the means from Step 2 and the covariances in Step 3, 100 samples are generated from the Gaussian distribution for each class to obtain a dataset with 300 samples.

5. Two information-less features are then added to the dataset obtained in Step 4: for all samples, two linear combinations (whose coefficients are in the interval $[0, 1]$) of the two features are obtained randomly to form two extra dimensions. Then, 16 random features are generated from the normal distribution $\mathcal{N}(0, 1)$ for each sample. Due to the fact that random features are appended to the original 2-dimensional samples, the covariance matrices are unequal among the classes.

6. The resulting dataset has 20 dimensions. Step 5 is necessary to simulate multicollinearity and noisy features that exist in many real world datasets and which lead to overfitting.

4.2.2 Artificial dataset 2 (DS2)

1. DS2 has a dimensionality of $d = 20$, and $K = 4$ classes with each of the classes being normally distributed.

2. The means of the classes are randomly sampled uniformly from a Latin hypercube of side length 6 in 3 dimensions. The following instances of the mean parameters are used in the experiment:

$$\mathbf{x}_1 = \begin{bmatrix} 0.6541 \\ 3.6482 \\ 1.9521 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 2.4828 \\ 2.8612 \\ 0.5296 \end{bmatrix}, \quad \mathbf{x}_3 = \begin{bmatrix} 3.6194 \\ 0.7111 \\ 2.4050 \end{bmatrix}, \quad \mathbf{x}_4 = \begin{bmatrix} 1.1658 \\ 1.9022 \\ 3.4388 \end{bmatrix} \tag{4.27}$$

3. The covariance matrix of each class is determined as the 3-dimensional identity matrix $\mathbf{I}_3$.

4. Using the means from Step 1 and the covariances in Step 2, 100 samples are generated from the Gaussian distribution for each class to obtain a dataset with 400 samples.

5. Three information-less features are then added to the dataset obtained in Step 4: for all samples, three linear combinations (whose coefficients are in the interval $[0, 1]$) of the three features are obtained randomly to form three extra dimensions. Then, 16 random features are generated from the normal distribution $\mathcal{N}(0, 1)$ for each sample. Due to the fact that random features are appended to the original 3-dimensional samples, the covariance matrices are unequal among the classes.

6. The resulting dataset has 20 dimensions.

The above procedures are based on the method of generating artificial datasets by Guyon [130]. The number of classes K for the two artificial datasets are chosen in such a way as to allow visualisation in $K - 1$ dimensions after LDR.

Table 4.1: List and characteristics of datasets
K is the number of classes, d is the dimensionality of the dataset, and n is the number of data points in the dataset.

Dataset	Label	d	n	K
Diabetes	(a)	8	768	2
Glass	(b)	9	214	6
Cleveland Heart	(c)	13	297	2
Vehicles	(d)	18	846	4
Image Segmentation (Statlog)	(e)	18	2310	7
Ionosphere	(f)	33	351	2
SPECTF Heart	(g)	44	267	2
Zernike Moments	(h)	47	2000	10
Optical Digits	(i)	62	5620	11
United States Postal Service	(j)	256	9298	10

4.2.3 Experimental procedure

First, the predictors are rescaled to the range $[0, 1]$. Then, dimensionality reduction is performed using the proposed algorithm and existing algorithms. This is followed by 10 independent trials of 10-fold cross-validation. On each training set after LDR, four Bayesian classifiers, namely QDA, LDA [5], the Naive Bayes classifier, and the GLD classifier (Algorithm 1, page 37) are trained. The average classification accuracy on the test set is then evaluated using the four classifiers.

For the proposed algorithm, the Bayes error is minimised using Algorithm (5) on page 78. A learning rate of $\alpha = 0.1$ over 10000 iterations is employed, although the procedure is terminated prematurely when the difference between two consecutive values of the Bayes error is less than 10^{-6}. The gradient

descent procedure uses only one initial solution at each step, given by:

$$w_i^{(0)} = \mathbf{S}_L^{-1}(\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_L)$$
$$t_i^{(0)} = \ln(\tau) + \frac{1}{2}(\bar{\mathbf{x}}_i^\top \mathbf{S}_L^{-1}\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_L^\top \mathbf{S}_L^{-1}\bar{\mathbf{x}}_L) \tag{4.28}$$

for every $i \in \mathcal{L}$, where

$$\mathbf{S}_L = \sum_{j\in\mathcal{L}\setminus\{i\}} \pi_i \mathbf{S}_j, \quad \bar{\mathbf{x}}_L = \frac{1}{|\mathcal{L}-1|}\sum_{j\in\mathcal{L}\setminus\{i\}} \bar{\mathbf{x}}_j \quad \text{and} \quad \tau = \frac{\sum_{j\in\mathcal{L}\setminus\{i\}} \pi_j}{\pi_i}. \tag{4.29}$$

The performance of the proposed algorithm is then compared with the following: PCA, F-LDR, M-LDR, C-LDR, as well as the case where there is no dimensionality reduction and the full dimensionality is used (No-LDR). Note that for PCA, F-LDR, M-LDR and C-LDR, the first $\bar{\rho}$ independent vectors after the matrix decomposition are taken to form the transformation matrix $\mathbf{M}$, where $\bar{\rho}$ is the effective rank of the matrix which is eigen-decomposed in the respective procedure. In the scenario where the effective rank cannot be easily determined (because all singular values happen to be non-zero, and it is not clear what values are large enough to be considered significant ranks), the first $K-1$ independent vectors are used. The results of these experiments can be seen in Tables 4.2, 4.4, 4.3 and 4.5. For every test dataset, the Wilcoxon's signed rank test is performed at a significant level of 0.01 to check for any significant differences between the classification accuracy of the best performing algorithm and those of the remaining algorithms. Based on the test results, an asterisk has been indicated against a value if that value is not statistically different from the best value in bold.

To provide a more meaningful perspective on the utility of the proposed algorithm, in Table 4.6, comparison is provided for the performance of the proposed LDR algorithm followed by the GLD classifier with that of the linear Support Vector Machine (SVM) that uses no dimensionality reduction. The SVM is implemented with the MATLAB function *fitcsvm* using the default settings for a linear SVM. The GLD classifier is used in this comparison because it shows the better performance among the two linear classifiers used in this section: LDA and the GLD classifier.

4.2.4 Results and discussions

The results of the dimensionality reduction for all the simulated algorithms for dataset DS1 with $K = 3$ classes are shown in Figs 4.1, 4.2, 4.3, 4.4 and 4.5, while the results for dataset DS2 with $K = 4$ classes are shown in Figs 4.6, 4.7, 4.8, 4.9 and 4.10. No axis labels are provided for these figures because the

datasets are artificially generated and the features have no specific meaning.

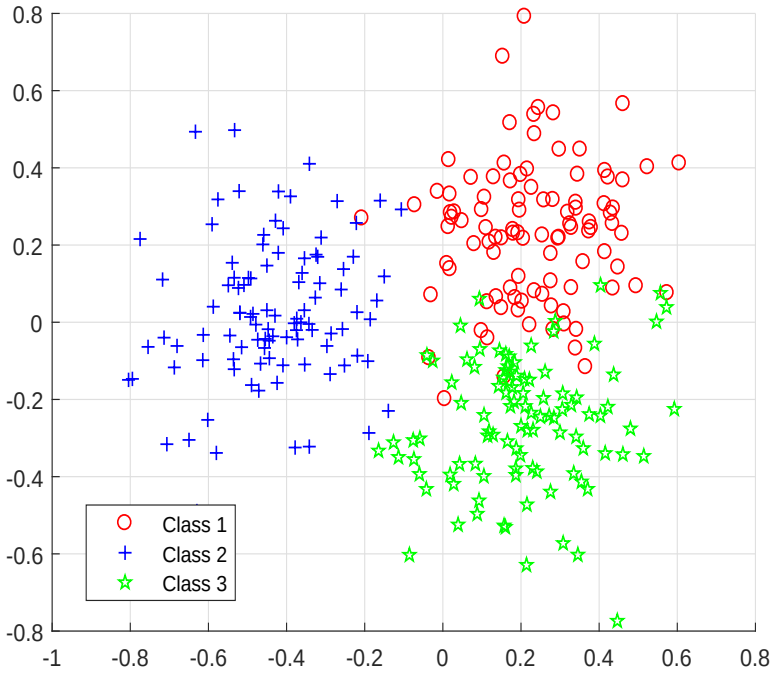

Figure 4.1: LDR using PCA on DS1

Table 4.2: Average classification accuracy (%) using QDA
Best values are in bold. The values with asterisk (*) are those that are statistically indiscernible from the best values based on the Wilcoxon's signed rank test at a significance level of 0.01.

Dataset	No-LDR	PCA	F-LDR	M-LDR	C-LDR	M-GLD
(DS1)	89.17	93.67	94.57	95.60	92.67	**96.80**
(DS2)	89.58	93.75	94.60	94.35	94.80	**95.50**
(a)	74.20	68.57	64.55	77.69	78.09	**78.17**
(b)	55.16	55.25	54.63	**61.89**	54.09	55.21
(c)	82.00	79.56	85.06	84.82	85.22*	**85.49**
(d)	**85.27**	45.78	63.80	78.61	75.97	81.95
(e)	88.82	89.32	90.36	88.09	89.82	**93.56**
(f)	87.51	61.14	89.37	90.43	89.12	**91.12**
(g)	79.42	72.61	79.42	82.96	78.42	**84.24**
(h)	80.14	77.76	79.08	77.47	82.96	**84.93**
(i)	96.44	96.00	96.30	92.55	79.94	**97.69**
(j)	88.09	91.67	88.96	57.70	61.37	**92.94**

The results in Tables 4.2, 4.3, 4.4 and 4.5 show that the classification performance can be improved

