# THE APPLICATION OF CLASSIFICATION TECHNIQUES IN MODELLING CREDIT RISK

**By**

**JONAH MUSHAVA**

Submitted in fulfillment of the academic requirements for the degree of Master of Science in Financial Mathematics in the School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Durban, South Africa.

March 2014


As the candidate's supervisor I have/have not approved this thesis/dissertation for submission.

Signed: ..... Name: ..... Date: .....

# Preface

The experimental work described in this dissertation was carried out in accordance with the rules and regulations of the School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Westville, from March 2012 to March 2014, under the supervision of Professor Mike Murray.

The research work represents original work by the author, and where use has been made of the work of others, it is duly acknowledged by special reference in the text. Any views expressed in the dissertation are those of the author and in no way represent those of the University of KwaZulu-Natal. The work has not been submitted in any form for any degree or diploma to any University.

Signed:..... Date.....
Jonah Mushava (Student)

Signed:..... Date.....
Prof. Mike Murray (Supervisor)

# Declaration - Plagiarism

I, Jonah Mushava, declare that

1. The research reported in this dissertation, except where otherwise indicated, is my original research.

2. This dissertation has not been submitted for any degree or examination at any other university.

3. This thesis does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.

4. This thesis does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:
   a. Their words have been re-written but the general information attributed to them has been referenced
   b. Where their exact words have been used, then their writing has been placed in italics and inside quotation marks, and referenced.

5. This dissertation does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the thesis and in the References sections.

Signed: ……

# Abstract

The aim of this dissertation is to examine the use of classification techniques to model credit risk through a practice known as credit scoring. In particular, the focus is on one parametric class of classification techniques and one non-parametric class of classification techniques. Since the goal of credit-scoring is to improve the quality of the decisions in evaluating a loan application, advanced and interesting methods that improve upon the performance of linear discriminant analysis (LDA) and classification and regression trees (CART) will be explored. For LDA these methods include a description of quadratic discriminant analysis (QDA), flexible discriminant analysis (FDA) and mixture discriminant analysis (MDA). Multivariate adaptive regression splines (MARS) are used in the FDA procedure. An Expectation Maximization (EM)-algorithm that estimates the model parameters in MDA will be developed thereof. Techniques that help to improve the performance of CART such as bagging, random forests and boosting are also discussed at length.

A real life dataset was used as an illustration to how these credit-scoring models can be used to classify a new applicant. The dataset shall be split into a 'learning sample' and a 'testing sample'. The learning sample will be used to develop the credit-scoring model (also known as a scorecard) whilst the testing sample will be used to test the predictive capability of the scorecard that would have been constructed. The predictive performance of the scorecards will be assessed using four measures; a classification error rate, a sensitivity measure, a specificity measure and the area under the ROC curve (AUC). Based on these four model performance measures, the empirical results reveal that there is no single ideal scorecard for modelling credit risk because such a conclusion depends on the aims and objectives of the lender, the details of the problem and the data structure.

# Table of Contents

# List of Figures

# List of Tables

# List of Algorithms

# Dedication

Mama naMati, you believed in me and here we are.

# Acknowledgements

This has been a journey, a learning curve and a bittersweet experience. Bitter in that it posed seemingly insurmountable challenges but by His grace, it is finally here. Much appreciation goes to my supervisor, Prof. Mike Murray without whom this dissertation would not have seen its decent completion. Your tireless guidance, patience and constructive criticism cannot be taken for granted, I thank you Prof!

I would also like to express my gratitude to the School of Mathematics, Statistics and Computer Science for funding this research. May the department grow in leaps and bounds.

I am also grateful to my friends and colleagues for their motivation, suggestions and assistance. J Sylaides, N Mupure, P.A Sibanda, H Kayiya, S Reade, T Makeke, M Kika and G S Rukanda and all those I did not get to mention by name, it is not because your contribution was not valuable, space inhibits me, I value and thank you.

# CHAPTER 1

# 1. Scope of the Study

## 1.1 Introduction

Credit risk is one of the major challenges that is threatening the growth of the financial markets today. Basel Committee on Banking Supervision (2000:1) defines credit risk as the likelihood that once a borrower is given a loan, they may fail to adhere to the terms of the credit agreement. A typical credit agreement obligates the borrower to pay back the principal and interest on the loan every month for a fixed period. A borrower is normally classified as a 'defaulter' if they miss three monthly instalments in a given period or if they miss three consecutive monthly instalments within the stipulated period (Crook, Edelman & Thomas, 2007). A lender's aim is therefore to avoid granting loans to borrowers who are likely to default. Therefore, a system that regulates this area of banking or finance is required in order to address this anomaly or problem. This study will amongst other things explore a method that can be used in this respect.

## 1.2 Motivation for this study

The birth of this study is attributable to many factors. These are given and explained in this section.

Determining a good or a bad borrower is not an easy task because information in the credit market is usually asymmetric which means that one party in the contract may have more or less information about the other party. The resulting imbalance in information may cause one party to take advantage of the other party, either before the transaction through a condition known as *adverse selection* or after the transaction through a condition known as *moral hazard*. A typical example of adverse selection occurs when a person who is bankrupt hides this information when acquiring a loan. On the other hand a typical example of moral hazard occurs when a person with a car insurance policy deliberately has a car accident in order to profit from the payout or compensation that has been promised by the insurance company.

In the respect of the above, many lenders especially in developing countries have attempted to alleviate this challenge by having credit applicants evaluated by a loan

officer. A loan officer uses his or her experience to decide if a person is creditworthy or not. Since this decision making is prone to a fair amount of subjectivity and thus possible bias, a set of Basel II banking regulations have been developed to help quantify and correctly price this credit risk giving rise to a practice known as *credit scoring* (Engelmann & Rauhmeier, 2006). In this study the goal is thus to develop classification techniques that can be used for credit scoring.

This study is also inspired by the interest of the researcher in credit risk related issues. Of particular concern is the recent financial crisis in the USA that crippled the credit markets globally. Poor credit risk management in the USA mortgage markets was cited as one of the major causes (Shahrokhi, 2011). Interestingly, scholars in the discipline have not shown much interest in this topic as evidenced by the limited number of literature (Thomas, Edelman & Crook, 2002). Therefore any study that attempts to understand how the effects of credit risk can be mitigated is essential.

Driven by this motive, it is the researcher's hope that this study will contribute to the extensive understanding of the theory that influences some of the classification techniques that can be employed in modelling credit risk. The stages in the model building strategy and the application of the classification procedures to a real life situation will be fully engaged to maximize comprehension.

### 1.3 Objectives

The objectives of this study can be highlighted as follows:

1. To develop a parametric and a non-parametric model that can 'accurately' classify a new credit applicant as being either a potential defaulter or non-defaulter based on their underlying baseline demographic factors/characteristics,

2. To improve the predictive capabilities of these aforementioned models.

3. To determine the effect that the above-mentioned baseline demographic factors have on a possible default or non-default.

### 1.4 Chapter layout

This research is made up of eight solid chapters, which will be broken down in this section. The first chapter was a broad contextual outline of the research. Chapter two provides an overview to some of the credit scoring concepts that are being used in the industry. The ensuing chapter then focuses on the development of a parametric classification method called linear discriminant analysis (LDA). Some new extensions have been developed to help improve the performance of the LDA classifier, chapter four is dedicated to the exploration of these. Chapter five focuses on developing a non-parametric classification technique that makes use of a classification and regression tree (CART). In chapter six a bagging, random forests and boosting procedure will be introduced to help improve the performance of the CART model. Chapter seven applies these classification techniques on a real life credit-related dataset with chapter eight as the summation of the study.

# CHAPTER 2

# 2. Conceptual and Contextual framework: An Overview of Credit Scoring

## 2.1 Introduction

The use of classification techniques to model credit risk dates back to the early 1940s. Durand (1941), was the first to employ this technique to distinguish between a good and a bad loan. Since then, advances in computer technology, improved data collection methods and competition in the financial industry have created a plethora of techniques for modelling credit risk. This process of modelling credit risk is generally known as credit scoring. However, it is difficult to define credit scoring with certainty, as there is no single attested definition of credit scoring. Therefore, it would be worthwhile to explore some of the definitions that have been attached to the term credit scoring.

Anderson (2007) defines credit scoring as the assignment of an appropriate score (numerical value) to a credit applicant that takes into account their baseline demographic characteristics. On the other hand, Hand & Jacka (1998) defines credit scoring as the process used by financial institutions in modelling creditworthiness. In unpacking the definition of credit scoring, Thomas *et al*. (2002:1) explains it using the concept of credit-scoring models, which are referred to as *scorecards*. These are defined as "*a set of decision models and their underlying techniques that aid lenders in the granting of consumer credit*."

Whilst there is an appreciation of the several definitions given, the definition proposed by Thomas *et al*. (2002:1) is quite relevant to this study. This is because the definition is in line with the research objectives of developing classification techniques, which will be used to decide whether a new credit applicant is likely to default or not. Anderson's (2007) definition also embeds crucial elements as it touches on the assignment of an appropriate score to an applicant based on their baseline demographic characteristics. These two therefore qualify as the working definition of this study.

Having defined credit scoring, it is important at this stage to consider some of the benefits of credit scoring, some of its common applications and popular techniques used

as credit scoring models. These will be looked at underneath, together with some of the procedures used to assess the performance of a scorecard. A consideration of these aspects is crucial inasmuch as giving one an insight of the concept of credit scoring is concerned.

### 2.2 Benefits of credit scoring

There are quite a number of benefits associated with credit scoring that have been discovered all over the world. Chief among these are cost effectiveness, efficiency and objectivity. Human based credit evaluation methods can be slow, costly and very subjective in nature whereas credit scoring on the other hand is fast, automated, cost-effective and objective in nature. Owing to this advantage associated with it, TransUnion (2007) has indicated that the use of credit scoring methods in the US mortgage market has managed to increase from 25% in 1996 to 90% in 2002. This translates to the conclusion that decisions that were taking weeks to be completed or passed in 1996 were now taking minutes to be completed in 2002. Apart from the above, the use of credit scoring models managed to reduce the cost of a loan application by an average amount of US$1500 per loan as asserted again by TransUnion (2007). Due to these lower operating costs, more credit could now be given to prospective clients.

### 2.3 Credit scorecard applications

The most prominent use of a credit scorecard relates to the processing of a loan application. In a typical loan application, the baseline characteristics of a prospective borrower are used to generate a score using a particular type of credit scoring model. According to Abdou, Pointon & El-Masry (2008), some of the characteristics that may be considered include the age of a particular applicant, their gender, ethnicity, marital status, house ownership, telephone ownership, occupation, monthly income, level of education and address location. In such a case, if the generated score lies below a particular benchmarked value, then the application ought to be rejected because the applicant is being classified as a potential defaulter on that loan. Bolton (2009) has discussed how this technique is being applied in a South African context with Kocenda & Vojtek (2009); Lee & Chen (2005) and Sustersic, Mramor & Zupan (2009) reinforcing this by citing various examples of where credit scoring can be used in the decision making process of granting a loan.

Quah & Srigaresh (2008) have shown how scorecards can also be used to help detect and prevent credit card fraud. Other applications of credit scoring models include:

- the issuing of mortgages (Feldman & Gross, 2005),
- bankruptcy prediction and classification (Nanni & Lumini, 2009),
- the rating of bonds (Altman, 2005),
- portfolio management (Xia *et al*., 2000),
- financial distress forecasting (Hamdi & Karaa, 2012),
- financial decision making (West, Dellana & Qian, 2005),
- stock price forecasting (Quah & Srigaresh, 1999)
- the granting small business loans (DeYoung *et al*., 2008).

**2.4 Credit-scoring methods**

In a typical scorecarding methodology, there is a set of baseline characteristics for the loan applicant. These vary from their age, marital status to their salary, which are then fed into an appropriate model from which a particular score is generated. Assignment as a potential defaulter depends then on this generated score lying below a particular benchmarked value. This benchmarked value however needs to be obtained from a sample of historical data, which is known as the *training* or *learning sample.* In this sense, the process of constructing credit-scoring models can be categorized as being a *pattern recognition* problem with *supervised learning* (Ripley, 1996).

In supervised learning, one is observing a set of $i = 1,2, \dots, N$ training observations, of the form: $-(\mathbf{x}_i, y_i)$. The p-dimensional vector $\mathbf{x}_i = \left(x_{1i}, x_{2i}, \dots x_{pi}\right)^T$ contains predictor variables for the $i^{th}$ case whose outcome on $y_i$ is known to belong to one of $K$ possible groups viz $y_i = k \in \{1,2, \dots, K\}$. Consequently, the learning sample $L$ would be of the form:

$$L = \{(\mathbf{x}_1, y_1)\,, (\mathbf{x}_2, y_2\,), \dots, (\mathbf{x}_N, y_N)\,\} \tag{2.1}$$

The objective of supervised learning then is to use $L$ to create a rule for classifying a new observation $\mathbf{x}_i^{new} \in R^p$ whose outcome on $y_i$ is unknown, to one of the above $K$ classes.

For the special case where an applicant $\mathbf{x}_i \in R^p$ can only belong to one of two possible classes $y_i = k \in \{0,1\}$, the following *logistic regression* model is popularly used to model the assignment mechanism of an applicant to a particular class:

$$\ln\left(\frac{p(y_i = 1|\mathbf{x}_i)}{p(y_i = 0|\mathbf{x}_i)}\right) = \hat{\beta}_o + \widehat{\boldsymbol{\beta}}^T\mathbf{x}_i \tag{2.2}$$

where $\hat{\beta}_o$ and $\widehat{\boldsymbol{\beta}}^T = \left(\hat{\beta}_1, \hat{\beta}_2, \dots, \hat{\beta}_p\right)$ are regression coefficients that need to be estimated (Hosmer & Lemeshow,1989).

As a result that,

$$p(y_i = 0|\mathbf{x}_i) + p(y_i = 1|\mathbf{x}_i) = 1$$

we can rewrite (2.2) in the following form

$$p(y_i = 1|\mathbf{x}_i) = \frac{e^{\widehat{\beta}_o + \widehat{\boldsymbol{\beta}}^T\mathbf{x}_i}}{1 + e^{\widehat{\beta}_o + \widehat{\boldsymbol{\beta}}^T\mathbf{x}_i}} = \quad \frac{1}{1 + e^{-(\widehat{\beta}_o + \widehat{\boldsymbol{\beta}}^T\mathbf{x}_i)}} \tag{2.3}$$

One can then assign a new observation $\mathbf{x}_i^{new}$ to one of the above two classes based on the following classification rule: Set

$$\mathrm{y}_i = \begin{cases} 1 & if \quad p(y_i = 1|\mathbf{x}_i^{new}) > c \\ 0 & otherwise \end{cases} \tag{2.4}$$

where $c$ is a threshold value for assigning this new observation $\mathbf{x}_i^{new}$ to a particular class that need to be determined. Abdou *et al*. (2008), Bolton (2009) and Lee & Jung (2000), among others, shows how this technique is being used in the field as a credit-scoring model.

*Artificial neural networks* (ANNs) are also being used to create credit scoring models (Abdou *et al*., 2008; Akkoç, 2012; Baesens *et al*., 2003 and Tsai & Wu, 2008). The basic building block of an ANN is a perceptron, the structure of which is illustrated in Figure 2.1.

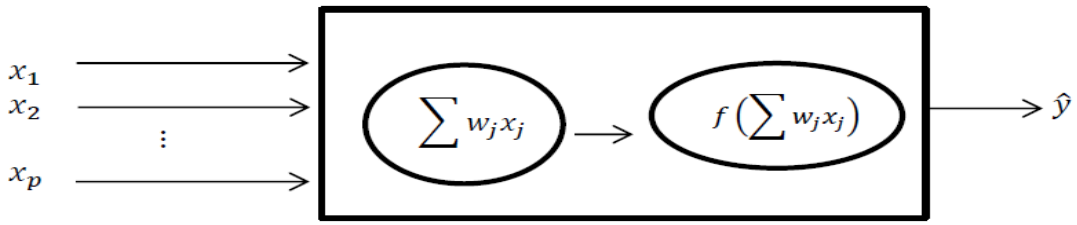


Figure 2.1**:** ANN perceptron

The $p$ attributes $\{x_1, x_2, \dots, x_p\}$ in Figure 2.1 represent the input features of the perceptron to which a weight $w_j$ is assigned. The weighted sum of these input features, $\sum_{j=1}^{p} w_j x_j$, then becomes an input to an *activation function*, $f\left(\sum_{j=1}^{p} w_j x_j\right)$, which then produces a predicted outcome, $\hat{y}$. The weights $w_j$ are generated from the learning sample using a *back propagation algorithm* that attempts to adjust the weights in such a way that the difference between the predicted outcome $\hat{y}$ and the corresponding known outcome $y$ is minimized. A sigmoid (S-shaped) function such as the logistic function,

$$\hat{y} = f\left(\sum_{j=1}^{p} w_j x_j\right) = \frac{1}{1 + e^{-\left(\sum_{j=1}^{p} w_j x_j\right)}} \tag{2.5}$$

is commonly used as an activation function. If the following identity activation function,

$$\hat{y} = f\left(\sum_{j=1}^{p} w_j x_j\right) = \sum_{j=1}^{p} w_j x_j \tag{2.6}$$

is used, then the ANN turn out to be the well-known multivariate linear regression problem.

The most commonly used ANN model for classification purposes is the multilayer perceptron (MLP), which comprises of an input layer, a hidden layer and an output layer. An illustration of a *feed forward* MLP artificial neural network is given in Figure 2.2 where the vector of predictor variables $\mathbf{x} = (x_1, x_2, \dots, x_p)^T$ serve as input features for a hidden layer with $M$ perceptrons denoted by $\{Z_k : k = 1,2, \dots, M\}$. The outcomes

from the perceptrons in the hidden layer then become the input features for each perceptron in the output layer, as denoted by $\{Y_k : k = 1,2, \dots, K\}$. Bishop (2006); Hastie, Tibshirani & Friedman (2009) and Ripley (1996) provide a more detailed description of how an artificial neural network works.

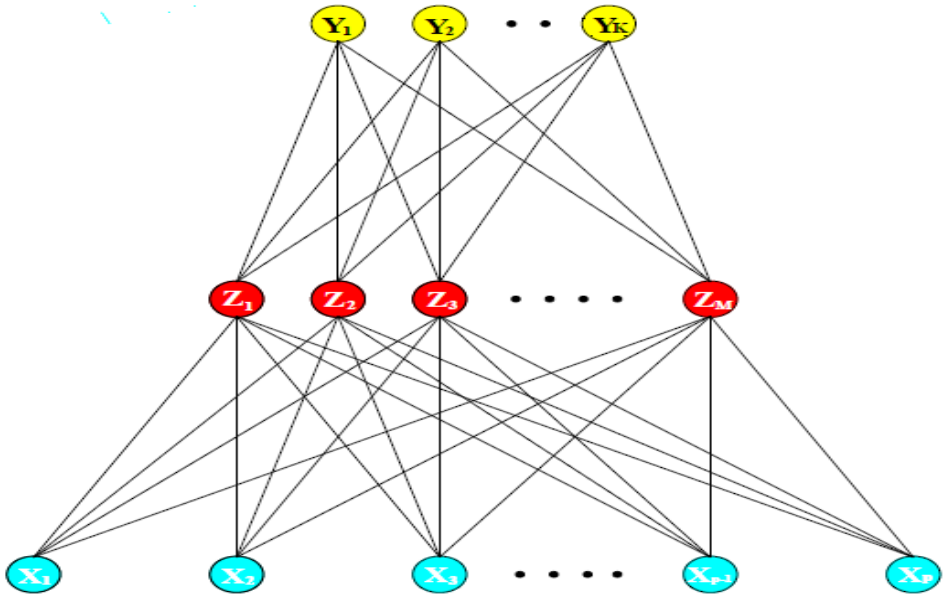


Figure 2.2: An illustration of a feed forward MLP artificial neural network (Source: Hastie *et al.*, 2009: 393)

ANNs have been shown to have excellent predictive power, performing better than conventional credit scoring techniques in many cases. However, their main drawback is that they are 'black box' methods meaning that they assign observations to classes without the operator knowing what has happened in-between. The weights $w_j$ are often difficult to interpret which creates a problem for lending institutions because the credit regulatory authorities often require lenders to provide reasons to new applicants for rejecting their loan applications.

This study focuses initially on developing a parametric classification technique known as Linear Discriminant Analysis (LDA) that uses linear combinations of predictor variables to come up with a class allocation rule. According to Hand (1997), LDA will have several desirable properties if the observations in each group $k \in \{1,2, \dots, K\}$ follow a multivariate normal distribution with mean $\boldsymbol{\mu}_k$ and each of the groups have a common covariance matrix $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}$. In particular, three different approaches that results

in the LDA classifier will be explored. These approaches will enable us to extend the LDA method so as to include the concept of *quadratic discriminant analysis* (QDA) (Geisser, 1964), *flexible discriminant analysis* (FDA) (Hastie, Tibshirani & Buja, 1994) and *mixture discriminant analysis* (MDA) (Hastie & Tibshirani, 1996).

Whilst focusing on the development of a classification technique that is non-parametric in nature the concept of a Classification And Regression Tree (CART) that is based on the work by Breiman *et al*. (1984) will be introduced. CART has become very popular because it outlines one's decision-making process using a tree-like structure, whose inherent logic is easy to interpret and understand. Small changes in the dataset may however produce an entirely different tree, thereby, casting doubts on CART's robustness as a classifier. To overcome this problem will be an exploration of some methods that have been developed for improving the performance of CART such as the concept of *bagging* (Breiman, 1996), *random forests* (Breiman, 2001) and *boosting* (Freund & Schapire, 1997).

Other techniques that may be considered when building credit-scoring models, though seldom used, include:

- linear regression (Hand & Henley, 1997; Orgler, 1970),
- probit analysis (Abdou *et al*., 2008),
- expert systems (Ben-David & Frank, 2009; Kumra, Stein & Assersohn, 2006),
- genetic programming (Lensberg, Eilifsen & McKee, 2006; Ong, Huang & Tzeng, 2005),
- support vector machines (Bellotti & Crook, 2009; Li, Shiue & Huang, 2006),
- *k*-nearest neighbor clustering (Baesens *et al.*, 2003; Henley & Hand, 1996).

Koh, Tan & Goh (2006), Lee & Chen (2005) and Lee *et al.* (2002) have all attempted to use a hybrid of one or more of the above models but their results are often difficult to interpret and time consuming to construct.

### 2.5 Credit scoring model performance measures

Having developed a method for scorecarding, the performance of this method needs to be evaluated. This leads to the concept of a specificity and sensitivity measure that we

will define using an example. Consider a credit-scoring model, which assigns a score $s_i \in \mathbb{R}$ to the $i^{th}$ applicant with a characteristic $x_i = \{\text{loan amount in rands}\}$. Assume that the actual class $y_i$ to which applicants belongs is also known:

$$y_i = \begin{cases} 0, & \text{for a non} - \text{defaulter} \\ 1, & \text{for a defaulter} \end{cases} \tag{2.7}$$

Table 2.1 shows a set of hypothetical results that could have been observed for such a credit-scoring model where someone is classified as being a non-defaulter (0) if their credit score lies above a threshold value $c = 0.5$, viz

$$\hat{y}_i = \begin{cases} 0, & if \quad s_i \geq c = 0.5 \\ 1, & otherwise \end{cases} \tag{2.8}$$

Table 2.1: A hypothetical sample of scored credit applicants

| Applicant ($i$) | Loan amount ($x_i$) | Actual class ($y_i$) | Credit score ($s_i$) | Assigned class ($\hat{y}_i$) |
|---|---|---|---|---|
| 1 | 20000 | 0 | 0.85 | 0 |
| 2 | 10000 | 1 | 0.70 | 0 |
| 3 | 30000 | 0 | 0.95 | 0 |
| 4 | 5000 | 1 | 0.35 | 1 |
| 5 | 12000 | 1 | 0.45 | 1 |
| 6 | 18000 | 1 | 0.50 | 0 |
| 7 | 7000 | 0 | 0.6 | 0 |
| 8 | 50000 | 1 | 0.8 | 0 |
| 9 | 5000 | 1 | 0.2 | 1 |
| 10 | 5000 | 0 | 0.25 | 1 |

This rule causes six applicants to be classified as non-defaulters (0) and the remaining four applicants to be classified as defaulters (1). A comparison of column (3) with column (5) shows that three out of the six applicants who were classified as being potential non-defaulters actually default on their loan obligations. A similar comparison

shows that one of the four applicants who were classified as being possible defaulters is in fact a non-defaulter. We can summarize the performance of this credit-scoring rule by cross tabulating what was predicted in terms of class membership with what actually happened with regard to class membership as shown in Table 2.2.

Known as a *classification matrix* or *confusion matrix*, Table 2.2 makes it easy to see that four out of the ten applicants were incorrectly classified. Conversely, six out of the ten applicants were correctly classified. Expressing these figures as a percentage one obtains an *error rate* of 0.4 and an *accuracy rate* of 0.6 for this classification rule.

Table 2.2: Classification matrix illustration

| | | | Predicted Default status | | |
|---|---|---|---|---|---|
| | | Default status | 0 | 1 | Total |
| Actual Default Status | Count | 0 | 3 | 3 | 6 |
| | | 1 | 1 | 3 | 4 |
| | % | 0 | 50 | 50 | 100.0 |
| | | 1 | 25 | 75 | 100.0 |

One can also use the classification matrix above to define the following model performance measures: a sensitivity measure, a specificity measure, a false alarm measure and a miss measure:

$$specificity = Pr\{s_i \geq c | x_i \in non - defaulter\} \quad (2.9)$$

$$sensitivity = Pr\{s_i < c | x_i \in defaulter\} \quad (2.10)$$

$$miss = Pr\{s_i \geq c | x_i \in defaulter\} \quad (2.11)$$

$$false\ alarm = Pr\{s_i < c | x_i \in non - defaulter\} \quad (2.12)$$

Due to the fact that the class allocation rule that was given in equation (2.8) classifies someone as being a non-defaulter if $s_i \geq c$, the specificity measure (2.9) can be interpreted as giving one the probability that a non-defaulter will be correctly classified as a non-defaulter. Likewise, the sensitivity measure can be interpreted as giving one the probability that a defaulter will be correctly classified as a defaulter. Thus, the

information in Table 2.2 shows a sensitivity measure of 75% and a specificity measure of 50%. The primary focus of this study is on the error rate, sensitivity and specificity measure that is being generated by a given classification method, noting that the other three measures (an accuracy rate, a miss and a false alarm) complement these three measures.

Table 2.3 shows how the sensitivity and specificity measures associated with the classification rule (2.8) can change as the cut-off value $c$ is varied across a range of values.

Table 2.3: Change in sensitivity and specificity across a range of cut-off values

| Cut-off value, $c$ | Sensitivity | Specificity |
|---|---|---|
| 0.2 | 1 | 0 |
| 0.25 | 1 | 0.166667 |
| 0.35 | 0.75 | 0.166667 |
| 0.45 | 0.75 | 0.333333 |
| 0.5 | 0.75 | 0.5 |
| 0.6 | 0.75 | 0.666667 |
| 0.7 | 0.5 | 0.666667 |
| 0.8 | 0.5 | 0.833333 |
| 0.85 | 0.5 | 1 |
| 0.95 | 0.25 | 1 |

A plot of the sensitivity values against the specificity values given in the above table is traced by the red curve in the Figure 2.3 (page 14). This is known as the *receiver operating characteristic* (ROC) curve.

A preferable model should have a high sensitivity and a high specificity value. Thus, its ROC curve must lie towards the top right corner. As a result, the area under the ROC curve (abbreviated AUC) would be greatest for the best model and lowest for the worst model. The blue curve in Figure 2.3 shows the ROC curve of a perfect model whose AUC value is one (1). Similarly, the diagonal green line is the ROC curve of a classification by chance model (i.e. random guessing) whose AUC value equals to half (0.5). The AUC value for our hypothetical model (red curve) which produced the values in Table 2.3 is 0.708, which is relatively high.

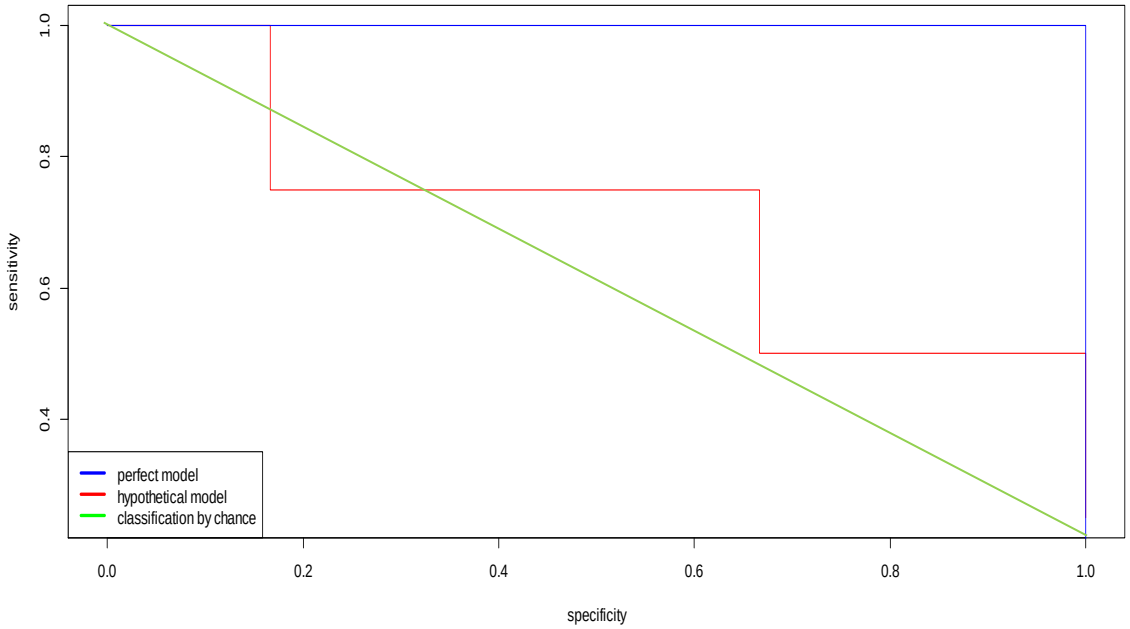


Figure 2.3: ROC curve for data in Table 2.3

We shall mainly use ROC curves to compare the overall discriminatory power (not accuracy) of the models we are going to develop, as the parameters of the classification rule varies.