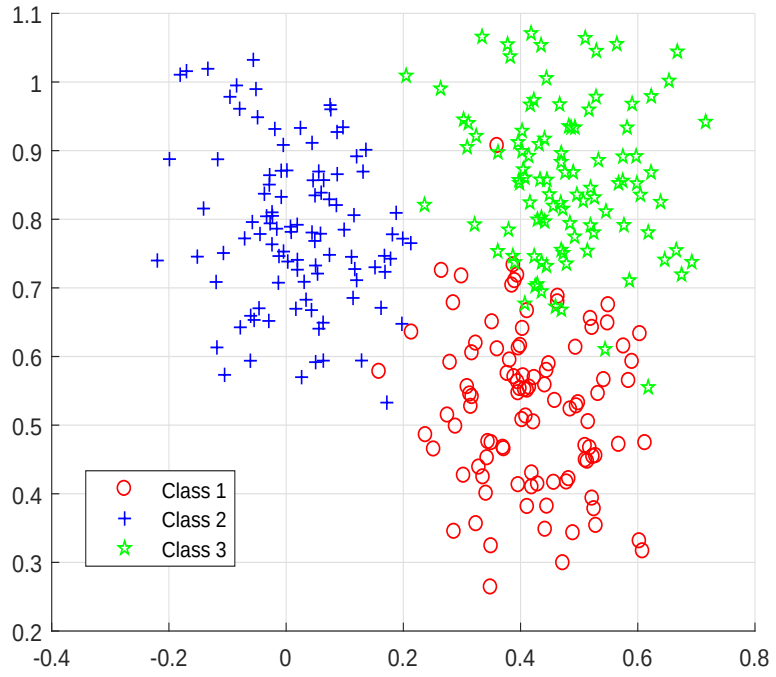

Figure 4.2: LDR based on Fisher's criterion on DS1

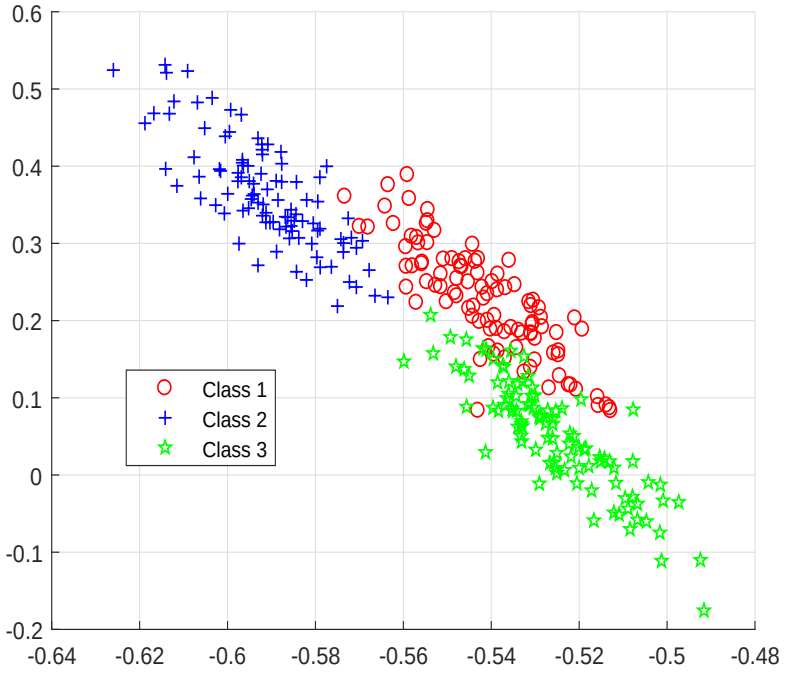

Figure 4.3: LDR based on Mahalanobis distance criterion on DS1

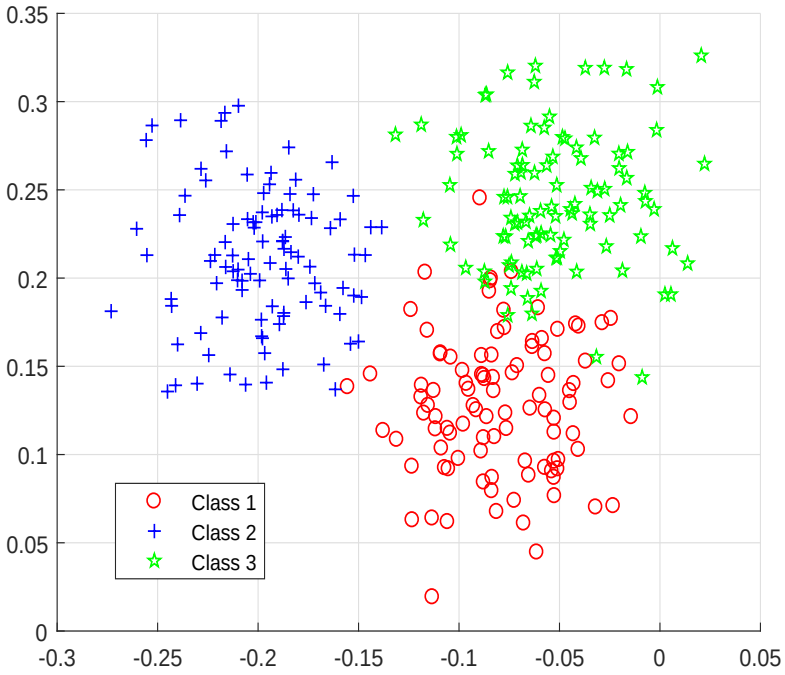

Figure 4.4: LDR based on Chernoff criterion on DS1

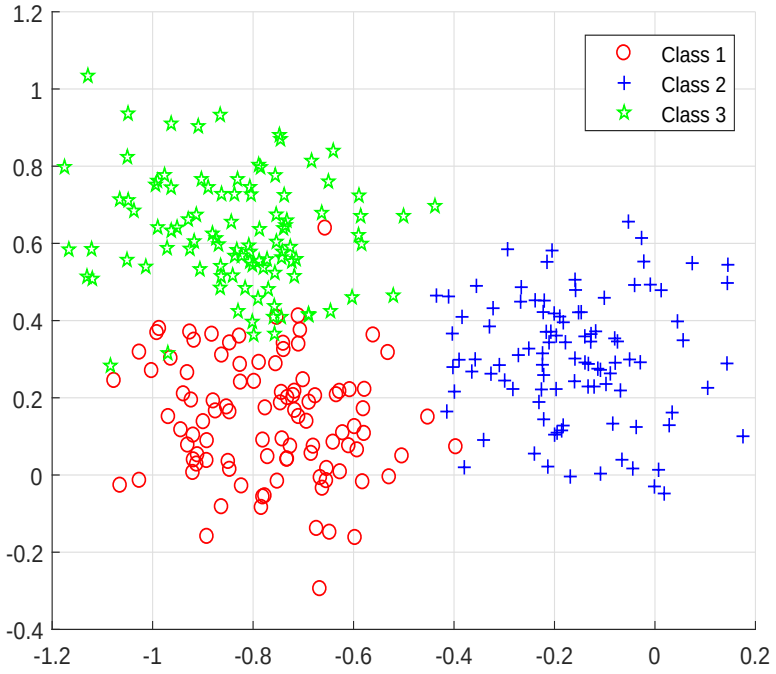

Figure 4.5: LDR based on M-GLD on DS1

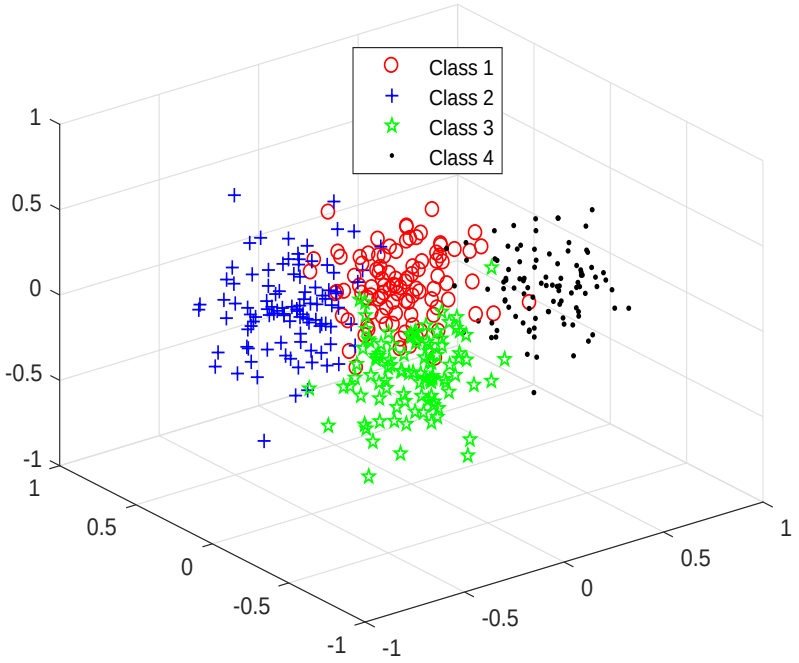

Figure 4.6: LDR using PCA on DS2

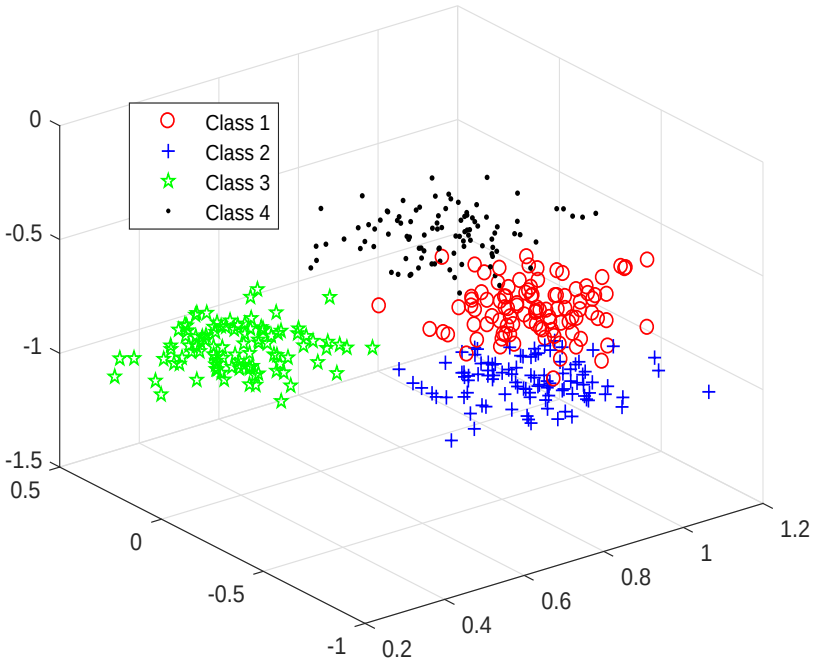

Figure 4.7: LDR based on Fisher's criterion on DS2

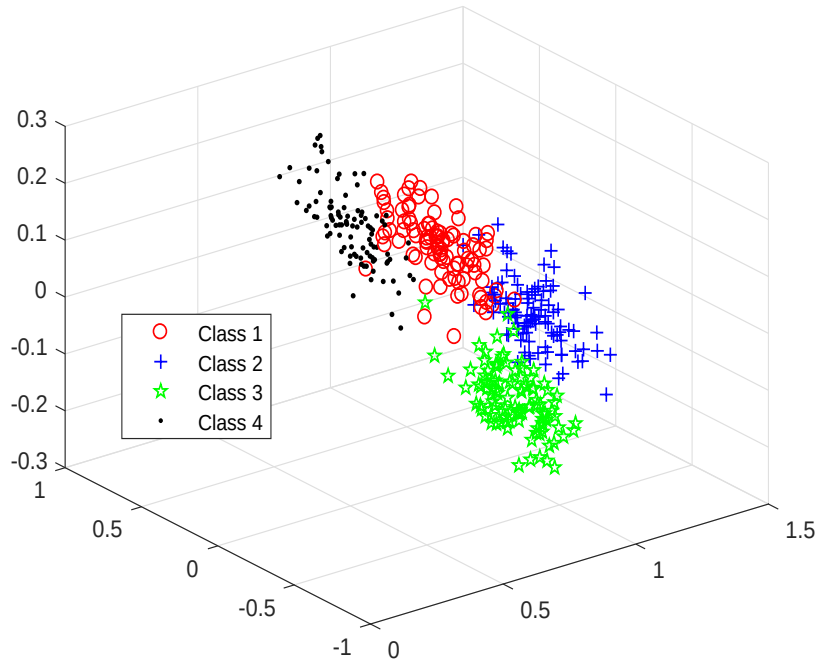

Figure 4.8: LDR based on Mahalanobis distance criterion on DS2

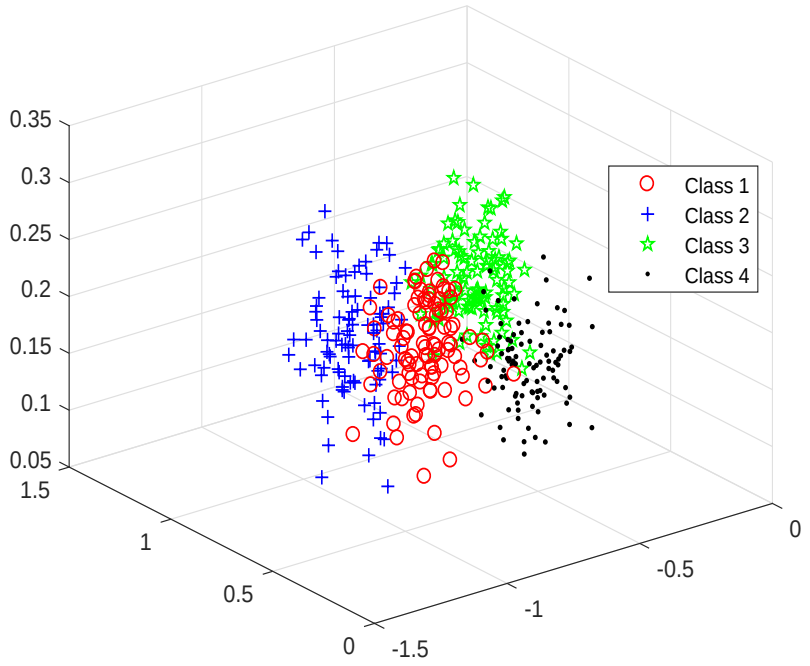

Figure 4.9: LDR based on Chernoff criterion on DS2

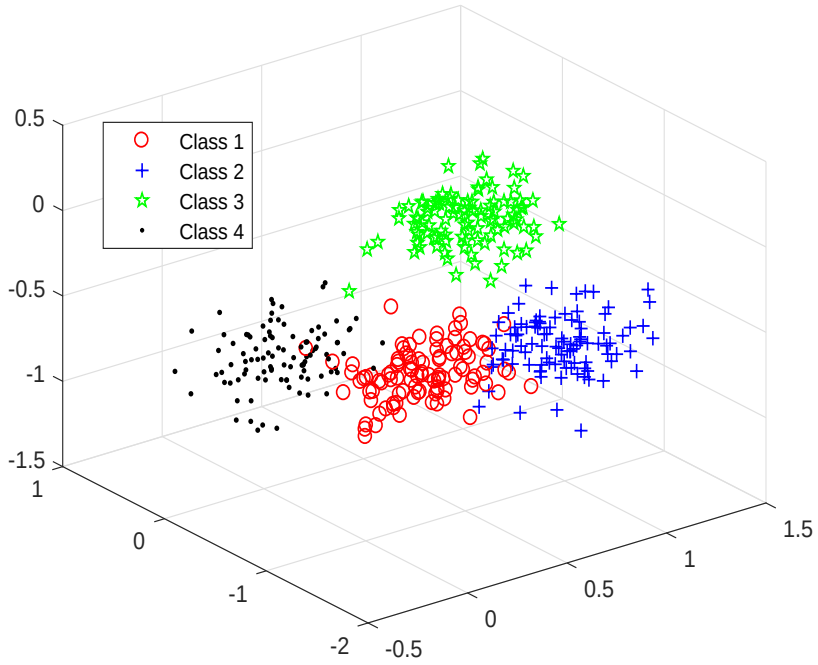

Figure 4.10: LDR based on M-GLD on DS2

Table 4.3: Average classification accuracy (%) using Naive Bayes classifier
Best values are in bold. The values with asterisk (*) are those that are statistically indiscernible from the best values based on the Wilcoxon's signed rank test at a significance level of 0.01.

Dataset	No-LDR	PCA	F-LDR	M-LDR	C-LDR	M-GLD
(DS1)	92.40	93.43	94.73	96.00	92.53	**96.90**
(DS2)	94.48	94.13	94.73	94.50	94.08	**96.38**
(a)	73.49	67.37	66.45	77.49*	**77.97**	77.71*
(b)	59.89	59.84	**70.91**	62.29	54.53	61.65
(c)	79.64	81.47	53.25	84.53	85.03*	**85.06**
(d)	61.44	54.43	65.41	73.80	76.47	**80.68**
(e)	89.98	89.13	90.48	81.07	89.16	**92.10**
(f)	90.19	79.57	89.62	89.71	**91.39**	90.76
(g)	73.64	78.85	79.42	83.35*	82.18	**83.84**
(h)	72.55	70.28	71.89	69.61	81.24	**82.25**
(i)	82.12	92.68	93.37	88.52	76.42	**96.04**
(j)	56.63	85.48	81.22	52.86	66.93	**92.09**

Table 4.4: Average classification accuracy (%) using LDA
Best values are in bold. The values with asterisk (*) are those that are statistically indiscernible from the best values based on the Wilcoxon's signed rank test at a significance level of 0.01.

Dataset	No-LDR	PCA	F-LDR	M-LDR	C-LDR	M-GLD
(DS1)	92.13	93.20	94.80	95.97	92.83	**97.00**
(DS2)	93.65	94.18	94.85	94.73	95.38*	**95.68**
(a)	77.39	68.37	65.11	77.76	77.87	**78.36**
(b)	63.09	60.66	61.08	62.76	59.26	**65.09**
(c)	83.55	79.61	85.29	84.89	85.36*	**85.84**
(d)	78.19	47.18	58.31	75.02	75.65	**79.30**
(e)	**91.48**	83.68	87.83	88.24	88.84	90.33
(f)	86.72	54.50	73.73	90.13*	74.92	**90.62**
(g)	75.27	79.42	79.42	84.77	79.30	**85.55**
(h)	81.79	70.35	74.02	70.78	82.00	**83.92**
(i)	95.32	91.49	93.06	88.86	56.81	**96.09**
(j)	**91.62**	84.29	81.62	27.41	60.96	90.21

Table 4.5: Average classification accuracy (%) using R-GLD classifier
Best values are in bold. The values with asterisk (*) are those that are statistically indiscernible from the best values based on the Wilcoxon's signed rank test at a significance level of 0.01.

Dataset	No-LDR	PCA	F-LDR	M-LDR	C-LDR	M-GLD
(DS1)	93.23	93.47	94.67	95.80	96.17	**97.13**
(DS2)	90.03	81.95	91.75	92.65	92.83	**95.63**
(a)	77.59	68.54	64.23	77.73	**78.07**	78.06*
(b)	62.89	59.85	**65.27**	63.10	57.37	62.52
(c)	83.78	79.62	85.22*	84.75	85.26*	**85.48**
(d)	80.75	46.72	60.87	75.98	76.30	**81.11**
(e)	94.99*	87.58	89.20	90.42	92.58	**95.01**
(f)	86.95	58.86	90.12	90.31	89.15	**91.48**
(g)	73.56	79.42	79.42	**84.10**	79.42	83.96*
(h)	83.77	75.20	78.51	75.45	83.71	**86.14**
(i)	**98.17**	94.54	95.08	93.12	80.69	97.16
(j)	94.46*	89.87	86.32	50.70	66.25	**94.52**

Table 4.6: Average classification accuracy (%): M-GLD+G-GLD vs Linear SVM
Best values are in bold. The values for both algorithms for all datasets are statistically different based on the Wilcoxon's signed rank test at a significance level of 0.01.

Dataset	SVM	M-GLD+G-GLD
(DS1)	93.40	**97.13**
(DS2)	93.98	**95.63**
(a)	76.94	**78.06**
(b)	57.79	**62.52**
(c)	83.45	**85.48**
(d)	74.11	**81.11**
(e)	92.89	**95.01**
(f)	87.73	**91.48**
(g)	79.60	**83.96**
(h)	82.94	**86.14**
(i)	**98.28**	97.16
(j)	**95.81**	94.52

after linear dimensionality reduction. This is due to the fact that the original datasets tend to show multicollinearity as well as contain noisy and useless features that cause overfitting. As an example, the two artificial datasets may be considered. These datasets have 14 and 16 noisy features respectively which can be easily overfit using a quadratic classifier such as QDA. Due to this, an improved performance is seen by employing any of the LDR procedures on these artificial datasets using the QDA classifier. Furthermore, the results also show the superiority of the proposed M-GLD LDR procedure over the existing C-LDR, M-LDR, F-LDR and PCA procedures on all four classifiers. This relative performance of the algorithms may be evident from the degree of class separation that is yielded by the M-GLD and other existing LDR procedures in Figure 4.1, 4.2, 4.3, 4.4 4.5, 4.6, 4.7, 4.8, 4.9 and 4.10.

Regarding the real-world datasets, Table 4.2 shows that the proposed M-GLD algorithm achieves the highest classification accuracy on 8 out of the 10 real-world datasets tested using a QDA classifier, as compared to the remaining LDR procedures. This superior performance is most marked on datasets (e), (g), (h), (i) and (j). On dataset (d), using the full dimensionality results in the best classification accuracy using the LDA classifier, as LDR seems to lose useful classification information. Yet, among the five dimensionality reduction techniques, the M-GLD achieves the best classification accuracy on this dataset.

A similar performance is seen in Table 4.3. The proposed algorithm once again achieves superior classification accuracy on 8 out of the 10 real-world datasets using the Naive Bayes classifier, with datasets,

(d), (e), (h), (i) and (j) showing the most significant performance. Furthermore, the relative superior performance of the proposed M-GLD LDR procedure in Tables 4.4 and 4.5 is consistent with the fact that the proposed algorithm is well-suited for Bayesian classification, unlike the existing heteroscedastic LDA approaches.

While the M-GLD is consistently superior on the four Bayesian classifiers, it will be noted that for the same linearly-reduced data arising from a given LDR procedure, the relative performance of the remaining LDR algorithms depends on the choice of the classifier. For example, the M-LDR procedure tends to outperform the F-LDR and C-LDR procedures for most of the test datasets using a QDA classifier, whereas the C-LDR procedure tend to yield superior classification performance over the C-LDR and F-LDR on most of the datasets using the Naive Bayes classifier and the two linear classifiers. It is also worth noting that when no dimensionality reduction is employed, the performance of the No-LDR procedure using a QDA classifier is significantly improved using a linear classifier such as LDA or the GLD. For instance, using the GLD classifier, the No-LDR is shown to be competitive with the LDR procedures on datasets (d), (e), (h), (i) and (j). This is because if a dataset does indeed contain repeated or noisy features, without linearly reducing the dimensionality, a linear classifier tends to reduce overfitting as compared to a quadratic classifier.

On the whole, the GLD classifier outperforms the LDA classifier, notably on datasets (d), (e), (f), (h), (i) and (j), due to the fact that the former does not make the assumption of homoscedasticity which is rarely seen in most real-world datasets. Moreover, in Table 4.6, it is seen that the proposed LDR algorithm, together with the GLD classifier, easily outperforms the linear SVM on 10 out of the 12 datasets tested.

The poor performance of PCA on most of the datasets across all the classifiers, e.g., datasets (b), (c), (d), (f), (h) and (i), is due to the fact that PCA reduces the dimensionality of the data without taking into account the class discriminatory information in the data. Also, as there is no guarantee that the choice of the first $K-1$ independent vectors (if there is no obvious effective rank) are those that mostly preserve the classification information in the reduced space for PCA, M-LDR and C-LDR, a reduction to some other dimensionality $q \neq K-1$ might result in a better classification performance. Yet, these algorithms do not provide the optimal dimensionality q to which to reduce the data. Thus, extensive trial and error is required to obtain an optimal dimensionality in these approaches. The proposed algorithm, on the other hand, obtains satisfactory classification performance after a reduction to a dimensionality of $K-1$, which is the optimal dimensionality required for Bayesian classification.

Even though the proposed algorithm has been shown to be superior to the existing procedures on

the datasets tested in terms of classification accuracies, it is built on an assumption of normality of the data in each of the K classes. Yet, since a lot of physical data tend to be nearly-normally distributed [10], the proposed M-GLD algorithm is well suited for a lot of applications particularly those involving measurement errors such as machine fault diagnosis or those involving physical measurements such as accelerometer-based human activity recognition. However, for data that are radically non-normal, the M-GLD procedure is expected to perform relatively poorly, as the Bayes error is not guaranteed to be minimised. Also, while the proposed procedure has been derived for Bayesian classification and is thus expected to perform well on Bayesian classifiers such as LDA, QDA and the Naive Bayes classifier, it is not suitable for other discriminative classifiers such as the SVM or logistic regression. Moreover, the M-GLD algorithm requires the successive construction of $(K^2 + K - 4)/2$ classifiers. Thus, the time complexity is quadratic in K which can be rather computationally costly for a dataset having a large number of classes. Nevertheless, this is not prohibitive, as the average training time for dataset (i), which has the largest number of classes, with $K = 11$ is $4.5s$.