## 2.6 Conclusion

The overview of credit scoring casts into light a number of pertinent issues. Arising from these issues is the need to develop better credit-scoring models that can accurately predict whether a new credit applicant is a potential defaulter. Subsequently, this study attempts to broaden one's scope of some of the classification models that can be used as scorecards. It also takes into account that credit scoring is increasingly becoming popular and that there is still limited knowledge of the underlying theory behind the credit-scoring models.

# CHAPTER 3

# 3. Linear Discriminant Analysis

## 3.1 Introduction

The objective of this chapter is to discuss a parametric classification technique called linear discriminant analysis (LDA). The term linear discriminant analysis refers to the way the classifier uses a linear combination of predictor variables to come up with a class allocation rule (Fisher, 1936). In particular, we will look at three different approaches that give rise to the LDA classifier. In developing these credit-scoring models, we will assume that we have a set of $i = 1,2, \dots, N$ training observations contained in a learning/training sample, $L$:

$$L = \{(\mathbf{x}_1, y_1)\,, (\mathbf{x}_2, y_2\,), \dots, (\mathbf{x}_N, y_N)\,\}$$

where the p-dimensional vector $\mathbf{x}_i = \left(x_{1i}, x_{2i}, \dots, x_{pi}\right)^T$ contain attributes of the $i^{th}$ observation. We will also assume that each training observation $\mathbf{x}_i$ belongs to one of $K$ possible classes, viz $y_i = k \in \{1,2, \dots, K\}$. Our goal is to develop a classification rule based on $L$ that assigns a new credit applicant $\mathbf{x}_i^{new} \in R^p$ in some optimal manner to one of the available $K$ classes.

## 3.2 The Bayesian approach

Our first approach, herein called the Bayesian classifier, uses Bayes' theorem to compute the posterior probability that a particular applicant belongs to one of the $k \in \{1,2, \dots, K\}$ groups (Geisser, 1964). Observations in a particular group are presumed to follow a $p$-dimensional multivariate normal distribution with a common covariance matrix. We will show that maximizing this posterior probability is equivalent to finding a value of $k \in \{1,2, \dots, K\}$ that maximizes a linear combination of the predictor variables in the vector $\mathbf{x}_i \in R^p$.

### 3.2.1 The procedure

Given a set of training observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, Bayes' theorem allows one to write

$$P(y_i = k|\mathbf{x}_i) = \frac{\mathrm{P}(y_i = k)P(\mathbf{x}_i|y_i = k)}{\sum_{l=1}^{K}\mathrm{P}(y_i = l)\mathrm{P}(\mathbf{x}_i|y_i = l)} = \frac{\pi_k f_k(\mathbf{x}_i)}{f(\mathbf{x}_i)} \qquad (3.1)$$

where, $P(y_i = k|\mathbf{x}_i)$ denotes the *posterior probability* that the $i^{th}$ observation with predictor variables $\mathbf{x}_i$ belongs to class $k$ and $\pi_k = \mathrm{P}(y_i = k)$ denotes the *prior probability* that this observation belongs to class $k$. Because the distribution of observations in each of the $K$ classes is presumed to follow a $p$-dimensional multivariate normal distribution with a common covariance matrix $\mathbf{\Sigma}_k = \mathbf{\Sigma}$, we have

$$f_k(\mathbf{x}_i) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_k)^T\mathbf{\Sigma}^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k)\right) \qquad (3.2)$$

where, $\boldsymbol{\mu}_k$ denotes the mean vector and $\mathbf{\Sigma}_k = \mathbf{\Sigma}$ the common covariance matrix for observations in class $k \in \{1,2,\dots,K\}$.

A classification rule for this Bayesian procedure assigns a new observation $\mathbf{x}_i^{new}$ to a class $k$ if that choice of value for $k \in \{1,2,\dots,K\}$ maximizes the posterior probability $P(y_i = k|\mathbf{x}_i^{new})$ given in equation (3.1). Since the denominator in equation (3.1) is the same for all values of $k$, we need only consider finding that value of $k$ that maximizes:

$$P(y_i = k|\mathbf{x}_i^{new}) \propto \pi_k f_k(\mathbf{x}_i^{new})$$

$$= \frac{\pi_k}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}_i^{new} - \boldsymbol{\mu}_k)^T\mathbf{\Sigma}^{-1}(\mathbf{x}_i^{new} - \boldsymbol{\mu}_k)\right) \qquad (3.3)$$

Because the natural logarithm function is monotonic, we need only find that value of $k$ that maximizes:

$$\ln[\pi_k f_k(\mathbf{x}_i^{new})] = \mathrm{In}\pi_k - \frac{p}{2}\mathrm{In}2\pi - \frac{1}{2}\mathrm{In}|\boldsymbol{\Sigma}| - \frac{1}{2}(\mathbf{x}_i^{new} - \boldsymbol{\mu}_k)^T\mathbf{\Sigma}^{-1}(\mathbf{x}_i^{new} - \boldsymbol{\mu}_k)$$

$$= \mathrm{In}\pi_k - \frac{p}{2}\mathrm{In}2\pi - \frac{1}{2}\mathrm{In}|\boldsymbol{\Sigma}| - \frac{1}{2}[(\mathbf{x}_i^{new})^T\mathbf{\Sigma}^{-1}\mathbf{x}_i^{new} - 2(\mathbf{x}_i^{new})^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_k + \boldsymbol{\mu}_k{}^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_k]$$

Omitting the expressions $\frac{p}{2}\mathrm{In}2\pi$ , $\frac{1}{2}\mathrm{In}|\boldsymbol{\Sigma}|$ and $(\mathbf{x}_i^{new})^T\mathbf{\Sigma}^{-1}\mathbf{x}_i^{new}$ because they do not depend on $k$, one only needs to find the value of $k$ that maximizes the following *classification function*:-

$$d_k(\mathbf{x}_i^{new}) = \mathrm{In}\pi_k - (\mathbf{x}_i^{new})^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k{}^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_k \qquad \forall k = 1,2,\dots,K \qquad (3.4)$$

### 3.2.2 Implementing the procedure

From a learning sample, $\{(\mathbf{x}_i, y_i)\}_i^N$ one can obtain the following parameter estimates (see, equation A.32 and A.34 in Appendix A for the derivation of the maximum likelihood based parameter estimates of sample mean $\bar{\mathbf{x}}$ and sample variance **S**):

$$\hat{\pi}_k = \frac{N_k}{N} \qquad \forall k = 1,2,\dots,K \tag{3.5}$$

$$\bar{\mathbf{x}}_k = \frac{1}{N_k}\sum_{y_i=k} \mathbf{x}_i \qquad \forall k = 1,2,\dots,K \tag{3.6}$$

$$\boldsymbol{S}_k = \frac{1}{N_k - 1}\sum_{y_i=k} (\mathbf{x}_i - \bar{\mathbf{x}}_k)\,(\mathbf{x}_i - \bar{\mathbf{x}}_k)^T \qquad \forall k = 1,2,\dots,K \tag{3.7}$$

$$\boldsymbol{S}_w = \frac{1}{N - K}\sum_{k=1}^{K} (N_k - 1)\boldsymbol{S}_k \qquad \forall k = 1,2,\dots,K \tag{3.8}$$

where $N_k$ and $N = \sum_{k=1}^{K} N_k$ denote the total number of observations in the $k^{th}$ group and overall sample, respectively. One would then assign a new observation $\mathbf{x}_i^{new}$ to that class $k$ which maximizes,

$$d_k(\mathbf{x}_i^{new}) = \ln\hat{\pi}_k - (\mathbf{x}_i^{new})^T \boldsymbol{S}_w^{-1}\bar{\mathbf{x}}_k - \frac{1}{2}\bar{\mathbf{x}}_k^T \boldsymbol{S}_w^{-1}\bar{\mathbf{x}}_k \tag{3.9}$$

over all values of $k \in \{1,2,\dots,K\}$.

When dealing with a $K = 2$ group classification problem, the above classification rule collapses into one where we can assign a new observation $\mathbf{x}_i^{new}$ to the first class (which we will label as class 1) if we have

$$d_1(\mathbf{x}_i^{new}) > d_2(\mathbf{x}_i^{new}) \Longrightarrow d_1(\mathbf{x}_i^{new}) - d_2(\mathbf{x}_i^{new}) > 0$$

$$\Longrightarrow (\mathbf{x}_i^{new})^T \boldsymbol{S}_w^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) - \ln\left(\frac{\hat{\pi}_1}{\hat{\pi}_2}\right) - \frac{1}{2}(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2)^T \boldsymbol{S}_w^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) > 0 \tag{3.10}$$

Setting,

$$\hat{\boldsymbol{\beta}} = \boldsymbol{S}_w^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{3.11}$$

and,

$$c = \left\{ \text{In}\left(\frac{\hat{\pi}_1}{\hat{\pi}_2}\right) + \frac{1}{2}(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2)^T \boldsymbol{S}_w^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \right\} \tag{3.12}$$

one would assign a new observation $\mathbf{x}_i^{new}$ to the class we have labeled 1 if we have

$$(\mathbf{x}_i^{new})^T \widehat{\boldsymbol{\beta}} > c \quad \Leftrightarrow \quad \widehat{\boldsymbol{\beta}}^T \mathbf{x}_i^{new} > c \tag{3.13}$$

If we assume equal prior probabilities ($\hat{\pi}_1 = \hat{\pi}_2$), then

$$c = \frac{1}{2}(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2)^T \widehat{\boldsymbol{\beta}} = \frac{1}{2}\widehat{\boldsymbol{\beta}}^T(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2) \tag{3.14}$$

and thus one would assign $\mathbf{x}_i^{new}$ to the class we have labeled 1 if we have

$$\widehat{\boldsymbol{\beta}}^T \mathbf{x}_i^{new} > \frac{1}{2}\widehat{\boldsymbol{\beta}}^T(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2) \tag{3.15}$$

### 3.2.3 Incorporating a misclassification cost

Granting a loan to someone who eventually defaults will result in the lender losing some important revenue. On the other hand, not granting a loan to someone who will not default will also result in the lender losing some potential revenue. To incorporate misclassification costs into the modelling process, we define a $K \times K$ cost matrix **C**, where $K$ denotes the number of classes in the sample. Let $C_{lm}$ represent the cost of assigning an observation $\mathbf{x}_i$ to a class $y = l$ when its true class is $y = m$. For a two class problem, the cost matrix would be as shown in Table 3.1.

Table 3.1: Cost matrix for a two-class problem

| | Classified as | |
|---|---|---|
| True class | k=1 | k=2 |
| k=1 | $C_{11}$ | $C_{21}$ |
| k=2 | $C_{12}$ | $C_{22}$ |

In order to minimize the expected cost associated with misclassification, Anderson (1958) suggested that one assign an observation $\mathbf{x}_i$ to a particular class $l$ if:

$$\sum_{k=1}^{K} \pi_k C_{lk} f_k(\mathbf{x}_i) < \sum_{k=1}^{K} \pi_k C_{mk} f_k(\mathbf{x}_i) \quad \forall 1 \leq k \leq K, l \neq m. \tag{3.16}$$

Assuming equal misclassification costs, for example

$$C_{lm} = c \in \mathbb{R} \qquad \forall\, 1 \leq k \leq K, l \neq m \tag{3.17}$$

the above decision rule (3.16) simplifies to one where we would assign an observation $\mathbf{x}_i$ to class $l$ if

$$\pi_l f_l(\mathbf{x}_i) = \max_k [\pi_k f_k(\mathbf{x}_i)] \qquad \forall 1 \leq k \leq K \tag{3.18}$$

Applying this rule to a two-class problem, one would assign an observation $\mathbf{x}_i$ to a class $k = 1$ if we have

$$\pi_1 C_{11} f_1(\mathbf{x}_i) + \pi_2 C_{12} f_2(\mathbf{x}_i) < \pi_1 C_{21} f_1(\mathbf{x}_i) + \pi_2 C_{22} f_2(\mathbf{x}_i)$$

$$\Leftrightarrow \pi_2 C_{12} f_2(\mathbf{x}_i) < \pi_1 C_{21} f_1(\mathbf{x}_i) \qquad \{if\ we\ assume\ \ C_{11} = C_{22} = 0\}$$

### 3.3 Fisher's approach: The K=2 class problem

Our second approach (known as Fisher's method) will attempt to use a set of optimally derived linear combinations of the predictor variables $\mathbf{x}_i \in R^p$ to map these observations which occupy a $p$ dimensional space onto a $(K-1)$ dimensional space where $K < p$. This section focusses on the special $K = 2$ class problem.

#### 3.3.1 The procedure

Given a set of training observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, let $N_1$ denote the number of observations belonging to the first group (which we will label group 1) and $N_2$ denote the number of observations belonging to the second group (which we will label group 2). Furthermore, let

$$\bar{\mathbf{x}}_1 = \frac{1}{N_1} \sum_{y_i=1} \mathbf{x}_i \tag{3.19}$$

and

$$\bar{\mathbf{x}}_2 = \frac{1}{N_2}\sum_{y_i=2}\mathbf{x}_i \tag{3.20}$$

denote the *p*-dimensional mean vectors associated with the observations in the first and second groups respectively and let

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{i}\mathbf{x}_i \tag{3.21}$$

denote an overall *p*-dimensional mean vector for all the observations in the learning sample (where the assignment to a particular group is being ignored). Fisher's method seeks to find a direction vector $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_{\mathrm{p}})^T$, which can then be used to project each observation $\mathbf{x}_i$, using the following function

$$d(\mathbf{x}_i) = \boldsymbol{\beta}^T\mathbf{x}_i = (\beta_1, \beta_2, \ldots, \beta_{\mathrm{p}})\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ . \\ x_{\mathrm{p}} \end{pmatrix} \tag{3.22}$$

onto a one-dimensional space (i.e. the real number line) where the separation between the projected observations from both groups is a maximum. For example, letting

$$|d(\bar{\mathbf{x}}_1) - d(\bar{\mathbf{x}}_2)| = |\boldsymbol{\beta}^T\bar{\mathbf{x}}_1 - \boldsymbol{\beta}^T\bar{\mathbf{x}}_2| = |\boldsymbol{\beta}^T(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)\,| \tag{3.23},$$

defines a distance measure between the mean of the projected observations in group 1 and those in group 2 that one may want to maximize with respect $\boldsymbol{\beta}$. One may also want the projected observations within each group to have a variance, which is as small as is possible. This suggests that one should rather attempt to find a projection vector $\boldsymbol{\beta}$ that maximizes

$$\lambda = \frac{[\boldsymbol{\beta}^T(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)]^2}{\tilde{\boldsymbol{S}}_w} \tag{3.24}$$

where,

$$\tilde{\boldsymbol{S}}_w = \sum_{y_i=1}\boldsymbol{\beta}^T(\mathbf{x}_i - \bar{\mathbf{x}}_1)(\mathbf{x}_i - \bar{\mathbf{x}}_1)^T\boldsymbol{\beta} + \sum_{y_i=2}\boldsymbol{\beta}^T(\mathbf{x}_i - \bar{\mathbf{x}}_2)(\mathbf{x}_i - \bar{\mathbf{x}}_2)^T\boldsymbol{\beta} \tag{3.25}$$

denotes a *measure of scatter* for the projected observations within each of the two different groups. Maximizing (3.24) with respect to $\boldsymbol{\beta}$ amounts to finding a projection vector that causes the projected observations $d(\mathbf{x}_i)$ in the same group to be as close as possible to each other in the transformed space (so that we have a small denominator appearing in equation (3.24)). At the same time, the projection vector must force the transformed groups' means to be as far apart as is possible (so that we have a large numerator appearing in equation (3.24)).

Letting,

$$\boldsymbol{S}_k = \sum_{y_i \in k} (\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_i - \bar{\mathbf{x}}_k)^T \qquad k = 1,2 \qquad (3.26)$$

denote a *within-class scatter* matrix for the *p*-dimensional observations belonging to class $k$ that come from our original space $\mathbf{x}_i \in R^p$ and

$$\boldsymbol{S}_b = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T \qquad (3.27)$$

a *between-class scatter* matrix for these original (unprojected) observations, one can write,

$$[\boldsymbol{\beta}^T(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)]^2 = \boldsymbol{\beta}^T(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T\boldsymbol{\beta} = \boldsymbol{\beta}^T\boldsymbol{S}_b\,\boldsymbol{\beta} \qquad (3.28)$$

and equation (3.25) becomes

$$\tilde{\boldsymbol{S}}_w = \boldsymbol{\beta}^T\boldsymbol{S}_w\,\boldsymbol{\beta} \qquad (3.29)$$

where,

$$\boldsymbol{S}_w = \boldsymbol{S}_1 + \boldsymbol{S}_2 \qquad (3.30)$$

Thus, Fisher's method can now be viewed as attempting to maximize

$$\lambda = \frac{[\boldsymbol{\beta}^T(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)]^2}{\tilde{\boldsymbol{S}}_w} = \frac{\boldsymbol{\beta}^T\boldsymbol{S}_b\,\boldsymbol{\beta}}{\boldsymbol{\beta}^T\boldsymbol{S}_w\,\boldsymbol{\beta}} \qquad (3.31)$$

with respect to $\boldsymbol{\beta}$ which can be done by solving the following set of first-order conditions,

$$\frac{\partial \lambda}{\partial \boldsymbol{\beta}} = \frac{2\boldsymbol{S}_b\boldsymbol{\beta}(\boldsymbol{\beta}^T\boldsymbol{S}_w\,\boldsymbol{\beta}) - 2(\boldsymbol{S}_w\,\boldsymbol{\beta})\boldsymbol{\beta}^T\boldsymbol{S}_b\boldsymbol{\beta}}{[\boldsymbol{\beta}^T\boldsymbol{S}_w\,\boldsymbol{\beta}]^2} = 0$$

$$\Longrightarrow \frac{\boldsymbol{S}_b\boldsymbol{\beta}(\boldsymbol{\beta}^T\boldsymbol{S}_w\,\boldsymbol{\beta})}{\boldsymbol{\beta}^T\boldsymbol{S}_w\boldsymbol{\beta}} - \frac{(\boldsymbol{S}_w\,\boldsymbol{\beta})\boldsymbol{\beta}^T\boldsymbol{S}_b\boldsymbol{\beta}}{\boldsymbol{\beta}^T\boldsymbol{S}_w\boldsymbol{\beta}} = 0$$

$$\Longrightarrow \boldsymbol{S}_b\boldsymbol{\beta} = \lambda\,\boldsymbol{S}_w\,\boldsymbol{\beta} \tag{3.32}$$

The above system of equations is a *generalized eigenvalue problem* that needs to be solved for $\boldsymbol{\beta}$. If $\boldsymbol{S}_w$ is invertible then the above problem can be rewritten as a standard eigenvalue problem:

$$\boldsymbol{S}_w^{-1}\boldsymbol{S}_b\boldsymbol{\beta} = \lambda\boldsymbol{\beta} \tag{3.33}$$

which is easier to solve.

In particular, for *any* vector $\mathbf{v} \in R^p$, we will always have

$$\boldsymbol{S}_b\mathbf{v} = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T\mathbf{v} = \alpha(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \Longrightarrow \boldsymbol{S}_w^{-1}\boldsymbol{S}_b\mathbf{v} = \alpha\boldsymbol{S}_w^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{3.34}$$

where $\alpha = (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^T\mathbf{v}$. Thus, one can write

$$\boldsymbol{S}_w^{-1}\boldsymbol{S}_b\boldsymbol{\beta} = \alpha\boldsymbol{S}_w^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{3.35}$$

implying that

$$\widehat{\boldsymbol{\beta}} = \boldsymbol{S}_w^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2) \tag{3.36}$$

is a solution vector to (3.32) and thus to (3.33) when $\boldsymbol{S}_w$ is invertible.

Multiplying the above solution vector $\boldsymbol{\beta}$ by an arbitrary constant $\gamma$ generates another vector $\boldsymbol{w} = \gamma\boldsymbol{\beta}$, which also produces the same maximum value for (3.31), viz

$$\lambda = \frac{\gamma\boldsymbol{\beta}^T\boldsymbol{S}_b\gamma\boldsymbol{\beta}}{\gamma\boldsymbol{\beta}^T\boldsymbol{S}_w\gamma\boldsymbol{\beta}} = \frac{\boldsymbol{\beta}^T\boldsymbol{S}_b\boldsymbol{\beta}}{\boldsymbol{\beta}^T\boldsymbol{S}_w\boldsymbol{\beta}} \tag{3.37}$$

Thus, our choice of criterion $\lambda$ is invariant with respect to a rescaling of the solution vector (3.36) that we have derived above. It should be noted therefore that what is of

essential importance is the direction of this solution vector rather than its overall magnitude. For this reason, one can recast the above classification problem as a constrained maximization problem where we want to maximize

$$\lambda' = \boldsymbol{\beta}^T \boldsymbol{S}_b \boldsymbol{\beta} \tag{3.38}$$

subject to the following normalization constraint

$$\boldsymbol{\beta}^T \boldsymbol{S}_w \boldsymbol{\beta} = 1$$

also being imposed on the entries in $\boldsymbol{\beta}$.

To have a better and insightful understanding of how Fisher's approach is able to work, consider Figure 3.1 and Figure 3.2 below. The figures contain a projection of two-dimensional observations $\mathbf{x}_i \in R^2$ onto a one-dimensional space that is being defined by a mapping $\boldsymbol{\beta}^T \mathbf{x}_i$, where $\boldsymbol{\beta}$ is being represented by the orientation of the line supporting the histograms of both groups of data in Figures 3.1 and 3.2, respectively.

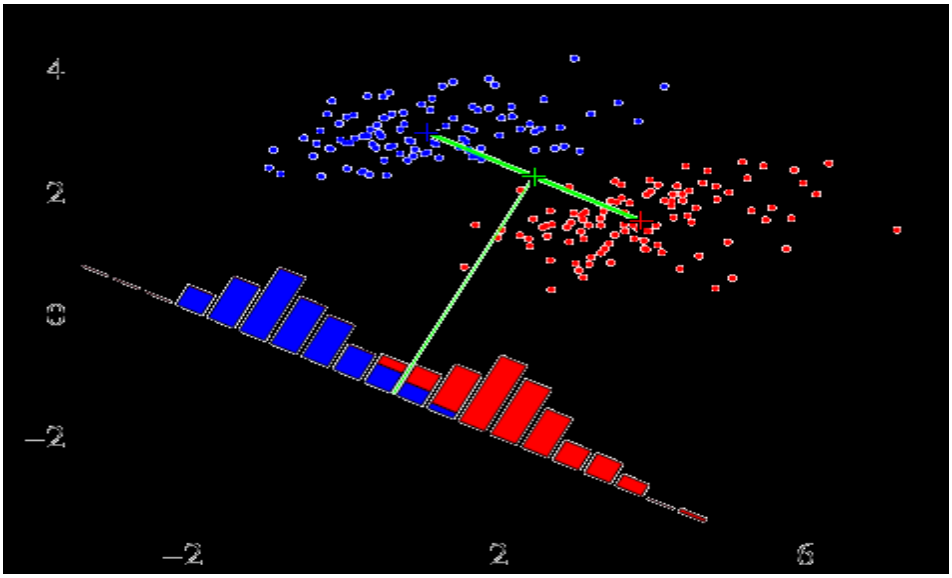


Figure 3.1: A projection of two-dimensional observations onto a one-dimensional space

In Figure 3.1, it can be observed that there is a considerable amount of overlap between the distribution functions of both groups. When we change the orientation of this slope, however, the separation between the two histograms becomes more apparent eventually producing the plot that we have in Figure 3.2.

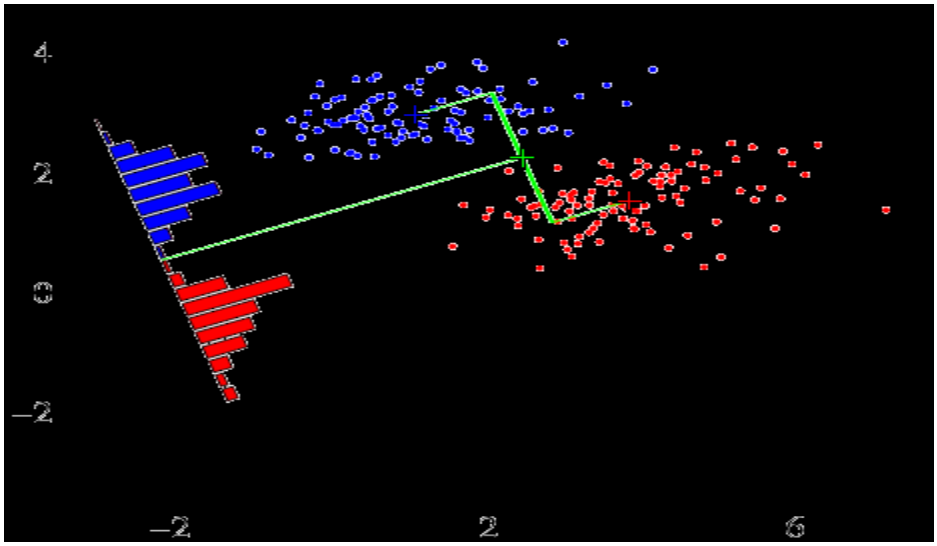


Figure 3.2: Changing the orientation of the projection vector

### 3.3.2 Implementing the procedure

Given a new observation $\mathbf{x}_i^{new} \in R^p$, one needs to project this observation onto a one-dimensional space using the projection vector $\widehat{\boldsymbol{\beta}} = \boldsymbol{S}_w^{-1}(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)$. Assign $\mathbf{x}_i^{new}$ to the group labeled '1' if the distance between this projected observation $d(\mathbf{x}_i^{new}) = \widehat{\boldsymbol{\beta}}^T\mathbf{x}_i^{new}$ and the mean value $\widehat{\boldsymbol{\beta}}^T\bar{\mathbf{x}}_1$ of all the projected observation in that group is smaller than the distance between this projected observation and the mean value $\widehat{\boldsymbol{\beta}}^T\bar{\mathbf{x}}_2$ of all the projected observation in the other group (that we have labeled group 2). Essentially, we can assign a new observation $\mathbf{x}_i^{new}$ to a particular class depending on whether the projected value $d(\mathbf{x}_i^{new}) = \widehat{\boldsymbol{\beta}}^T\mathbf{x}_i^{new}$ lies to the left or right of a cut-off point $c$ that lies exactly half way between the two-*projected* group means (see Figure 3.3);

$$c = \left\{\frac{1}{2}\widehat{\boldsymbol{\beta}}^T(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2)\right\} \tag{3.39}$$

Thus, we will assign a new observation $\mathbf{x}_i^{new} \in R^p$ to class 1 if we have

$$\widehat{\boldsymbol{\beta}}^T\mathbf{x}_i^{new} > \frac{1}{2}\widehat{\boldsymbol{\beta}}^T(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2) \tag{3.40}$$

Otherwise, one would assign this observation to the other class, which we have labeled class 2.

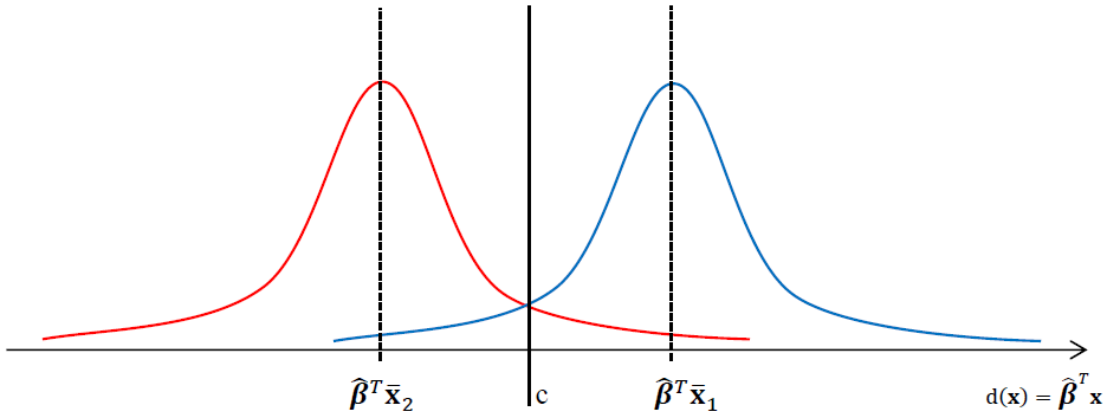


Figure 3.3: Distribution of the projected observations on the one-dimensional space (assuming $\widehat{\boldsymbol{\beta}}^T\bar{\mathbf{x}}_2 < \widehat{\boldsymbol{\beta}}^T\bar{\mathbf{x}}_1$)

Because of the equivalence of Fisher's canonical discriminant coefficients (3.36) and the Bayesian based discriminant coefficients (3.11), one may want to choose the following cut-off point from the Bayesian classifier that is given in equation (3.12),

$$c = \left\{\ln\left(\frac{\hat{\pi}_1}{\hat{\pi}_2}\right) + \frac{1}{2}\widehat{\boldsymbol{\beta}}^T(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2)\right\} \tag{3.41}$$

and use it in (3.40) in place of the cut-off point computed in (3.39). This has the effect of adjusting Fisher's classification function (3.40) to take into account the possibility that the size of the two groups may not be the same. If the size of the two groups are the same (implied by the similar shapes of the distribution curves in Figure 3.3), then the cut-off point (3.41) would be the same as the one in (3.39) since we will now have $\hat{\pi}_1 = \hat{\pi}_2$.

### 3.3.3 Equivalence between the Bayesian and Fisher approach

Formula (3.15) indicates that the classification rule that has been derived under the Bayesian approach becomes equivalent to Fisher's class allocation rule (3.40) when we assume equal prior probabilities.

## 3.4 Fisher's approach: The K>2 class problem

For the sake of completeness, in this section we shall generalize Fisher's approach to LDA discussed in the previous section to the $K > 2$ class problem.

### 3.4.1 The procedure

Instead of working with a single projection vector, for a $K > 2$ class problem one may want to consider an approach that attempts to include, as columns of a $p \times J$ matrix $\boldsymbol{B}$, an appropriately chosen set of $J \leq K - 1$ projection vectors $\{\boldsymbol{\beta}_k : k = 1,2,\cdots,J\}$ that can then be used to project each observation $\mathbf{x}_i \in R^p$ onto a smaller *J*- dimensional subspace (providing $p > J$) with coordinates

$$\boldsymbol{d}(\mathbf{x}_i) = \boldsymbol{B}^T \mathbf{x}_i \tag{3.42}$$

where the separation between the *K* different groups becomes easier to identify.

Using observations $\{(\mathbf{x}_i, y_i)\}_i^N$ from a learning sample to produce the following estimate for the mean vector of all the observations belonging to class *k*

$$\bar{\mathbf{x}}_k = \frac{\sum_{\mathbf{x}_i \in k} \mathbf{x}_i}{N_k} \qquad \forall k = 1, \dots, K \tag{3.43}$$

a total within-class scatter matrix for all the observations that are being projected onto this smaller *J*- dimensional subspace can be given by

$$\tilde{\boldsymbol{S}}_w \equiv \sum_{\mathrm{k}=1}^{\mathrm{K}} \sum_{\mathrm{y}_i \in \mathrm{k}} \boldsymbol{B}^T (\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_i - \bar{\mathbf{x}}_k)^T \boldsymbol{B} = \boldsymbol{B}^T \boldsymbol{S}_w \boldsymbol{B} \tag{3.44}$$

where,

$$\boldsymbol{S}_w = \sum_{k=1}^{K} \sum_{y_i \in k} (\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_i - \bar{\mathbf{x}}_k)^T \tag{3.45}$$

denotes a within-class scatter matrix for all the observations in the learning sample that have been collected in the original p-dimensional space. Similarly, setting

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \tag{3.46}$$

a suitably weighted between class scatter matrix for all the observations in our learning sample can be given by,

$$\boldsymbol{S}_b = \sum_{k=1}^{K} N_k(\bar{\mathbf{x}}_k - \bar{\mathbf{x}})(\bar{\mathbf{x}}_k - \bar{\mathbf{x}})^T \tag{3.47}$$

with

$$\tilde{\boldsymbol{S}}_b \equiv \sum_{k=1}^{K} N_k \boldsymbol{B}^T(\bar{\mathbf{x}}_k - \bar{\mathbf{x}})(\bar{\mathbf{x}}_k - \bar{\mathbf{x}})^T \boldsymbol{B} = \boldsymbol{B}^T \boldsymbol{S}_b \boldsymbol{B} \tag{3.48}$$

representing a between class scatter matrix for all the projected observations, $\boldsymbol{B}^T\mathbf{x}_i \in R^J$ in our learning sample . An approach that mirrors the maximization of (3.31) for the *K=2* class problem would then attempt to maximize

$$\lambda = \frac{\boldsymbol{B}^T \boldsymbol{S}_b \boldsymbol{B}}{\boldsymbol{B}^T \boldsymbol{S}_w \boldsymbol{B}} \tag{3.49}$$

with respect to $\boldsymbol{B}$. However, this is no longer possible since both $\boldsymbol{B}^T\boldsymbol{S}_b\boldsymbol{B}$ and $\boldsymbol{B}^T\boldsymbol{S}_w\boldsymbol{B}$ are now square matrices of order $J$. To overcome this problem, Fukunaga (1990:448) has suggested that one attempt (for each fixed value of *J*) to find a projection matrix $\boldsymbol{B} \in R^{p \times J}$ that will maximize:

$$\lambda(J) = tr\{(\boldsymbol{B}^T\boldsymbol{S}_w\boldsymbol{B})^{-1}(\boldsymbol{B}^T\boldsymbol{S}_b\boldsymbol{B})\} \tag{3.50}$$

However, it is important to note that because $\boldsymbol{S}_b$ is a sum of *K* matrices each having a rank equal to one (1), the following constraint

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i = \frac{1}{N}\sum_{i=1}^{N} N_k\bar{\mathbf{x}}_k \tag{3.51}$$

ensures that the $p \times p$ dimensional matrix $\boldsymbol{S}_b$ can have a rank at most equal to $(K-1)$. Thus, in (3.50) one is attempting to maximize the trace of a $J \times J$ dimensional matrix $(\boldsymbol{B}^T\boldsymbol{S}_w\boldsymbol{B})^{-1}(\boldsymbol{B}^T\boldsymbol{S}_b\boldsymbol{B})$ whose rank can equal at most $(K-1)$. Since the trace of a matrix equals the sum of its eigenvalues and the rank of a matrix equals the number of non-zero eigenvalues in that matrix, one will not be able to further increase the value of $\lambda(J)$ (when viewed as a function of *J*) by projecting the *p*-dimensional observations $\mathbf{x}_i$ using $d(\mathbf{x}_i) = \boldsymbol{B}^T\mathbf{x}_i$ into a space of dimension higher than $J = K-1$. With the above choice of criterion in mind, we thus need to consider solely what happens when we

project a set of observations $\mathbf{x}_i$ into a space of dimension $J$ where $J$ is less than or equal to $K-1$. If we include all $K-1$ discriminant vectors as column vectors in the projection matrix $\boldsymbol{B}$ then we get what is termed a *full-rank* LDA classification rule. If we only use $J < K-1$ vectors then we get what is called a *reduced-rank* LDA classification rule.

For a fixed $J \in \{1, \ldots, K-1\}$, now consider the problem of finding a $p \times J$ projection matrix $\boldsymbol{B}$ that maximizes (3.50). One needs to solve the following first order conditions that generate that maximum, viz

$$\frac{\partial tr\{(\boldsymbol{B}^T\boldsymbol{S}_w\boldsymbol{B})^{-1}(\boldsymbol{B}^T\boldsymbol{S}_b\,\boldsymbol{B})\}}{\partial \boldsymbol{B}} = \mathbf{0}$$

$$\Longrightarrow \frac{d}{d\boldsymbol{B}_1}[tr\{(\boldsymbol{B}_1^T\boldsymbol{S}_w\boldsymbol{B}_1)^{-1}(\boldsymbol{B}^T\boldsymbol{S}_b\boldsymbol{B})\}]_{\boldsymbol{B}_1=\boldsymbol{B}}$$

$$+\frac{d}{d\boldsymbol{B}_2}[tr\{(\boldsymbol{B}^T\boldsymbol{S}_w\boldsymbol{B})^{-1}(\boldsymbol{B}_2^T\boldsymbol{S}_b\boldsymbol{B}_2)\}]_{\boldsymbol{B}_2=\boldsymbol{B}} = \mathbf{0} \quad (3.52)$$

Since (see Fukunaga (1990: 566))

$$\frac{d}{d\boldsymbol{B}}[tr\{(\boldsymbol{B}^T\boldsymbol{S}\,\boldsymbol{B})^{-1}\boldsymbol{R}\}] = -\boldsymbol{S}\boldsymbol{B}(\boldsymbol{B}^T\mathbf{S}\,\boldsymbol{B})^{-1}(\boldsymbol{R}+\boldsymbol{R}^T)(\boldsymbol{B}^T\mathbf{S}\,\boldsymbol{B})^{-1} \quad (3.53)$$

equation (3.52) takes on the form (where $\boldsymbol{S}_b$ is symmetric $\Longrightarrow \boldsymbol{B}^T\boldsymbol{S}_b\boldsymbol{B} = \boldsymbol{B}^T\boldsymbol{S}_b^T\boldsymbol{B}$):

$$-\boldsymbol{S}_w\boldsymbol{B}(\boldsymbol{B}^T\boldsymbol{S}_w\,\boldsymbol{B})^{-1}[\boldsymbol{B}^T\boldsymbol{S}_b\,\boldsymbol{B}+\boldsymbol{B}^T\boldsymbol{S}_b^T\boldsymbol{B}](\boldsymbol{B}^T\boldsymbol{S}_w\boldsymbol{B})^{-1}+(\boldsymbol{B}^T\boldsymbol{S}_w\boldsymbol{B})^{-1}2\boldsymbol{S}_b\boldsymbol{B} = \mathbf{0}$$

$$\Longrightarrow -2\boldsymbol{S}_w\boldsymbol{B}(\boldsymbol{B}^T\boldsymbol{S}_w\,\boldsymbol{B})^{-1}[\boldsymbol{B}^T\boldsymbol{S}_b\boldsymbol{B}](\boldsymbol{B}^T\boldsymbol{S}_w\,\boldsymbol{B})^{-1}+2\boldsymbol{S}_b\boldsymbol{B}(\boldsymbol{B}^T\boldsymbol{S}_w\,\boldsymbol{B})^{-1} = \mathbf{0}$$

$$\Longrightarrow \boldsymbol{S}_b\boldsymbol{B}(\boldsymbol{B}^T\boldsymbol{S}_w\,\boldsymbol{B})^{-1} = \boldsymbol{S}_w\boldsymbol{B}(\boldsymbol{B}^T\boldsymbol{S}_w\,\boldsymbol{B})^{-1}[\boldsymbol{B}^T\boldsymbol{S}_b\boldsymbol{B}](\boldsymbol{B}^T\boldsymbol{S}_w\boldsymbol{B})^{-1} \quad (3.54)$$

Multiplying both sides of (3.54) by $\boldsymbol{B}^T\boldsymbol{S}_w\boldsymbol{B}$ gives,

$$\boldsymbol{S}_b\boldsymbol{B} = \boldsymbol{S}_w\boldsymbol{B}(\boldsymbol{B}^T\boldsymbol{S}_w\,\boldsymbol{B})^{-1}(\boldsymbol{B}^T\boldsymbol{S}_b\boldsymbol{B}) \quad (3.55)$$

If $\boldsymbol{S}_w$ is a full rank matrix, multiplying both sides of (3.55) by $\boldsymbol{S}_w^{-1}$ will produce the following system of equations that will need to be solved for $\boldsymbol{B}$;

$$\boldsymbol{S}_w^{-1}\boldsymbol{S}_b\boldsymbol{B} = \boldsymbol{B}(\boldsymbol{B}^T\boldsymbol{S}_w\,\boldsymbol{B})^{-1}(\boldsymbol{B}^T\boldsymbol{S}_b\boldsymbol{B}) \quad (3.56)$$

**Theorem**: Any two symmetric matrices $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ can always be simultaneously diagonalized as,

$$\boldsymbol{A}^T \boldsymbol{S}_1 \boldsymbol{A} = \boldsymbol{I} \quad \text{and} \quad \boldsymbol{A}^T \boldsymbol{S}_2 \boldsymbol{A} = \boldsymbol{\Lambda} \tag{3.57}$$

where, $\boldsymbol{\Lambda}$ and $\boldsymbol{A}$ denote the diagonalized eigenvalue and eigenvector matrices of $\boldsymbol{S}_1^{-1}\boldsymbol{S}_2$, respectively.

**Proof**: (see Fukunaga (1990:31-32))
Note that $\boldsymbol{S}_1 = (\boldsymbol{A}^T)^{-1}\boldsymbol{A}^{-1} = (\boldsymbol{A}\boldsymbol{A}^T)^{-1}$ and $\boldsymbol{S}_2 \boldsymbol{A} = (\boldsymbol{A}^T)^{-1}\boldsymbol{\Lambda}$ thus,

$$\Longrightarrow \boldsymbol{S}_1^{-1}\boldsymbol{S}_2\boldsymbol{A} = \boldsymbol{A}\boldsymbol{A}^T(\boldsymbol{A}^T)^{-1}\boldsymbol{\Lambda} \;=\; \boldsymbol{A}\,\boldsymbol{\Lambda} \tag{3.58}$$

implying that the entries in the diagonalized matrix $\boldsymbol{\Lambda}$ and the column vectors of **A** are the eigenvalue and eigenvector matrices associated with the matrix $\mathbf{S}_1^{-1}\mathbf{S}_2$, respectively.

After setting

$$\boldsymbol{S}_1 = \boldsymbol{B}^T \boldsymbol{S}_w \boldsymbol{B} \tag{3.59}$$

and

$$\boldsymbol{S}_2 = \boldsymbol{B}^T \boldsymbol{S}_b \boldsymbol{B} \tag{3.60}$$

the above theorem implies that the equation system (3.56) that we are attempting to solve for $\boldsymbol{B}$ can be rewritten as

$$\boldsymbol{S}_w^{-1}\boldsymbol{S}_b \boldsymbol{B} = \boldsymbol{B}\,\boldsymbol{S}_1^{-1}\boldsymbol{S}_2 \;= \boldsymbol{B}\;\boldsymbol{A}\,\boldsymbol{\Lambda}\boldsymbol{A}^{-1} \tag{3.61}$$

or

$$\boldsymbol{S}_w^{-1}\boldsymbol{S}_b(\boldsymbol{B}\boldsymbol{A}) = (\boldsymbol{B}\boldsymbol{A})\boldsymbol{\Lambda} \tag{3.62}$$

where the diagonal components of $\boldsymbol{\Lambda}$ and the column vectors in $\boldsymbol{B}\boldsymbol{A}$ are eigenvalues and eigenvectors of the matrix $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$, respectively.

Since the trace of a matrix equals the sum of the eigenvalues associated with that matrix, if we want to maximize

$$\lambda(J) = tr\{(\boldsymbol{B}^T\boldsymbol{S}_w\boldsymbol{B})^{-1}(\boldsymbol{B}^T\boldsymbol{S}_b\boldsymbol{B})\} = tr\,(\boldsymbol{\Lambda}\boldsymbol{A}^{-1}\boldsymbol{A}) = tr(\boldsymbol{\Lambda}) \tag{3.63}$$

then we must choose as the $J$ column vectors making up the $p \times J$ matrix $\boldsymbol{BA}$ those eigenvectors of $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$ that correspond with the largest $J$ eigenvalues of $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$.

Given any orthogonal matrix $\boldsymbol{P}$ (i.e. $\boldsymbol{P}\boldsymbol{P}^T = \boldsymbol{I}$), because

$$\begin{aligned}\lambda(J) &= tr\{(\boldsymbol{B}^T\boldsymbol{S}_w\boldsymbol{B})^{-1}(\boldsymbol{B}^T\boldsymbol{S}_b\boldsymbol{B})\} \\ &= tr\{(\boldsymbol{P}\boldsymbol{P}^T\boldsymbol{B}^T\boldsymbol{S}_w\boldsymbol{B})^{-1}(\boldsymbol{P}\boldsymbol{P}^T\boldsymbol{B}^T\boldsymbol{S}_b\boldsymbol{B})\} \\ &= tr\{(\boldsymbol{P}^T\boldsymbol{B}^T\boldsymbol{S}_w\boldsymbol{B}\boldsymbol{P})^{-1}(\boldsymbol{P}^T\boldsymbol{B}^T\boldsymbol{S}_b\boldsymbol{B}\boldsymbol{P})\}\end{aligned} \tag{3.64}$$

we have a lack of uniqueness relating to the projection matrix $\boldsymbol{B}$ that one can use to maximize (3.50). One can therefore recast the above problem in a constrained maximization framework, where we attempt to maximize $tr\{(\boldsymbol{B}^T\boldsymbol{S}_b\boldsymbol{B})\}$ subject to the following orthogonality and normalization constraints being applied to all the column vectors $\{\boldsymbol{\beta}_k; k = 1,2,\dots,J\}$ that make up $\boldsymbol{B}$, viz

$$\boldsymbol{B}^T\boldsymbol{S}_w\,\boldsymbol{B} = \boldsymbol{I} \tag{3.65}$$

or where we successively maximize $\boldsymbol{\beta}_k^T\boldsymbol{S}_b\boldsymbol{\beta}_k$ subject to

$$\boldsymbol{\beta}_k^T\boldsymbol{S}_w\boldsymbol{\beta}_k = 1, \boldsymbol{\beta}_k^T\boldsymbol{S}_w\boldsymbol{\beta}_j = 0 \quad \forall\, j < k = 1,2,\dots,J \tag{3.66}$$

### 3.4.2 Implementing the procedure

A new observation $\mathbf{x}_i^{new} \in \mathrm{R}^{\mathrm{p}}$ can be assigned to a particular class using the following set of rules:

**Step 1**: Use the observations $\{(\mathbf{x}_i, y_i)\}_i^N$ in one's training sample to compute

$$\boldsymbol{S}_b = \sum_{k=1}^{K} N_k(\bar{\mathbf{x}}_k - \bar{\mathbf{x}})(\bar{\mathbf{x}}_k - \bar{\mathbf{x}})^T$$

$$\boldsymbol{S}_w = \sum_{k=1}^{K}\sum_{y_i \in k}(\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_i - \bar{\mathbf{x}}_k)^T$$

and fix the dimension $J \in \{1, \ldots, K-1\}$, which you want to project your *p*-dimensional observations $\{\mathbf{x}_i; i = 1,2, \ldots, N\}$.

**Step 2**: Find that projection matrix $\widehat{\boldsymbol{B}} \in R^{p \times J}$ that maximizes

$$\lambda = tr\left\{\left(\widehat{\boldsymbol{B}}^T \boldsymbol{S}_w \widehat{\boldsymbol{B}}\right)^{-1}\left(\widehat{\boldsymbol{B}}^T \boldsymbol{S}_b \widehat{\boldsymbol{B}}\right)\right\}$$

applying an appropriate set of normalizing constraints to $\widehat{\boldsymbol{B}}$ that will ensure the uniqueness of this solution. Thus, one need to find those *J* eigenvectors of $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$ that correspond with the *J* largest eigenvalues of $\boldsymbol{S}_w^{-1}\boldsymbol{S}_b$ and arrange them in descending order as the column vectors in the $(p \times J)$-dimensional solution matrix $\widehat{\mathbf{B}}$.

**Step 3**: Use this projection matrix $\widehat{\boldsymbol{B}}$ to compute a mean vector (centroid)

$$\bar{\boldsymbol{d}}_k = \frac{1}{N_k}\sum_{y_i=k} \widehat{\boldsymbol{B}}^T \mathbf{x}_i \qquad \forall\, k = 1,2, \ldots, J$$

in this new *J*-dimensional subspace for all those observations in the training sample that occur in each of the *K* different classes of our classification problem.

**Step 4**: Now use this projection matrix $\widehat{\boldsymbol{B}}$ to map a new observation $\mathbf{x}_i^{new} \in R^p$ into this *J*-dimensional subspace using

$$\boldsymbol{d}(\mathbf{x}_i^{new}) = \widehat{\boldsymbol{B}}^T \mathbf{x}_i^{new}$$

**Step 5**: Assign $\mathbf{x}_i^{new}$ to that class whose centroid $\bar{\boldsymbol{d}}_k$ is 'closest' to the projected value $\widehat{\boldsymbol{B}}^{\mathrm{T}}\mathbf{x}_i^{new}$ of this observation. This means that one assigns this new observation $\mathbf{x}_i^{new}$ to that class $k$ for which the following distance measure is a minimum:

$$d_k(\mathbf{x}_i^{new}) = \left\|\boldsymbol{d}(\mathbf{x}_i^{new}) - \bar{\boldsymbol{d}}_k\right\|^2 \qquad \forall k = 1,2, \ldots, J \tag{3.67}$$

### 3.5 An Optimal scoring approach

Our third approach, developed by Hastie *et al*. (1994), will attempt to make use of a regression based argument, known as *optimal scoring* to recast LDA as a linear regression problem. We will show how this method produces a set of discriminant functions that are proportional to Fisher's (1936) discriminant functions coefficients.

The benefit of using this approach is that it allows one to include non-parametric regression methods in the model, which may lead to the creation of a better classifier.

### 3.5.1 The procedure

Classification can also be viewed as being a problem of prediction where we have a set of characteristics $\mathbf{x}'$ that we want to use to predict the outcome of an associated but discrete valued random variable $y$ that assigns a class label to that particular observation. Let,

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1' \\ \mathbf{x}_2' \\ \vdots \\ \mathbf{x}_N' \end{pmatrix} \tag{3.68}$$

contain as row entries the observed outcomes of the predictor variables $\{\mathbf{x}_i'\colon\ i = 1,2,\dots,N\}$ that one has collected for one's training sample. Given that $y_i$ records the class to which the observation $\mathbf{x}_i'$ belongs, let us now create a $N \times p$ class indicator matrix $\mathbf{Y}$ for all the observations in the training sample that sets $y_{ij} = 1$ if the $i^{th}$ observation $\mathbf{x}_i'$ lies in class $j$, i.e.

$$y_{ij} = 1_{\{y_i = 1\}}$$

where $1_{\{A\}}$ denotes an indicator function for the set A. To illustrate this coding concept more clearly, consider a $K = 3$ class problem with the following class based outcomes being recorded in one's training sample

$$y_1 = 1, y_2 = 3, y_3 = 2, \dots \dots y_{N-1} = 3, y_N = 1$$

This training sample would then have the following class indicator matrix $\mathbf{Y}$ representing their outcomes

$$\mathbf{Y} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ & \vdots & \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \tag{3.69}$$

In principle, $K-1$ columns for the matrix $\mathbf{Y}$ would be sufficient to record all the possible outcomes that one could observe in a $K$ class problem but we will use all $K$ columns in the discussion that follows.

Let $\boldsymbol{\theta}$ represent a $K$-dimensional vector that we will be using to map each row of $\mathbf{Y}$ onto the real number line by making use of the following mapping$-\mathbf{Y}\boldsymbol{\theta}$. Due to the nature of the indicator matrix $\mathbf{Y}$, if the $i^{th}$ observation $\mathbf{x}_i'$ in $\mathbf{X}$ belongs to class $k$ then the $i^{th}$ component of the vector $\mathbf{Y}\boldsymbol{\theta}$ will be using the $k^{th}$ component of $\boldsymbol{\theta}$ as an optimal score for that class.

Similarly, let $\boldsymbol{\beta}$ represent a $p$-dimensional vector that maps each row of our learning sample based outcomes $\mathbf{X}$ onto the real number line by making use of the following mapping$-\mathbf{X}\boldsymbol{\beta}$.

With this notation in hand, the method of optimal scoring attempts to assign a set of values to $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ that will minimize the following *average squared residual* (ASR) :-

$$ASR(\boldsymbol{\theta},\boldsymbol{\beta}) = \frac{1}{N}\|\mathbf{Y}\boldsymbol{\theta} - \mathbf{X}\boldsymbol{\beta}\|^2 \equiv \frac{1}{N}(\mathbf{Y}\boldsymbol{\theta} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{Y}\boldsymbol{\theta} - \mathbf{X}\boldsymbol{\beta}) \tag{3.70}$$

but with the following normalization constraint

$$\frac{1}{N}\|\mathbf{Y}\boldsymbol{\theta}\|^2 = \frac{1}{N}\boldsymbol{\theta}^T\mathbf{Y}^T\mathbf{Y}\boldsymbol{\theta} = 1 \tag{3.71}$$

being imposed on $\boldsymbol{\theta}$ so that we do not have a trivial solution to the above problem arising. Without the normalization constraint (3.71), $\boldsymbol{\theta} = \mathbf{0}$ and $\boldsymbol{\beta} = \mathbf{0}$ would minimize ASR($\boldsymbol{\theta}$ , $\boldsymbol{\beta}$).

Instead of using a pair of vectors $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ to map each row of $\mathbf{Y}$ and $\mathbf{X}$ onto the real number line, one could consider an extension of the above scoring algorithm where a collection of vector pairs,

$$\{(\boldsymbol{\theta}_k, \boldsymbol{\beta}_k);\ k = 1,2,\dots,J = \min(p, K-1)\} \tag{3.72}$$

are used as columns of a matrix $\boldsymbol{\Theta} \in R^{K\times J}$ and $\mathbf{B} \in R^{p\times J}$ respectively, to map each row of $\mathbf{Y}$ and $\mathbf{X}$ into a *J*-dimensional space with the entries in $\boldsymbol{\Theta}$ and $\mathbf{B}$ being chosen so as to minimize:

$$ASR(\boldsymbol{\Theta}, \mathbf{B}) = \frac{1}{N}\|\mathbf{Y}\boldsymbol{\Theta} - \mathbf{X}\mathbf{B}\|^2 \equiv \quad \frac{1}{N} tr[(\mathbf{Y}\boldsymbol{\Theta} - \mathbf{X}\mathbf{B})^T(\mathbf{Y}\boldsymbol{\Theta} - \mathbf{X}\mathbf{B})] \qquad (3.73)$$

To prevent a trivial solution $\boldsymbol{\Theta} = \mathbf{O}$ and $\mathbf{B} = \mathbf{O}$ from occurring, the following set of normalization constraints will have to be added to this minimization problem:

$$\frac{1}{N}\boldsymbol{\Theta}^T\mathbf{Y}^T\mathbf{Y}\,\boldsymbol{\Theta} = \boldsymbol{I}_J \qquad (3.74)$$

Keeping $\boldsymbol{\Theta}$ fixed at a known set of values, minimizing the resulting ASR with respect to $\boldsymbol{B}$ produces a multivariate regression problem where we want to regress $\mathbf{Y}\boldsymbol{\Theta}$ on $\mathbf{X}$. Providing $\mathbf{X}$ is of full rank,

$$\widehat{\mathbf{B}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}\boldsymbol{\Theta} \qquad (3.75)$$

will minimize this ASR. Substituting $\widehat{\mathbf{B}}$ back into (3.73),

$$\Longrightarrow ASR\left(\boldsymbol{\Theta}, \widehat{\mathbf{B}}\right) = \frac{1}{\mathrm{N}}\|\mathbf{Y}\boldsymbol{\Theta} - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}\boldsymbol{\Theta}\|^2$$

$$= \frac{1}{\mathrm{N}}\|(\mathbf{I} - \mathbf{H}_\mathbf{x})\mathbf{Y}\boldsymbol{\Theta}\|^2 \qquad \{\text{where } \mathbf{H}_\mathbf{x} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\}$$

$$= \frac{1}{\mathrm{N}} tr[\boldsymbol{\Theta}^T\mathbf{Y}^T(\mathbf{I} - \mathbf{H}_\mathbf{x})\mathbf{Y}\boldsymbol{\Theta}]$$

$$= \frac{1}{\mathrm{N}} tr\left[[\boldsymbol{\Theta}^T\mathbf{Y}^T\mathbf{Y}\boldsymbol{\Theta}] - [\boldsymbol{\Theta}^T\mathbf{Y}^T\mathbf{H}_\mathbf{x}\mathbf{Y}\boldsymbol{\Theta}]\right]$$

$$= \frac{1}{N} tr[\boldsymbol{\Theta}^T\mathbf{Y}^T\mathbf{Y}\boldsymbol{\Theta} - \boldsymbol{\Theta}^T\mathbf{Y}^T\mathbf{H}_\mathbf{x}\mathbf{Y}\boldsymbol{\Theta}]$$

$$= J - \frac{1}{N} tr(\boldsymbol{\Theta}^T\mathbf{Y}^T\mathbf{H}_\mathbf{x}\mathbf{Y}\boldsymbol{\Theta}) \qquad (3.76)$$

since the restriction in equation (3.74) implies that we have

$$\frac{1}{N}\mathrm{tr}[\boldsymbol{\Theta}^T\mathbf{Y}^T\mathbf{Y}\boldsymbol{\Theta}] = tr\,(\boldsymbol{I}_J) = J$$

Differentiating the following Lagrangian with respect to $\boldsymbol{\Theta}$ and the Lagrange multiplier $\gamma$ viz:

$$\mathrm{L}(\boldsymbol{\Theta},\gamma) = J - \frac{1}{N} tr(\boldsymbol{\Theta}^T \mathbf{Y}^T \mathbf{H_x} \mathbf{Y} \boldsymbol{\Theta}) + \gamma tr\left(\frac{1}{N} \boldsymbol{\Theta}^T \mathbf{Y}^T \mathbf{Y} \boldsymbol{\Theta} - \boldsymbol{I}_J\right) \tag{3.77}$$

produces the following first order condition for minimizing ASR

$$\frac{\partial \mathrm{L}(\boldsymbol{\Theta},\gamma)}{\partial \boldsymbol{\Theta}} = \mathbf{O}$$

$$\Longrightarrow -\frac{2}{N} \mathbf{Y}^T \mathbf{H_x} \mathbf{Y} \boldsymbol{\Theta} + \frac{2}{N} \gamma \mathbf{Y}^T \mathbf{Y}\, \boldsymbol{\Theta} = \mathbf{O} \quad \left\{\text{since;} \frac{\partial tr(\boldsymbol{\Theta}^T \mathbf{A} \boldsymbol{\Theta})}{\partial \boldsymbol{\Theta}} = (\boldsymbol{A} + \mathbf{A}^T)\boldsymbol{\Theta}\right\}$$

$$\Longrightarrow \mathbf{Y}^T \mathbf{H_x} \mathbf{Y} \boldsymbol{\Theta} = \gamma \mathbf{Y}^T \mathbf{Y}\, \boldsymbol{\Theta}$$

$$\Longrightarrow \mathbf{Y}^T \widehat{\mathbf{Y}} \boldsymbol{\Theta} = \gamma \mathbf{Y}^T \mathbf{Y}\, \boldsymbol{\Theta} \qquad \{\text{since; } \widehat{\mathbf{Y}} = \mathbf{H_x} \boldsymbol{Y}\} \tag{3.78}$$

Furthermore, differentiating the Lagrangian (3.77) with respect to the Lagrange multiplier $\gamma$ produces

$$\frac{\partial \mathrm{L}(\boldsymbol{\Theta},\gamma)}{\partial \gamma} = \mathbf{O} \Longrightarrow \frac{1}{N} \boldsymbol{\Theta}^T \mathbf{Y}^T \mathbf{Y}\, \boldsymbol{\Theta} = \boldsymbol{I}_J$$

where in the context of our discussion $\frac{1}{N} \mathbf{Y}^T \mathbf{Y}$ is a diagonal matrix whose $k^{th}$ diagonal element equals $\frac{N_k}{N}$. From expression (3.78), it can be observed that minimizing $ASR\left(\boldsymbol{\Theta}, \widehat{\boldsymbol{B}}\right)$ with respect to $\boldsymbol{\Theta}$ amounts to finding those $J$ eigenvectors of $\mathbf{Y}^T \widehat{\mathbf{Y}}$ that correspond with the $J$ largest eigenvalues of $\mathbf{Y}^T \widehat{\mathbf{Y}}$ and arranging them in descending order as the column vectors in the $K \times J$ matrix $\boldsymbol{\Theta}$.