4.3 Chapter summary

This chapter has introduced a supervised linear dimensionality reduction (LDR) procedure that extends linear discriminant analysis (LDA) to the multi-class case while maintaining heteroscedasticity. This procedure requires the construction of multiple GLD classifiers by sequentially minimising the Bayes error in the multi-class scenario. The optimality conditions for this minimisation problem have been derived in the chapter. For a K-class problem, the procedure is shown to require $(K^2 + K - 4)/2$ classifiers. This is first described for the special cases of $K = 2$ and $K = 3$, before it is detailed for any general value of K. The resulting procedure known as M-GLD projects the data unto a $K - 1$ dimensional space which is optimal for Bayesian classification. Thus, the algorithm is shown to be well-suited for Bayesian classifiers, such as quadratic discriminant analysis (QDA), Naive Bayes and LDA classifiers.

The chapter concludes with an extensive experimental validation of the proposed LDR algorithm on 2 artificial datasets and 10 real-world UCI datasets. Using four different Bayesian classifiers for classification, the M-GLD is shown to yield the best classification accuracies on most of the datasets, as compared to the existing LDR procedures such as LDR based on PCA, Fisher's criterion, Mahalanobis distance and the Chernoff-criterion.

4.A Appendix to Chapter 4

4.A.1 Rank inequalities

Consider the dataset $\mathcal{X} \in \mathbb{R}^{d\times n}$ having n d-dimensional vectors, which are all labelled such that $\mathcal{X}$ can be partitioned into K classes thus: $\mathcal{X} = [\mathcal{D}_1, ..., \mathcal{D}_K]$, with the sample covariance matrix of the kth class given by:

$$\mathbf{\Sigma}_k = \frac{1}{n_k - 1}\sum_{i=1}^{n_k}(\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_i - \bar{\mathbf{x}}_k)^\top, \quad \forall \mathbf{x}_i \in \mathcal{D}_k \tag{4.30}$$

where $\bar{\mathbf{x}}_k$ is the sample mean of $\mathcal{D}_k$, and n_k is the number of samples in the kth class. Define a within-class covariance matrix $\mathbf{S}_w$ and a between-class covariance matrix $\mathbf{S}_b$ as:

$$\mathbf{S}_w = \sum_{k=1}^{K}\pi_k\mathbf{\Sigma}_k \quad \text{and} \quad \mathbf{S}_b = \sum_{k=1}^{K}\pi_k(\bar{\mathbf{x}}_k - \bar{\mathbf{x}})(\bar{\mathbf{x}}_k - \bar{\mathbf{x}})^\top. \tag{4.31}$$

where

$$\bar{\mathbf{x}} = \sum_{k=1}^{K}\pi_k\mathbf{x}_k \quad \text{and} \quad \pi_k = \frac{n_k}{n}. \tag{4.32}$$

Let $\rho(\mathbf{A})$ denote the rank of matrix $\mathbf{A}$.

Theorem 2. $\rho(\mathbf{\Sigma}_k) \leq n_k - 1$

Proof. The subadditivity property of a rank [131] holds that for any two comformable matrices $\mathbf{A}$ and $\mathbf{B}$,

$$\rho(\mathbf{A} + \mathbf{B}) \leq \rho(\mathbf{A}) + \rho(\mathbf{B}) \tag{4.33}$$

Since the rank of a matrix is unchanged by a scale factor, it follows that

$$\rho(\mathbf{\Sigma}_k) \leq \sum_{i=1}^{n_k}\rho\big[(\mathbf{x} - \bar{\mathbf{x}}_k)(\mathbf{x} - \bar{\mathbf{x}}_k)^\top\big] \tag{4.34}$$

However, note that for any two conformable matrices, $\mathbf{A}$ and $\mathbf{B}$ [131]

$$\rho(\mathbf{AB}) \leq \min\big[\rho(\mathbf{A}), \rho(\mathbf{B})\big] \tag{4.35}$$

Therefore,

$$\rho\big[(\mathbf{x} - \bar{\mathbf{x}}_k)(\mathbf{x} - \bar{\mathbf{x}}_k)^\top\big] \leq \min\big[\rho(\mathbf{x} - \bar{\mathbf{x}}_k), \rho(\mathbf{x} - \bar{\mathbf{x}}_k)^\top\big] \tag{4.36}$$

Since $\mathbf{x} - \bar{\mathbf{x}}_k$ is a vector, it has a rank of at most 1. Therefore (4.36) becomes,

$$\rho\big[(\mathbf{x} - \bar{\mathbf{x}}_k)(\mathbf{x} - \bar{\mathbf{x}}_k)^\top\big] \leq 1 \tag{4.37}$$

Substituting (4.37) into (4.34), results in the following:

$$\rho(\boldsymbol{\Sigma}_k) \leq n_k \tag{4.38}$$

However, this bound can be tightened by noticing that the sample covariance matrix is mean-centred and hence it contains only $n_k - 1$ independent terms in the sum. To verify this, observe that:

$$\sum_{i=1}^{n_k}(\mathbf{x}_i - \bar{\mathbf{x}}_k) = \sum_{i=1}^{n_k}\mathbf{x}_i - \sum_{i=1}^{n_k}\bar{\mathbf{x}}_k = 0 \tag{4.39}$$

Therefore,

$$\sum_{i=1}^{n_k-1}(\mathbf{x}_i - \bar{\mathbf{x}}_k) = -(\mathbf{x}_{n_k} - \bar{\mathbf{x}}_k) \tag{4.40}$$

Since the covariance of $\boldsymbol{\Sigma}_k$ in (4.30) can be rewritten as:

$$\boldsymbol{\Sigma}_k = \frac{1}{n_k - 1}\Bigg[\sum_{i=1}^{n_k-1}(\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_i - \bar{\mathbf{x}}_k)^\top + (\mathbf{x}_{n_k} - \bar{\mathbf{x}}_k)(\mathbf{x}_{n_k} - \bar{\mathbf{x}}_k)^\top\Bigg], \quad \forall \mathbf{x}_i \in \mathcal{D}_k, \tag{4.41}$$

substituting (4.40) into (4.41) yields the following:

$$\boldsymbol{\Sigma}_k = \frac{1}{n_k - 1}\Bigg[\sum_{i=1}^{n_k-1}(\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_i - \bar{\mathbf{x}}_k)^\top - \sum_{i=1}^{n_k-1}(\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_{n_k} - \bar{\mathbf{x}}_k)^\top\Bigg] \quad \forall \mathbf{x}_i \in \mathcal{D}_k \tag{4.42}$$

which can be simplified as:

$$\boldsymbol{\Sigma}_k = \frac{1}{n_k - 1}\sum_{i=1}^{n_k-1}(\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_i - \bar{\mathbf{x}}_{n_k})^\top, \quad \forall \mathbf{x}_i \in \mathcal{D}_k \tag{4.43}$$

Since $\boldsymbol{\Sigma}_{\boldsymbol{k}}$ has been expressed as a sum of $n_k - 1$ rank-1 matrices, it follows from (4.33),

$$\rho(\boldsymbol{\Sigma}_k) \leq n_k - 1 \tag{4.44}$$

□

Theorem 3. $\rho(\boldsymbol{S}_w) \leq n - K$

Proof. $\mathbf{S}_w$ is as defined in (4.31). Therefore, from (4.34),

$$\rho(\mathbf{S}_w) \leq \sum_{k=1}^{K} \rho(\pi_k \boldsymbol{\Sigma}_k) \tag{4.45}$$

Since scalar multiplication does not change the rank of a matrix, and $\boldsymbol{\Sigma}_k$ has been shown to have a rank of at most $n_k - 1$,

$$\rho(\mathbf{S}_w) \leq \sum_{k=1}^{K} n_k - 1 = n - K \tag{4.46}$$

since $\sum_{k=1}^{K} n_k = n$. Therefore, $\rho(\mathbf{S}_w) \leq n - K$ □

Theorem 4. $\rho(\boldsymbol{S}_b) \leq K - 1$

Proof. Notice that $\mathbf{S}_b$ (4.31) is a covariance matrix of the form $\boldsymbol{\Sigma}_k$ (4.30) where

$$K \equiv n_k, \quad \bar{\mathbf{x}}_k \equiv \mathbf{x}_i, \quad \bar{\mathbf{x}} \equiv \bar{\mathbf{x}}_k \quad \text{and} \quad \pi_k \equiv \frac{1}{n_k - 1} \tag{4.47}$$

Therefore, since $\rho(\boldsymbol{\Sigma}_k) \leq n_k - 1$, in the same way, $\rho(\mathbf{S}_b) \leq K - 1$ □

Theorem 5. $\rho(\boldsymbol{S}_w^{-1}\boldsymbol{S}_b) \leq K - 1$

Proof. From (4.33),

$$\rho(\mathbf{S}_w^{-1}\mathbf{S}_b) \leq \min[\rho(\mathbf{S}_w^{-1}), \rho(\mathbf{S}_b)] \tag{4.48}$$

But $\rho(\mathbf{S}_b) \leq K - 1$, and $\rho(\mathbf{S}_w^{-1}) = \rho(\mathbf{S}_w) \leq n - K$ since an invertible matrix has full rank and so does its inverse. Therefore,

$$\rho(\mathbf{S}_w^{-1}\mathbf{S}_b) \leq \min[n - K, K - 1] \tag{4.49}$$

However, in order for the sample covariance matrix per class $\boldsymbol{\Sigma}_k$ given by (4.30) to be non-zero, notice that there has to be at least two samples per class, i.e., $n_k \geq 2$. This implies that for all K classes, $n \geq 2K$, and therefore,

$$n - K \geq K > K - 1 \tag{4.50}$$

Hence,

$$\rho(\mathbf{S}_w^{-1}\mathbf{S}_b) \leq K - 1 \tag{4.51}$$

□

4.A.2 Optimality conditions for minimisation of Bayes error

The first-order optimality condition for the minimisation of ϵ_i requires the gradient of ϵ_i to be zero, i.e.,

$$\nabla\epsilon_i(\mathbf{w}_i, t_i) = \left[\frac{\partial\epsilon_i}{\partial\mathbf{w}_i^\top}, \frac{\partial\epsilon_i}{\partial t_i}\right]^\top = \mathbf{0} \tag{4.52}$$

From (4.20), it can be shown that:

$$\frac{\partial\epsilon_i}{\partial\mathbf{w}_i} = \pi_i\left(\frac{1}{\sqrt{2\pi}}e^{-\frac{z_i^2}{2}}\frac{\partial z_i}{\partial\mathbf{w}_i}\right) - \sum_{j\in\mathcal{L}\setminus\{i\}}\pi_j\left(\frac{1}{\sqrt{2\pi}}e^{-\frac{z_j^2}{2}}\frac{\partial z_j}{\partial\mathbf{w}_i}\right) \tag{4.53}$$

where $\partial z_k/\partial\mathbf{w}_i$ can be obtained from (4.21) as:

$$\frac{\partial z_k}{\partial\mathbf{w}_i} = \frac{-\sigma_k\bar{\mathbf{x}}_k - z_k\mathbf{S}_k\mathbf{w}_i}{\sigma_k^2} \quad \text{for every} \quad k\in\mathcal{L}. \tag{4.54}$$

Therefore,

$$\frac{\partial\epsilon_i}{\partial\mathbf{w}_i} = \frac{\pi_i}{\sqrt{2\pi}}e^{-\frac{z_i^2}{2}}\left(\frac{-\sigma_i\bar{\mathbf{x}}_i - z_i\mathbf{S}_i\mathbf{w}_i}{\sigma_i^2}\right) - \sum_{j\in\mathcal{L}\setminus\{i\}}\frac{\pi_j}{\sqrt{2\pi}}e^{-\frac{z_j^2}{2}}\left(\frac{-\sigma_j\bar{\mathbf{x}}_j - z_j\mathbf{S}_j\mathbf{w}_i}{\sigma_j^2}\right) \tag{4.55}$$

Also,

$$\frac{\partial\epsilon_i}{\partial t_i} = \frac{\pi_i}{\sqrt{2\pi}}e^{-\frac{z_i^2}{2}}\frac{\partial z_i}{\partial t_i} - \sum_{j\in\mathcal{L}\setminus\{i\}}\frac{\pi_j}{\sqrt{2\pi}}e^{-\frac{z_j^2}{2}}\frac{\partial z_j}{\partial t_i} \tag{4.56}$$

where,

$$\frac{\partial z_k}{\partial t_j} = \frac{1}{\sigma_k}, \quad \text{for every} \quad k\in\mathcal{L} \tag{4.57}$$

which can also be obtained from (4.21). Therefore,

$$\frac{\partial\epsilon_i}{\partial t_i} = \frac{\pi_i}{\sqrt{2\pi}\sigma_i}e^{-\frac{z_i^2}{2}} - \sum_{j\in\mathcal{L}\setminus\{i\}}\frac{\pi_j}{\sqrt{2\pi}\sigma_j}e^{-\frac{z_j^2}{2}} \tag{4.58}$$

By equating the gradient to zero, (4.55) yields the following:

$$\left(\sum_{j\in\mathcal{L}\setminus\{i\}}\frac{\pi_j}{\sigma_j}e^{-\frac{z_j^2}{2}}\frac{z_j}{\sigma_j}\mathbf{S}_j - \frac{\pi_i}{\sigma_i}e^{-\frac{z_i^2}{2}}\frac{z_i}{\sigma_i}\mathbf{S}_i\right)\mathbf{w}_i = \frac{\pi_i}{\sigma_i}e^{-\frac{z_i^2}{2}}\bar{\mathbf{x}}_i - \sum_{j\in\mathcal{L}\setminus\{i\}}\frac{\pi_j}{\sigma_j}e^{-\frac{z_j^2}{2}}\bar{\mathbf{x}}_j \tag{4.59}$$

while (4.58) results in:

$$\frac{\pi_i}{\sigma_i}e^{-\frac{z_i^2}{2}} = \sum_{j\in\mathcal{L}\setminus\{i\}}\frac{\pi_j}{\sigma_j}e^{-\frac{z_j^2}{2}} \tag{4.60}$$

Substituting (4.58) into (4.55), the following is obtained:

$$\Bigg(\sum_{j\in\mathcal{L}\setminus\{i\}}\frac{\pi_j}{\sigma_j}e^{-\frac{z_j^2}{2}}\frac{z_j}{\sigma_j}\mathbf{S}_j-\sum_{j\in\mathcal{L}\setminus\{i\}}\frac{\pi_j}{\sigma_j}e^{-\frac{z_j^2}{2}}\frac{z_i}{\sigma_i}\mathbf{S}_i\Bigg)\mathbf{w}_i=$$
$$\sum_{j\in\mathcal{L}\setminus\{i\}}\frac{\pi_j}{\sigma_j}e^{-\frac{z_j^2}{2}}\bar{\mathbf{x}}_i-\sum_{j\in\mathcal{L}\setminus\{i\}}\frac{\pi_j}{\sigma_j}e^{-\frac{z_j^2}{2}}\bar{\mathbf{x}}_j, \tag{4.61}$$

i.e.,

$$\sum_{j\in\mathcal{L}\setminus\{i\}}\frac{\pi_j}{\sigma_j}e^{-\frac{z_j^2}{2}}\left(\frac{z_j}{\sigma_j}\mathbf{S}_j-\frac{z_i}{\sigma_i}\mathbf{S}_i\right)\mathbf{w}_i=\sum_{j\in\mathcal{L}\setminus\{i\}}\frac{\pi_j}{\sigma_j}e^{-\frac{z_j^2}{2}}(\bar{\mathbf{x}}_i-\bar{\mathbf{x}}_j) \tag{4.62}$$

$\mathbf{w}_i$ may then be obtained as:

$$\mathbf{w}_i=\Bigg[\sum_{j\in\mathcal{L}\setminus\{i\}}\frac{\pi_j}{\sigma_j}e^{-\frac{z_j^2}{2}}\left(\frac{z_j}{\sigma_j}\mathbf{S}_j-\frac{z_i}{\sigma_i}\mathbf{S}_i\right)\Bigg]^{-1}\sum_{j\in\mathcal{L}\setminus\{i\}}\frac{\pi_j}{\sigma_j}e^{-\frac{z_j^2}{2}}(\bar{\mathbf{x}}_i-\bar{\mathbf{x}}_j) \tag{4.63}$$

But for the fact that z_i and z_j are functions of t_i as can be seen from (4.21), $\mathbf{w}_i$ could have been solved for iteratively from (4.63) starting from an initial solution. To overcome this problem, t_i can be solved for from (4.58), expressing it as a function of $\mathbf{w}_i$, to allow for the iterative solution of $\mathbf{w}_i$ from (4.63).