It is important to note that the matrix $\boldsymbol{\Theta}$ in (3.75) that was initially fixed at a known set of values will need to be replaced with the matrix containing those $J$ eigenvectors of $\mathbf{Y}^T \widehat{\mathbf{Y}}$ that correspond with the $J$ largest eigenvalues of $\mathbf{Y}^T \widehat{\mathbf{Y}}$.

The above optimal scoring routine is summarized in Algorithm 3.1 on the following page.

Algorithm 3.1: Optimal scoring routine for LDA

1. **Initialize:** Create the $N \times K$ indicator response matrix $\mathbf{Y}$.

2. **Run a multivariate regression:** Regress $\mathbf{Y}$ on $\mathbf{X}$ producing a $p \times K$ dimensional matrix, $\widehat{\boldsymbol{B}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ and thus an $N \times K$ dimensional matrix of fitted responses:

$$\widehat{\mathbf{Y}} = \mathbf{X}\widehat{\boldsymbol{B}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{Y} = \mathbf{H}_{\mathbf{x}}\boldsymbol{Y}$$

3. **Optimal scores:** Solve $\mathbf{Y}^T\widehat{\mathbf{Y}}\boldsymbol{\Theta} = \gamma\mathbf{Y}^T\mathbf{Y}\,\boldsymbol{\Theta}$ for $\boldsymbol{\Theta}$ subject to the normalizing condition $\frac{1}{N}\boldsymbol{\Theta}^T\mathbf{Y}^T\mathbf{Y}\,\boldsymbol{\Theta} = \boldsymbol{I}_J$. In other words, find those $J$ eigenvectors of $\mathbf{Y}^T\widehat{\mathbf{Y}}$ that correspond with the $J$ largest eigenvalues of $\mathbf{Y}^T\widehat{\mathbf{Y}}$ and arrange them in descending order as the column vectors in the $(K \times J)$-dimensional solution matrix $\widehat{\boldsymbol{\Theta}}$.

4. Perform a multivariate regression of $\mathbf{Y}\widehat{\boldsymbol{\Theta}}$ on $\mathbf{X}$:
   Since $\mathbf{Y} = \mathbf{XB} \Longrightarrow \mathbf{Y}\widehat{\boldsymbol{\Theta}} = \mathbf{XB}\widehat{\boldsymbol{\Theta}}$, there is no need to re-fit a regression of $\mathbf{Y}\widehat{\boldsymbol{\Theta}}$ on $\mathbf{X}$. One can simply update the estimate $\widehat{\mathbf{B}}$ obtained in step 2 to the $p \times J$ matrix:

$$\widehat{\mathbf{B}}_{OS} = \widehat{\mathbf{B}}\widehat{\boldsymbol{\Theta}} \tag{3.79}$$

   Therefore, the optimally scaled vector containing $J \leq K - 1$ regression (or canonical discriminant) functions is given by:

$$\boldsymbol{d}(\mathbf{x}_i) = \widehat{\mathbf{B}}_{OS}^T\mathbf{x}_i \tag{3.80}$$

   where, $\mathbf{x}_i = \left(x_{1i}, x_{2i}, \ldots, x_{pi}\right)^T$ are the arguments of the p-dimensional predictor variables.

### 3.5.2 Using the optimal scoring routine for classifying a new observation

Given a new observation $\mathbf{x}_i^{new} \in R^p$, the assignment of this observation to a particular class $k \in \{1,2,\ldots,K\}$ can be done by making use of (3.80) to project this new observation into this $J$-dimensional subspace, viz:

$$\boldsymbol{d}(\mathbf{x}_i^{new}) = \widehat{\mathbf{B}}_{\mathrm{OS}}^T \mathbf{x}_i^{new} \tag{3.81}$$

and to compute a mean vector (centroid),

$$\bar{\boldsymbol{d}}_k = \frac{1}{N_k} \sum_{y_i=k} \widehat{\mathbf{B}}_{\mathrm{OS}}^T \mathbf{x}_i \qquad \forall k = 1,2 \ldots, J \tag{3.82}$$

in this new $J$-dimensional subspace for all those observations in the training sample that occur in each of the $K$ different classes of our classification problem. One would then assign this new observation, $\mathbf{x}_i^{new}$ to that class $k$ for which the following distance measure is a minimum:

$$d_k(\mathbf{x}_i^{new}) = \left\| \boldsymbol{d}(\mathbf{x}_i^{new}) - \bar{\boldsymbol{d}}_k \right\|^2 \qquad \forall k = 1,2, \ldots, J \tag{3.83}$$

### 3.5.3 Proof of the equivalence between Fisher's and the optimal scoring approach

Fisher's LDA seeks to find the vector $\boldsymbol{\beta}$ such that:

$$\boldsymbol{\beta} = arg\, \underset{\boldsymbol{\beta}}{max} (\boldsymbol{\beta}^T \boldsymbol{S}_b \boldsymbol{\beta}) \text{ subject to } \quad \boldsymbol{\beta}^T \boldsymbol{S}_w \boldsymbol{\beta} = 1 \tag{3.84}$$

whilst optimal scoring seeks to find a pair of vectors $(\boldsymbol{\theta}, \boldsymbol{\beta})$such that:

$$\boldsymbol{\theta}, \boldsymbol{\beta} = arg\, \underset{\boldsymbol{\theta},\boldsymbol{\beta}}{min} \left( \frac{1}{N} \|\boldsymbol{Y\theta} - \boldsymbol{X\beta}\|^2 \right) \text{subject to} \frac{1}{N} \boldsymbol{\theta}^T \boldsymbol{Y}^T \boldsymbol{Y\theta} = 1 \tag{3.85}$$

Differentiating the following Lagrangian with respect to $\boldsymbol{\theta}$ and the Lagrange multiplier φ viz:

$$L(\boldsymbol{\theta}, \boldsymbol{\beta}, \varphi) = \frac{1}{N} (\boldsymbol{Y\theta} - \boldsymbol{X\beta})^T (\boldsymbol{Y\theta} - \boldsymbol{X\beta}) - \varphi \left( \frac{1}{N} \boldsymbol{\theta}^T \boldsymbol{Y}^T \boldsymbol{Y\theta} - 1 \right)$$

$$= \frac{1}{N} [\boldsymbol{\theta}^T \boldsymbol{Y}^T \boldsymbol{Y\theta} - 2\boldsymbol{\theta}^T \boldsymbol{Y}^T \boldsymbol{X\beta} + \boldsymbol{\beta}^T \boldsymbol{X}^T \boldsymbol{X\beta}] - \varphi \left( \frac{1}{N} \boldsymbol{\theta}^T \boldsymbol{Y}^T \boldsymbol{Y\theta} - 1 \right) \tag{3.86}$$

produces the following first order condition for minimizing (3.85)

$$\widehat{\boldsymbol{\theta}} = \frac{1}{\alpha} (\boldsymbol{Y}^T \boldsymbol{Y})^{-1} \boldsymbol{Y}^T \boldsymbol{X\beta} \quad \text{with } \alpha = (1 - \varphi) \in \mathbb{R} \tag{3.87}$$

On substituting $\widehat{\boldsymbol{\theta}}$ for $\boldsymbol{\theta}$ in the optimization problem (3.85), we get the partially optimized criterion:

$$\boldsymbol{\beta} = arg\,\min_{\boldsymbol{\beta}} \left( \frac{1}{N} \left\| \frac{1}{\alpha} \boldsymbol{Y} (\boldsymbol{Y^T Y})^{-1} \boldsymbol{Y}^T \boldsymbol{X\beta} - \boldsymbol{X\beta} \right\|^2 \right)$$

$$= arg\,\min_{\boldsymbol{\beta}} \left( \frac{1}{N} \left( \frac{1}{\alpha} \boldsymbol{Y} (\boldsymbol{Y}^T \boldsymbol{Y})^{-1} \boldsymbol{Y}^T \boldsymbol{X\beta} - \boldsymbol{X\beta} \right)^T \left( \frac{1}{\alpha} \boldsymbol{Y} (\boldsymbol{Y^T Y})^{-1} \boldsymbol{Y}^T \boldsymbol{X\beta} - \boldsymbol{X\beta} \right) \right)$$

$$= arg\,\min_{\boldsymbol{\beta}} \left( \frac{1}{\alpha^2} \boldsymbol{\beta}^T \boldsymbol{S}_b \boldsymbol{\beta} - \frac{2}{\alpha} \boldsymbol{\beta}^T \boldsymbol{S_b \beta} + \boldsymbol{\beta}^T \boldsymbol{S}_T \boldsymbol{\beta} \right) \tag{3.88}$$

where, $\boldsymbol{S}_b = \frac{1}{N} \boldsymbol{X}^T \boldsymbol{Y} (\boldsymbol{Y}^T \boldsymbol{Y})^{-\mathbf{1}} \boldsymbol{Y}^T \boldsymbol{X}$ is the between-class covariance matrix and $\boldsymbol{S}_T = \frac{1}{N} \boldsymbol{X}^T \boldsymbol{X}$ is the total covariance matrix. Since the total-class covariance matrix ($\boldsymbol{S}_T$) is the sum of the within-class covariance matrix ($\boldsymbol{S}_w$) and between-class covariance matrix ($\boldsymbol{S}_b$),

$$\boldsymbol{S}_T = \boldsymbol{S}_w + \boldsymbol{S}_b$$

equation (3.88) becomes

$$\boldsymbol{\beta} = arg\,\min_{\boldsymbol{\beta}} \left( \frac{1}{\alpha^2} \boldsymbol{\beta}^T \boldsymbol{S}_b \boldsymbol{\beta} - \frac{2}{\alpha} \boldsymbol{\beta}^T \boldsymbol{S}_b \boldsymbol{\beta} + \boldsymbol{\beta}^T \boldsymbol{S}_b \boldsymbol{\beta} + \boldsymbol{\beta}^T \boldsymbol{S}_w \boldsymbol{\beta} \right)$$

$$= arg\,\min_{\boldsymbol{\beta}} \left( \left( \frac{1}{\alpha^2} - \frac{2}{\alpha} + 1 \right) \boldsymbol{\beta}^T \boldsymbol{S}_b \boldsymbol{\beta} + \boldsymbol{\beta}^T \boldsymbol{S}_w \boldsymbol{\beta} \right)$$

$$= arg\,\min_{\boldsymbol{\beta}} \left( \left( \frac{\alpha - 1}{\alpha} \right)^2 \boldsymbol{\beta}^T \boldsymbol{S}_b \boldsymbol{\beta} + \boldsymbol{\beta}^T \boldsymbol{S}_w \boldsymbol{\beta} \right) \tag{3.89}$$

For notational convenience, let:

$$\widetilde{\boldsymbol{\beta}} = \boldsymbol{S}_w^{\frac{1}{2}} \boldsymbol{\beta} \implies \boldsymbol{\beta} = \boldsymbol{S}_w^{-\frac{1}{2}} \widetilde{\boldsymbol{\beta}} \tag{3.90}$$

$$\tilde{\boldsymbol{S}}_b = \boldsymbol{S}_w^{\frac{1}{2}} \boldsymbol{S}_b \boldsymbol{S}_w^{\frac{1}{2}} \implies \boldsymbol{S}_b = \boldsymbol{S}_w^{-\frac{1}{2}} \tilde{\boldsymbol{S}}_b \boldsymbol{S}_w^{-\frac{1}{2}} \tag{3.91}$$

The optimal scoring vector $\boldsymbol{\beta}$ in equation (3.89) can then be found as a solution to the following problem:

$$\widetilde{\boldsymbol{\beta}} = arg\,\min_{\widetilde{\boldsymbol{\beta}}}\left(\left(\frac{\alpha-1}{\alpha}\right)^2 \widetilde{\boldsymbol{\beta}}^T\tilde{\boldsymbol{S}}_b\widetilde{\boldsymbol{\beta}}+\widetilde{\boldsymbol{\beta}}^T\widetilde{\boldsymbol{\beta}}\right) \tag{3.92}$$

The minimizing $\widetilde{\boldsymbol{\beta}}$ for (3.92) can be found by differentiating its objective function

$$\left(\frac{\alpha-1}{\alpha}\right)^2 \widetilde{\boldsymbol{\beta}}^T\tilde{\boldsymbol{S}}_b\widetilde{\boldsymbol{\beta}}+\widetilde{\boldsymbol{\beta}}^T\widetilde{\boldsymbol{\beta}}$$

with respect to $\widetilde{\boldsymbol{\beta}}$ and equating the derivative to the zero vector to get:

$$2\left(\frac{\alpha-1}{\alpha}\right)^2 \tilde{\boldsymbol{S}}_b\widetilde{\boldsymbol{\beta}} + 2\widetilde{\boldsymbol{\beta}} = \mathbf{0}$$

$$\Longrightarrow \tilde{\boldsymbol{S}}_b\widetilde{\boldsymbol{\beta}} = \omega\widetilde{\boldsymbol{\beta}} \qquad \text{with} \qquad \omega = -\left(\frac{\alpha}{\alpha-1}\right)^2 \in \mathbb{R} \tag{3.93}$$

Therefore, $\omega$ is the eigenvalue of $\tilde{\boldsymbol{S}}_b$ and $\widetilde{\boldsymbol{\beta}}$ is the eigenvector of $\tilde{\boldsymbol{S}}_b$. Substituting $\tilde{\boldsymbol{S}}_b\widetilde{\boldsymbol{\beta}} = \omega\widetilde{\boldsymbol{\beta}}$ into the objective function in equation (3.92) we get:

$$\widetilde{\boldsymbol{\beta}} = arg\,\min_{\widetilde{\boldsymbol{\beta}}}\left(\left(\frac{\alpha-1}{\alpha}\right)^2 \omega\widetilde{\boldsymbol{\beta}}^T\widetilde{\boldsymbol{\beta}}+\widetilde{\boldsymbol{\beta}}^T\widetilde{\boldsymbol{\beta}}\right) = arg\,\min_{\widetilde{\boldsymbol{\beta}}}\left(-\frac{1}{\omega}(\omega+1)\widetilde{\boldsymbol{\beta}}^T\widetilde{\boldsymbol{\beta}}\right) \tag{3.94}$$

which is minimized when $\omega$ is large. Therefore, $\widetilde{\boldsymbol{\beta}}$ is the first eigenvector of $\tilde{\boldsymbol{S}}_b$.

Making use of the notation that we have introduced in (3.90) and (3.91), Fisher's LDA vector $\boldsymbol{\beta}$ in (3.84) is obtained by solving:

$$\widetilde{\boldsymbol{\beta}} = arg\,\max_{\boldsymbol{\beta}}\left(\widetilde{\boldsymbol{\beta}}^T\tilde{\boldsymbol{S}}_b\widetilde{\boldsymbol{\beta}}\right) \quad subject\ to \quad \widetilde{\boldsymbol{\beta}}^T\widetilde{\boldsymbol{\beta}} = 1 \tag{3.95}$$

The maximizing $\widetilde{\boldsymbol{\beta}}$ for (3.95) can be found by differentiating the following Lagrangian with respect to $\widetilde{\boldsymbol{\beta}}$:

$$L(\widetilde{\boldsymbol{\beta}}, \gamma) = \widetilde{\boldsymbol{\beta}}^T\tilde{\boldsymbol{S}}_b\widetilde{\boldsymbol{\beta}} - \gamma\left(\widetilde{\boldsymbol{\beta}}^T\widetilde{\boldsymbol{\beta}} - 1\right) \tag{3.96}$$

where $\gamma \in \mathbb{R}$ is the Lagrange multiplier, which on equating the first derivative to the zero vector gives the eigenvalue-eigenvector equation:

$$\tilde{\boldsymbol{S}}_b\widetilde{\boldsymbol{\beta}} = \gamma\widetilde{\boldsymbol{\beta}} \tag{3.97}$$

Substituting $\tilde{\boldsymbol{S}}_b\widetilde{\boldsymbol{\beta}} = \gamma\widetilde{\boldsymbol{\beta}}$ into the objective function in (3.95), we get:

$$\widetilde{\boldsymbol{\beta}} = arg\,\max_{\widetilde{\boldsymbol{\beta}}}\left(\gamma \widetilde{\boldsymbol{\beta}}^T \widetilde{\boldsymbol{\beta}}\right) \quad subject\ to \quad \widetilde{\boldsymbol{\beta}}^T \widetilde{\boldsymbol{\beta}} = 1 \tag{3.98}$$

which is maximized when the eigenvalue $\gamma$ is large as well. Therefore, the associated $\widetilde{\boldsymbol{\beta}}$ is the first eigenvector of $\tilde{\boldsymbol{S}}_b$.

To this end, the optimal $\boldsymbol{\beta} = \boldsymbol{S}_w^{-\frac{1}{2}} \widetilde{\boldsymbol{\beta}}$ for both Fisher's LDA and the optimal scoring approach to LDA is the first eigenvector of $\tilde{\boldsymbol{S}}_b$. Thus,

$$\Longrightarrow \boldsymbol{\beta}_{OS} \propto \boldsymbol{\beta}_{Fisher} \tag{3.99}$$

where, $\boldsymbol{\beta}_{OS}$ and $\boldsymbol{\beta}_{Fisher}$ is being used to denote the optimal scoring and Fisher's LDA based canonical discriminant coefficients.

### 3.6 Judging variable importance

Variable importance can be assessed by considering the magnitude of the canonical discriminant function coefficients. However, the value of these discriminant coefficients can be misleading if the predictor variables have different units of measurement. To measure variable importance, one needs first to standardize the predictor variables in one's dataset. This standardization can be achieved by computing the following z-scores for all the observations in the training sample:

$$z_{ij} = \frac{\left(x_{ij} - \mu_j\right)}{\sigma_j} \qquad \forall\, i = 1,2,\dots,N \text{ and } j = 1,2,\dots,p \tag{3.100}$$

where $x_{ij}$ denotes an observed value for the $j^{th}$ component of the $i^{th}$ vector $\mathbf{x}_i$ that one observes in the learning sample, $\mu_j$ and $\sigma_j$ denote the mean and standard deviation of observations in the training sample belonging to the predictor variable $x_j$, respectively.

### 3.7 Conclusion

This chapter brought to the fore a number of observations amongst them, the following: In the $K = 2$ class case, the Bayesian and Fisher's approach to LDA produce the same classifier when we assume equal prior probabilities. However, the Bayesian approach also provides a convenient way of incorporating misclassification costs into the LDA model. On the other hand, since Fisher's approach is able to transform the observations from a higher $p$-dimensional space into a much lower $(K-1)$-dimensional space, this

method becomes particularly more useful when dealing with a high dimensional set of data ($p \gg K$). The optimal scoring approach provides one with another way of producing Fisher's canonical discriminant functions. The main advantage of using this optimal scoring technique is that it allows one to replace the linear regression functions with a class of far more flexible non-parametric regression functions. This idea will be explored in depth in the next chapter.

# CHAPTER 4

# 4. Quadratic, Flexible and Mixture Discriminant Analysis

## 4.1 Introduction

The preoccupation of this chapter is to explore three techniques that have been designed to handle some of the limitations of LDA. The first approach is to relax the assumption of equal covariance matrices in the Bayesian classifier we have developed, which results in the creation of quadratic decision boundaries (Geisser, 1964). This modification is known as quadratic discriminant analysis (QDA). Hastie *et al.* (1994) proposed that a class of even more flexible models could be created by replacing the linear regression functions in the optimal scoring approach to LDA with a set of non-parametric or semi-parametric regression functions. This approach is known as flexible discriminant analysis (FDA). For multi-modal data, Hastie & Tibshirani (1996) have developed another modelling approach called mixture discriminant analysis (MDA) where each class is modelled as a Gaussian mixture of two or more subgroups within that class. These extensions of LDA will be looked at underneath, together with Friedman's (1991) multivariate adaptive regression splines (MARS) procedure that we will use in the FDA classifier in place linear regression functions.

## 4.2 Quadratic discriminant analysis

Quadratic discriminant analysis (QDA) follows directly from the Bayesian approach to LDA discussed in section (3.2) were one assigns a new observation $\mathbf{x}_i^{new}$ to a class $k \in \{1,2,\dots K\}$ that maximizes

$$\ln[\pi_k f_k(\mathbf{x}_i^{new})] = \text{In}\pi_k - \frac{p}{2}\text{In}2\pi - \frac{1}{2}\text{In}|\boldsymbol{\Sigma}| - \frac{1}{2}(\mathbf{x}_i^{new} - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_i^{new} - \boldsymbol{\mu}_k) \quad (4.1)$$

One can relax the assumption of equal covariance matrices (i.e. $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}$ ) such that a new observation $\mathbf{x}_i^{new}$ is assigned to a class $k \in \{1,2,\dots K\}$ that maximizes

$$\ln[\pi_k f_k(\mathbf{x}_i^{new})] = \text{In}\pi_k - \frac{p}{2}\text{In}2\pi - \frac{1}{2}\text{In}|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\mathbf{x}_i^{new} - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_i^{new} - \boldsymbol{\mu}_k) \quad (4.2)$$

Since $-\frac{p}{2}\text{In}2\pi$ in (4.2) is independent of $k$, one would then assign a new observation $\mathbf{x}_i^{new}$ to that class $k \in \{1,2 \ldots, \text{K}\}$ that maximizes the following discriminant function:

$$d_k(\mathbf{x}_i^{new}) = \text{In}\pi_k - \frac{1}{2}\text{In}|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\mathbf{x}_i^{new} - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{x}_i^{new} - \boldsymbol{\mu}_k) \tag{4.3}$$

where sample based estimates of the parameters $\pi_k$, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are computed from the learning sample using equations (3.5)-(3.7) given in chapter three. Thus, assign a new observation $\mathbf{x}_i^{new}$ to that class $k \in \{1,2 \ldots, \text{K}\}$ that maximizes:

$$d_k(\mathbf{x}_i^{new}) = \text{In}\hat{\pi}_k - \frac{1}{2}\text{In}|\mathbf{S}_k| - \frac{1}{2}(\mathbf{x}_i^{new} - \bar{\mathbf{x}}_k)^T\mathbf{S}_k^{-1}(\mathbf{x}_i^{new} - \bar{\mathbf{x}}_k) \tag{4.4}$$

Figure 4.1 below is an illustration of two-dimensional observations $\mathbf{x}_i \in R^2$ that belong to one of the classes, $y_i = k \in \{1,2\}$. The red line illustrates a quadratic decision boundary (QDA) while the green line illustrates a linear decision boundary (LDA) on the same dataset. A visual inspection shows that QDA fits the data better than LDA in this case.

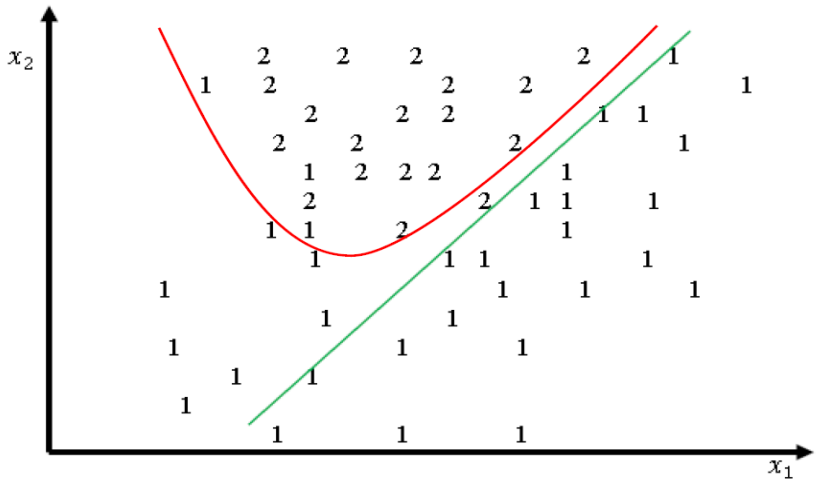


Figure 4.1: An illustration of LDA and QDA decision boundaries for a two-class problem

### 4.3 Flexible discriminant analysis

Flexible discriminant analysis (FDA) is a generalization of the optimal scoring approach to LDA that was developed in section (3.5). It allows one to substitute other appropriate regression procedures in place of linear regression functions (Hastie *et al.*, 1994).

To recap, the optimal scoring approach recasts LDA as a regression problem by using a $K$-dimensional vector $\boldsymbol{\theta}$ to map each row of an $N \times K$ indicator type matrix of outcomes $\mathbf{Y}$ onto the real number line using the following mapping$-\mathbf{Y}\boldsymbol{\theta}$. By initially fixing $\boldsymbol{\theta}$ at a set of known values**,** a linear regression of the derived responses $\mathbf{Y}\boldsymbol{\theta}$ against the $N \times p$ matrix of predictor variables $\mathbf{X}$ produces a $p$-dimensional vector of regression coefficients:

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}\boldsymbol{\theta}$$

The value of $\boldsymbol{\theta}$ is then updated to its optimal value which was shown to be the first eigenvector of $\mathbf{Y}^T\widehat{\mathbf{Y}} = \mathbf{Y}^T\,\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} = \mathbf{Y}^T\mathbf{H}_{\mathbf{x}}\boldsymbol{Y}$**,** where $\mathbf{H}_{\mathbf{x}}$ is a so-called *linear operator* that maps $\mathbf{Y}$ to $\widehat{\mathbf{Y}}$ (i.e. $\widehat{\mathbf{Y}} = \mathbf{H}_{\mathbf{x}}\mathbf{Y}$ ).

The limitation with the aforementioned optimal scoring technique however is that, the relationship between the optimally derived responses $\mathbf{Y}\boldsymbol{\theta}$ and the p-dimensional predictor variables $\mathbf{x}_i$ contained in the matrix $\mathbf{X}$ may not be linear in nature. To deal with this limitation, we will repeat the optimal scoring approach to LDA, only this time using a non-parametric regression technique in place of linear regression. The regression technique we have in mind is Friedman (1991)'s Multivariate Adaptive Regression Splines (MARS). The MARS procedure is promising because it does not assume a linear relationship between the covariates $\mathbf{x}_i$ and the response variables $y_i$, but instead approximates the relationship entirely from the learning sample observations.

#### 4.3.1 The MARS regression procedure

As a motivation for the MARS regression procedure, consider Figure 4.2, which shows a simple linear regression model fitted to a dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ on the left hand side and a MARS model fitted to the same dataset on the right hand side. A visual inspection of Figure 4.2 suggests that the MARS procedure provides a better fit to this dataset.

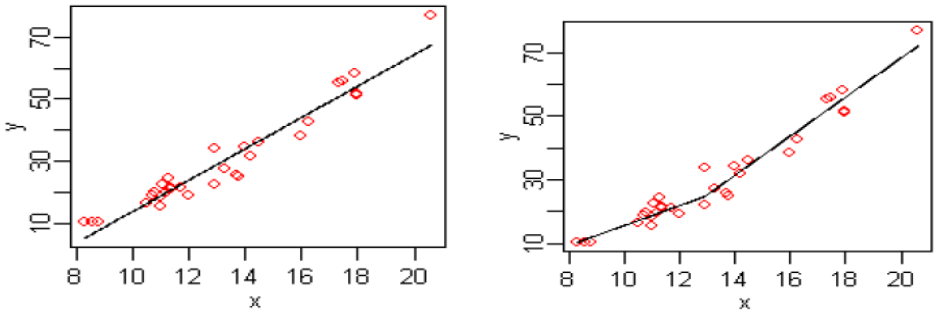


Figure 4.2: A simple linear regression model (left) and a MARS model (right)

A MARS model typically takes on the form:

$$\hat{f}(\mathbf{x}_i) = \hat{\beta}_0 + \sum_{q=1}^{Q} \hat{\beta}_q M_q(\mathbf{x}_i) \tag{4.5}$$

where, $\{M_q(\mathbf{x}_i): q = 1,2,\dots,Q\}$ denotes a set of basis functions that will be generated from the following pool of paired hinge functions (which we shall call conjugate pairs) by taking cross products in these functions:

$$C = \left\{(x_j - x_{ij})_+, (x_{ij} - x_j)_+\right\} \qquad \forall\, j = 1,2,\dots,p \text{ and } i = 1,2,\dots,N$$

where $x_{ij}$ denotes an observed value for the $j^{th}$ component of the $i^{th}$ vector $\mathbf{x}_i$ that one observes in our learning sample and $x_j$ denotes an argument for the hinge function that has been given in the set *C* above.

The coefficients that are given in (4.5) are obtained by minimizing the following residual sum-of-squares (RSS) that one can associate the MARS model formulation with:

$$RSS = \sum_{i=1}^{N} \left(y_i - \hat{f}(\mathbf{x}_i)\right)^2 = \sum_{i=1}^{N} \left(y_i - \sum_{q=1}^{Q} \hat{\beta}_q M_q(\mathbf{x}_i)\right)^2 \tag{4.6}$$

The notation $(\ \ )_+$ on the hinge function considers the positive difference obtained from $(x_j - x_{ij})$, viz:

$$(x_j - x_{ij})_+ = \begin{cases} x_j - x_{ij} & \text{if } x_j > x_{ij} \\ 0, & \text{otherwise} \end{cases} \quad ; \quad (x_{ij} - x_j)_+ = \begin{cases} x_{ij} - x_j, & \text{if } x_{ij} > x_j \\ 0, & \text{otherwise} \end{cases}$$

For a one-dimensional observation $x$, the conjugate pair $(x-0.5)_+$ and $(0.5-x)_+$ is illustrated in Figure 4.3, where the values $x=0.4$ and $x=0.6$ results in the same basis function value, $M(x)=0.1$. The basis functions have a value of zero at the knot point, $x=0.5$.