From (4.58), the following can be derived:

$$\ln\left(\frac{\pi_i}{\sigma_i}\right)-\frac{z_i^2}{2}=\ln\sum_{j\in\mathcal{L}\setminus\{i\}}\frac{\pi_j}{\sigma_j}e^{-\frac{z_j^2}{2}} \tag{4.64}$$

If the cardinality of $\mathcal{L}$, $|\mathcal{L}|>2$, the right hand side of (4.64) is a logarithmic sum of exponentials, and (4.64) has no closed-form solution. Note, however, that (4.64) can be rewritten as

$$\ln\left(\frac{\pi_i}{\sigma_i}\right)-\frac{z_i^2}{2}-\ln(|\mathcal{L}|-1)=\ln\left(\frac{1}{|\mathcal{L}|-1}\sum_{j\in\mathcal{L}\setminus\{i\}}\frac{\pi_j}{\sigma_j}e^{-\frac{z_j^2}{2}}\right). \tag{4.65}$$

Then, as a consequence of Jensen's inequality,

$$\ln\left(\frac{\pi_i}{\sigma_i}\right)-\frac{z_i^2}{2}-\ln(|\mathcal{L}|-1)\geq\frac{1}{|\mathcal{L}|-1}\sum_{j\in\mathcal{L}\setminus\{i\}}\ln\left(\frac{\pi_j}{\sigma_j}\right)-\frac{z_j^2}{2}, \tag{4.66}$$

By approximating (4.64) using the lower bound in (4.66), we obtain:

$$\frac{1}{|\mathcal{L}|-1}\sum_{j\in\mathcal{L}\setminus\{i\}}\frac{z_j^2}{2}-\frac{z_i^2}{2}+\ln\left(\frac{\pi_i}{\sigma_i}\right)-\frac{1}{|\mathcal{L}|-1}\sum_{j\in\mathcal{L}\setminus\{i\}}\ln\left(\frac{\pi_j}{\sigma_j}\right)-\ln(|\mathcal{L}|-1)$$
$$=0 \tag{4.67}$$

which can be simplified to:

$$\left(\frac{1}{|\mathcal{L}|-1}\sum_{j\in\mathcal{L}\setminus\{i\}}\frac{1}{\sigma_j^2}-\frac{1}{\sigma_i^2}\right)t_i^2+2\left(\frac{\mu_i}{\sigma_i^2}-\frac{1}{|\mathcal{L}|-1}\sum_{j\in\mathcal{L}\setminus\{i\}}\frac{\mu_j}{\sigma_j^2}\right)t_i-\frac{\mu_i^2}{\sigma_i^2}$$
$$+\frac{1}{|\mathcal{L}|-1}\sum_{j\in\mathcal{L}\setminus\{i\}}\frac{\mu_j^2}{\sigma_j^2}+2\left[\ln\left(\frac{\pi_i}{\sigma_i}\right)-\frac{1}{|\mathcal{L}|-1}\sum_{j\in\mathcal{L}\setminus\{i\}}\ln\left(\frac{\pi_j}{\sigma_j}\right)-\ln(|\mathcal{L}|-1)\right], \tag{4.68}$$

a quadratic in t_i. Thus, by solving (4.68), t_i can be expressed as a function of $\mathbf{w}_i$ through the variables μ_i, μ_j, σ_i and σ_j. Being a quadratic, there are two solutions to (4.68). Yet, by choosing the solution that yields the smaller Bayes error, (4.68) is then expressed solely in terms of $\mathbf{w}_i$, so that $\mathbf{w}_i$ can be solved for iteratively.

Note two computational issues with the above procedure. First, (4.63) is derived from the first-order optimality condition. Therefore, there is no certainty that iteratively solving for $\mathbf{w}_i$ would converge to a local minimum of ϵ_i, as the optimality condition of (4.52) from which (4.63) is derived is also satisfied for a local maximum or a saddle point. For this reason, the iterative procedure requires the use of several different initial solutions to improve the chances of convergence to a local minimum. Moreover, there is no guarantee that (4.68) has any real solution, for any given dataset $\mathcal{D}$.

Chapter 5

LDA for flowmeter fault diagnosis[1]

This chapter follows from Section 1.1, and it describes the applicability of the linear classification and dimensionality methods proposed in Chapters 3 and 4 to flowmeter diagnostics. The need for flowmeter diagnostics is described first. This is followed by a description of experiments performed with the project partner, National Engineering Laboratory (NEL), Glasgow, United Kingdom, to investigate the relationship between diagnostic variables and the health states of four flowmeters. Using the data from these experiments, the chapter describes how the machine learning techniques of linear dimensionality reduction (LDR) and linear classification may be used to improve the diagnostic capabilities of the flowmeters. Using the proposed methods in Chapters 3 and 4, diagnostic accuracies of between 97.2% and 100% are achieved on four flowmeter types.

5.1 Need for flowmeter diagnostics

As mentioned in Chapter 1, condition-based management (CBM) of flowmeters promises to mitigate the problem of incorrect measurements in the oil and gas industry, avoid costly shut-downs of a flow rig, and reduce the recalibration frequency of a flowmeter. To enable CBM, the condition of the flowmeter has to be known at all times. Fortunately, with the recent emergence of flowmeters that provide diagnostic information that are secondary to the primary flow measurement, it is possible to infer the health state of a flowmeter by monitoring the values of the diagnostic variables.

However, making sense of the wealth of diagnostic data accessible from a given flowmeter usually requires end-user expertise, which is not often available in the oilfield. Thus, it becomes necessary for flowmeter manufacturers to provide an expert system as part of the meter's diagnostic capabilities to eliminate the need for end-user expertise. Such an expert system would summarise the plethora of diagnostic variables from a given flowmeter and relate them to known health states of the meter. Since flowmeters are used in varied environments, establishing a relationship between the diagnostic variables

[1]Most of the work presented in this chapter first appeared in: K. S. Gyamfi, J. Brusey, A. Hunt and E. Gaura, "Linear dimensionality reduction for classification via a sequential Bayes error minimisation with an application to flowmeter diagnostics," *Expert Systems with Applications* (2017), vol. 91, Sep. 2017, pp. 252-262.

and the different health states of the meter requires testing the flowmeter under ideal and non-ideal conditions in a test facility.

Only one representative flowmeter from a line of similarly-manufactured flowmeters needs to undergo these tests in a test facility on the assumption that any relationships derived from the tests would be applicable to all similar flowmeters. (This assumption is tested in Section 5.4). Moreover, only the most prevalent health states of the meter need to be simulated. Along this line, NEL have conducted a series of experiments, in collaboration with three flowmeter manufacturers. Four representative liquid ultrasonic flowmeter types were tested under ideal and non-ideal conditions to investigate how the diagnostic variables change from their baseline values under three conditions, namely: non-ideal installations, presence of a second phase, and wax-filled ports.

Training the proposed model or expert system can be formulated as a classification task where it is desired to classify the flowmeter under one of K classes (or health states), based on a set of diagnostic variables that form the feature vector $\mathbf{x}$ (see Section 2.1). For the liquid ultrasonic flowmeters (USM) used in the NEL experiments, the K classes include: healthy, waxing, installation effects, and presence of second phase [1].

Nevertheless, the diagnostic variables available from a given flowmeter can be varied and many, so that the feature vector $\mathbf{x}$ lives in a rather high-dimensional space. For the USM shown in Figure 5.1, the diagnostic variables include the flow profile, symmetry, cross-flow, swirl angle, flow velocity (as measured by each of the eight paths), speed of sound (as measured by each of the eight paths), signal strength (as measured at both ends of each of the eight paths), turbulence (as measured by each of the eight paths), signal quality (as measured at both ends of each of the eight paths), gain (as measured at both ends of each of the eight paths) and transit time (as measured at both ends of each of the eight paths). Thus, the feature vector $\mathbf{x}$ has 92 diagnostic variables in total.

However, some of the diagnostic variables like the swirl angle or turbulence have been shown experimentally to contain little or no classification information required to classify the meter under the most prevalent health states, such as those simulated in the NEL experiments [1]. Besides, it is not known if the diagnostic variables measured from all eight paths in Figure 5.1 are useful for classification, or whether the average of all eight paths would suffice. The effect of having too many nuisance features is that the learning model can over-fit the data leading to poor diagnostic accuracy, especially if the data is noisy, which is inevitable due to an imperfect measurement system. Linear dimensionality reduction (LDR) alleviates this problem, and if reduction to two or three dimensions is possible, LDR makes visualisation and analysis of the diagnostics data easier for flowmeter operators.

Furthermore, some of these diagnostic variables tend to be correlated. For example, the speed of sound, flow velocity and transit time have a known dependence [132]. Dimensionality reduction is therefore useful to reduce the effects of multicollinearity from the features, before the data is trained for statistical classification.

Performing LDR for the purpose of flowmeter diagnostics involves linearly reducing the dimensionality of the high-dimensional diagnostics data, while maximising the class discriminatory information contained in it, so that the diagnostic capabilities of a given flowmeter are not compromised. Still, given the wide-ranging dimensionality reduction procedures in existence, there is no obvious choice of which procedure would yield satisfactory performance in terms of class separation. Thus, this chapter evaluates the performance of existing LDR procedures such as principal components analysis (PCA) and LDR based on Fisher's criterion (F-LDR), Chernoff criterion (C-LDR) and Mahalanobis distance (M-LDR) (see Section 2.2), as compared to the proposed multi-class Gaussian linear discriminant (M-GLD) LDR procedure.

5.2 Description of NEL experiments [1]

5.2.1 Meter description

As indicated earlier, four different liquid USMs were tested in all. These are denoted as Meter A, Meter B, Meter C and Meter D. Meter A has 8 paths, as indicated in Figure 5.1, while Meters B, C and D have 4 paths as shown in Figure 5.2.

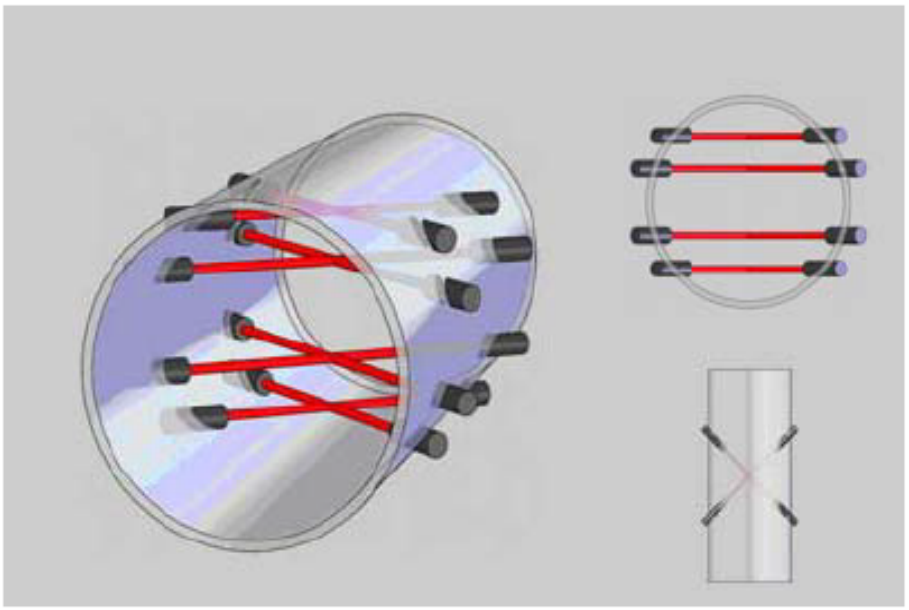

Figure 5.1: An 8-path ultrasonic flowmeter transducer configuration [1]

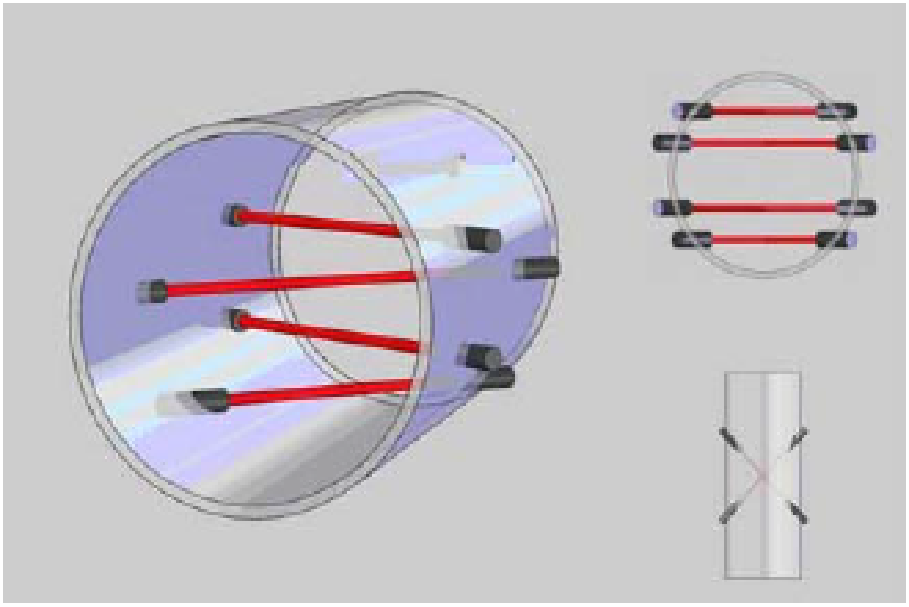

Figure 5.2: A 4-path ultrasonic flowmeter transducer configuration [1]

All four meters were tested to monitor their diagnostic values under the following conditions: installation effects, waxing, and two-phase flow, with the exception of Meter A that was tested only for installation effects. Before and after each test, the diagnostic variables of the healthy meter are recorded over the various reference flow rates to indicate the baseline performance of the meter.

5.2.2 Installation effects tests

These tests were aimed at exploring the impact an imperfect installation of a flowmeter may have on the performance of the meter. The imperfect installations simulated included both vertical and horizontal misalignment of the flange of the flowmeter, as well as the use of a different pipe schedule[2] at the upstream port. It was expected that changes in some of the diagnostic variables might indicate this health state of the meter. Each of the flowmeters was tested over its specified flow range, and single readings of the measured flow rate and all the diagnostic variables were taken at each reference flow rate.

5.2.3 Waxing tests

These tests were aimed at investigating the effect of wax build-up (due to the accumulation of hydrocarbon fluids in the transducer ports) on the performance of liquid USMs, and the changes in the diagnostic variables. Paraffin or candle wax was heated until molten and used to fill the transducer ports. The

[2]The pipe schedule is a number that expresses the thickness of the walls of a pipe.

waxing tests included the waxing of upstream ports only, as well as waxing of all transducer ports. Tests were conducted at 8 flow rates across the range of 40 to 140 litres/second. At each reference flow rate, single readings of the measured flow rate and the diagnostic variables were taken.

5.2.4 Two-phase flow tests

Since the performance of a liquid USM can be adversely affected by the presence of a second phase [1], these tests were aimed at investigating the relationship between the meter's performance, diagnostic variables and the presence of a second phase. The second phase used was gas. A gas measurement system was employed to allow the injection of gas into the liquid whose flow rate was being measured by the liquid USM. Over flow rates that range from 25 to 100 litres/second, different amounts of gas were injected, whose gas volume fractions (GVF) ranged from 0.1% to 10%. The diagnostic variables were recorded for every GVF and reference flow rate settings.

5.2.5 Diagnostic variables [1]

The diagnostic variables recorded from these tests are explained below:

1. Flatness ratio or profile factor: This parameter compares the amount of flow on the outer paths to the centre paths. It quantifies how peaked or flattened the flow profile is.

2. Profile symmetry: It compares the amount of flow on the top planes to the bottom planes.

3. Swirl: It describes the amount of transversal flow that is rotating in the pipe. Typically, this describes flow profile in a pipe after an out of plane double elbow. A positive number means swirl flow is clockwise looking downstream.

4. Cross-flow: This parameter describes the amount of transversal flow that is generating a double swirl pattern with individual vortices in the top and bottom of the pipe. Typically, this describes flow profile after a single bend. The sign of the number indicates the direction of the cross-flow. The cross-flow compares velocities in the chords in one plane to velocities in the plane at right angles. In good flow conditions, the ratio should be close to unity.

5. Standard deviation: This describes the stability of flow measurement in each path, and is sometimes used as a measure of turbulence in the flow.

6. Speed of sound (SoS): It is calculated from transit-time measurements. The calculated value is compared to a theoretical value. The SoS should be approximately the same for each path.

7. Gain: This is a measure of how much amplification is being applied by the electronics to the received ultrasonic signal to ensure an effective level. This is controlled by the automatic gain control (AGC) function built into the software. The AGC function is programmed to maintain a constant amplitude of received signal.

8. Performance or signals percentage: This value describes how many of the ultrasonic signals are acceptable to be used for flow measurement. The value is displayed as a percentage indicating how many of the transmitted signals are being used.

9. Signal to noise ratio (SNR): The SNR is the ratio of the amplitude of the received signal to the amplitude of the background noise. The signal amplitude should be significantly greater to ensure good measurement.

Other diagnostic variables include the transit time and the flow velocity.

5.3 Characteristics of diagnostics data

Following the receipt of the diagnostics data gathered from the above tests from NEL, the procedures applied to pre-process the data prior to training an expert system for flowmeter diagnostics are detailed in this section.

The first task was to clean the data. Data cleaning consisted in deleting all rows with missing data, removing outliers, as well as deleting any diagnostics variables that were constant for all tests or that were not recorded for each health state of a given flowmeter; the entire data collected under the "Installation effects" state for Meter B were deleted, as there were too many columns with missing data. The original and processed data can be accessed at the University of California, Irvine (UCI) machine learning repository `http://archive.ics.uci.edu/ml/datasets.html` or at `http://cogentee.coventry.ac.uk/~kojo/`. The cleaned data are summarised in Table 5.1 for each of the flowmeters tested.

With this data, a learning model is trained to learn the relationship between the diagnostic variables and the health states of the flowmeters. Specifically, an expert system in the form of a statistical classifier is trained using the above data to make predictions on the health state of a flowmeter given a particular set of diagnostic variables.

Prior to training the classifier, however, LDR is performed to reduce the possibility of over-fitting in order to improve the diagnostic accuracy of a given flowmeter. This is done using the M-GLD LDR procedure proposed in Chapter 4 on page 78, after which the linearly reduced data is trained for stat-

Table 5.1: USM diagnostics data
The table shows the number of data samples collected for each health state of a given flowmeter. n is the total number of data samples for a meter, d represents the number of diagnostic variables, and K, the number of health states. "−" indicates the non-availability of data.

	Meter A	Meter B	Meter C	Meter D
Healthy	35	19	54	51
Gas injection	−	24	23	23
Installation effects	52	−	54	55
Waxing	−	24	49	51
n	87	92	181	180
d	36	51	43	43
K	2	3	4	4

istical classification using the Gaussian linear discriminant optimised with gradient descent (G-GLD) as proposed in Chapter 3 on page 39.

Since both of the proposed models are based on linear discriminant analysis (LDA), it is important to first test for the within-class normality and the homoscedasticity assumptions that LDA makes, i.e., the data gathered for every health state of a given flowmeter is normally distributed, and that the covariance matrices of the data are the same across all health states for that flowmeter. For this purpose, the Royston test is used to test for multivariate normality [133, 134], while the Box M test is used to test for homoscedasticity [135, 136]. The results of these tests are shown in Tables 5.2 and 5.3.

Table 5.2: Royston test
This table indicates whether or not the null hypothesis of within-class normality is accepted, based on the Royston multivariate normality test at a significance level of 0.01. "−" indicates the non-availability of data.

	Meter A	Meter B	Meter C	Meter D
Healthy	Yes	Yes	Yes	Yes
Gas injection	−	No	No	No
Installation effects	Yes	−	Yes	Yes
Waxing	−	No	No	No

The results of the normality and homoscedasticity tests show that the data for the "Healthy" and "Installation effects" health states of all four flowmeters tend to be normally distributed, whereas the "Waxing" and "Gas injection" health states do not show normality. Moreover, with the exception of Meter A, none of the meters has a common covariance across all its simulated health states. While satisfying the assumptions of normality and homoscedasticity is not critical to the application of LDA, any unsatisfactory performance can be attributed to the extent of violation of the assumptions. In the

Table 5.3: M Box test
This table indicates whether or not the null hypothesis of homoscedasticity is accepted, based on the M Box test for equality of covariances at a significance level of 0.01.

	Homoscedasticity accepted
Meter A	Yes
Meter B	No
Meter C	No
Meter D	No

same way, the proposed algorithms are expected to show an improved performance over the original LDA procedure for heteroscedastic data, since they account for the violation of the homoscedasticity assumption.

Finally, since the different diagnostic variables take on different ranges of values, it is important to first normalise all the variables in order to improve the speed and accuracy of the learning algorithm [137]. This is done by using the following transformation:

$$p' = \frac{p - p_{\max}}{p_{\max} - p_{\min}}, \tag{5.1}$$

where p is any given diagnostic parameter, $p_{\max}$ and $p_{\min}$ are respectively the maximum and minimum values of that parameter across all the data samples for a given flowmeter, and p' is the normalised value of p. The formula in (5.1) ensures that all the variables are rescaled to the range $[0, 1]$.

5.4 Cross-validation performance for USMs

In order to show that any relationships derived from the tests would be applicable to similarly-manufactured flowmeters, cross-validation is employed [138]. Cross-validation involves partitioning the diagnostics data for a given flowmeter into a number of folds: one fold, known as the *test set*, is used for testing or validating the model, while the remaining folds, known as the *training set* is used to train the model or expert system. The test set is therefore an unseen independent set of samples which represents the diagnostics data expected from similarly-manufactured flowmeters in the field.

5.4.1 Linear dimensionality reduction (LDR)

The proposed LDR technique, M-GLD (Algorithm 5 on page 78), is applied to the normalised data to reduce the dimensionality to $K - 1$ for each of the flowmeters: the data for Meter A is reduced to 1

dimension, that for Meter B is reduced to 2 dimensions, and those for Meters C and D are reduced to 3 dimensions (see Table 5.1). In Figure 5.3, the 1-dimensional representation of the diagnostics data for Meter A is shown. The implication of this linear transformation is that, instead of the 36 diagnostic variables that were recorded for Meter A, only one parameter (dimension) is necessary to tell the "Healthy" state from the "Installation effects" state of Meter A; note that this one parameter is a linear combination of all 36 diagnostic variables, and it does not rely solely on one. The existing algorithms of PCA, F-LDR, M-LDR, and C-LDR are also used to reduce the dimensionality of Meter A's diagnostic data to 1, as shown in Figure 5.3

In Figures 5.4, 5.5, 5.6, 5.7 and 5.8, the 3-dimensional representation of the diagnostic data for Meter C are shown using the proposed M-GLD and existing LDR procedures. However, only the "Healthy" and "Installation effects" health states are shown, since they are the ones that show the most marked difference in the degree of class separation among the LDR algorithms.

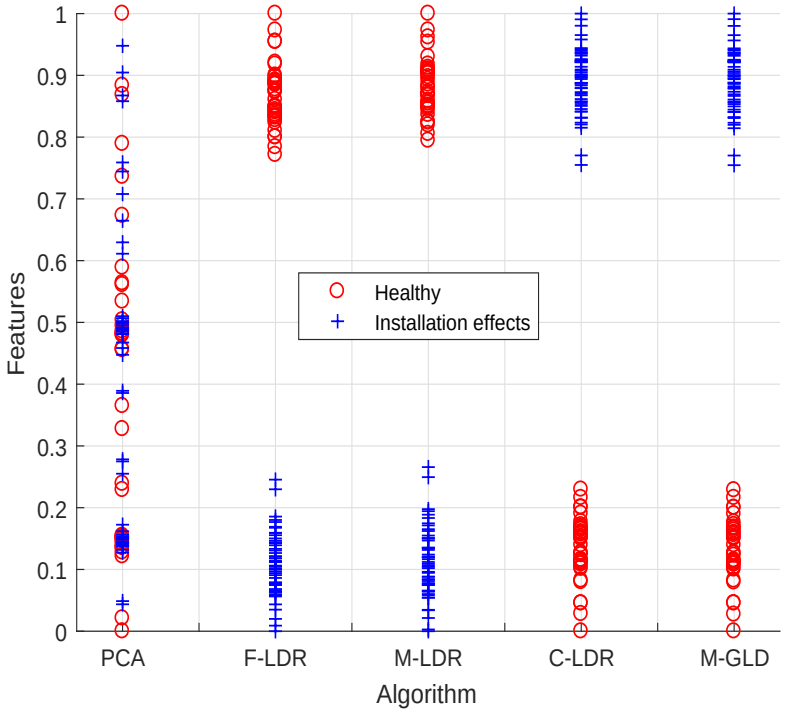

Figure 5.3: LDR performance on Meter A diagnostics data

5.4.2 Statistical classification

Following dimensionality reduction with the proposed M-GLD and existing LDR procedures, the diagnostics data for the flowmeters are trained for statistical classification, after which the classification (or

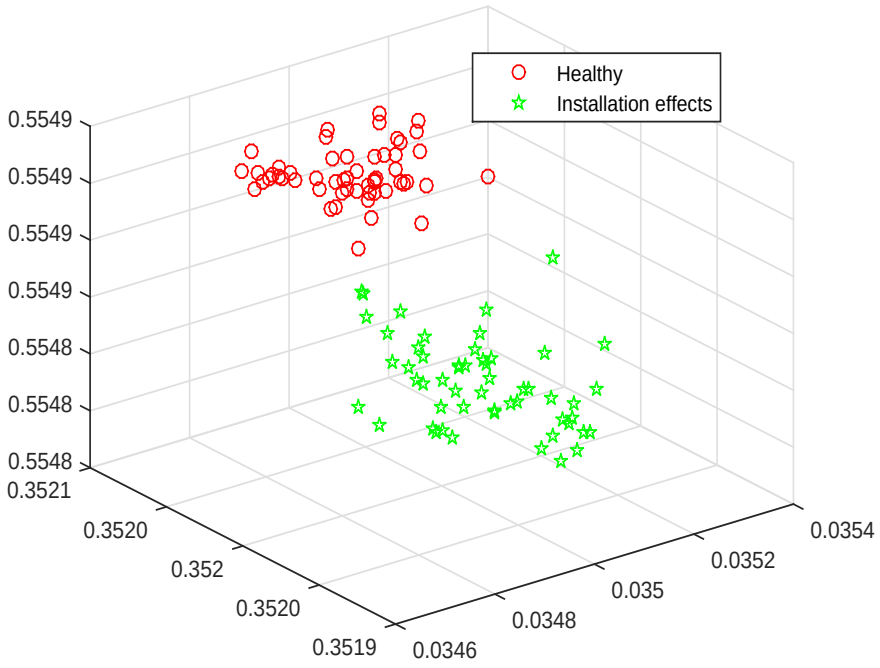

Figure 5.4: LDR performance on Meter C diagnostics data: M-GLD

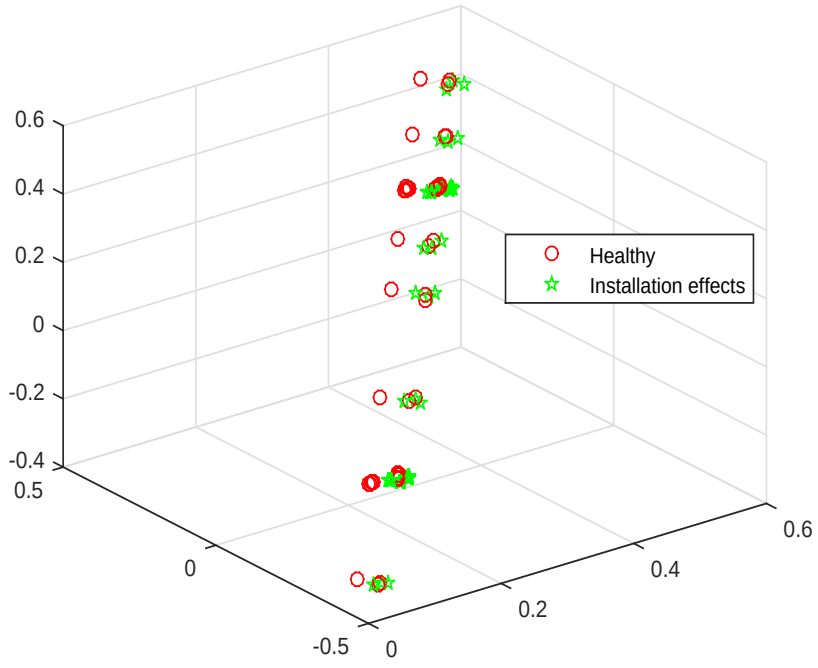

Figure 5.5: LDR performance on Meter C diagnostics data: PCA

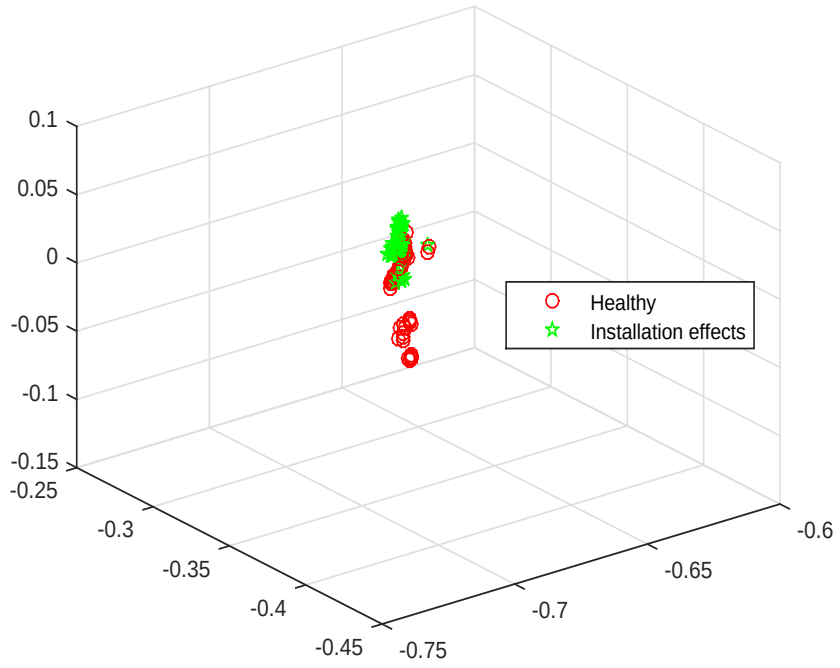

Figure 5.6: LDR performance on Meter C diagnostics data: F-LDR

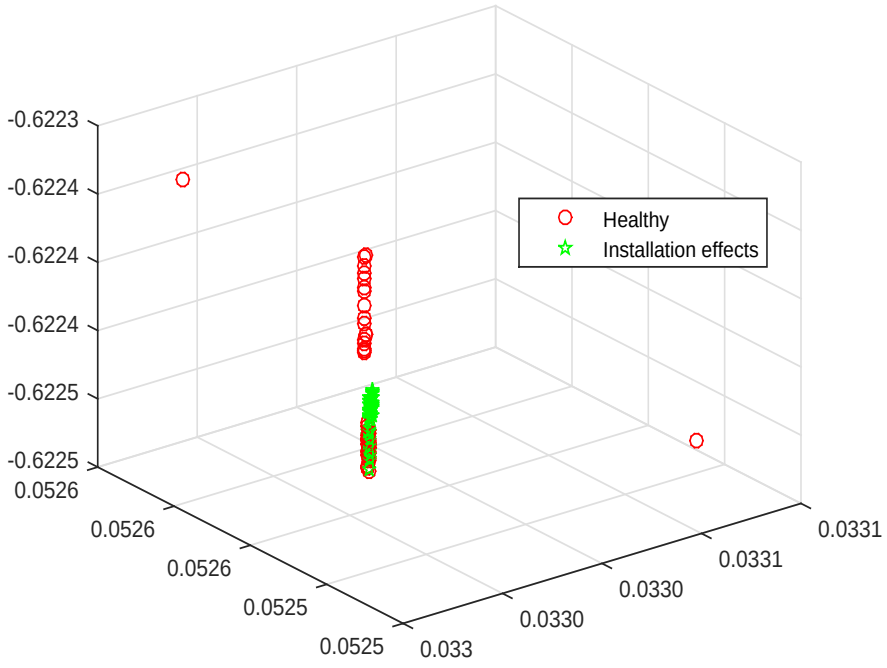

Figure 5.7: LDR performance on Meter C diagnostics data: M-LDR

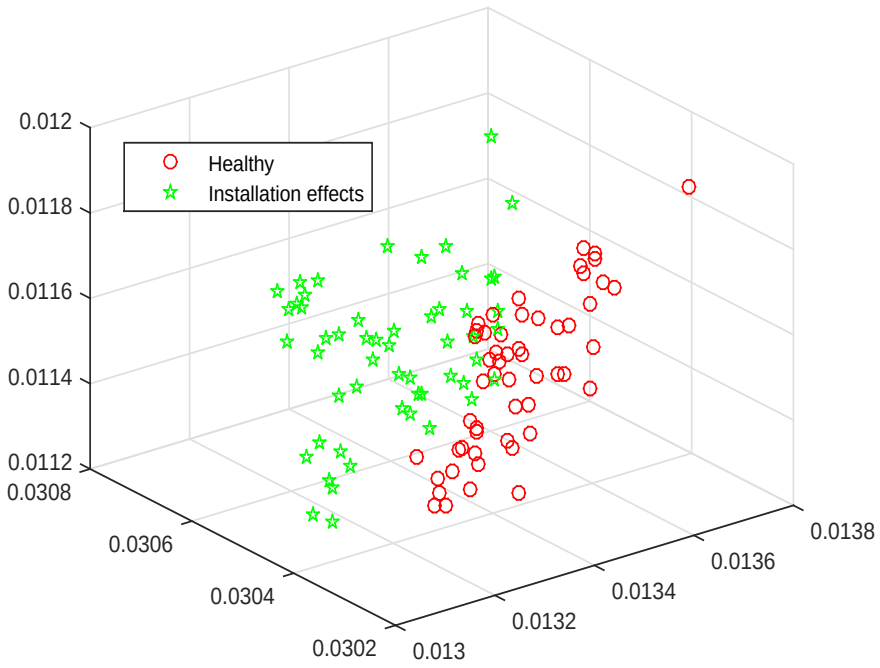

Figure 5.8: Proposed M-GLD LDR performance on Meter C diagnostics data: C-LDR

diagnostic) accuracy is evaluated. For this purpose, 10 independent trials of 10−fold cross validation are performed. 10-fold cross validation involves randomly partitioning the entire data for a given flowmeter into 10 folds. For every fold $\mathcal{F}$ out of the 10 folds, the data contained in the remaining 9 folds are used to train the algorithm. The classifier obtained from training is then tested on the fold $\mathcal{F}$ that was excluded from training. On each test fold $\mathcal{F}$, the correctly classified samples are tallied and expressed as a percentage of the total samples in the fold; this is the classification accuracy.

The classifier is trained using the G-GLD algorithm (Algorithm 2), as well as the original LDA and QDA procedures (Section 2.1) for comparison. For Meters B, C and D which have more than two classes, the One-vs-One multi-class classification strategy is used (See Section 2.1.4). For all the algorithms, the prior probabilities of a meter's health states (classes) are estimated from their empirical distribution in the data, even though in practice, the healthy state of the meter may be far more probable than the unhealthy states; if the true prior probabilities are known, they can easily be substituted in place of the empirical estimates in the respective algorithms.

The results of classification using the three classifiers (QDA, LDA, G-GLD) on all flowmeters, without any dimensionality reduction, are shown in Figure 5.9. The classification results after the various LDR procedures have been applied to the original datasets are shown in Figure 5.10 to Figure 5.14.

The performance of the proposed G-GLD algorithm, after dimensionality reduction with M-GLD, is also compared with the linear Support Vector Machine (SVM) in Figure 5.15. SVM is implemented with the MATLAB function *fitcsvm* using the default settings for a linear SVM; no dimensionality reduction is performed prior to using SVM.