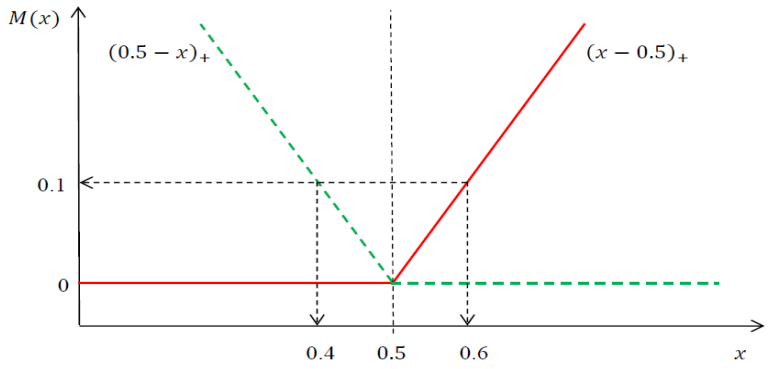


Figure 4.3: Conjugate pair $(x-0.5)_+$ and $(0.5-x)_+$

It is important to note that the hinge functions that make up the basis functions have a value of zero for part of their range. For example, the hinge function $(0.5-x)_+$ in Figure 4.3 is zero when *x* is greater than 0.5. Likewise, the hinge function $(x-0.5)_+$ is zero when *x* is less than 0.5. It is because of this nature of these hinge functions that they can be used to partition the dataset into mutually disjoint regions, each of which can be treated independently.

As an example, the following one-dimensional MARS model

$$\hat{f}(x)=0.2+2(x-0.5)_+-0.5(0.5-x)_+$$

is plotted in Figure 4.4.

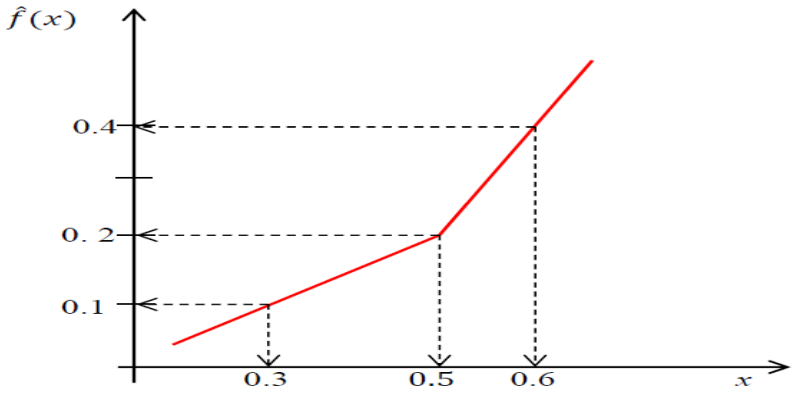


Figure 4.4: Illustration of MARS model

Figure 4.4 shows that the MARS model has partitioned the dataset into two disjoint regions, one defined for values of $x < 0.5$ and one defined for values of $x > 0.5$. Thus, the MARS model in Figure 4.4 can be viewed as dividing the dataset into two mutually disjoint regions, and then fitting a linear regression model in each of two regions.

We can create piecewise non-linear regression functions by multiplying two or more hinge functions together. In particular, the model building process is done using a *forward pass* to add more hinge-pair based cross products to the model until a pre-determined stopping criterion is satisfied. A *backward pass* is then implemented where a pruning process takes place, removing the hinge functions themselves (rather than the hinge function pairs) until only a constant term is remaining. Model subsets from the pruning process are compared using a generalized cross-validation measure and the optimal model selected.

**Step 1: The forward pass**

We will explain the MARS forward pass with the aid of an example. Consider Figure 4.5, which shows the set *C* of all the candidate basis functions (which are the conjugate pairs of hinge functions such as the one in Figure 4.3) on the right hand side and the basis functions that have been selected to be in the model are presented on the left hand side.

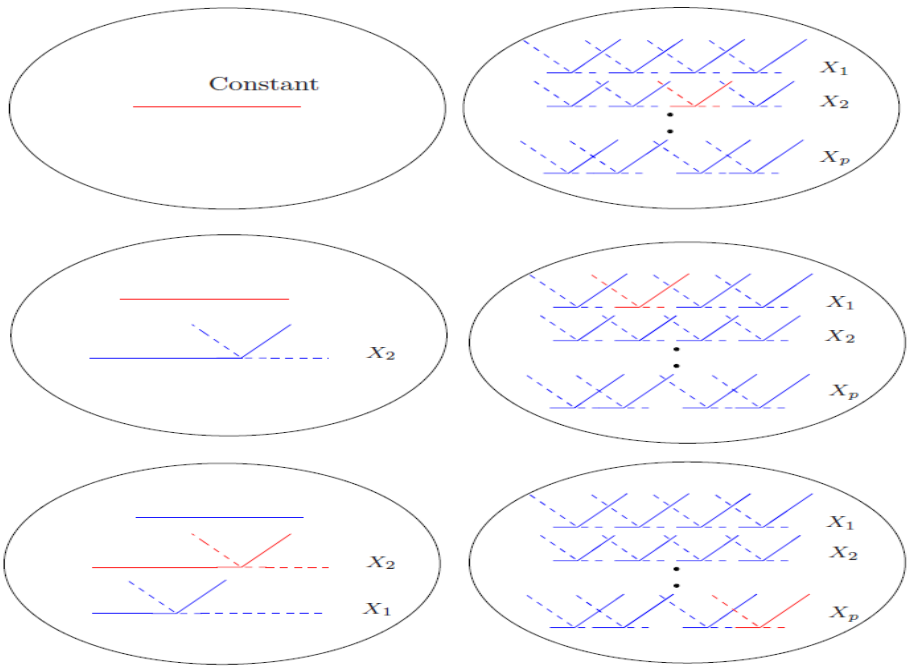


Figure 4.5: Illustration of MARS forward process (Source: Hastie *et al.*, 2009: 323)

We will denote by $x_{ij}$ the observed value for the $j^{th}$ component of the $i^{th}$ vector $\mathbf{x}_i$ in the learning sample, where for the example in Figure 4.5 we have $i = 1,2,3,4$ observations and $j = 1,2,\ldots,p$ components. Thus, the components of the vector $\mathbf{x}_i$ are (dropping the subscript *i* which label the vector) $\mathbf{x} = (x_1, x_2 \ldots, x_p)$.

Starting with a basis function $M_0(\mathbf{x}_i) = 1$, minimizing (4.6) will produce a fitted model structure,

$$\hat{f}(\mathbf{x}_i) = \hat{\beta}_0 \tag{4.7}$$

where $\hat{\beta}_0 = \frac{1}{N}\sum_{i=1}^{N} y_i$, which is the constant shown to be initially in the model in Figure 4.5. Letting $RSS(\hat{\beta}_0)$ denote a residual sum of squares for this fitted model, a conjugate pair from the set *C* is then added to the model producing a new MARS model that is shown below:

$$\hat{f}(\mathbf{x}_i) = \hat{\beta}_0 \ + \hat{\beta}_1\big(x_j - x_{ij}\big)_+ + \hat{\beta}_2\big(x_{ij} - x_j\big)_+ \tag{4.8}$$

Parameter estimates for the model that minimizes (4.6) can then be produced and a resulting residual sum of squares $RSS(\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, x_j, x_{ji})$ can be computed. It is important to note that the above residual sum of squares may change if we were to have chosen another variable $x_k$ and a knot point $x_{ik}$ from the conjugate pair set *C*. Thus, for all the conjugate pairs in *C* that we have available, one can (if the knot points are all different) produce a total of $Np$ residual sum of squares, viz

$$RSS\big(\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2, x_j, x_{ji}\big) \qquad \forall\, j = 1,2,\ldots,p \ \text{ and } \ i = 1,2,\ldots,N \tag{4.9}$$

The conjugate pair generating the smallest value of RSS is then chosen as one's best MARS model for the first step of this forward pass routine.

Our example in Figure 4.5 shows that the following choice of index values $x_2$ and $x_{32}$ has produced the smallest RSS so that our MARS model for the end of the first step assumes the following form

$$\hat{f}(\mathbf{x}_i) = \hat{\beta}_0 \ + \hat{\beta}_1(x_2 - x_{32})_+ + \hat{\beta}_2(x_{32} - x_2)_+ \tag{4.10}$$

Given $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ from a model fitting point of view (4.10) indicates that a predicted value for this observation will now be given by

$$\hat{f}(\mathbf{x}_i) = \hat{\beta}_0 \ + \hat{\beta}_1(x_{i2} - x_{32})_+ + \hat{\beta}_2(x_{32} - x_{i2})_+ \tag{4.11}$$

The second step in this forward pass routine now involves revisiting the set of hinge-pairs that we have available in *C* and adding to (4.10) that hinge-pair function whose addition causes the largest decrease in the RSS of the new model to occur. Our example in Figure 4.5 shows that the following choice of index values $x_1$ and $x_{21}$ has produced the smallest RSS when added to (4.10). Therefore, we consider adding to the model (4.10) a term of the form

$$\beta_q(x_1 - x_{21})_+ M_l(\mathbf{x}_i) + \ \beta_{q+1}(x_{21} - x_1)_+ M_l(\mathbf{x}_i) \tag{4.12}$$

where, $M_l(\mathbf{x}_i) \in \{1, (x_2 - x_{32})_+, (x_{32} - x_2)_+\}$ are basis functions helping form (4.10).

Therefore, the largest possible MARS model from the second step will look something like this:

$$\begin{aligned}\hat{f}(\mathbf{x}_i) = \hat{\beta}_0 \ &+ \hat{\beta}_1(x_2 - x_{32})_+ + \hat{\beta}_2(x_{32} - x_2)_+ + \ \beta_3(x_1 - x_{21})_+ + \ \beta_4(x_{21} - x_1)_+ \\ &+\beta_5(x_1 - x_{21})_+(x_2 - x_{32})_+ + \ \beta_6(x_{21} - x_1)_+(x_2 - x_{32})_+ \\ &+\beta_7(x_1 - x_{21})_+(x_{32} - x_2)_+ + \ \beta_8(x_{21} - x_1)_+(x_{32} - x_2)_+ \end{aligned} \tag{4.13}$$

The model (4.13) reveals that at the end of the second step it is possible that one will have cross products entering into the model depending on what hinge functions we choose from the MARS model (4.10) to create cross products with new hinge functions.

In the model building process, each predictor variable $x_j$ is allowed to appear at most once in a basis function. This has the effect of preventing higher-order powers of the predictor variables from appearing which increase or decrease quickly near boundary values of the domain of the covariates. For example, the following polynomial basis function is not allowed:

$$(x_j - x_{ij})_+(x_j - x_{ij})_+ \quad or \quad (x_j - x_{ij})_+(x_{ij} - x_j)_+ \tag{4.14}$$

In the third step of our example in Figure 4.5, the choice of index values $x_p$ and $x_{4p}$ has produced the smallest RSS when added to the model in the second step. As a result, we now add a term of the form:

$$\hat{\beta}_q(x_p - x_{4p})_+ M_l(\mathbf{x}_i) + \hat{\beta}_{q+1}(x_{4p} - x_p)_+ M_l(\mathbf{x}_i) \quad (4.15)$$

where, $M_l(\mathbf{x}_i) \in \{1, (x_2 - x_{32})_+, (x_{32} - x_2)_+, (x_1 - x_{21})_+, (x_{21} - x_1)_+\}$ are basis functions helping form the MARS model in the second step.

An advantage with the MARS procedure is that the user can set an upper limit (which we shall denote by B) that controls the degree of interactions of the hinge functions allowed. For example, a MARS model of degree $B = 1$ makes (4.5) an additive model because interaction between hinge functions will not be permitted. Using the same idea, a MARS model of degree $B = 2$ forces the hinge functions to interact at most twice with each other. Higher orders of $B$ however makes the model complex and thus difficult to interpret.

The following rules can be used to control when the forward pass should stop:

- Set a maximum number of terms that the MARS model (4.5) must have in the forward pass (including the constant term $\hat{\beta}_0$).
- Set a threshold value $\alpha$ (the default is $\alpha = 0.001$) such that the forward pass stops when adding a term changes $RSS$ by less than $\alpha$.

At the end of the MARS forward pass, we have a large model which probably over fits the data (i.e. provides the best fit to training sample observations but not generalizing new data well). Thus, a backward pass is used to delete insignificant terms.

**Stage two: The backward pass**

The backward pass removes the hinge functions themselves (rather than the conjugate pairs) one by one. This process occurs until we are left with the initial model, $\hat{f}(\mathbf{x}_i) = \hat{\beta}_0$ which is associated with the constant basis function $M_0(\mathbf{x}_i) = 1$. It should be noted that once a single hinge function has been removed, the following generalized cross-validation (GCV) measure is calculated for a model with $\lambda$ hinge functions:

$$GCV(\lambda) = \frac{\sum_{i=1}^{N}\left(y_i - \hat{f}_\lambda(\mathbf{x}_i)\right)^2}{\left(1 - \frac{Eff(\lambda)}{N}\right)^2} \tag{4.16}$$

where, $\hat{f}_\lambda(\mathbf{x}_i)$ denotes a MARS model with $\lambda$ hinge functions such that the numerator of (4.16) is the RSS associated with this model. The term $Eff(\lambda)$ is the effective number of parameters in the MARS model with $\lambda$ hinge functions, which are defined as

$$Eff(\lambda) = r + cT \tag{4.17}$$

where for the model $\hat{f}_\lambda(\mathbf{x}_i)$, *r* is the number of linearly independent basis functions, *c* is a penalty parameter (usually $c = 2\ or\ 3$) and *T* is the number of hinge function knots. Thus, (4.17) means a MARS model 'pays' a penalty of *c* for having additional knots. The model $\hat{f}_\lambda(\mathbf{x}_i)$ that gives the lowest value of GCV ($\lambda$) is chosen to be optimal, viz

$$\hat{f}_\lambda(\mathbf{x}_i) = \hat{\beta}_0 + \sum_{q=1}^{Q} \hat{\beta}_q M_q(\mathbf{x}_i) \tag{4.18}$$

### 4.3.2 Performing FDA using the MARS regression procedure

Mirroring the optimal scoring approach to LDA developed in section (3.5), suppose we have a set of training observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$. As usual, $\mathbf{x}_i = \{x_{1i}, x_{2i}, \ldots, x_{pi}\}$ are p-dimensional predictor variables associated with the class label $y_i = k \in \{1,2, \ldots, K\}$. The following steps can be used to perform FDA using the MARS procedure outlined in the preceding section.

**Step 1:** Create a $N \times K$ class indicator matrix **Y** for all the observations in the training sample that sets $y_{ij} = 1$ if the $i^{th}$ observation $\mathbf{x}_i$ lies in class *j* and otherwise sets $y_{ij} = 0$ (see equation (3.69) for example). Fixing $\boldsymbol{\Theta}$ to be the following diagonal matrix of order *K*,

$$\boldsymbol{\Theta}_0 = \text{diag}\left\{\frac{1}{\sqrt[2]{N_1}}, \frac{1}{\sqrt[2]{N_2}}, \cdots, \frac{1}{\sqrt[2]{N_K}}\right\} \tag{4.19}$$

such that the following normalization constrain is satisfied,

$$\boldsymbol{\Theta}_0^T \frac{1}{N} \mathbf{Y}^T \mathbf{Y} \boldsymbol{\Theta}_0 \;=\; \mathbf{I}_K \tag{4.20}$$

to prevent trivial solutions, let $\boldsymbol{\Theta}_0^* = \mathbf{Y}\boldsymbol{\Theta}_\mathbf{0}$ be the initial $N \times K$ matrix of 'scored' responses.

**Step 2:** Perform a (multi-response) multivariate regression of $\boldsymbol{\Theta}_0^*$ on the $N \times p$ matrix of predictor variables **X** using the MARS technique to get the $N \times K$ matrix of fitted values $\widehat{\boldsymbol{\Theta}}_0^*$. Let $\mathbf{S}(\hat{\lambda})$ be a linear operator that fits the final chosen model such that we have:

$$\widehat{\boldsymbol{\Theta}}_0^* = \mathbf{S}\big(\hat{\lambda}\big)\boldsymbol{\Theta}_0^* \tag{4.21}$$

where $\hat{\lambda}$ represents the estimated optimal size of the MARS model $\hat{f}_\lambda(\mathbf{x}_i)$ selected using the GCV criterion (4.16).

It is important to note that (4.21) means that the same procedure $\mathbf{S}\big(\hat{\lambda}\big)$ is being used to fit $K$ models to each of the $K$ levels of scored responses in $\boldsymbol{\Theta}_0^*$. In particular, for each of the $K$ levels of scored responses in $\boldsymbol{\Theta}_0^*$, $K$ MARS models of the same size $\hat{\lambda}$ are simultaneously fit sharing the same basis functions $\{M_q(\mathbf{x}_i): q = 0,1,\dots,Q\}$ but may have different coefficients $\big\{\hat{\beta}_{qk}: q = 0,1,\dots,Q; k = 1,2,\dots,K\big\}$. Thus, a (multi-response) multivariate regression of the optimal scores matrix $\boldsymbol{\Theta}_0^*$ on the original matrix of predictor variables **X** using the MARS technique produces the following $K$ regression functions:

$$\hat{f}_1(\mathbf{x}_i) = \hat{\beta}_{01} + \sum_{q=1}^{Q} \hat{\beta}_{q1} M_q(\mathbf{x}_i)$$

$$\hat{f}_2(\mathbf{x}_i) = \hat{\beta}_{02} + \sum_{q=1}^{Q} \hat{\beta}_{q2} M_q(\mathbf{x}_i)$$

$$\vdots$$

$$\hat{f}_K(\mathbf{x}_i) = \hat{\beta}_{0K} + \sum_{q=1}^{Q} \hat{\beta}_{qK} M_q(\mathbf{x}_i)$$

Let,

$$\boldsymbol{f}(\mathbf{x}_i) = \left(\hat{f}_1(\mathbf{x}_i) \ldots . \hat{f}_K(\mathbf{x}_i)\right)^T \tag{4.22}$$

be a *K*-dimensional vector of fitted regression functions.

**Step 3**: Generate a $K \times J$ eigenvector matrix $\boldsymbol{\Phi}$ by performing an eigenvalue-based decomposition on the following matrix,

$$\boldsymbol{\Theta}_0^{*T} \widehat{\boldsymbol{\Theta}}_0^* = \boldsymbol{\Theta}_0^{*T} \mathbf{S}(\hat{\lambda})\, \boldsymbol{\Theta}_0^* \tag{4.24}$$

to get the new $K \times J$ matrix of optimal scores as, $\boldsymbol{\Theta} = \boldsymbol{\Theta}_0 \boldsymbol{\Phi}$.

**Step 4:** Update the *K*-vector of regression functions in step 2 using the eigenvector matrix $\boldsymbol{\Phi}$ to get the optimal $J \leq K - 1$ dimensional vector of regression (canonical discriminant) functions as

$$\boldsymbol{f}^*(\mathbf{x}_i) = \boldsymbol{\Phi}^T \boldsymbol{f}(\mathbf{x}_i) \tag{4.25}$$

Because the basis functions $\{M_q(\mathbf{x}_i) : q = 1, \ldots, Q\}$ are treated as fixed once selected ($M_0(\mathbf{x}_i) = 1$), we can replace the $N \times p$ original data matrix

$$\mathbf{X} = \begin{pmatrix} x_{11} & \cdots & x_{p1} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{p1} \end{pmatrix} \tag{4.26}$$

used in the optimal scoring Algorithm 3.1 with an $N \times Q$ data matrix

$$\mathbf{P} = \begin{pmatrix} M_1(\mathbf{x}_1) & \cdots & M_Q(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ M_1(\mathbf{x}_N) & \cdots & M_Q(\mathbf{x}_N) \end{pmatrix} \tag{4.27}$$

containing basis expansions of the original predictor variables $\mathbf{x}_i \in R^p$ (i.e. we have expanded the predictor variables from a p-dimensional space to a Q-dimensional space where $Q > p$).Using the optimal scoring Algorithm 3.1 with $\boldsymbol{P}$ in place of $\boldsymbol{X}$ should in theory produce the following optimal $J \leq K - 1$ dimensional vector of regression functions that are equivalent to the ones in (4.25):

$$\boldsymbol{d}(\mathbf{x}_i) = \widehat{\boldsymbol{B}}_{OS}^T M(\mathbf{x}_i) \tag{4.28}$$

where for all the training sample observations $i = 1.2 \ldots, N$;

$$M(\mathbf{x}_i) = \left(M_1(\mathbf{x}_i), \ldots, M_Q(\mathbf{x}_i)\right)^T \tag{4.29}$$

is a Q-dimensional vector containing arguments of the fixed basis function variables that have been created using the MARS technique and $\widehat{\boldsymbol{B}}_{OS} \in R^{Q\times J}$ is a matrix of regression coefficients. In this regard, we can think of the optimal scoring approach to LDA summarized in Algorithm 3.1 as having the following fixed basis function variables:

$$M(\mathbf{x}_i) = \mathbf{x}_i = \left(x_{i1}, \ldots, x_{ip}\right)^T \quad \forall i = 1,2, \ldots, N \tag{4.30}$$

The FDA algorithm is summarized below:

---

Algorithm 4.1**:** FDA algorithm

---

1. **Initialize:** Create the $N \times K$ indicator response matrix $\mathbf{Y}$ and as an initial value, set $\boldsymbol{\Theta} = \boldsymbol{\Theta}_0$ that satisfy the restriction $\boldsymbol{\Theta}_0^T \frac{1}{N}\mathbf{Y}^T\mathbf{Y}\boldsymbol{\Theta}_0 = \mathbf{I}_J$. Let $\boldsymbol{\Theta}_0^* = \mathbf{Y}\boldsymbol{\Theta}_\mathbf{0}$.

2. **Multivariate non-parametric regression:** Perform a (multi-response) multivariate regression of $\boldsymbol{\Theta}_0^*$ on the original matrix of predictor variables $\mathbf{X}$ producing fitted values $\widehat{\boldsymbol{\Theta}}_0^*$. Let $\mathbf{S}(\hat{\lambda})$ be the linear operator that fit the final model and $\boldsymbol{f}(\mathbf{x}_i)$ be the vector of fitted regression functions.

3. **Optimal scores:** Generate an eigenvector matrix $\boldsymbol{\Phi}$ by performing an eigenvalue based decomposition on the following matrix,

   $$\boldsymbol{\Theta}_0^{*\mathbf{T}}\widehat{\boldsymbol{\Theta}}_0^* = \boldsymbol{\Theta}_0^{*\mathbf{T}}\mathbf{S}(\hat{\lambda})\boldsymbol{\Theta}_0^*$$

   to get the new matrix of optimal scores as, $\boldsymbol{\Theta} = \boldsymbol{\Theta}_0\boldsymbol{\Phi}$.

4. **Update:** Update the fitted regression functions in step 2 using the eigenvector matrix $\boldsymbol{\Phi}$ to get the optimal regression functions vector as,

   $$\boldsymbol{f}^*(\mathbf{x}_i) = \boldsymbol{\Phi}^T\boldsymbol{f}(\mathbf{x}_i)$$

---

The advantage of using this approach is that some of the features of the multivariate regression technique used are inherited. In our case, using MARS to perform discriminant analysis via optimal scoring means that model selection and regularization can be performed by varying the degree of interaction terms, B and/or the penalty parameter, *c*. Other candidate non-parametric regression techniques suggested by Hastie *et al.*(1994) are neural networks (Lippman, 1989), multi-response projections pursuit regression (Friedman & Stuetzle, 1981) and hinge functions (Breiman, 1993).

### 4.3.3 Using the FDA routine for classifying a new observation

Having determined $\boldsymbol{f}^*(\mathbf{x}_i)$, a new applicant $\mathbf{x}_i^{new} \in R^p$ is assigned to a class $k$ that minimizes the following distance measure:

$$d_k(\mathbf{x}_i^{new}) = \left\|\left(\boldsymbol{f}^*(\mathbf{x}_i^{new}) - \bar{\boldsymbol{f}}_k^*\right)\right\|^2 \qquad \forall k = 1,2,\dots,J \tag{4.31}$$

where,

- $\boldsymbol{f}^*(\mathbf{x}_i^{new})$: are the coordinates of the new applicant $\mathbf{x}_i^{new}$ in this new $J$-dimensional subspace and
- $\bar{\boldsymbol{f}}_k^* = \frac{1}{N_k}\sum_{y_i=k} \boldsymbol{f}^*(\mathbf{x}_i)$ ; $\forall k = 1,2,\dots,J$ are the fitted group centroids in this new $J$-dimensional subspace for all those observations in the training sample that occur in each of the $K$ different classes of our classification problem.

## 4.4 Mixture discriminant analysis

The LDA classifier that we have developed requires each class $k \in \{1,2,\dots K\}$ to have a single mean, $\boldsymbol{\mu}_k$ and a common pooled within-class covariance matrix, $\boldsymbol{\Sigma}_w$. If the data is multi-modal (multiple group centroids per class), then this classifier may not perform very well. Mixture discriminant analysis (MDA) exists to handle such a limitation.

### 4.4.1 The procedure

Suppose an observation $\mathbf{x}_i \in R^p$ belongs to one of $r \in \{1,2,\dots,R_k\}$ latent classes of class $k \in \{1,2,\dots,K\}$(which we shall call subclasses of class $k$). In other words, we are assuming that each of the $K$ groups is made up of unobserved $R_k$ subgroups to which the observation $\mathbf{x}_i$ may belong. In order to model the multimodal response data that is

generated by this class, the following mixture model (assuming equal covariance matrices: $\boldsymbol{\Sigma}_{kr} = \boldsymbol{\Sigma}$) is presumed to be generating the outcomes $\mathbf{x}_i$ that we are observing:

$$f_k(\mathbf{x}_i) = P(\mathbf{x}_i|y_i = k) = \sum_{r=1}^{R_k} \omega_{kr} p(\mathbf{x}_i, \boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}) \tag{4.32}$$

where, $\{\omega_{kr}; \sum \omega_{kr} = 1\}$ denote the weights that are being associated with each of the $r$ Gaussian components of class $k$ with probability density function,

$$p(\mathbf{x}_i, \boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}) \sim N(\boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}_{kr} = \boldsymbol{\Sigma}) \tag{4.33}$$

where $\boldsymbol{\mu}_{kr}$ is the mean vector of the $r^{th}$ subclass of class $k$ and $\boldsymbol{\Sigma}_{kr} = \boldsymbol{\Sigma}$ is the common pooled-within class $p \times p$ covariance matrix.

It is important to note that we are assuming that one of the $R_k$ Gaussian distributions pertaining to class $k$ is generating the observation $\mathbf{x}_i$ that we are observing as belonging to class $k$. In order to be able to assign a new observation to a particular class applying the Bayesian approach to LDA that we discussed in section (3.2), the posterior probability of an observation $\mathbf{x}_i$ belonging to class $k$ is given by

$$p(y_i = k|\mathbf{x}_i) = \frac{p(\mathbf{x}_i|y_i = k)\mathrm{p}(y_i = \mathrm{k})}{\sum_{l=1}^{K} p(\mathbf{x}_i|y_i = l)\ p(y_i = l)} = \frac{\sum_{r=1}^{R_k} \omega_{kr} p(\mathbf{x}_i, \boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma})\, \pi_k}{\sum_{l=1}^{K} \sum_{r=1}^{R_l} \omega_{lr} p(\mathbf{x}_i, \boldsymbol{\mu}_{lr}, \boldsymbol{\Sigma})\, \pi_l} \tag{4.34}$$

where, $\pi_k = p(y_i = k)$ denotes the prior probability of an observation belonging to class $k$. One can then assign an observation $\mathbf{x}_i \in R^p$ to that class $y_i = k$ that maximizes $p(y_i = k|\mathbf{x}_i)$ for all $k \in \{1,2,\dots,K\}$. Noting that the denominator in (4.30) does not depend on $k$, one need only find that value of $k$ that maximizes,

$$p(\mathbf{x}_i|y_i = k)\mathrm{p}(y_i = \mathrm{k}) = \sum_{r=1}^{R_k} \omega_{kr} p(\mathbf{x}_i, \boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma})\, \pi_k \tag{4.35}$$

which is equivalent to finding $k \in \{1,2,\dots,K\}$ that maximizes,

$$d_k(\mathbf{x}_i) = \pi_k \sum_{r=1}^{R_k} \omega_{kr} \exp\left(-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_{kr})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_{kr})\right) \tag{4.36}$$

### 4.4.2 Implementing the MDA procedure using the EM algorithm

Given a set of $i = 1,2,\dots,N$ training observations $(\mathbf{x}_i, y_i)$, an estimate of the prior probability can be obtained from the training observations as,

$$\hat{\pi}_k = \frac{N_k}{N} \tag{4.37}$$

where, $N_k$ denotes the number of observations in class $k$ and $N$ is the overall number of observations in the training sample. Hastie & Tibshirani (1996) suggested an iterative technique known as the Expectation Maximization (EM)-algorithm be used to compute the maximum likelihood parameter estimates of $\omega_{kr}$, $\boldsymbol{\mu}_{kr}$ and $\boldsymbol{\Sigma}$, which are otherwise difficult to compute directly (see Appendix A for a detailed discussion of the EM-algorithm).