Note that in all the figures, the classification accuracies of the highest achieving algorithms are statistically different from those of the remaining algorithms at the 0.01 confidence level based on the Wilcoxon's signed rank test.

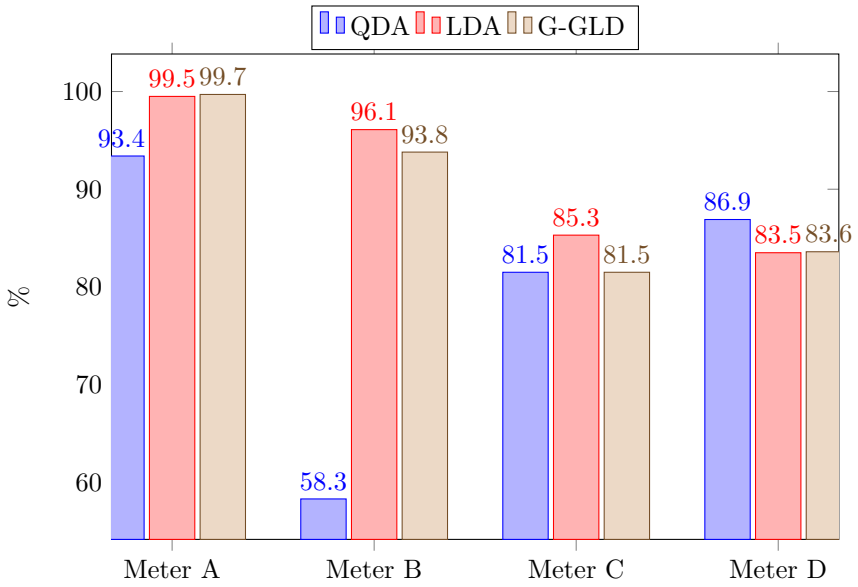

Figure 5.9: Average classification accuracy for all flowmeters with no LDR (No-LDR) (%)

5.4.3 Results and discussion

Meter A

After linearly reducing the dimensionality of Meter A's diagnostic data to 1 using the proposed M-GLD procedure, Figure 5.3 shows the two health states of the flowmeter to be very well-separated. Thus, there is no ambiguity in classifying unknown diagnostic data from this meter under the "Healthy" class or the "Installation effects" class; there is a 100% classification accuracy for Meter A using the QDA, LDA and G-GLD classifiers, after dimensionality reduction with M-GLD (Figure 5.14).

Due to the fact that Meter A's diagnostic data shows within-class normality and homoscedasticity (Tables 5.2, 5.3), as well as having $K = 2$ classes, the existing LDR procedures of F-LDR, M-LDR and C-LDR are optimal in terms of minimising the Bayes error (see Section 2.2), and hence they achieve the same

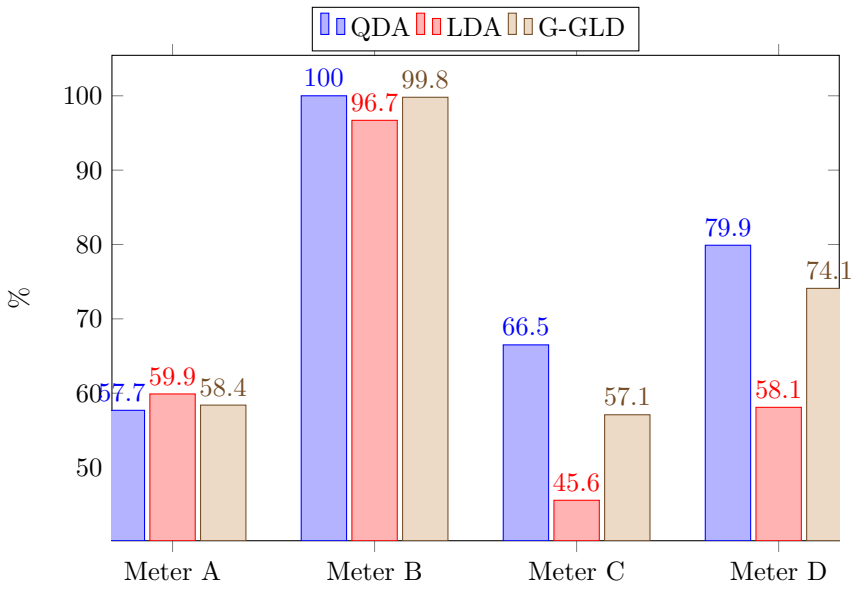

Figure 5.10: Average classification accuracy for all flowmeters after LDR by PCA (%)

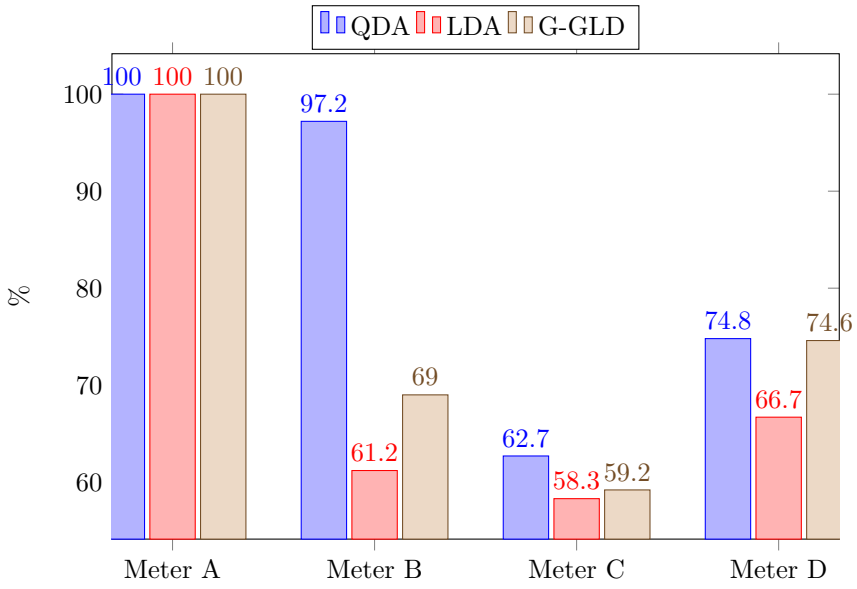

Figure 5.11: Average classification accuracy for all flowmeters after LDR by F-LDR (%)

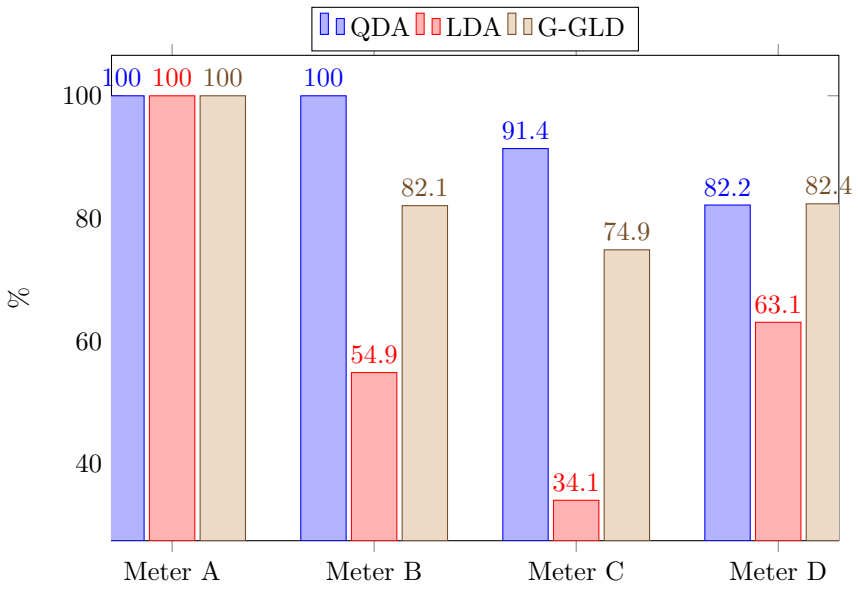

Figure 5.12: Average classification accuracy for all flowmeters after LDR by M-LDR (%)

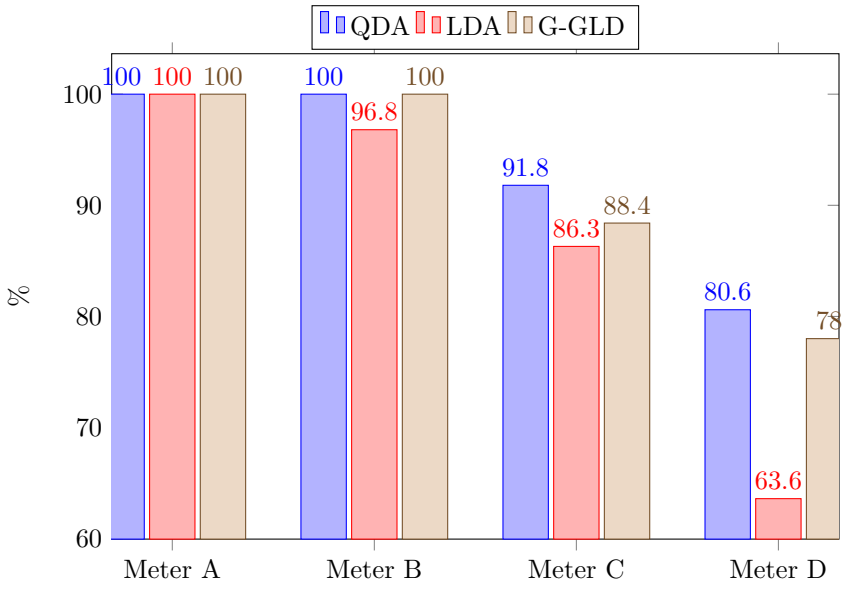

Figure 5.13: Average classification accuracy for all flowmeters after LDR by C-LDR (%)

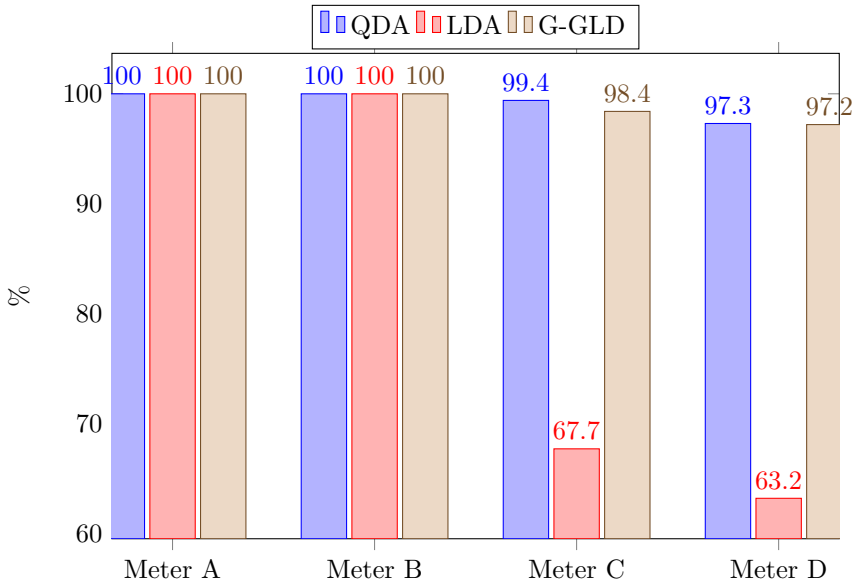

Figure 5.14: Average classification accuracy for all flowmeters after LDR by M-GLD (%)

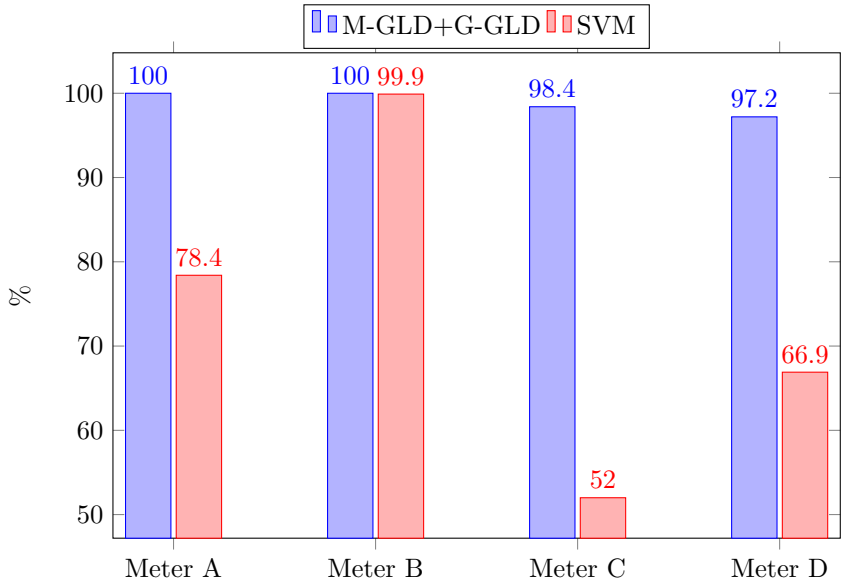

Figure 5.15: Average classification accuracy using M-GLD+G-GLD classifier and linear SVM for all flowmeters (%)

degree of class separation as the proposed M-GLD procedure, as can be seen in Figure 5.3. Consequently, these algorithms achieve classification accuracies of 100%, equivalent to the proposed M-GLD procedure, using all three classifiers, as shown in Figure 5.11, Figure 5.12 and Figure 5.13.

PCA, while being a common dimensionality reduction procedure, is unsupervised in the sense that it does not make use of the class labels, but rather, it linearly projects the 36-dimensional data of Meter A onto a linear subspace in such a way as to maximise the variance. However, maximising the variance results in significant class overlap in the 1-dimensional space, as can be seen in Figure 5.3. Therefore, the performance of PCA is unsatisfactory on Meter A using all three classifiers (Figure 5.10).

Without any dimensionality reduction, the degree of class separation obtained for Meter A by M-GLD as well as F-LDR, M-LDR and C-LDR may not be possible in the original 36-dimensional space, as the original data tends to be noisy and full of redundant features. It is for this reason that the classification accuracy for Meter A without any dimensionality reduction (No-LDR), at 93.4%, is lower than those of F-LDR, M-LDR, C-LDR and M-GLD using the QDA classifier (Figure 5.9, 5.10, 5.11, 5.12, 5.13, 5.14).

However, since linear classifiers tend to be more robust under noisy data, the classification performance for Meter A without any dimensionality reduction using the LDA and G-GLD classifiers show significant improvement over its performance with the QDA classifier (Figure 5.9).

Note that there is little performance difference between the QDA, LDA and G-GLD classifiers on Meter A for the F-LDR, M-LDR, C-LDR and M-GLD procedures. This is due to the fact that under within-class normality and homoscedasticity, both QDA and G-GLD decompose to become equivalent to the LDA classifier.

Meter B

In the case of Meter B, which does not satisfy the homoscedasticity and within-class normality assumptions of LDA (Tables 5.2, 5.3), F-LDR shows a reduced classification accuracy of 97.2% (Figure 5.11) as compared to M-GLD (Figure 5.14) and C-LDR (Figure 5.13) which both achieve 100% classification accuracy using the QDA classifier. This is because, while F-LDR assumes normally distributed and homoscedastic data, both M-GLD and C-LDR account for the violation of homoscedasticity. Though M-LDR also assumes homoscedastic data, the Mahalanobis distance criterion, which is maximised in the M-LDR procedure, results in a better class separation than Fisher's criterion, which is employed in F-LDR. As a result, M-LDR also achieves 100% classification accuracy using the QDA classifier (Figure 5.12).

PCA performs fairly well on Meter B as compared to Meter A (Figure 5.10), because for Meter B, the

directions of maximum variance onto which the original data is linearly reduced, coincide with the directions that maximise the class-discriminatory information, thereby yielding a satisfactory classification performance using the QDA classifier.

The QDA classifier performs poorly on Meter B (58.3%) without any dimensionality reduction (No-LDR) (Figure 5.9) due to the fact the original data is noisy, contains redundant features, is not normally distributed in each class, and does not have equal covariance among the classes. Using the two linear classifiers (LDA and G-GLD), the classification performance is significantly improved for No-LDR, since linear classifiers are more robust to noise and the distribution of the data, while QDA easily over-fits.

After dimensionality reduction, however, especially with M-GLD, QDA tends to not over-fit (Figure 5.14).

Note that with the exception of No-LDR, there is a general decrease in classification accuracy among all the LDR procedures on Meter B using an LDA classifier as compared to using a QDA classifier (Figure 5.10 to Figure 5.14). This is attributable to the fact that the LDA classifier assumes equal covariance whereas the QDA classifier does not. The loss in performance due to using an LDA classifier is recouped to a large extent by the G-GLD, which although being a linear classifier, accounts for unequal covariances among the different health states of the meter.

5.4.4 Meters C and D

Meters C and D, being heteroscedastic, show a similar performance to Meter B for all LDR procedures and all three classifiers, except for the fact that M-GLD (Figure 5.14) significantly outperforms C-LDR (Figure 5.13) and M-LDR (Figure 5.12) using the QDA and G-GLD classifiers. This is because the sequential minimisation of the Bayes error that is followed by the M-GLD procedure in reducing the dimensionality, results in a better class discrimination than maximising the Chernoff criterion (as employed in C-LDR) or the Mahalanobis distance (as employed in M-LDR).

5.4.5 All flowmeters

A comparison of M-GLD+G-GLD with the linear SVM in Figure 5.15 shows that the M-GLD LDR procedure, followed by linear classification with G-GLD, easily outperforms the linear SVM on all four flowmeters; the SVM performs rather poorly on all flowmeters but Meter B. This shows that the SVM, while being one of the most widely used algorithms for linear classification, is unsuitable for the diagnosis of the health states of the four meter types tested.