The EM-algorithm oscillates between the following E-step and M-step until the parameter estimates $\boldsymbol{\Phi} = \{\omega_{kr}, \bar{\mathbf{x}}_{kr}, \mathbf{S}_w\}$ converge:

**E-step:**

Given that the $i^{th}$ observation belongs to class $k$ and initial parameter estimates $\boldsymbol{\Phi} = \{\omega_{kr}, \bar{\mathbf{x}}_{kr}, \mathbf{S}_w\}$, one estimates the probability that the $r^{th}$ subclass of class $k$ (which we will denote by $c_{kr}$) is generating the observation $\mathbf{x}_i$ as:

$$p(c_{kr}|\mathbf{x}_i \in k, \boldsymbol{\Phi}) = \frac{\omega_{kr} p(\mathbf{x}_i|\bar{\mathbf{x}}_{kr}, \mathbf{S}_w)}{\sum_{l=1}^{m} \omega_{kr} p(\mathbf{x}_i|\bar{\mathbf{x}}_{kr}, \mathbf{S}_w)}$$

$$= \frac{\omega_{kr} \exp\left(-\frac{1}{2}(\mathbf{x}_i - \bar{\mathbf{x}}_{kr})^T (\mathbf{S}_w)^{-1} (\mathbf{x}_i - \bar{\mathbf{x}}_{kr})\right)}{\sum_{l=1}^{R_k} \omega_{kl}^{j} \exp\left(-\frac{1}{2}(\mathbf{x}_i - \bar{\mathbf{x}}_{kl})^T (\mathbf{S}_w)^{-1} (\mathbf{x}_i - \bar{\mathbf{x}}_{kl})\right)} \quad \forall r = 1, \dots, R_k \tag{4.38}$$

**M-step:**

The parameter estimates of the mixing probabilities $\omega_{kr}$, subclass-specific mean vectors $\bar{\mathbf{x}}_{kr}$ and the common within-class covariance matrix $\mathbf{S}_w$ are updated using the probabilities from the E-step as follows:

$$\omega_{kr} = \frac{1}{N_k} \sum_{i=1}^{N_k} p(c_{kr}|\mathbf{x}_i \in k, \boldsymbol{\Phi}) \tag{4.39}$$

$$\bar{\mathbf{x}}_{kr} = \frac{\sum_{i=1}^{N_k} \mathbf{x}_i p(c_{kr}|\mathbf{x}_i \in k, \boldsymbol{\Phi})}{\sum_{i=1}^{N_k} p(c_{kr}|\mathbf{x}_i \in k, \boldsymbol{\Phi})} \tag{4.40}$$

$$\mathbf{S}_{kr} = \frac{\sum_{i=1}^{N_k} \mathbf{x}_i p(c_{kr}|\mathbf{x}_i \in k, \boldsymbol{\Phi})(\mathbf{x}_i - \bar{\mathbf{x}}_{kr})(\mathbf{x}_i - \bar{\mathbf{x}}_{kr})^T}{\sum_{i=1}^{N_k} \mathbf{x}_i p(c_{kr}|\mathbf{x}_i \in k, \boldsymbol{\Phi})} \tag{4.41}$$

The parameter estimates $\boldsymbol{\Phi} = \{\omega_{kr}, \bar{\mathbf{x}}_{kr}, \mathbf{S}_{kr}\}$ from the M-step are then used to update subclass probabilities $p(c_{kr}|\mathbf{x}_i \in k, \boldsymbol{\Phi})$ in the E-step and the ensuing subclass probabilities in turn used to re-calculate new parameter estimates in the M-step, repeating the steps until convergence or a number of times. The iterative processes are repeated $K$ times in order to compute the parameter estimates for all the $k = 1,2, \dots, K$ classes and the pooled within covariance matrix can then be computed as:

$$\mathbf{S}_w = \frac{1}{N} \sum_{k=1}^{K} \sum_{r=1}^{R_k} N_k \mathbf{S}_{kr} \tag{4.42}$$

A new observation $\mathbf{x}_i^{new} \in R^p$ will then be assigned to that class $k \in \{1,2, \dots, K\}$ that maximizes

$$d_k(\mathbf{x}_i^{new}) = \pi_k \sum_{r=1}^{m} \omega_{kr} \exp\left(-\frac{1}{2}(\mathbf{x}_i^{new} - \bar{\mathbf{x}}_{kr})^T \mathbf{S}_w^{-1} (\mathbf{x}_i^{new} - \bar{\mathbf{x}}_{kr})\right) \tag{4.43}$$

### 4.4.3 Integrating the optimal scoring routine into the MDA procedure

One can also integrate the optimal scoring routine introduced in section (3.5) into the MDA procedure. This is achieved by replacing the dummy coded $N \times K$ response matrix $\mathbf{Y}$ in equation (3.69) with an $N \times R$ response matrix $\mathbf{Z}$ that contains probabilities associated with membership of an observation $\mathbf{x}_i$ to a particular mixing distribution within a particular class. Since each class $k$ has $R_k$ unobserved classes,

$$R = \sum_{k=1}^{K} R_k \tag{4.44}$$

is the total number of classes in our new classification problem.

As an example, a $K = 3$ class problem with each of the $K$ classes containing $R_k = 2$ unobserved subclasses to which $\mathbf{x}_i$ can belong would have a matrix $\mathbf{Z}$ of the form:

$$\mathbf{Z} = \begin{array}{c} \\ y_1 = 2 \\ y_2 = 1 \\ y_3 = 3 \\ \vdots \\ \vdots \\ y_{N-1} = 1 \\ y_N = 3 \end{array} \begin{array}{c} \begin{array}{cccccc} c_{11} & c_{12} & c_{21} & c_{22} & c_{31} & c_{32} \end{array} \\ \begin{pmatrix} 0 & 0 & 0.4 & 0.6 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.7 & 0.3 \\ & & & \vdots & & \\ & & & \vdots & & \\ 0.1 & 0.9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.8 & 0.2 \end{pmatrix} \end{array} \tag{4.45}$$

In (4.45), $c_{kr}$ denotes the $r^{th}$ latent subclass of class $k$ and $y_i = k$ denotes that the $i^{th}$ case $\mathbf{x}_i$ is being observed as belonging to class $k$. Therefore, focusing on the first row of the matrix $\boldsymbol{Z}$ above, a case $\mathbf{x}_i$ being observed as belonging to class $k = 2$ actually belongs to either the first unobserved subclass $c_{21}$ with probability 0.4 or the second unobserved subgroup $c_{22}$ with probability 0.6. All the other row entries are set to zero such that probabilities in a particular row add up to one. A similar interpretation can be applied to the other rows of the matrix $\mathbf{Z}$.

Algorithm 4.2 shows how the optimal scoring routine provided in Algorithm 3.1 can be modified to incorporate the MDA procedure (Clemmensen *et al*., 2001).

---

Algorithm 4.2: Optimal scoring routine for MDA

---

1. **Initialization:** Initialize the $N \times R$ response matrix $\mathbf{Z}$ containing the subclass membership probabilities. For example, let the initial $\boldsymbol{Z}$ be a {0/1} indicator type matrix.

2. Iterate until convergence or a maximum number of iterations:
   I. Regress $\mathbf{Z}$ on $\mathbf{X}$ to get fitted values $\hat{\mathbf{Z}} = \mathbf{X}\hat{\boldsymbol{B}}$ where $\hat{\boldsymbol{B}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{Z}$
   II. Find those $J$ eigenvectors of $\mathbf{Z}^T\hat{\mathbf{Z}}$, that correspond with the $J \leq R - 1$ largest eigenvalues of $\mathbf{Z}^T\hat{\mathbf{Z}}$ and arrange them in descending order as the column vectors in a $(R \times J)$-dimensional solution matrix $\hat{\boldsymbol{\Theta}}$. This is subject to the normalizing condition: $\frac{1}{N}\boldsymbol{\Theta}^T\mathbf{Z}^T\mathbf{Z}\,\boldsymbol{\Theta} = \boldsymbol{I}_J$

III. Compute the $p \times J$ optimal matrix of regression coefficients : $\widehat{\boldsymbol{B}}_{OS} = \widehat{\boldsymbol{B}}\widehat{\boldsymbol{\Theta}}$

IV. Calculate the $N \times J$ transformed data matrix: $\widetilde{\mathbf{X}} = \mathbf{X}\widehat{\boldsymbol{B}}_{OS}$

V. Compute the parameter estimates $\{\omega_{kr}, \bar{\mathbf{x}}_{kr}, \mathbf{S}_w : \forall r = 1,2,\dots,R_k \;\&\; k = 1,2,\dots,K\}$ using equations $(4.39) - (4.42)$ and the transformed data matrix $\widetilde{\mathbf{X}}$ instead of $\mathbf{X}$, viz:

$$\omega_{kr} = \frac{1}{N_k}\sum_{i=1}^{N_k} p(c_{kr}|\tilde{\mathbf{x}}_i \in k)$$

$$\bar{\mathbf{x}}_{kr} = \frac{\sum_{i=1}^{N_k} \tilde{\mathbf{x}}_i p(c_{kr}|\tilde{\mathbf{x}}_i \in k)}{\sum_{i=1}^{N_k} p(c_{kr}|\tilde{\mathbf{x}}_i \in k)}$$

$$\mathbf{S}_w = \frac{1}{N}\sum_{k=1}^{K}\sum_{r=1}^{R_k} N_k \mathbf{S}_{kr}$$

where:

$$\mathbf{S}_{kr} = \frac{\sum_{i=1}^{N_k} \tilde{\mathbf{x}}_i p(c_{kr}|\tilde{\mathbf{x}}_i \in k)(\tilde{\mathbf{x}}_i - \bar{\mathbf{x}}_{kr})(\tilde{\mathbf{x}}_i - \bar{\mathbf{x}}_{kr})^T}{\sum_{i=1}^{N_k} \tilde{\mathbf{x}}_i p(c_{kr}|\tilde{\mathbf{x}}_i \in k)}$$

VI. Calculate new estimates of the probabilities of latent subclass membership (using the transformed data matrix $\widetilde{\mathbf{X}}$ instead of $\mathbf{X}$) and update matrix $\boldsymbol{Z}$:

$$p(c_{kr}|\tilde{\mathbf{x}}_i \in k) = \frac{\omega_{kr}\exp\left(-\frac{1}{2}(\tilde{\mathbf{x}}_i - \bar{\mathbf{x}}_{kr})^T \boldsymbol{S}_w^{-1}(\tilde{\mathbf{x}}_i - \bar{\mathbf{x}}_{kr})\right)}{\sum_{l=1}^{R_k} \omega_{kr}\exp\left(-\frac{1}{2}(\tilde{\mathbf{x}}_i - \bar{\mathbf{x}}_{kr})^T \boldsymbol{S}_w^{-1}(\tilde{\mathbf{x}}_i - \bar{\mathbf{x}}_{kr})\right)}$$

$$\forall r = 1,2,\dots,R_k \,;\, k = 1,2,\dots,K$$

3. Given a new observation $\mathbf{x}_i^{new} \in R^p$, compute $\tilde{\mathbf{x}}_i^{new} = \widehat{\boldsymbol{B}}_{OS}\,\tilde{\mathbf{x}}_i^{new}$ and assign the new observation to a class $k \in \{1,2,\dots,K\}$, whose value of $k$ results in the largest value of

$$\pi_k \sum_{r=1}^{R_k} \omega_{kr}\exp\left(-\frac{1}{2}(\tilde{\mathbf{x}}_i^{new} - \bar{\mathbf{x}}_{kr})^T \mathbf{S}_w^{-1}(\tilde{\mathbf{x}}_i^{new} - \bar{\mathbf{x}}_{kr})\right)$$

Using the optimal scoring technique to perform MDA produces up to $J \leq (R - 1)$ canonical discriminant functions. These provide a good low-dimensional pictorial view of the dataset when used as axis to plot the dataset.

### 4.5 Conclusion

In this chapter, it has been revealed that a desirable classifier is one that is flexible enough to model both linear and non-linear separations between classes. QDA allows us to fit quadratic decision boundaries in grouping observations in a dataset. Even more flexible decision boundaries can be fitted by using non-parametric regression procedures in the optimal scoring approach to LDA, a technique known as FDA. MDA provides a way of handling multi-modal datasets. This has the potential to improve the accuracy of the classifier.

# CHAPTER 5

# 5. Classification and Regression Trees

## 5.1 Introduction

The classification techniques that have been looked into so far are parametric in nature. This means that the data is presumed to follow a particular probability distribution that is being governed by certain parameters. A response variable *y* is then estimated using a parametric function $\hat{f}(\mathbf{x}_i, a)$ whose model parameters have to be estimated based on the minimization of some appropriately defined goodness of fit function:

$$\mathrm{R(a)} = \frac{1}{N}\sum_{i=1}^{N} L[\, y_i, f(\mathbf{x}_i, a)]$$

This chapter therefore focuses on the development of a classification/regression technique that is now non-parametric in nature. More specifically the domain of the covariate vector $\mathbf{x}_i$ will be partitioned into a series of mutually disjoint rectangular regions using a series of rules to identify regions that have the most homogeneous responses to these predictor variables. A constant value is then fitted to each region with classification trees fitting the most probable class as that constant value and regression trees fitting the mean response for observation in that region.

More specifically,

$$\hat{f}(\mathbf{x}_i, c) = \sum_{m=1}^{M} \hat{c}_m \, 1_{\{\mathbf{x}_i \in R_m\}}$$

is used as a predictor function for *y* with $\hat{c}_m$ being used as a predictor value for *y* if $\mathbf{x}_i$ falls in the region being defined by $R_m$. This procedure is known as Classification and Regression Trees (CART). CART produces a classification tree if the outcome variable is qualitative and a regression tree if the outcome variable is quantitative. A set of 'yes/no' responses to questions relating to the state of the predictor variables is used to recursively split the data into subgroups resulting in an 'upside-down tree-like' structure that is easy to interpret.

Figure 5.1 shows a hypothetical tree that can be constructed from a dataset with predictor variables $\mathbf{x} = \{x_1, x_2, x_3\}$ and a binary outcome variable $y = \{0/1\}$.

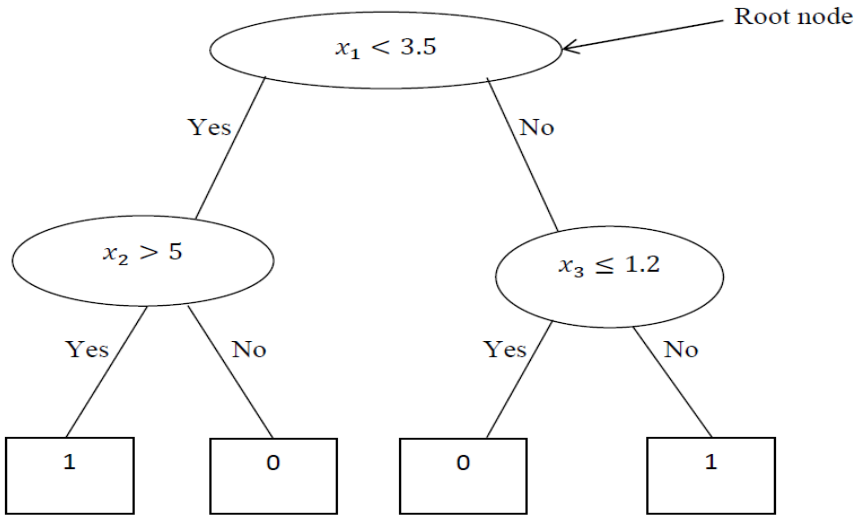


Figure 5.1: An illustration of a classification tree

To create the tree in Figure 5.1, the CART algorithm starts with the entire learning sample being assigned to the *parent or root node of the tree.* The entire dataset assigned to this root node is then partitioned into two mutually disjointed subsets, which form the two *child nodes* in the above tree (as represented by ovals in Figure 5.1). The observations assigned to the same child node are as similar as is possible but the observations assigned to the two differing child nodes are as different as is possible based on some appropriately chosen measure of dissimilarity (or impurity index). The covariates in $\mathbf{x}_i$ are used to determine a splitting rule for the parent node with the process of splitting continuing until some stopping criterion has been satisfied. We will term child nodes that are not split any further *terminal nodes* or *leaves* (as represented by squares in Figure 5.1) and the lines connecting the nodes shall be referred to as being *branches* in the tree.

## 5.2 Growing the Tree

Given a set of training observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, the CART procedure uses the following guidelines to 'grow' a tree.

- a standard set of questions for splitting the nodes,
- a criterion for splitting the nodes,

- a rule to control when to stop splitting the nodes,
- a technique for assigning class labels to the terminal nodes in the tree.

We shall elaborate on the above guidelines and more in this section.

### 5.2.1 A standard set of questions for splitting the nodes

A splitting rule for node *t* will be denoted by *s(t)*. CART splits the sample associated with node $t$ based on a decision rule that uses one of the predictor variables $\{x_1, x_2, \dots, x_p\}$. If the predictor variable $x_j$ is quantitative in value, the splitting rule, $s(t)$ takes on the form:

$$\text{Is the condition: } \{x_j \leq c\} \text{ or } \{x_j > c\} \text{ true?} \qquad (5.1)$$

where, *c denotes* a real number in the domain of $x_j$ that needs to be determined. If the predictor variable $x_j$ is categorical in value, then the splitting rule *s(t)* takes on the form:

$$\text{Is the condition :} \{x_j = m\} \text{ or } \{x_j \neq m\} \text{ true?} \qquad (5.2)$$

for some value $m$ in the domain of $x_j$ that also needs to be determined. For a particular observation in the parent node, if the response to the questions in (5.1) and/or (5.2) is a 'yes' then this observation is cascaded down to the left child node of that tree. Alternatively, the observation is cascaded down to the right child node of that tree.

### 5.2.2 The criterion for splitting the nodes

The underlying objective behind CART is to create a splitting rule that best splits the dataset being associated with the parent node of a tree into two child nodes that are even more *homogeneous (or pure)* in nature. This concept of homogeneity is defined in terms of an *impurity function*, denoted by $i(t)$ that ideally will have the following properties:

1. a unique maximum value for a K-class problem at the point $\left(\frac{1}{K}, \frac{1}{K}, \dots, \frac{1}{K}\right)$,
2. a unique minimum value at the points (1,0,…,0) ,(0,1,…,0),….(0,0,…,1).

Letting $p(k|t)$ denote the *posterior* probability that the observations in a node *t* belong to class $k \in \{1,2,\ldots,K\}$ an impurity function that one could consider using is the following Gini index:

$$Gini(t) = i(t) = 1 - \sum_{k} p^2(k|t) \tag{5.3}$$

Figure 5.2 shows the relationship that exists between this Gini index and $p(k|t)$ for a $K = 2$ class problem.

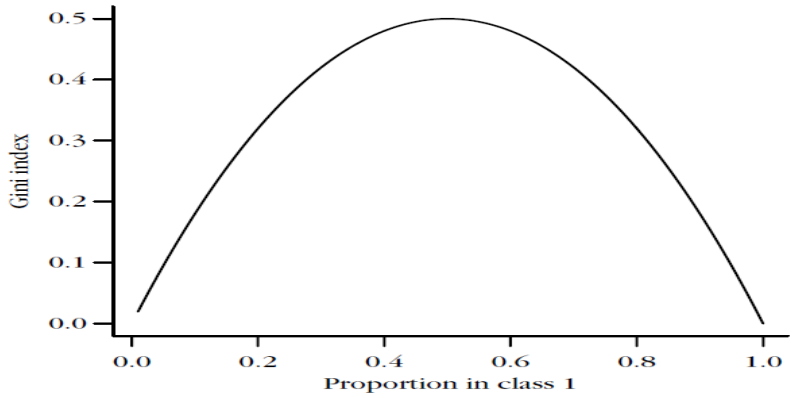


Figure 5.2: Relationship between the Gini index and the proportion of observations in class 1 for a two-class problem

It can be observed that the impurity at any given node *t* reaches its highest value when this node contains an equal proportion of observations from each class in that node and reaches a minimum value when the observations in that node belong to only one of the two possible class.

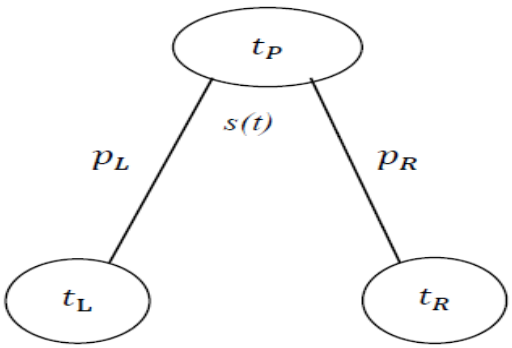


Figure 5.3: Change in Impurity

Consider the diagram that has been given in Figure 5.3 and let $i(t)$ denote the measure of impurity that exists at a node *t*. The goal of CART is to use a rule $s(t)$ to split a parent node $t = t_P$ into a left child node $t_L$ and right child node $t_R$ in such a manner that this split $s(t)$ causes the greatest reduction in the overall impurity of the parent node $t = t_P$, measured by:

$$\Delta i[s(t)] = p(t)i(t) - p(t_L)\, i(t_L) - p(t_R)i(t_R) = i(t) - p_L\, i(t_L) - p_R i(t_R) \quad (5.4)$$

where, $p(t_L) = p_L$ and $p(t_R) = p_R$ denote the proportions of observations in the parent node $t = t_P$ that will go to the left node $t_L$ and to the right node $t_R$, respectively. Note that in (5.4) we have used the fact that;

$$p(t = t_P) = p(t_L) + p(t_R) = 1 \quad (5.5)$$

where $p(t = t_P)$ denotes the proportions of observations contained in the parent node $t_P$ that is being split into the left node $t_L$ and into the right node $t_R$.

Letting $N_P$ denote the number of observations in the parent node $t_P$, if the number of observations that fall into the child nodes $t_L$ and $t_R$ is $N_L$ and $N_{\mathrm{R}}$ respectively, then

$$p_L = \frac{N_L}{N_P} \qquad \text{and} \qquad p_R = \frac{N_{\mathrm{R}}}{N_P} \quad (5.6)$$

where it follows that, $N_P = N_L + N_{\mathrm{R}}$ and $p(t = t_P) = \frac{N_P}{N_P}$ =1.

Criterion (5.4) provides one with a measure of how good a particular splitting rule will be, with the best split of a node *t* being given by

$$s^*(t) = \arg\max_{s(t)} \Delta i[s(t)] = \arg\max_{s(t)} \{i(t) - p_L i(t_L) - p_R i(t_R)\}$$

$$\Leftrightarrow s^*(t) = \operatorname{argmin}_{s(t)} \{p_L i(t_L) + p_R i(t_R)\} \quad (5.7)$$

The product of the number of observations in a node *t*, denoted by $N_t$; and $\Delta i[s(t)]$ produces what represents an *improvement* value for implementing that split in the tree, viz:

$$improve[s(t)] = N_t \Delta i[s(t)] \quad (5.8)$$

Substituting the Gini index (5.3) into the optimization problem in (5.7) produces

$$s^*(t) = \underset{s(t)}{\operatorname{argmin}} \left\{ - \left( p_L \sum_{k=1}^{K} p^2 \left(k|t_L\right) + p_R \sum_{k=1}^{K} p^2 \left(k|t_R\right) \right) \right\} \tag{5.9}$$

Some other impurity functions that one could consider using include:

1. Entropy: $i(t) = \sum_k \mathrm{p}(k|t) \log_2 p(k|t)$
2. Classification error: $i(t) = 1 - \max_k [p(k|t)]$

Bayes' theorem allows us to develop the following expression for the conditional probability $p(k|t)$ that an observation in node *t* belongs to class $k \in \{1,2, \dots, K\}$:

$$p(k|t) = \frac{p(k)p(t|k)}{p(t)} = \frac{\pi_k \left(\frac{N_k(t)}{N_k}\right)}{\sum_{l=1}^{K} \pi_l \left(\frac{N_l(t)}{N_l}\right)} \tag{5.10}$$

where,

$$p(t|k) = \frac{N_k(t)}{N_k} \tag{5.11}$$

denotes the conditional probability that an observation will be assigned to node $t$ given that it belongs to class $k$, $N_k(t)$ denotes the number of observations in node *t* that belong to class *k* and $N_k$ denotes the total number of training sample observations that belong to class *k.*

In (5.10), $p(k) = \pi_k$ denotes the prior probability that an observation belongs to class *k*, which can be estimated from the training sample with $N$ observations as,

$$\hat{\pi}_k = \frac{N_k}{N} \tag{5.12}$$

or supplied by the user if known. If the sample based estimate (5.12) is used, then (5.10) becomes,

$$p(k|t) = \frac{\left(\frac{N_k}{N}\right)\left(\frac{N_k(t)}{N_k}\right)}{\sum_{l=1}^{K}\left(\frac{N_l}{N}\right)\left(\frac{N_l(t)}{N_l}\right)} = \frac{\frac{N_k(t)}{N}}{\sum_{l=1}^{K}\frac{N_l(t)}{N}} = \frac{N_k(t)}{N(t)} \tag{5.13}$$

Thus if sample based estimates of the prior probability are used, the conditional probability $p(k|t)$ is simply the fraction of the number of class $k$ observations falling into node t and the total number of observations falling into node $t$.

### 5.2.3 A rule for controlling when to stop splitting the nodes

A simple rule that one could consider employing is one where the recursive partitioning of nodes is terminated when each terminal node contains only observations from one particular class. However, employing such a rule is likely to result in the generation of an excessively large tree. To prevent this from occurring, one could consider declaring a node $t$ as being a terminal node if splitting that node does not change the impurity beyond a threshold value $\alpha$. Thus, declare a node terminal if

$$\max_{s(t)} \Delta i[s(t)] < \alpha \tag{5.14}$$

Another rule that one could consider using is a rule that stops splitting a node *t*, if the number of observations in the node $N(t)$, is less than a positive real number *r*.

### 5.2.4 A technique for assigning a class label to a particular node

The assignment of a class label to a particular node proceeds as follows: Let $k_t$ denote the class label that is being assigned to node *t*, then

$$k_t = \underset{k}{\text{argmax}}[p(k|t)] \tag{5.15}$$

where, $p(k|t)$ denotes the posterior probability associated with an observation in node *t* belonging to class *k* as given by equation (5.10). With the above notation in hand, given that an observation falls into node *t* a conditional probability of misclassification, denoted by $r(t)$ , and called a *resubstitution error rate* can be given by

$$r(t) = 1 - p(k_t|t) = 1 - \max_{k} p(k|t) \tag{5.16}$$

and an unconditional probability of misclassification at a node *t* by,

$$R(t) = r(t)p(t) \tag{5.17}$$

where, $p(t) = \sum_{k=1}^{K} \pi_k \left(\frac{N_k(t)}{N_k}\right)$ denotes the probability of any observation falling into node *t*, regardless of its class label. For an entire tree *T*, with terminal nodes $\tilde{T}$, an overall misclassification rate can be given by

$$R(T) \quad = \sum_{t \in \tilde{T}} R(t) \tag{5.18}$$

**Theorem**

For any split of the parent node $t = t_p$ into its left and right child nodes $t_L$ and $t_R$ respectively, we have:

$$R(t) \geq R(t_L) + R(t_R)$$

**Proof**:

Let, $k^* = k_t$ be the class label being assigned to a node *t* according to the class allocation rule (5.15). The probability that an observation belongs to class $k^*$ given that it is in node *t* then takes on the value

$$\begin{aligned} p(k^*|t) &= p(k^*, t_L|t) + p(k^*, t_R|t) \\ &= p(k^*|t_L)p(t_L|t) + p(k^*|t_R)p(t_R|t) \\ &= p_L p(k^*|t_L) + p_R p(k^*|t_R) \\ &\leq p_L \max_k p(k|t_L) + p_R \max_k p(k|t_R) \end{aligned} \tag{5.19}$$

It follows that,

$$\begin{aligned} r(t) &= 1 - p(k^*|t) \\ &\geq 1 - \left[p_L \max_k p(k|t_L) + p_R \max_k p(k|t_R)\right] \\ &= p_L \left(1 - \max_k p(k|t_L)\right) + p_R \left(1 - \max_k p(k|t_R)\right) \\ &= p_L r(t_L) + p_R r(t_R) \end{aligned} \tag{5.20}$$

Thus,

$$\begin{aligned} R(t) &= r(t)p(t) \\ &\geq p(t)p_L r(t_L) + p(t)p_R r(t_R) \\ &= p(t_L)r(t_L) + p(t_R)r(t_R) \\ &= R(t_L) + R(t_R) \end{aligned} \tag{5.21}$$

The above theorem shows that the recursive splitting of nodes in the tree growing process will always produce a new tree with a lower overall misclassification rate. Thus, any splitting routine that focuses on minimizing the overall misclassification rate will always produce a bigger tree (with more terminal nodes).

### 5.2.5 Incorporating misclassification costs

The theory developed section (5.2.4) assumes a cost $C_{lm}$ of misclassifying an observation $\mathbf{x}_i$ as belonging to a class $k = l$, when the observation actually belongs to a class $k = m$ , as being the same for all $l \neq m$. As has already been discussed in section (3.2.3), a lender may want to incorporate *misclassification costs* into the classification procedure that one develops. This is because in a credit-scoring context, it is usually far less damaging to have incorrectly classified a non-defaulter as a defaulter than the other way round.

Letting,

$$\sum_{k=1}^{K} C_{lk} p(k|t) \tag{5.22}$$

denote an expected misclassification cost for this problem, a class label $k_t$ can be assigned to node $t$ that minimizes the expected misclassification cost (5.22), viz:

$$k_t = \underset{l}{\operatorname{argmin}} \left[ \sum_{k=1}^{K} C_{lk} p(k|t) \right] \tag{5.23}$$

Analogously, one can define

$$r(t) = \min_{l} \sum_{k=1}^{K} C_{lk} p(k|t) \tag{5.24}$$

and,

$$R(T) = \sum_{t \in \tilde{T}} r(t)p(t) \tag{5.25}$$

as an overall misclassification cost for this model.