In general however, the best classification (diagnostic) accuracy for all four flowmeters is achieved

by performing dimensionality reduction with the proposed M-GLD procedure on the original datasets, followed by quadratic discrimination with QDA. With this approach, Meters A and B have the best diagnostic capabilities (100%), followed by Meter C at 99.4% and Meter D at 97.3%, as shown in Figure 5.14. Also in the same figure, it will be noted that the G-GLD classifier shows a competitive performance with QDA after dimensionality reduction with M-GLD.

For other flowmeters in the field, different from the ones tested in this chapter, G-GLD may have the advantages of robustness in terms of diagnostic performance, and a faster testing (or diagnosing) time, since G-GLD is a linear classifier.

5.5 Chapter summary

This chapter discusses the application of the Gaussian linear discriminant (GLD) proposed in Chapter 3 and extended for linear dimensionality reduction in the multi-class case in Chapter 4 to flowmeter diagnostics. The chapter begins by justifying the need for flowmeter diagnostics in the oil and gas industry as a measure to avoid costly shut-downs of a flow facility, and to reduce costs incurred as a result of incorrect measurements and frequent recalibration of a flowmeter.

Following this, the chapter describes experiments conducted at the National Engineering Laboratory (NEL), UK, to test the diagnostic capabilities of four different ultrasonic flowmeter types, by monitoring the changes in their diagnostic variables under ideal and non-ideal flow conditions.

The data gathered from these tests are preprocessed and used to train an expert system to predict the health states of similar flowmeters in the field based on the values their diagnostic variables take. Training the expert system involves linear dimensionality reduction of the original diagnostics data to reduce over-fitting, followed by statistical classification using the following Bayesian classifiers: linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), and the Gaussian linear discriminant optimised by gradient descent (G-GLD).

By performing 10-fold cross validation, the classification or diagnostic accuracy of the four flowmeters are evaluated on the test folds, using various dimensionality reduction procedures together with the three Bayesian classifiers. The proposed multi-class Gaussian linear discriminant (M-GLD) LDR procedure followed by quadratic discrimination with the QDA classifier is shown to achieve the best diagnostic performance on all four flowmeters, with Meters A and B showing the best diagnostic capabilities, followed by Meters C and D respectively.

The results shown in this chapter indicate that, with the adoption of the proposed M-GLD dimen-

sionality reduction procedure and G-GLD linear classifier, it is possible to eliminate the need for end-user expertise required to interpret the wealth of diagnostic variables from a flowmeter in order to make sense of the health state of the meter. Moreover, the high diagnostic accuracies achieved for the flowmeters with the proposed algorithms have huge implications for the recalibration frequency of a flowmeter. Specifically, if evidence is provided to the regulatory body that a meter is predicted to be in its healthy state even at the point of its scheduled recalibration period, recalibration may be delayed [1], resulting in substantial savings (see page 1). In the same way, if a meter is predicted to be in a particular unhealthy state, even before its scheduled recalibration period, recalibration may be recommended earlier to prevent incorrect flow measurement and its associated costs (see page 1).

Chapter 6

Conclusions and future work

Linear discriminant analysis (LDA) has been applied to several problems such as medical diagnosis, face recognition and spam filtering, either for supervised linear dimensionality reduction (LDR) of a dataset, or for linear classification. This thesis addresses the issue of unequal covariance (heteroscedasticity), which causes LDA to be suboptimal in its performance, since LDA assumes equal covariance and normally distributed classes. The work presented in this thesis details how to account for heteroscedasticity in LDA for linear classification in a computationally efficient procedure termed Gaussian linear discriminant (GLD), that minimises the Bayes error—the minimum achievable error rate by a classifier that makes predictions from the knowledge of the true distribution of the data. The GLD procedure is then extended for dimensionality reduction in a procedure that involves the sequential minimisation of the Bayes error, while accounting for heteroscedasticity. Experimental validation of the proposed algorithms, using several publicly available datasets, indicate significant performance improvement over LDA when heteroscedasticity is accounted for.

This thesis further demonstrates the utility of the proposed algorithms by describing their applicability to flow meter diagnostics, which involves classifying a flow meter under one of a given number of health states.

6.1 Research questions answered

Motivated by the following peculiarities with flow meters, namely:

1. the covariance matrices of the health states of a given flow meter are not equal;
2. there is the potential issue of class imbalance, which has been claimed to negatively affect the performance of LDA [11];
3. the individual health states tend to be nearly normally distributed; with knowledge of this distribution, optimum linear classification and dimensionality reduction can be achieved by minimising the Bayes error [8],

This thesis has sought to answer the following research questions:

1. How can heteroscedasticity be accounted for in LDA while minimising the Bayes error for classification?
2. What is the effect of class imbalance on LDA when heteroscedasticity has been accounted for?
3. How can heteroscedasticity be accounted for in LDA while minimising the Bayes error for dimensionality reduction?
4. Does accounting for heteroscedasticity in LDA improve the accuracy of diagnosis for a given flow meter?

6.1.1 How can heteroscedasticity be accounted for in LDA while minimising the Bayes error for classification?

To answer this question, the first and second-order optimality conditions for the minimisation of the Bayes error are derived for linear classification, while accounting for heteroscedasticity under an assumption of normally distributed classes. From the optimality conditions, it is shown that there is no closed-form solution for a linear classifier that minimises the Bayes error in heteroscedastic LDA, thus, requiring the use of iterative methods. An iterative procedure that involves recursive matrix inversions to obtain the Bayes-optimal linear classifier is therefore derived from the optimality conditions. This procedure is referred to as recursive Gaussian linear discriminant (R-GLD). While R-GLD is computationally efficient for low-dimensional data, it is shown that it can sometimes converge on solutions that do not minimise the Bayes error at all, but rather maximise it. A gradient descent procedure, for which convergence to a local minimum is guaranteed, is therefore proposed to minimise the Bayes error under heteroscedasticity in a procedure referred to as gradient descent Gaussian linear discriminant (G-GLD). G-GLD requires no matrix inversions, and as such, is well-suited for high-dimensional data, for which matrix inversions can be computationally prohibitive.

An analysis of the Hessian matrix of the Bayes error under heteroscedasticity shows that the Bayes error is non-convex, which implies that the local optimum solution on which R-GLD or G-GLD converges is not unique, and there may be several other local minima. Non-convexity of the Bayes error therefore requires R-GLD or G-GLD be run several times with different initial solutions, to improve the quality of the overall solution.

R-GLD and G-GLD are validated experimentally, and are shown to achieve improved performance over LDA.

6.1.2 What is the effect of class imbalance on LDA when heteroscedasticity has been accounted for?

The effect of class imbalance on LDA is to cause the prior probability of the majority class to be much higher than that of the minority class. By studying the limiting behaviour of the ratio of the prior probabilities of the classes, it is seen that classification accuracy of LDA is biased in favour of the majority class, whether or not heteroscedasticity has been accounted for.

However, under heteroscedasticity, the optimality conditions of the Bayes error imply that the Bayes-optimal discriminating hyperplane depends on the choice of the discriminating threshold. Thus, a dynamic linear model, referred to as dynamic Gaussian linear discriminant (D-GLD), is derived, which directly expresses the discriminating hyperplane in terms of the discriminating threshold.

Under class imbalance, when the discriminating threshold is typically varied to improve the detection of more minority samples, D-GLD is shown experimentally to improve the area under the receiver operating characteristics curve (AUC). The improved performance is due to the fact that D-GLD optimises the discriminating hyperplane for every choice of the discriminating threshold. The AUC is an evaluation metric that measures the trade-off between the probability of correct detection of minority samples and the probability of false alarm, and is preferred over the classification accuracy under class imbalance, since the classification accuracy is skewed toward the majority class.

6.1.3 How can heteroscedasticity be accounted for in LDA while minimising the Bayes error for dimensionality reduction?

In the two-class case, the linear classifier obtained using G-GLD or R-GLD can be directly employed for linear dimensionality reduction, since it minimises the Bayes error under heteroscedasticity.

For more than two classes however, the Bayes error is not minimised in LDA, even if the assumptions of equal covariance and normally distributed classes are satisfied. To solve this problem, a procedure that involves the sequential minimisation of the Bayes error under heteroscedasticity is proposed for the multi-class case. The optimality conditions for this minimisation are derived. The procedure, termed multi-class Gaussian linear discriminant (M-GLD), involves the successive construction of G-GLD classifiers to discriminate one class from the remaining classes in $K-1$ stages, where K is the number of classes. M-GLD consequently reduces the dimensionality of a given dataset to $K-1$, which is necessary and sufficient to preserve the classification information in the original dataset, if a Bayesian classifier such as LDA is to be employed after LDR.

6.1.4 Does accounting for heteroscedasticity in LDA improve the accuracy of diagnosis for a given flow meter?

Yes.

To answer this question, the proposed M-GLD dimensionality reduction procedure is applied on four different ultrasonic flow meter (USM) diagnostics datasets. The datasets are obtained from experiments performed by National Engineering Laboratory (NEL), United Kingdom. Dimensionality reduction with M-GLD is then followed by linear classification with the proposed G-GLD classifier. The cross-validation results show that, by accounting for heteroscedasticity, M-GLD followed by G-GLD achieve substantial performance improvement over LDA in terms of the classification or diagnostic accuracy on all flow meters. The performance difference observed between the proposed algorithms and LDA are significant at the 0.01 confidence level, based on the Wilcoxon's signed rank test.

6.2 Future work

By addressing the research questions posed, the work presented in this thesis has led to significant contributions to knowledge. Yet, it still opens several windows for future work.

Firstly, the proposed GLD linear classifier, while it accounts for heteroscedasticity, still makes the assumption of within-class normality, in line with LDA. For many physical quantities such as measurement errors, the assumption of normality is often valid. Still, there are a lot others that do not satisfy this assumption. To account for non-normality, this thesis proposes a local neighbourhood search (LNS) procedure that searches in the neighbourhood of the GLD classifier to obtain a discriminative classifier that is robust to the distribution of the data. However, the computational time of LNS does not scale well with large amounts of training data, and it only works well on data that are nearly normally distributed, and not on those that are radically non-normal. One possible direction for future work is therefore to derive the GLD procedure for arbitrary non-normal distributions by minimising their Bayes error (or some upper bounds of it, given that the Bayes error can be analytically intractable for an arbitrary distribution). Alternatively, future research can be aimed at deriving a kernel function that implicitly transforms some data of a known non-normal distribution into a feature space where the data in each class is nearly normally distributed.

Secondly, it is implicitly assumed in LDA and the proposed GLD procedures that the data in any two classes are linearly separable, thus necessitating the construction of a linear decision boundary. For datasets with classes that are not linearly separable, there has been the application of the kernel

trick to LDA to implicitly transform the data into a new feature space where linear separability is guaranteed; this procedure, referred to as kernel Fisher discriminant (KFD), still does not account for heteroscedasticity. A heteroscedastic kernel LDA model, dubbed kernel Gaussian linear discriminant (K-GLD), which minimises the Bayes error in the transformed space, has been proposed in this thesis. However, neither K-FD nor K-GLD scales well with large amounts of training data, as they require the inversion of matrices the size of the training data. More efficient optimisation procedures are therefore needed to minimise the Bayes error in the K-GLD procedure.

Thirdly, while the proposed M-GLD procedure has been shown to be superior to LDA and other existing heteroscedastic LDA procedures for dimensionality reduction, it requires the construction of $(K^2+K-4)/2$ GLD classifiers for a K-class problem, which can be computationally costly for a dataset having a lot of classes. As future work, information theoretic approaches can be explored to reduce the total number of classifiers constructed in each stage of the proposed M-GLD dimensionality reduction procedure. This would decrease the computational complexity of the algorithm and improve its speed.

Fourthly, LDA relies on knowledge of the prior probabilities of the classes in a dataset. Since these prior probabilities are often unknown, they are commonly estimated from the relative frequency of each class in the dataset. However, for the four flow meter datasets used in this work, the relative frequencies of the health states are roughly balanced, and not indicative of the actual prior probabilities expected in the field. This is because, for a given flow meter in operation, there is a much higher probability that it is healthy than it is in a particular unhealthy state. Thus, it is expected that typical flow meter diagnostics data would show class imbalance in favour of the healthy class. Further research work is therefore required to estimate the prior probabilities of each health state of a given flow meter.

Finally, as an application to flow meter diagnostics, future work can focus on leveraging the correct diagnosis of a flow meter in the estimation of the error associated with each flow measurement, with reasonable accuracy. That is, with the knowledge of the true health state of a given flow meter, the associated flow measurement errors can be estimated with improved accuracy. This will allow erroneous flow measurements to be self-validated, thus resulting in significant cost cuts due to incorrect flow measurements in oil and gas operations.

6.3 Summary

In conclusion, this thesis describes the design of an optimal linear classifier and linear dimensionality reduction procedure which account for unequal covariance (heteroscedasticity) in LDA, while minimising

the Bayes error. Further adaptations of these basic models are made for the scenarios of non-normal distributions, class imbalance and linearly non-separable classes. The experimental validation of the proposed heteroscedastic LDA models on several real-world and artificial datasets show that accounting for heteroscedasticity while minimising the Bayes error results in significant performance improvement over LDA and other existing heteroscedastic LDA procedures.

The thesis further demonstrates the utility of the proposed heteroscedastic LDA models, by employing them in diagnosing the health states of four ultrasonic flow meters. The high diagnostic accuracies achieved with the proposed algorithms on the flow meters promise significant cost benefits in oil and gas operations.

Bibliography

[1] TUV-NEL. *Testing the diagnostic capabilities of liquid ultrasonic flow meters.* National Measurement System, 2012.

[2] Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Scholkopf, and Klaus-Robert Mullers. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop.*, pages 41–48. IEEE, 1999.

[3] Peter N. Belhumeur, João P Hespanha, and David J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.

[4] Håkan Brunzell and Jonny Eriksson. Feature reduction for classification of multidimensional data. *Pattern Recognition*, 33(10):1741–1748, 2000.

[5] R. P. W. Duin and M. Loog. Linear dimensionality reduction via a heteroscedastic extension of LDA: the Chernoff criterion. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):732–739, 2004.

[6] Mairead L. Bermingham, Ricardo Pong-Wong, Athina Spiliopoulou, Caroline Hayward, Igor Rudan, Harry Campbell, Alan F. Wright, James F. Wilson, Felix Agakov, Pau Navarro, et al. Application of high-dimensional feature selection: evaluation for genomic prediction in man. *Scientific reports*, 5, 2015.

[7] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 6. Springer, 2013.

[8] Keinosuke Fukunaga. *Introduction to statistical pattern recognition.* Academic press, 2013.

[9] FMI-EURMAET. *Joint FMI-EURAMET TC-Flow Research Collaboration Workshop.* Flow Measurement Institute, 2017.

[10] Aidan Lyon. Why are normal distributions normal? *The British Journal for the Philosophy of Science*, 65(3):621–649, 2013.

[11] Jigang Xie and Zhengding Qiu. The effect of imbalanced data sets on LDA: A theoretical and empirical analysis. *Pattern recognition*, 40(2):557–562, 2007.

[12] Jing-Hao Xue and D. Michael Titterington. Do unbalanced data have a negative effect on LDA? *Pattern Recognition*, 41(5):1558–1571, 2008.

[13] Henry Schneiderman and Takeo Kanade. A statistical method for 3D object detection applied to faces and cars. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 746–751. IEEE, 2000.

[14] Chengjun Liu and Harry Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Transactions on Image processing*, 11(4):467–476, 2002.

[15] Stan Z. Li and ZhenQiu Zhang. Floatboost learning and statistical face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1112–1123, 2004.

[16] Henry Schneiderman and Takeo Kanade. Object detection using the statistics of parts. *International Journal of Computer Vision*, 56(3):151–177, 2004.

[17] Anil K. Jain, Robert P. W. Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, 22(1):4–37, 2000.

[18] Christopher Kanan and Garrison Cottrell. Robust classification of objects, faces, and flowers using natural image statistics. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2472–2479. IEEE, 2010.

[19] Igor Kononenko. Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in medicine*, 23(1):89–109, 2001.

[20] Igor Kononenko, Ivan Bratko, and Matjaž Kukar. Application of machine learning to medical diagnosis. *Machine Learning and Data Mining: Methods and Applications*, 389:408, 1997.

[21] Akin Ozcift and Arif Gulten. Classifier ensemble construction with rotation forest to improve medical diagnosis performance of machine learning algorithms. *Computer methods and programs in biomedicine*, 104(3):443–451, 2011.

[22] Philip K. Chan, Wei Fan, Andreas L. Prodromidis, and Salvatore J. Stolfo. Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems and Their Applications*, 14(6):67–74, 1999.

[23] Sam Maes, Karl Tuyls, Bram Vanschoenwinkel, and Bernard Manderick. Credit card fraud detection using Bayesian and neural networks. In *Proceedings of the 1st international naiso congress on neuro fuzzy technologies*, pages 261–270, 2002.

[24] R. Brause, T. Langsdorf, and Michael Hepp. Neural data mining for credit card fraud detection. In *Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference on*, pages 103–106. IEEE, 1999.

[25] Nader Mahmoudi and Ekrem Duman. Detecting credit card fraud by modified Fisher discriminant analysis. *Expert Systems with Applications*, 42(5):2510–2516, 2015.

[26] Achmad Widodo and Bo-Suk Yang. Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical systems and signal processing*, 21(6):2560–2574, 2007.

[27] Leo H. Chiang, Mark E. Kotanchek, and Arthur K. Kordon. Fault diagnosis based on Fisher discriminant analysis and support vector machines. *Computers & chemical engineering*, 28(8):1389–1401, 2004.

[28] V. Sugumaran, V. Muralidharan, and K. I. Ramachandran. Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing. *Mechanical systems and signal processing*, 21(2):930–942, 2007.

[29] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 2:841–848, 2002.

[30] Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[31] Alan J. Izenman. *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. Springer Science & Business Media, 2009.

[32] Guo-Xun Yuan, Chia-Hua Ho, and Chih-Jen Lin. Recent advances of large-scale linear classification. *Proceedings of the IEEE*, 100(9):2584–2603, 2012.

[33] Christopher M. Bishop. Pattern recognition. *Machine Learning*, 128:1–58, 2006.

[34] Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, 99(10):6567–6572, 2002.

[35] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[36] Onur C. Hamsici and Aleix M. Martinez. Bayes optimality in linear discriminant analysis. *IEEE transactions on pattern analysis and machine intelligence*, 30(4):647–657, 2008.

[37] Alok Sharma and Kuldip K. Paliwal. Cancer classification by gradient LDA technique using microarray gene expression data. *Data & Knowledge Engineering*, 66(2):338–347, 2008.

[38] Danny Coomans, M. Jonckheer, Désiré Luc Massart, Ivo Broeckaert, and Pièrre Blockx. The application of linear discriminant analysis in the diagnosis of thyroid diseases. *Analytica chimica acta*, 103(4):409–415, 1978.

[39] Abdulkadir Sengur. An expert system based on linear discriminant analysis and adaptive neuro-fuzzy inference system to diagnosis heart valve diseases. *Expert Systems with Applications*, 35(1):214–222, 2008.

[40] Kemal Polat, Salih Güneş, and Ahmet Arslan. A cascade learning system for classification of diabetes disease: Generalized discriminant analysis and least square support vector machine. *Expert systems with applications*, 34(1):482–487, 2008.

[41] Fengxi Song, David Zhang, Jizhong Wang, Hang Liu, and Qing Tao. A parameterized direct LDA and its application to face recognition. *Neurocomputing*, 71(1):191–196, 2007.

[42] Li-Fen Chen, Hong-Yuan Mark Liao, Ming-Tat Ko, Ja-Chen Lin, and Gwo-Jong Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern recognition*, 33(10):1713–1726, 2000.

[43] Jun Liu, Songcan Chen, Xiaoyang Tan, and Daoqiang Zhang. Efficient pseudoinverse linear discriminant analysis and its nonlinear form for face recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 21(08):1265–1278, 2007.

[44] Hua Yu and Jie Yang. A direct LDA algorithm for high-dimensional dataâĂŤwith application to face recognition. *Pattern recognition*, 34(10):2067–2070, 2001.

[45] Xiaohang Jin, Mingbo Zhao, Tommy W. S. Chow, and Michael Pecht. Motor bearing fault diagnosis using trace ratio linear discriminant analysis. *IEEE Transactions on Industrial Electronics*, 61(5):2441–2451, 2014.

[46] Bulent Ayhan, M.-Y. Chow, and M.-H. Song. Multiple discriminant analysis and neural-network-based monolith and partition fault-detection schemes for broken rotor bar in induction motors. *IEEE Transactions on Industrial Electronics*, 53(4):1298–1308, 2006.

[47] Ashkan Moosavian, Hojat Ahmadi, and Ahmad Tabatabaeefar. 814. fault diagnosis of main engine journal bearing based on vibration analysis using Fisher linear discriminant, K-nearest neighbor and support vector machine. *Journal of Vibroengineering*, 14(2), 2012.

[48] Adil Mehmood Khan, Y.-K. Lee, S. Y. Lee, and T.-S. Kim. Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis. In *Future Information Technology (FutureTech), 2010 5th International Conference on*, pages 1–6. IEEE, 2010.

[49] Yan Yan, Elisa Ricci, Ramanathan Subramanian, Gaowen Liu, and Nicu Sebe. Multitask linear discriminant analysis for view invariant action recognition. *IEEE Transactions on Image Processing*, 23(12):5599–5611, 2014.

[50] Alexandros Iosifidis, Anastasios Tefas, Nikolaos Nikolaidis, and Ioannis Pitas. Multi-view human movement recognition based on fuzzy distances and linear discriminant analysis. *Computer Vision and Image Understanding*, 116(3):347–360, 2012.

[51] Corinna Cortes and Vladimir Vapnik. Support vector machine. *Machine learning*, 20(3):273–297, 1995.

[52] Bharath Hariharan, Jitendra Malik, and Deva Ramanan. Discriminative decorrelation for clustering and classification. In *European Conference on Computer Vision*, pages 459–472. Springer, 2012.

[53] Alok Sharma and Kuldip K. Paliwal. Linear discriminant analysis for the small sample size problem: an overview. *International Journal of Machine Learning and Cybernetics*, 6(3):443–454, 2015.

[54] Juwei Lu, Kostas N. Plataniotis, and Anastasios N. Venetsanopoulos. Regularized discriminant analysis for the small sample size problem in face recognition. *Pattern Recognition Letters*, 24(16):3079–3087, 2003.

[55] Wenming Zheng, Li Zhao, and Cairong Zou. An efficient algorithm to solve the small sample size problem for LDA. *Pattern Recognition*, 37(5):1077–1079, 2004.

[56] Rui Huang, Qingshan Liu, Hanqing Lu, and Songde Ma. Solving the small sample size problem of LDA. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 29–32. IEEE, 2002.

[57] Kuldip K. Paliwal and Alok Sharma. Improved pseudoinverse linear discriminant analysis method for dimensionality reduction. *International Journal of Pattern Recognition and Artificial Intelligence*, 26(01):1250002, 2012.

[58] Jerome H. Friedman. Regularized discriminant analysis. *Journal of the American statistical association*, 84(405):165–175, 1989.

[59] W. Wu, Y. Mallet, B. Walczak, W. Penninckx, D. L. Massart, S. Heuerding, and F. Erni. Comparison of regularized discriminant analysis linear discriminant analysis and quadratic discriminant analysis applied to NIR data. *Analytica Chimica Acta*, 329(3):257–265, 1996.

[60] Trevor Hastie and Robert Tibshirani. Discriminant analysis by Gaussian mixtures. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 155–176, 1996.

[61] Geoffrey McLachlan. *Discriminant analysis and statistical pattern recognition*, volume 544. John Wiley & Sons, 2004.

[62] Junchang Ju, Eric D. Kolaczyk, and Sucharita Gopal. Gaussian mixture discriminant analysis and sub-pixel land cover characterization in remote sensing. *Remote Sensing of Environment*, 84(4):550–560, 2003.

[63] Deng Cai, Xiaofei He, Kun Zhou, Jiawei Han, and Hujun Bao. Locality sensitive discriminant analysis. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 708–713. Morgan Kaufmann Publishers Inc., 2007.

[64] K. Fukunaga and J. M. Mantock. Nonparametric discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):671–678, 1983.

[65] Zhifeng Li, Dahua Lin, and Xiaoou Tang. Nonparametric discriminant analysis for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):755–761, 2009.

[66] Zheng Zhao, Liang Sun, Shipeng Yu, Huan Liu, and Jieping Ye. Multiclass probabilistic kernel discriminant analysis. In *Proceedings of the 21st international jont conference on Artifical intelligence*, pages 1363–1368. Morgan Kaufmann Publishers Inc., 2009.

[67] Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. *Learning from data*, volume 4. AMLBook New York, NY, USA:, 2012.

[68] John Aitchison and Colin G. G. Aitken. Multivariate binary discrimination by the kernel method. *Biometrika*, 63(3):413–420, 1976.

[69] Tommi S. Jaakkola, Mark Diekhans, and David Haussler. Using the Fisher kernel method to detect remote protein homologies. In *ISMB*, volume 99, pages 149–158, 1999.

[70] Sidney Marks and Olive Jean Dunn. Discriminant functions when covariance matrices are unequal. *Journal of the American Statistical Association*, 69(346):555–559, 1974.

[71] Theodore W Anderson and R. R. Bahadur. Classification into two multivariate normal distributions with different covariance matrices. *The annals of mathematical statistics*, pages 420–431, 1962.

[72] D. Peterson and R. Mattson. A method of finding linear discriminant functions for a class of performance criteria. *IEEE Transactions on Information Theory*, 12(3):380–387, 1966.

[73] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2009.

[74] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[75] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. *Advances in intelligent computing*, pages 878–887, 2005.

[76] Olukunle Ojetola, Elena Gaura, and James Brusey. Data set for fall events and daily activities from inertial sensors. In *Proceedings of the 6th ACM Multimedia Systems Conference*, pages 243–248. ACM, 2015.

[77] Mitja Luštrek and Boštjan Kaluža. Fall detection and activity recognition with machine learning. *Informatica*, 33(2), 2009.

[78] Norbert Noury, Anthony Fleury, Pierre Rumeau, A. K. Bourke, G. O. Laighin, Vincent Rialle, and J. E. Lundy. Fall detection-principles and methods. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 1663–1666. IEEE, 2007.

[79] Jonathan Burez and Dirk Van den Poel. Handling class imbalance in customer churn prediction. *Expert Systems with Applications*, 36(3):4626–4636, 2009.

[80] Yaya Xie, Xiu Li, E. W. T. Ngai, and Weiyun Ying. Customer churn prediction using improved balanced random forests. *Expert Systems with Applications*, 36(3):5445–5449, 2009.

[81] Koen W. De Bock and Dirk Van den Poel. An empirical evaluation of rotation-based ensemble classifiers for customer churn prediction. *Expert Systems with Applications*, 38(10):12293–12301, 2011.

[82] Jordan McBain and Markus Timusk. Feature extraction for novelty detection as applied to fault detection in machinery. *Pattern Recognition Letters*, 32(7):1054–1061, 2011.

[83] Atefeh Dehghani Ashkezari, Hui Ma, Tapan K. Saha, and Chandima Ekanayake. Application of fuzzy support vector machine for determining the health index of the insulation system of in-service power transformers. *IEEE Transactions on Dielectrics and Electrical Insulation*, 20(3):965–973, 2013.

[84] Michael Buckland and Fredric Gey. The relationship between recall and precision. *Journal of the American society for information science*, 45(1):12, 1994.

[85] I. Dan Melamed, Ryan Green, and Joseph P. Turian. Precision and recall of machine translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003–short papers-Volume 2*, pages 61–63. Association for Computational Linguistics, 2003.

[86] Michael Steinbach, George Karypis, Vipin Kumar, et al. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston, 2000.

[87] Yuanpeng J. Huang, Robert Powers, and Gaetano T. Montelione. Protein NMR recall, precision, and F-measure scores (RPF scores): structure quality assessment measures based on information retrieval statistics. *Journal of the American Chemical Society*, 127(6):1665–1674, 2005.

[88] David Martin Powers. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. 2011.

[89] Peter A. Flach. The geometry of ROC space: understanding machine learning metrics through ROC isometrics. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 194–201, 2003.

[90] Jesse Davis and Mark Goadrich. The relationship between precision-recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

[91] Andrew P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[92] Tom Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[93] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks*, 13(2):415–425, 2002.

[94] John P. Cunningham and Zoubin Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *Journal of Machine Learning Research*, 16:2859–2900, 2015.

[95] Mark Richardson. Principal component analysis. *URL: http://people. maths. ox. ac. uk/richardsonm/SignalProcPCA. pdf (last access: 3.5. 2013). Aleš Hladnik Dr., Ass. Prof., Chair of Information and Graphic Arts Technology, Faculty of Natural Sciences and Engineering, University of Ljubljana, Slovenia ales. hladnik@ ntf. uni-lj. si*, 6:16, 2009.

[96] Sebastian Mika, Bernhard Schölkopf, Alex J. Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel PCA and de-noising in feature spaces. In *Advances in neural information processing systems*, pages 536–542, 1999.

[97] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer, 1997.

[98] Heiko Hoffmann. Kernel PCA for novelty detection. *Pattern Recognition*, 40(3):863–874, 2007.

[99] Wikipedia. Diagonalizable matrix — Wikipedia, the free encyclopedia, 2017. [Online; accessed 1-August-2017].

[100] L. J. Buturovic. Toward Bayes-optimal linear dimension reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):420–424, 1994.

[101] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern classification*. John Wiley & Sons, 2012.

[102] David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[103] Wei Bian. *Supervised linear dimension reduction*. PhD thesis, 2012.

[104] Masashi Sugiyama. Local fisher discriminant analysis for supervised dimensionality reduction. In *Proceedings of the 23rd international conference on Machine learning*, pages 905–912. ACM, 2006.

[105] Wikipedia. Rayleigh quotient — Wikipedia, the free encyclopedia, 2017. [Online; accessed 10-August-2017].

[106] Haifeng Li, Tao Jiang, and Keshu Zhang. Efficient and robust feature extraction by maximum margin criterion. In *Advances in neural information processing systems*, pages 97–104, 2004.

[107] Barbara G. Tabachnick, Linda S. Fidell, and Steven J. Osterlind. Using multivariate statistics. 2001.

[108] Joseph F. Hair, William C. Black, Barry J. Babin, Rolph E. Anderson, Ronald L. Tatham, et al. *Multivariate data analysis*, volume 5. Prentice Hall Upper Saddle River, NJ, 1998.

[109] Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7:15, 2008.

[110] Wikipedia. Gradient descent — Wikipedia, the free encyclopedia, 2017. [Online; accessed 11-August-2017].

[111] Prasanta C. Mahalanobis. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2(1):49–55, 1936.

[112] Goeffrey J. McLachlan. Mahalanobis distance. *Resonance*, 4(6):20–26, 1999.

[113] Roy De Maesschalck, Delphine Jouan-Rimbaud, and Désiré L Massart. The Mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18, 2000.

[114] Henry P. Decell Jr and Salma K. Marani. Feature combinations and the Bhattacharyya criterion. *Communications in Statistics-Theory and Methods*, 5(12):1143–1152, 1976.

[115] Frank Nielsen. Generalized Bhattacharyya and Chernoff upper bounds on Bayes error using quasi-arithmetic means. *Pattern Recognition Letters*, 42:25–34, 2014.

[116] Kerstin Bunte, Sven Haase, Michael Biehl, and Thomas Villmann. Stochastic neighbor embedding (SNE) for dimension reduction and visualization using arbitrary divergences. *Neurocomputing*, 90:23–45, 2012.

[117] Michael Collins, Sanjoy Dasgupta, and Robert E. Schapire. A generalization of principal components analysis to the exponential family. In *Advances in neural information processing systems*, pages 617–624, 2002.

[118] Henry P. Decell and Shailesh M. Mayekar. Feature combinations and the divergence criterion. *Computers & Mathematics with Applications*, 3(1):71–76, 1977.

[119] A. Kai Qin, Ponnuthurai N. Suganthan, and Marco Loog. Uncorrelated heteroscedastic LDA based on the weighted pairwise Chernoff criterion. *Pattern recognition*, 38(4):613–616, 2005.

[120] Koel Das and Zoran Nenadic. Approximate information discriminant analysis: A computationally simple heteroscedastic feature extraction technique. *Pattern Recognition*, 41(5):1548–1557, 2008.

[121] Marco Loog and Robert P. W. Duin. Non-iterative heteroscedastic linear dimension reduction for two-class data. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 508–517. Springer, 2002.

[122] Marco Loog. Approximate pairwise accuracy criteria for multiclass linear dimension reduction: Generalisations of the Fisher criterion. *WBBM Report Series 44*, 1999.

[123] Marco Ramoni and Paola Sebastiani. Robust Bayes classifiers. *Artificial Intelligence*, 125(1-2):209–226, 2001.

[124] Jorge Nocedal and Stephen J. Wright. *Sequential quadratic programming*. Springer, 2006.

[125] Govind S. Mudholkar and Alan D. Hutson. The epsilon–skew–normal distribution for analyzing near-normal data. *Journal of Statistical Planning and Inference*, 83(2):291–309, 2000.

[126] Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. Applying support vector machines to imbalanced datasets. *Machine learning: ECML 2004*, pages 39–50, 2004.

[127] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[128] Sebastian Raschka. Linear discriminant analysis bit by bit. *Blog, August*, 2014.

[129] Kojo Sarfo Gyamfi, James Brusey, Andrew Hunt, and Elena Gaura. Linear classifier design under heteroscedasticity in linear discriminant analysis. *Expert Systems with Applications*, 79:44–52, 2017.

[130] Isabelle Guyon. Design of experiments of the NIPS 2003 variable selection benchmark, 2003.

[131] Wikipedia. Rank (linear algebra) — Wikipedia, the free encyclopedia, 2017. [Online; accessed 12-August-2017].

[132] Marcel J. M. Vermeulen, Jan G. Drenthen, and den Hilko Hollander. *Understanding diagnostic and expert systems in ultrasonic flow meters*. KROHNE Oil and Gas, CT Products, 2012.

[133] J. P. Royston. Some techniques for assessing multivarate normality based on the Shapiro-Wilk W. *Applied Statistics*, pages 121–133, 1983.

[134] Antonio trujillo ortiz, Rafael Hernandez-Walls, K. Barba-Rojo, and L. Cupul-Magana. Roystest. Royston's multivariate normality test, 10 2007.

[135] James P. Stevens. *Applied multivariate statistics for the social sciences*. Routledge, 2012.

[136] Antonio trujillo ortiz and Rafael Hernandez-Walls. Multivariate statistical testing for the homogeneity of covariance matrices by the Box's M., 12 2003.

[137] Andrew Ng. Machine learning. coursera, 2016.

[138] Wikipedia. Cross-validation (statistics) — wikipedia, the free encyclopedia, 2017. [Online; accessed 29-November-2017].