### 5.3 Pruning the Tree

Pruning the tree is done to remove those branches that do not contribute much to its predictive power. The following *cost-complexity pruning* criterion,

$$\epsilon(t) = \frac{R(T_{-t}) - R(T)}{size(T) - size(T_{-t})} \tag{5.26}$$

can be used to determine whether branches of a particular node *t* should be removed from the tree (Breiman *et al*., 1984:66). In (5.26), $R(T)$ represents the resubstitution error estimate for the entire tree *T*, $R(T_{-t})$ the resubstitution error estimate for this tree *T* with the branches of node *t* removed, $size(T)$ and $size(T_{-t})$ the number of terminal nodes in the trees *T* and $T_{-t}$ , respectively

The theorem in section (5.2.4) shows that a larger tree will always have a lower resubstitution error estimate. Thus:

$$R(T_{-t}) \geq R(T)$$

which means that $\epsilon(t) \geq 0$ because $size(T) \geq size(T_{-t})$. The cost-complexity function (5.26) represents a tradeoff between an increase in the resubstitution error estimate (cost) that results from removing the branches of node *t* from the tree *T* and the benefit of using a smaller (but less complex) tree $T_{-t}$ that results from removing the branches of that particular node *t*. A small value of $\epsilon(t)$ means that the removal of branches of node *t* does not cause a significant increase in the resubstitution error estimate of the tree *T*. Likewise, a large value of $\epsilon(t)$ means that the branches attached to node *t* are significant.

Letting $T_0$ denote the original unprunned tree grown using the guidelines in section (5.2), $\epsilon(t)$ is computed for each of its non-terminal nodes *t* using the formulae in equation (5.26). That node *t*, generating the smallest value of $\epsilon(t)$ then has its branches

cut-off to produce a new tree (which we will call subtree $T_1$). $T_1$ is then taken as the original unprunned tree and $\epsilon(t)$ is computed again for every non-terminal node with that node generating the smallest value of $\epsilon(t)$ having its branches pruned to produce the subtree $T_2$. This pruning process is repeated until all the branches have been pruned off with the remaining root node being $T_k$. Thus, this pruning mechanism results in a series of simpler trees, $T_0 > T_1 > T_2 > \cdots > T_k$, each of which is subsequently smaller than the preceding tree. The smallest value of $\epsilon(t)$ that is being produced at each stage is called the *cost-complexity parameter* or simply the *cp-value* (denoted by $\alpha$) for that stage.

### 5.4 Selecting an Optimal Tree

Two approaches have been suggested in the literature, depending on the size of the dataset. For large datasets, one uses part of that dataset to build a tree and prune it, and the remaining part of the data to select the optimal tree. For small datasets, a technique based on N-fold cross validation becomes more appropriate.

#### 5.4.1 Testing sample validation

Testing sample validation is a technique utilized for evaluating the accuracy of the tree by randomly splitting the dataset into a learning sample and a testing sample. The learning sample is used for building and pruning the tree resulting in a series of simpler trees $T_0 > T_1 > T_2 > \cdots > T_k$ as outlined in section (5.3). The testing sample is then applied on each of the subtrees from the pruning process and the tree that gives the lowest error rate is selected as being optimal. Breiman *et al.* (1984:72-80), recommended that this technique be used for large datasets, typically greater than 900 observations.

Figure 5.4 shows a typical evolution of the error rate that can occur when both the learning and testing samples are passed through each of the subtrees that have been developed using the learning sample. As the size of the tree increases, the classification error rate associated with each new tree decreases monotonically when applied to the learning data. This supports the more formal result that we have derived in the theorem in section (5.2.4). When applied to the testing data, however, the classification error rate generally declines steadily before increasing as the size of the tree through which this training sample has been passed increases.

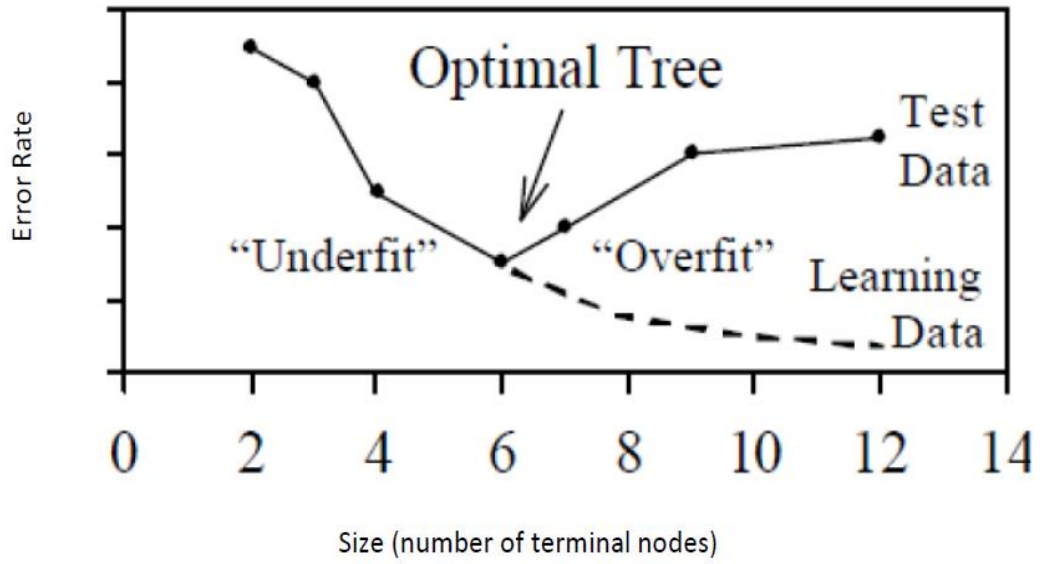


Figure 5.4: Change in the learning and testing sample -based error rate plotted against the size of the tree

Convention detects that it is advisable to choose a tree that produces the lowest error rate when applied to the testing sample as the optimal tree.

### 5.4.2 N-fold cross-validation

N-fold cross-validation can be used if the dataset that we have available is too small to be split into a learning and a testing sample. One begins by randomly splitting the dataset into *N* subsets. One of these subsets is used as a testing sample, while the other $N-1$ subsets are combined and used as a learning sample for the model building procedure. This model building procedure is then repeated a total of *N* times (see Figure 5.6), with a different subset of the data being reserved for use as a testing sample.

The proportion of times that a wrong classification has been made for the testing sample $j \in \{1,2 \dots, N\}$ can serve as an estimate for the error rate $e_j$ for the tree constructed using the other $N-1$ subsets that have been combined to give the learning sample. An average cross-validation error estimate for these *N* models (which we shall denote by $CV_{error}$) can then be given by the equation,

$$CV_{error} = \frac{1}{N}\sum_{j=1}^{N} e_j \tag{5.27}$$

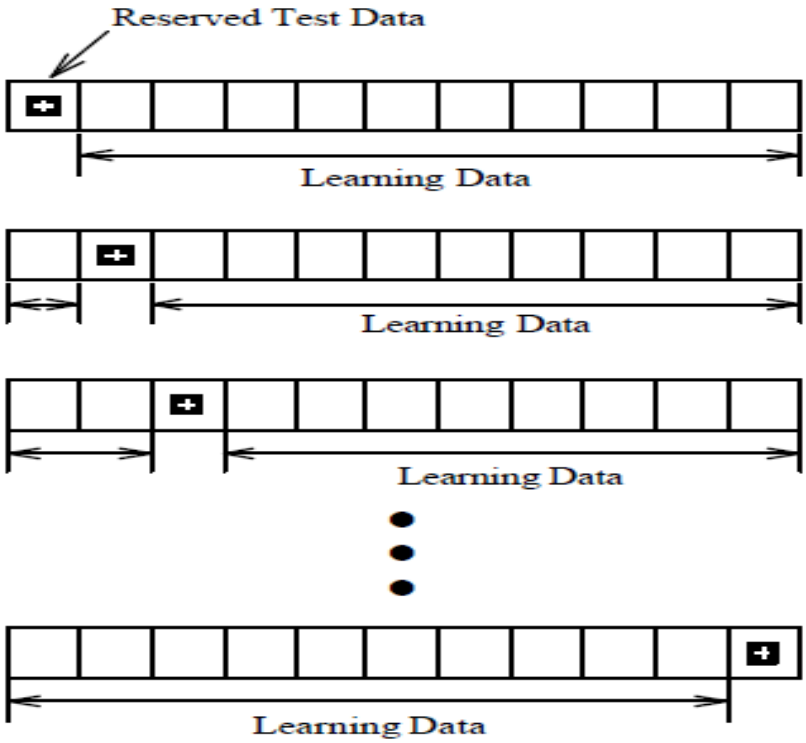


Figure 5.5**:** N-fold Cross-validation

### 5.5 Judging variable importance

According to equation (5.8) importance of a predictor variable $x_j$ can be measured by considering the following improvement in the impurity of node *t*,

$$improve[s(t)] = N_t \Delta i[s(t)]$$

that occurs when a predictor variable $x_j$ is used to split the node $t$. Let $s^*_{x_j}(t)$ denote the best split of a node $t$ that is being based on the predictor variable $x_j$ , then

$$\mathrm{VI}_{CART,t}(x_j) = N_t \Delta i \left[s^*_{x_j}(t)\right] \tag{5.28}$$

will represent the improvement in impurity that occurs when $x_j$ is used as a splitting variable at node *t* . Summing these improvements over all the non-terminal nodes of the tree *T*, gives rise to an overall variable importance measure for the predictor variable $x_j$ that is given by

$$\mathrm{VI}_{CART}(x_j) = \sum_{t \in T} \mathrm{VI}_{CART,t}(x_j) \qquad \forall j = 1, \dots, p \tag{5.29}$$

## 5.6 Conclusion

Derivable from the above discussion is evidence to the effect that using CART is an easier and comprehensible method that can be interpreted without much statistical background or knowhow. However, a slight change in the dataset may result in an entirely different tree being generated, rendering CART an unstable classifier. The forthcoming chapter proffers a consideration of a set of methods that have been specifically designed to help overcome this shortcoming in the model.

# CHAPTER 6

# 6. Bagging, Random Forests and Boosting

## 6.1 Introduction

In this chapter, the researcher delves into some techniques that have been developed to help improve the predictive capability of unstable classifiers such as CART. The basic idea behind each technique is to create a set of classifiers during the model development process. Each of these classifiers is then combined in some optimal way to produce a single predicted value for a new observation $\mathbf{x}_i^{new}$ that one wants to classify.

Two types of ensemble methods will be discussed in this section:-

- a *bagging* and a *random forests* method where, for a new observation $\mathbf{x}_i^{new}$ , one attempts to build several models independent of each other using datasets generated by a *bootstrapping* technique and then combine the predictions derived from these models in some optimal way.
- a *boosting* method where we fit a model to the data, modify the data in response to the type of result that we have achieved and then refit the model repeating this process a number of times and then combine the results that we have obtained in some optimal way.

## 6.2 Bootstrapping

Bootstrapping is a technique developed by Efron (1979) where one randomly draws (with replacement) $B$ samples each of size $N$ (which we will denote by $L_1$,$L_2$,…,$L_B$) from the learning sample $L = \{g_1, g_2, \dots, g_N\}$ where $g_i = (\mathbf{x}_i, y_i)$. This procedure is illustrated in Table 6.1 for a learning sample $L$ with $N = 8$ observations where $B = 7$ bootstrap samples are being drawn from $L$. The probability that an observation $g_i$ will not be selected equals

$$\left(1 - \frac{1}{N}\right)^N \longrightarrow e^{-1} = 0.368 \quad \text{as } N \to \infty \tag{6.1}$$

Table 6.1: Illustration of bootstrap sampling

| $L$ | $g_1$ | $g_2$ | $g_3$ | $g_4$ | $g_5$ | $g_6$ | $g_7$ | $g_8$ |
|---|---|---|---|---|---|---|---|---|
| $L_1$ | $g_4$ | $g_2$ | $g_1$ | $g_7$ | $g_2$ | $g_1$ | $g_2$ | $g_4$ |
| $L_2$ | $g_4$ | $g_1$ | $g_7$ | $g_7$ | $g_5$ | $g_5$ | $g_6$ | $g_1$ |
| $L_3$ | $g_6$ | $g_8$ | $g_1$ | $g_5$ | $g_5$ | $g_1$ | $g_7$ | $g_1$ |
| $L_4$ | $g_2$ | $g_5$ | $g_2$ | $g_1$ | $g_2$ | $g_1$ | $g_4$ | $g_1$ |
| $L_5$ | $g_7$ | $g_1$ | $g_7$ | $g_4$ | $g_6$ | $g_2$ | $g_3$ | $g_8$ |
| $L_6$ | $g_3$ | $g_1$ | $g_4$ | $g_3$ | $g_4$ | $g_2$ | $g_8$ | $g_7$ |
| $L_7$ | $g_8$ | $g_6$ | $g_8$ | $g_2$ | $g_4$ | $g_1$ | $g_3$ | $g_2$ |

Figure 6.1 below indicates that $B$ models are constructed using the bootstrap samples, $L_1, L_2, \ldots, L_B$. In classifying a new observation, each of the $B$ models is then used to generate an outcome.

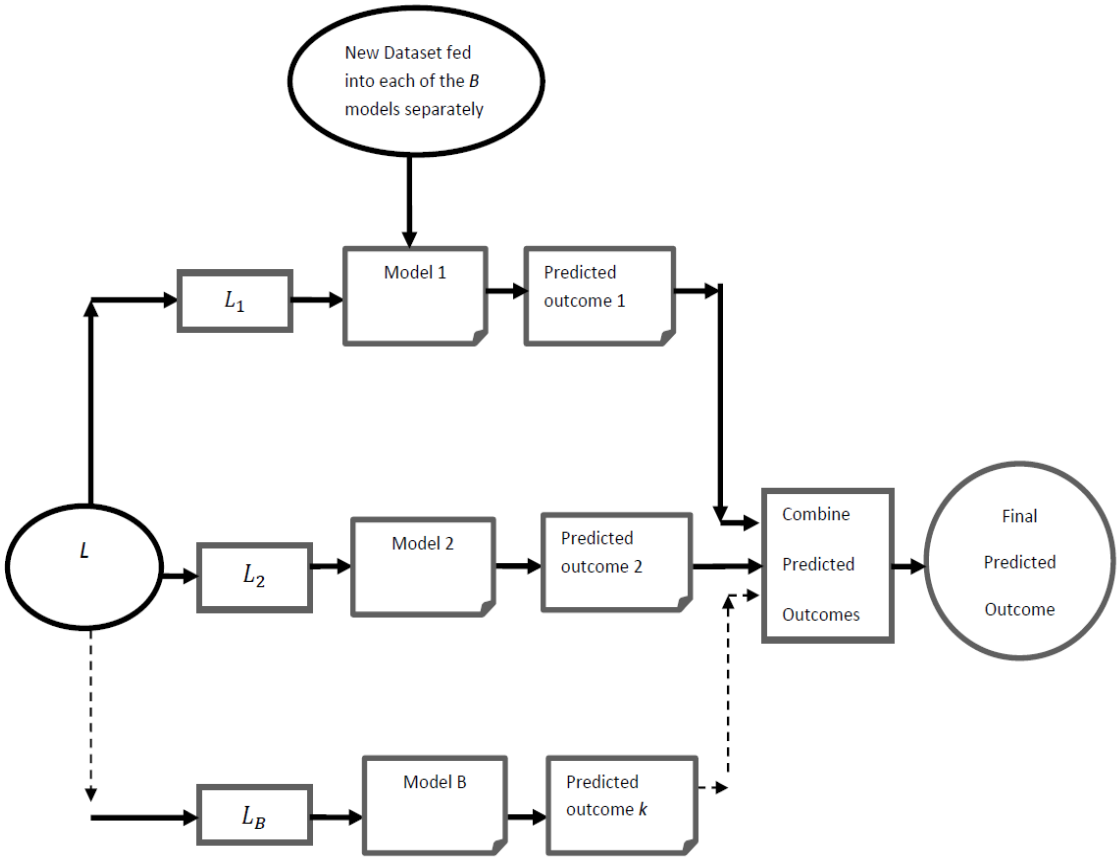


Figure 6.1: Using bootstrapping to improve the performance of a classifier

For a classification problem, the predicted outcomes in Figure 6.1 may then be combined by selecting that class label that has been chosen the most amongst the *B* models as one's predicted outcome. For a regression problem, a predicted outcome for that individual may be obtained by averaging all *B* outcomes that have been produced by the procedure in Figure 6.1.

## 6.3 Bagging

Bagging is an acronym for 'Bootstrap Aggregating' that uses the above bootstrapping technique to improve the performance of an unstable classifier. It achieves this by averaging a particular method of classification over *B* bootstraps samples in an attempt to reduce the variance associated with such a classifier. We will introduce the concept of bagging with the aid of an example (see section 6.3.1).

### 6.3.1 The procedure

Given a learning sample *L*, suppose we generate six bootstrap samples from *L* and on applying the CART procedure to each of the six bootstrap samples we produce the classification trees that appear in Figure 6.2 below.

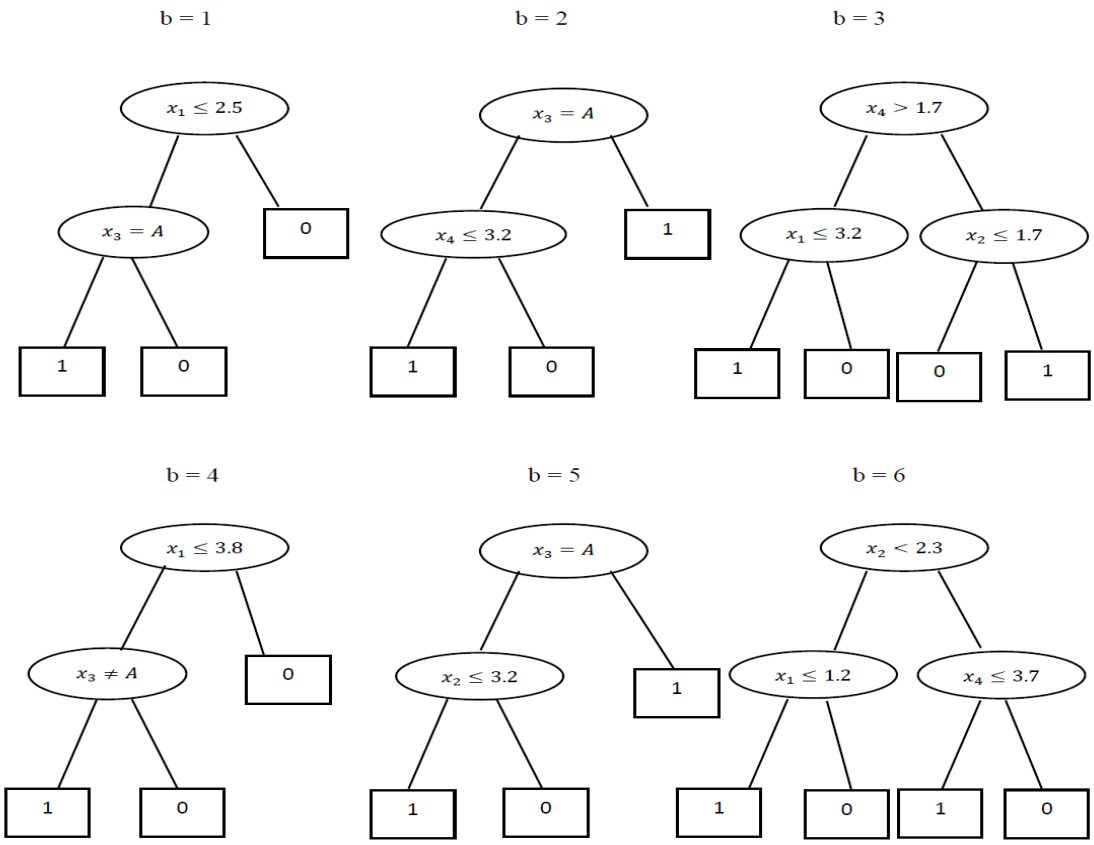


Figure 6.2: Illustration of bagging

Let $C(L_b, \mathbf{x}_i^{new})$ denote the class label that is being predicted for a new observation $\mathbf{x}_i^{new}$ when it is passed through the tree that has been constructed from the bootstrap sample $L_b$. In particular, assume that we want to assign a class label to a new observation $\mathbf{x}_i^{new} = \{x_1 = 1.7, x_2 = 3.5, x_3 = A, x_4 = 2\}$ using the method of bagging. Running this $\mathbf{x}_i^{new}$ down the six trees in Figure 6.2 produced the classifications in Table 6.2.

Table 6.2: Classifying new observations using bagging

| Tree based on bootstrap sample $L_b$ | Predicted outcome;$C(L_b, \mathbf{x}_i^{new})$ |
|---|---|
| b = 1 | 1 |
| b = 2 | 1 |
| b = 3 | 1 |
| b = 4 | 0 |
| b = 5 | 0 |
| b = 6 | 1 |

A bagging estimate for this new observation will then set,

$$\hat{C}_{bag}(\mathbf{x}_i^{new}) = \text{ Majority Vote}\{C(L_b, \mathbf{x}_i^{new})\}_{b=1}^{B=6} = 1 \tag{6.2}$$

For an outcome variable that is quantitative, the bagging estimate $\hat{f}_{bag}(\mathbf{x}_i^{new})$ becomes the average of all these predicted outcomes:

$$\hat{f}_{bag}(\mathbf{x}_i^{new}) = \frac{1}{B}\sum_{b=1}^{B} \text{f}(L_b, \mathbf{x}_i^{new}) \tag{6.3}$$

We outline the bagging algorithm below as follows:

Algorithm 6.1: Bagging Algorithm

1. Generate $B$ bootstrap samples $L_1, L_2, \ldots, L_B$ each of the same size $N$ from the learning sample $L$.
2. Fit a CART to each of the $B$ bootstrap samples. Do not prune the trees.
3. Predict the outcome of a new observation $\mathbf{x}_i^{new}$ by selecting as a class label for this new observation the class label that produces the majority of votes from all $B$ trees that have been generated for a classification problem or by selecting as its value the average of all the predicted outcomes for a regression problem.

### 6.3.2 Proof that Bagging works

An insight into why aggregating models would work is best understood for a quantitative response variable. Suppose the training observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $y_i$ is a quantitative variable, are being independently drawn from a population distribution $Q$ that puts an equal weight of $\frac{1}{N}$ on each observation in the above sample. An outcome for $y_i$ can be then be estimated using the following relationship:

$$y_i = f_{bag}(\mathbf{x}_i) + \varepsilon_i \quad \text{with error term} \quad \varepsilon_i \sim N(0, \sigma_\varepsilon^2) \tag{6.4}$$

where, $f_{bag}(\mathbf{x}_i) = E_Q[\,\hat{f}^*(\mathbf{x}_i)\,]$ denotes the 'true' bagging estimate that has been computed using the bootstrap datasets, $\{(\mathbf{x}_1^*, y_1^*),(\mathbf{x}_2^*, y_2^*),\ldots,(\mathbf{x}_N^*, y_N^*)\}$ that have been drawn from the population $Q$ (and not from the learning sample, $L$). An expected squared prediction error evaluated at an input point $\mathbf{x}_i$ for the model $\hat{f}^*$ that has been constructed from a bootstrap sample $\{(\mathbf{x}_1^*, y_1^*),(\mathbf{x}_2^*, y_2^*),\ldots,(\mathbf{x}_N^*, y_N^*)\}$ is given by

$$E_Q[y_i - \hat{f}^*(\mathbf{x}_i)]^2 = E_Q[y_i - f_{bag}(\mathbf{x}_i) + f_{bag}(\mathbf{x}_i) - \hat{f}^*(\mathbf{x}_i)]^2$$

$$= E_Q\big[y_i - f_{bag}(\mathbf{x}_i)\big]^2 + 2E_Q\big[y_i - f_{bag}(\mathbf{x}_i)\big]E_Q\big[\hat{f}^*(\mathbf{x}_i) - f_{bag}(\mathbf{x}_i)\big] + E_Q[\hat{f}^*(\mathbf{x}_i) - f_{bag}(\mathbf{x}_i)]^2$$

Since, $f_{bag}(\mathbf{x}_i) = E_Q\big[\hat{f}^*(\mathbf{x}_i)\big]$ it follows that the middle term becomes

$$E_Q\big[\hat{f}^*(\mathbf{x}_i) - f_{bag}(\mathbf{x}_i)\big] = f_{bag}(\mathbf{x}_i) - f_{bag}(\mathbf{x}_i) = 0$$

$$\Longrightarrow E_Q[y_i - \hat{f}^*(\mathbf{x}_i)]^2 = E_Q[y_i - f_{bag}(\mathbf{x}_i)]^2 + E_Q[\hat{f}^*(\mathbf{x}_i) - f_{bag}(\mathbf{x}_i)]^2 \tag{6.5}$$

The term $E_Q[\hat{f}^*(\mathbf{x}_i) - f_{bag}(\mathbf{x}_i)]^2 \geq 0$ denotes the variance of the predicted outcome $\hat{f}^*(\mathbf{x}_i)$ about its mean $f_{bag}(\mathbf{x}_i)$ thus

$$E_Q[\hat{y}_i - \hat{f}^*(\mathbf{x}_i)]^2 \geq E_Q\big[\hat{y}_i - f_{bag}(\mathbf{x}_i)\big]^2 \tag{6.6}$$

The above result implies that the bagging estimate $f_{bag}(\mathbf{x}_i) = E_Q[\,\hat{f}^*(\mathbf{x}_i)\,]$ will always have a lower mean square error than the estimate $\hat{f}^*(\mathbf{x}_i)$ that has been based on a single sample. In practice, it is very difficult to get the ideal bagging estimate $f_{bag}(\mathbf{x}_i)$ because

the actual population $Q$ from which the data is being generated is not available. The estimate given in equation (6.3) is usually used as an approximation for $f_{bag}(\mathbf{x}_i)$, i.e.

$$\hat{f}_{bag}(\mathbf{x}_i) = \frac{1}{B}\sum_{b=1}^{B} \mathrm{f}(L_b, \mathbf{x}_i) \tag{6.7}$$

where $\mathrm{f}(L_b, \mathbf{x}_i)$ denotes the outcome that is being generated from the model that is being constructed from the bootstrap sample $L_b$.

The underlying principle behind bagging is that by averaging the estimates that are being produced by many noisy models one may be able to reduce the variance associated with this estimator. The rationale for this is as follows: An average of B independent and identically distributed random variables $\{X_i : i = 1,2,\dots,B\}$, each with a common variance $\sigma^2$ will have a variance

$$\mathrm{Var}\left(\frac{1}{B}\sum_{i=1}^{B} X_i\right) = \frac{1}{B^2} * \sum_{i=1}^{B} Var(X_i) = \frac{1}{B^2} * B = \frac{\sigma^2}{B} \tag{6.8}$$

It follows that as $B \to \infty$, $\mathrm{Var}\left(\frac{1}{B}\sum_{i=1}^{B} X_i\right) \to 0$. Thus, by combining many classifiers, we may subsequently be able to reduce the variance associated with using the average of these classifiers to assign a class label to a new observation.

To help understand why bagging may also work for a categorical response based outcome, suppose each of the observations $\mathbf{x}_i \in R^p$ belong to one of two possible groups $y_i = k \in \{0,1\}$. Consider an observation $\mathbf{x}_i$ whose true class label is known to be $y_i = 1$. Given that a set of $B$ individual classifiers $\{C(L_b, \mathbf{x}_i) : b = 1,2,\dots,B\}$ independently assign this observation $\mathbf{x}_i$ to a class $k = \{0,1\}$,

$$\mathrm{S} = \sum_{b=1}^{B} 1_{\{C(L_b, \mathbf{x}_i)) = 1\}} \tag{6.9}$$

will represent the total number of correctly predicted class classifications for this observation amongst all $B$ of the bootstrap sample based trees. Because the classifiers are presumed to independent, (6.9) follows a binomial distribution:

$$\mathrm{S}\sim Bin(B, 1-\ \varepsilon) \tag{6.10}$$

where

$$\varepsilon_b = \varepsilon \equiv 1 - P(C(L_b, \mathbf{x}_i)\ =\ 1)$$

denotes a constant error rate $\varepsilon$ associated with the misclassification of $\mathbf{x}_i$ to class $y_i = 0$ when the true class label is $y_i = 1$. Consequently,

$$Pr\left(\mathrm{S} > \frac{B}{2}\right) = 1 \text{ as } B \longrightarrow \infty \tag{6.11}$$

provided that all $B$ of the bootstrap sample based trees, have a misclassification rate better than random guessing in this $K = 2$ class problem (i.e. $\varepsilon_b = \varepsilon < 0.5$).

The proof of (6.11) can be established by noting that for large B, the *central limit theorem* allows one to approximate the binomial distribution (6.10) using the following normal distribution (Hansen, 2011):

$$\mathrm{S}\sim N(Bp, Bp(1-\ p)) \qquad \text{with } p = 1 - \varepsilon$$

where $Bp$ and $Bp\ (1-p)$ are the mean and variance of S, repectively.

$$\Longrightarrow Pr\left(\mathrm{S} > \frac{B}{2}\right) = Pr\left(Z \geq \frac{\left(\frac{B}{2} - Bp\right)}{\sqrt[2]{Bp\ (1-p)}}\right) \qquad \text{where } Z\sim N(0,1)$$

$$= Pr\left(Z \geq \frac{\sqrt[2]{B}\left(\frac{1}{2} - p\right)}{\sqrt[2]{p\ (1-p)}}\right) = Pr\left(Z \geq \frac{\sqrt[2]{B}\left(\varepsilon_b - \frac{1}{2}\right)}{\sqrt[2]{(1-\varepsilon_b)\varepsilon_b}}\right) = Pr\left(Z \geq c\sqrt[2]{B}\right)$$

where $c = \frac{\left(\varepsilon_b - \frac{1}{2}\right)}{\sqrt[2]{(1-\varepsilon_b)\varepsilon_b}}$ is a negative constant for values of $\varepsilon < 0.5$ . Therefore,

$$\lim_{B\to\infty} Pr\left(Z \geq c\ \sqrt[2]{B}\right) = \Pr(Z \geq -\infty) = 1$$

This means that as the number B of class based predictions that are being combined increases, more than half of the cases will be correctly classified. This result is based on

the principle of "collective wisdom amongst a crowd" as discussed by Surowiecki (2000). According to this principle, a committee composed of many people is more likely to reach the correct decision compared to a committee with fewer people.

### 6.3.3 Judging variable importance

Since the bagging estimate represents a collection of CART based trees, the same variable importance measure (5.29),

$$\mathrm{VI}_{CART}(x_j) = \sum_{t \in T} \mathrm{VI}_{CART,t}(x_j) \qquad \forall j = 1, \ldots, p$$

that was developed in section (5.5) can be computed for each tree in the bagging estimate. The average variable importance measure for the predictor variable $x_j$ in the bagging estimate is then given by:

$$VI_{bag}(x_j) = \frac{1}{B} \sum_{b=1}^{B} \mathrm{VI}_{CART}(x_{jb}) \qquad \forall\, j = 1,2, \ldots, p \tag{6.12}$$

where, $\mathrm{VI}_{CART}(x_{jb})$ denotes the variable importance of $x_j$ in the $b^{th}$ tree. If $VI_{bag}(x_j) \approx 0$, then the predictor variable $x_j$ is not considered as being an important variable to include in one's classification algorithm.

## 6.4 Random Forests

The concept of a 'random forest' has been developed as a generalization of bagging with the objective of reducing the amount of correlation that may exist amongst the trees being grown from the bootstrap samples. As is the case with a bagging technique, each tree in the forest is built from a bootstrap sample that has been drawn with replacement from the learning sample.

However, it should be noted that with random forests when splitting a node during the construction of a tree the split that is chosen is no longer the best split that results from use of all *p* predictor variables to form a splitting rule. Instead, the best split results from using a randomly generated subset $m < p$ of these predictor variables to form the splitting rule. By randomly selecting these variables one is helping to reduce the correlation that may exist between the trees that are being generated if one were to make

use of all $p$ predictor variables when creating one's splitting rule. Reducing the correlation between the terms that compose the components of equation (6.7) should in turn reduce the variance associated with the averaging of these components that eventually produce a final classifier.

The rationale for this is as follows: An average of B independent and identically distributed random variables $\{X_i: i = 1,2, \dots, B\}$, each with a common variance $\sigma^2$ will have a variance $\sigma^2/B$ as shown in equation (6.8). If the variables are identicallly distributed but have a positive correlation $\rho$ then the variance of this average will be

$$
\begin{aligned}
&\mathrm{Var}\left(\frac{1}{B}\sum_{i=1}^{B} X_i\right) = \\
&\quad = \frac{1}{B^2}\left[\{Var(X_1 + X_2 + \cdots X_B)\} + \left\{2Cov\left(X_1, X_2\right) + \cdots + 2Cov(X_1, X_B)\right\}\right. \\
&\qquad\qquad \left. + \{2Cov(X_2, X_3) + \cdots 2Cov(X_2, X_B)\} + \cdots + 2Cov(X_{B-1}, X_B)\right] \\
&\quad = \frac{1}{B^2}[B\sigma^2 + 2B\rho\sigma^2 + 2(B-1)\rho\sigma^2 + 2(B-2)\rho\sigma^2 + \cdots + 2\rho\sigma^2] \\
&\quad = \frac{1}{B^2}\sigma^2[B + 2\rho\sum_{k=1}^{B-1}(B-k)] \\
&\quad = \frac{1}{B^2}\sigma^2\left[B + 2\rho\left\{B(B-1) - \frac{B(B-1)}{2}\right\}\right] \\
&\quad = \frac{1}{B^2}\sigma^2\left[B + 2\rho B(B-1)\left\{1 - \frac{1}{2}\right\}\right] \\
&\quad = \frac{1}{B^2}\sigma^2[B + \rho B(B-1)] \\
&\quad = \frac{1}{B^2}\sigma^2 * B[1 + \rho B - \rho] \\
&\quad = \frac{\sigma^2}{B}[\rho B + (1-\rho)] \\
&\quad = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2 \qquad (6.13)
\end{aligned}
$$

It follows that as $B \to \infty$, then $\mathrm{Var}\left(\frac{1}{B}\sum_{i=1}^{B} X_i\right) \to \rho\sigma^2$. Thus, reducing the correlation $\rho$ between the trees is desirable if one wants the resultant estimator to have a smaller variance.

### 6.4.1 Implementing the procedure

Given the learning sample observations $\{(\mathbf{x}_i, y_i)\}_i^N$, random forests attempt to improve on the bagging Algorithm 6.1 by decorrelating the trees using the following adjustment:

---

Algorithm 6.2: Random Forests Algorithm

---

1. Generate $B$ bootstrap samples $L_1, L_2, \ldots, L_B$ each of size $N$ from the learning sample $L = \{(\mathbf{x_1}, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$.

2. For each bootstrap sample, fit a tree with the following adjustments being made at each node, *t*:

    - Randomly select $m < p$ predictor variables.
    - Use only the randomly selected predictor variables in finding the best split of the $t^{th}$ node.

    Do not prune the trees.

3. Predict the outcome of a new observation $\mathbf{x}_i^{new}$ by selecting as a class label for this new observation the class label that produces the majority of votes from all $B$ trees that have been generated for a classification problem or by selecting as its value the average of all the predicted outcomes for a regression problem.

---

The performance of these trees can be assessed by running the out-of-the-bag (OOB) sample (those observations that have not been used in the construction of a particular tree) down the constructed tree. The proportion of times that a wrong classification has been made in this OOB sample can serve as an estimate for the error rate $e_b$ for that tree. An average OOB error estimate for the entire forest can then be given by the equation,

$$\text{OOB} - \text{ER} = \frac{1}{B}\sum_{b=1}^{B} e_b \tag{6.14}$$

When the OOB-ER of the forest begins to stabilize, it is recommended to stop adding more trees to the forest to avoid over fitting.

### 6.4.2 Judging variable importance

We now shift our focus on the problem of deciding which predictor variables to include in our classification algorithm. It should be noted that the random forests technique also uses a similar Gini index based variable importance measure to rank the variables that one may want to include in one's classification algorithm. Thus,

$$VI_{RF}(x_j) = \frac{1}{B}\sum_{b=1}^{B} \mathrm{VI}_{CART}(x_{jb}) \qquad \forall\, j = 1,2,\ldots,p \tag{6.15}$$

where, $\mathrm{VI}_{CART}(x_{jb})$ denotes the CART based variable importance measure that we have developed in (5.29) for the predictor variable $x_j$ in the $b^{th}$ tree of the forest, can be used as the random forests based variable importance measure (also known as the mean decrease in Gini). If $VI_{RF}(x_j) \approx 0$, then the predictor variable $x_j$ is not considered as being an important variable to include in one's classification algorithm.

## 6.5 Boosting

Boosting is a method for improving the accuracy of a classification model. It is based on the idea that it is often easier to find and average many 'rough rule of thumb' predictions than it is to find a single highly accurate prediction rule. Whereas bagging also seeks to combine the predictions that are being produced by several models, boosting is different in that when it fits a decision tree to the training sample it seeks to identify an area of poor fit and then update the next tree accordingly. Instead of working with a newly generated bootstrap sample, the boosting algorithm re-weights the observations in the original learning sample. This has the practical effect of giving more weight to those observations that were misclassified in the previous iteration and less weight to those observations that have been correctly classified. Subsequently, the next classifier to be fitted need only concentrate on observations incorrectly classified in the previous round. The classifiers obtained by this method of successive reweighting are then combined to produce a final classifier with better properties.

Boosting algorithms differ in how they quantify a lack of fit and how they then adjust their settings for the next iteration of the algorithm. The original boosting algorithm was developed for a classification problem by Schapire (1990). Schapire's original algorithm was limited to a two-class problem and combined the outcomes of only three

classifiers produced from three filtered versions of the learning sample by simple majority voting. Freund (1995) improved upon Schapire's (1990) algorithm using a variation called *boost by majority* that combined many weak learners at the same time. However both these algorithms required that the base classifiers have a constant error rate. Freund & Schapire's (1997) collaboration led to the development of the very popular *adaptive boosting* algorithm termed *AdaBoost*, which dropped the assumption of a fixed error rate. Breiman (1998) later developed *adaptive resampling and combining* (*arcing*) algorithms, which generalized the overall technique of boosting. Freund & Schapire's (1997) AdaBoost algorithm is a special case of arcing algorithms. For the purposes of this study, we will focus on Freund & Schapire's (1997) AdaBoost algorithm.

### 6.5.1 The AdaBoost procedure for a K=2 class problem

Given a set of observations $\{(\mathbf{x}_i, y_i)\}_i^N$ in a learning sample, with $y_i$ being given the following binary coding $\{-1, +1\}$, the AdaBoost algorithm proceeds as follows:

**Step 1**: As a starting point for the algorithm ($t = 1$), a weight $\omega_{t=1}(i) = \frac{1}{N}$ is assigned to each of the $i = 1,2, \dots, N$ observations in the learning sample.

**Step 2:** A base classifier (CART in our case) is then fitted to the learning sample containing those observations that have been weighted by a factor $\omega_t(i)$ to give the classification model $C_t$ for this $t^{th}$ iteration whose resubstitution error estimate is given by

$$\epsilon_t = \frac{\sum_{i=1}^{N} \omega_t(i).\, 1_{\{C_t(\mathbf{x}_i) \neq y_i\}}}{\sum_{i=1}^{N} \omega_t(i)} \tag{6.16}$$

where $C_t(\mathbf{x}_i)$ represents the predicted outcome at input point $\mathbf{x}_i$.

**Step 3:** The error rate $\epsilon_t$ is then used to create the following positive-valued scaling factor

$$\alpha_t = \frac{1}{2} In\left(\frac{1 - \epsilon_t}{\epsilon_t}\right) \tag{6.17}$$

The scaling factors $\alpha_t$ are strictly positive for values of $1 - \epsilon_t > {}^1/_2$ (this is the same as demanding the model to be boosted to have an accuracy rate $1 - \epsilon_t$ that is slightly better than random guessing in the K=2 class problem). If $\alpha_t$ becomes negative, the weights will be updated in the opposite direction in the next step resulting in the failure of the boosting procedure.

**Step 4:** The observations which have been incorrectly classified, then have the weights associated with them inflated while the weights associated with those, which have been correctly classified are deflated in value using the following function:

$$\omega_{t+1}(i) = \frac{\omega_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } \; y_i = C_t(\mathbf{x}_i) \\ e^{\alpha_t} & \text{if } \; y_i \neq C_t(\mathbf{x}_i) \end{cases} \tag{6.18}$$

where $Z_t$ is a normalization constant that ensures that the weights add up to one ($\sum_t \omega_t = 1$) thereby making $\omega_t$ a probability distribution function, viz

$$\Longrightarrow Z_t = \sum_{i=1}^{N} \left[ \omega_t(i) \times \begin{cases} e^{-\alpha_t} & \text{If } \; y_i = C_t(\mathbf{x}_i) \\ e^{\alpha_t} & \text{If } \; y_i \neq C_t(\mathbf{x}_i) \end{cases} \right]$$

$$= \sum_{i:y_i = C_t(\mathbf{x}_i)} \omega_t(i) e^{-\alpha_t} + \sum_{i:y_i \neq C_t(\mathbf{x}_i)} \omega_t(i) e^{\alpha_t}$$

$$= (1 - \epsilon_t) e^{-\alpha_t} + \epsilon_t e^{\alpha_t}$$

$$= (1 - \epsilon_t) \left( \frac{1 - \epsilon_t}{\epsilon_t} \right)^{-\frac{1}{2}} + \epsilon_t \left( \frac{1 - \epsilon_t}{\epsilon_t} \right)^{\frac{1}{2}} \qquad \{\text{on inserting (6.17)}\}$$

$$= (\epsilon_t)^{\frac{1}{2}} (1 - \epsilon_t)^{\frac{1}{2}} + (\epsilon_t)^{\frac{1}{2}} (1 - \epsilon_t)^{\frac{1}{2}}$$

$$= 2\sqrt{\epsilon_t (1 - \epsilon_t)} \tag{6.19}$$

**Step 5:** The steps $2 - 4$ are repeated $t = 1, \ldots, T$ times updating the weights at each stage using (6.18) to give $\alpha_t$ weights and $C_t$ fitted classifiers. An observation $\mathbf{x}_i$ is assigned to the class $y_i = k \in \{-1, +1\}$ by considering the sign of the following weighted combinations of the predicted outcomes:

$$C_{boost}(\mathbf{x}_i) = sign\left\{ \sum_t \alpha_t \, C_t(\mathbf{x}_i) \right\} \tag{6.20}$$

An appropriate maximum number of iterations $T$ one may consider would be when there is no significant change in the learning error rate of $C_{boost}(\mathbf{x}_i)$ viz:

$$learning\ error[C_{boost}(\mathbf{x}_i)] = \frac{1}{N}\sum_{i=1}^{N} 1_{\{y_i \neq C_{boost}(\mathbf{x}_i)\}} \tag{6.21}$$

We summarize the AdaBoost algorithm for a two-class problem below:

---

Algorithm 6.3: AdaBoost algorithm for a two-class problem

---

1. Initialize: $\omega_{t=1}(i) = \frac{1}{N}$ for all $i = 1, \ldots, N$.

2. Repeat for $t = 1,\ldots, T$ .

   I. Fit the training algorithm (CART in our case) to the learning sample weighted by $\omega_t$, in order to obtain the model $C_t: \mathbf{x}_i \rightarrow y_i = \{-1, +1\}$.

   II. Compute the weighted error rate $\epsilon_t = P_{\omega_t}\{C_t(\mathbf{x}_i) \neq y_i\}$

   $$\Longrightarrow \epsilon_t = \frac{\sum_{i=1}^{N} \omega_t(i).\, 1_{\{C_t(\mathbf{x_i}) \neq y_i\}}}{\sum_{i=1}^{N} \omega_t(i)}$$

   III. Compute $\alpha_t = \frac{1}{2} In\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

   IV. Compute the new weights $\omega_{t+1}(i)$ for all the observations $i = 1, \ldots, N$

   $$\omega_{t+1}(i) = \frac{\omega_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } y_i = C_t(\mathbf{x}_i) \\ e^{\alpha_t} & \text{if } y_i \neq C_t(\mathbf{x}_i) \end{cases}$$

3. Assign a new observation $\mathbf{x}_i^{new} \in R^P$ to the class $y_i = k \in \{-1, +1\}$ by using the sign of the weighted combination of the base classifiers:

$$C_{boost}(\mathbf{x}_i^{new}) = sign\left\{\sum_t \alpha_t\, C_t(\mathbf{x}_i^{new})\right\} \tag{6.22}$$

---

### 6.5.2 Extending the AdaBoost algorithm to the K>2 class problem

The AdaBoost algorithm outlined in Algorithm 6.3 above can also be applied to a $K > 2$ class problem, as long as for the resubstitution error rate in (6.16) we have

$$1 - \epsilon_t > \frac{1}{2} \qquad \forall t = 1, \dots, T \tag{6.23}$$

by making the following adjustment to (6.20):

$$C_{boost}(\mathbf{x}_i) = \underset{y_i}{arg\,max} \sum_{t=1}^{T} \alpha_t . 1_{C_t(\mathbf{x}_i)=y_i} \tag{6.24}$$

The above class allocation rule means that one now assigns an observation $\mathbf{x}_i$ to the class label $y_i \in \{1,2, \dots, K\}$ that receives the most weighted votes. Without the restriction imposed in (6.23), the weights $\alpha_t$ become negative whenever $\epsilon_t > \frac{1}{2}$ leading to the failure of the AdaBoost proposal.

Zhu *et al.*(2009) further modified the scaling factors (6.17) to

$$\alpha_t = In\left(\frac{1 - \epsilon_t}{\epsilon_t}\right) + In(K - 1) \tag{6.25}$$

such that $\alpha_t$ is now positive whenever $(1 - \epsilon_t) > \frac{1}{K}$. This means that the accuracy rate $(1 - \epsilon_t)$ of each of the base classifiers $\{C_t : t = 1,2, \dots, T\}$ must be slightly better than classification by chance in the *K*-class problem. For $K = 2$, the scaling factors (6.17) and (6.25) are the same since we will now have $\ln 1 = 0$ in (6.25). A summary of the AdaBoost algorithm for a multi-class problem is provided in Algorithm 6.4 (page 91).

### 6.5.3 Judging variable importance

Since the boosting estimate of the outcome $C_{boost}(\mathbf{x}_i)$ is a weighted sum of predictions by CART based classifiers, variable importance (which we shall denote by $VI_{boost}(x_j)$ ) is measured using the following formulae:

$$VI_{boost}(x_j) = \frac{1}{T} \sum_{t=1}^{T} \alpha_t \mathrm{VI}_{CART}(x_{jt}) \qquad \forall\ j = 1,2, \dots, p \tag{6.26}$$

where, $\alpha_t$ is the weight associated with the $t^{th}$ iteration and $\mathrm{VI}_{CART}(x_{jt})$ is the CART based variable importance measure in equation (5.29) of the predictor variable $x_j$ in the

tree of the $t^{th}$ iteration. If $VI_{boost}(x_j) \approx 0$, then the predictor variable $x_j$ is not considered as being an important variable to include in one's classification algorithm.

---

Algorithm 6.4: Multi-class AdaBoost algorithm

---

1. Initialize: $\omega_1(i) = \frac{1}{N}$ for $i = 1, \ldots, N$.

2. Repeat for $t$ = 1,2,..., $T$.

   I. Fit the classification technique (CART in our case) to the learning sample weighted by $\omega_t$ in order to obtain the classifier $C_t : \mathbf{x}_i \rightarrow y_i = \{1,2, \ldots, K\}$

   II. Compute the weighted error rate $\epsilon_t = P_{\omega_t}\{C_t(\mathbf{x}_i) \neq y_i\}$

$$\Longrightarrow \epsilon_t = \frac{\sum_{i=1}^{N} \omega_t(i).\, 1_{\{C_t(\mathbf{x}_i) \neq y_i\}}}{\sum_{i=1}^{N} \omega_t(i)}$$

   III. Compute; $\alpha_t = In\left(\frac{1-\epsilon_t}{\epsilon_t}\right) + In(K-1)$

   IV. Compute the new weight $\omega_{t+1}(i)$ for all the observations $i = 1, \ldots, N$

$$\omega_{t+1}(i) = \frac{\omega_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{If } y_i = C_t(\mathbf{x}_i) \\ e^{\alpha_t} & \text{If } y_i \neq C_t(\mathbf{x}_i) \end{cases}$$

3. Assign a new observation $\mathbf{x}_i^{new} \in R^P$ to the class $y_i \in \{1,2, \ldots, K\}$ that get the most weighted votes :

$$C_{boost}(\mathbf{x}_i^{new}) = \underset{y_i}{arg\,max} \sum_{t=1}^{T} \alpha_t.\, 1_{C_t(\mathbf{x}_i^{new}) = y_i}$$

---

**6.6 Conclusion**

It can be concluded from the above that applying the bagging, random forests and boosting procedures to CART destroys its interpretability appeal. This is because there will not exist a single 'combined' tree to interpret afterwards. Nevertheless, the three procedures tapped into in this chapter have a great potential to improve the predictive capabilities of an unstable classifier such as CART.

# CHAPTER 7

# 7. Applications and Results

## 7.1 Introduction

This chapter makes use of a publicly available dataset that was compiled by Hofmann (1994) to demonstrate the concepts that we have been discussing in the previous chapters. The dataset is widely known as the 'German credit data'. The discussion that follows essentially attempts to fulfill the research objectives of this study as outlined in chapter one. Firstly, a description and preliminary analysis of the dataset is provided in section (7.2). Sections (7.3) to (7.10) then focus on the application of each of the classification techniques to the credit related dataset and the results obtained. In section (7.11), a summary and comparison of the results will be provided before offering our conclusion in section (7.12). The analysis is done using the following software packages: SPSS (version 21), R-programming language (version 2.5.1) and Microsoft Excel (2010 edition).

## 7.2 The dataset and preliminary analysis

The German credit dataset used to build the scorecards in this study consists of 1000 past credit applicants classified as either non-defaulters, denoted by '0', or defaulters, denoted by '1'. There are 700 non-defaulters and 300 defaulters in the dataset. Each of these credit applicants has twenty measured characteristics, which are displayed in Table 7.1 (page 93). We randomly split the dataset into a 'learning sample' and a 'testing sample' in the ratio 0.7:0.3 respectively.

The subsequent learning sample contains 489 non-defaulters and 211 defaulters whilst the testing sample contains 211 non-defaulters and 89 defaulters. All the scorecards in this study are developed using the learning sample. The testing sample is reserved for evaluating (or testing) the predictive capabilities of the developed scorecards. Testing sample based results give us an indication of how the developed scorecards will perform in classifying new credit applicants.

Table 7.1: Characteristics of credit applicants

| | **Characteristic** | **Abbreviation** |
|---|---|---|
| 1 | balance of current account | bankbal |
| 2 | duration of loan | durloan |
| 3 | payment of previous credits | payprevdebt |
| 4 | purpose of credit | purcred |
| 5 | loan amount | loanamt |
| 6 | values of savings or stock | savings |
| 7 | time employed | timeempl |
| 8 | instalment in percentage of available income | instalmnt |
| 9 | value of asset | valasset |
| 10 | age | age |
| 11 | further running credits | curcred |
| 12 | foreign worker | alien |
| 13 | house ownership | hsetype |
| 14 | number of previous credits at the bank | histcred |
| 15 | occupation | jobtype |
| 16 | number of dependents | dependents |
| 17 | sex/marital status | marital.sex |
| 18 | duration in current house | durhse |
| 19 | guarantor | guarantor |
| 20 | telephone ownership | tel |

There are seventeen (17) categorical predictor variables in the dataset that were assigned an appropriate ordinal value or rank according to a method that is outlined in Appendix B. The average rank of each of the 17 categorical predictor variables in the learning sample is plotted in Figure 7.1 as a function of non-defaulters and defaulters.

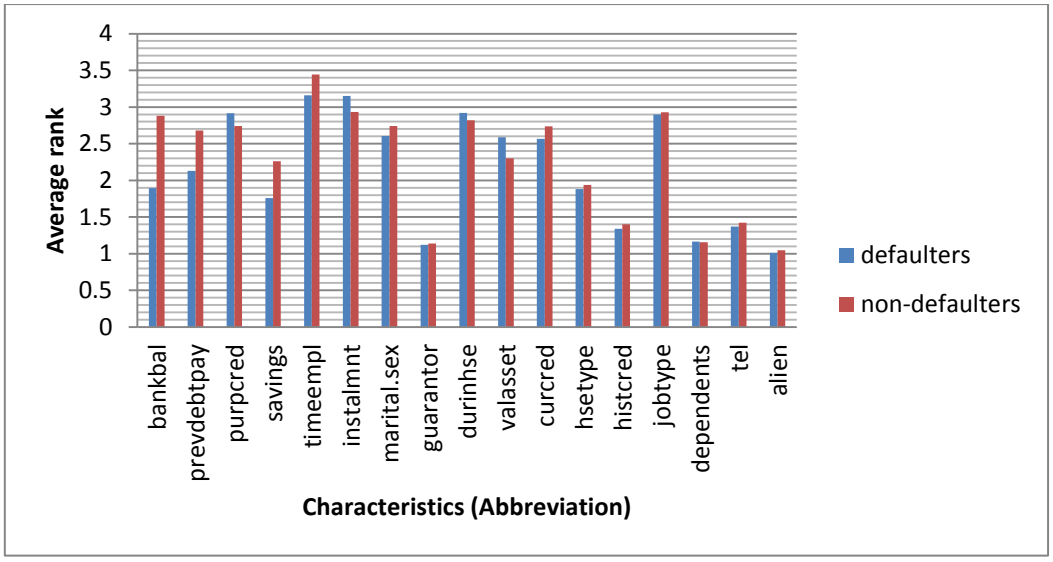


Figure 7.1: The average rank of categorical predictor variables as a function of defaulters and non-defaulters

A plot of the absolute difference between the average rank assigned to the applicants classified as defaulters and those classified as non-defaulters (in ascending order) for each of the categorical predictor variables in Figure 7.1 is given in Figure 7.2.

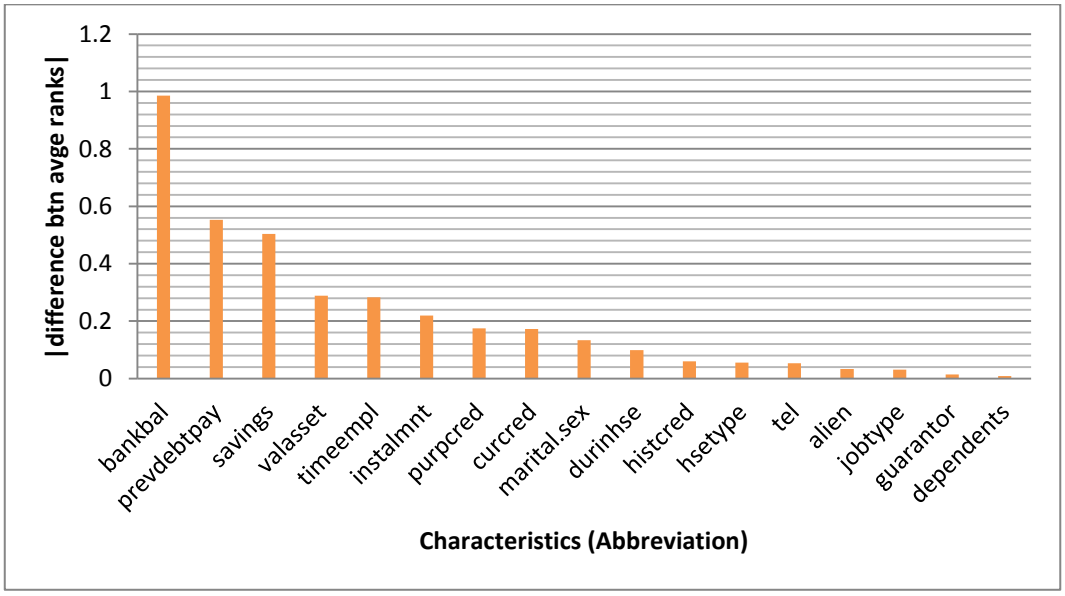


Figure 7.2: Variable importance as measured by absolute value of the difference between average ranks of non-defaulters and defaulters

Intuitively, the bigger the absolute difference between the average ranks of defaulters and non-defaulters for a given categorical predictor variable, the more 'important' it may be in distinguishing between non-defaulters and defaulters.

The remaining three continuous predictor variables are; duration of loan in months, amount of the loan in German deutsche marks (DM) and age in years. Figures 7.3 to Figure 7.5 are box plots of the distribution of these three continuous predictor variables as a function of non-defaulters (0) and defaulters (1).

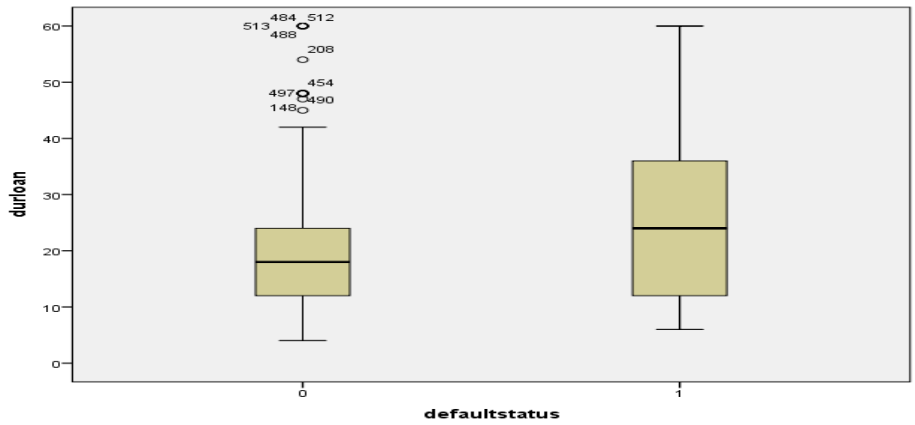


Figure 7.3: Duration of loan (in months) boxplot

As expected, Figure 7.3 reveals that the longer duration based loans are associated with defaulters as evidenced by the higher median value, higher third quartile value and a longer upper tail with a maximum value at 60 months. On the other hand, shorter duration loans are associated with non-defaulters with a few outlier applicants having longer duration of loans. Outlier applicants are those whose behavior deviates significantly from the other applicants in the same group (for example applicants numbered 484,512,513).

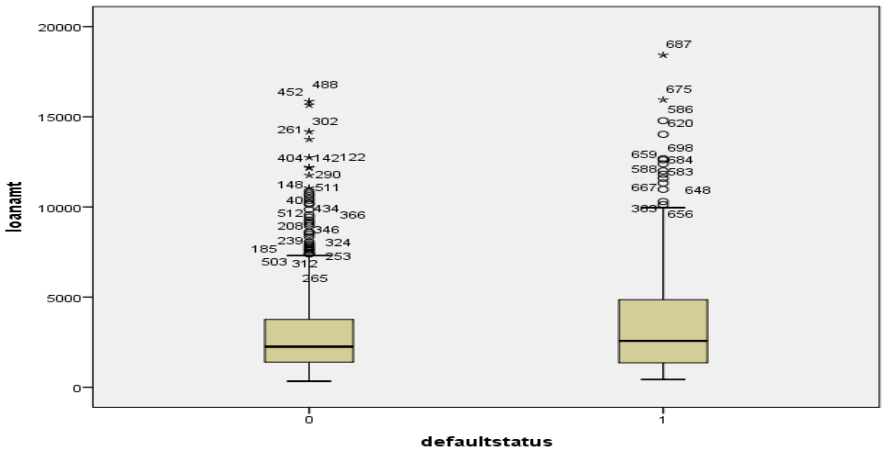


Figure 7.4: Loan amount in DM boxplot

In Figure 7.4, we observe that the median amount of loan taken by both defaulters and non-defaulters is similar. However, defaulters have a larger third quartile value and a longer upper tail compared to non-defaulters. This is also anticipated because those applicants who borrow large amounts of money are likely to default.

Focusing on the age boxplot in Figure 7.5 below, the distribution of the data in the two groups is similar, with older people inclined towards being non-defaulters as evidenced by the slightly higher median value.

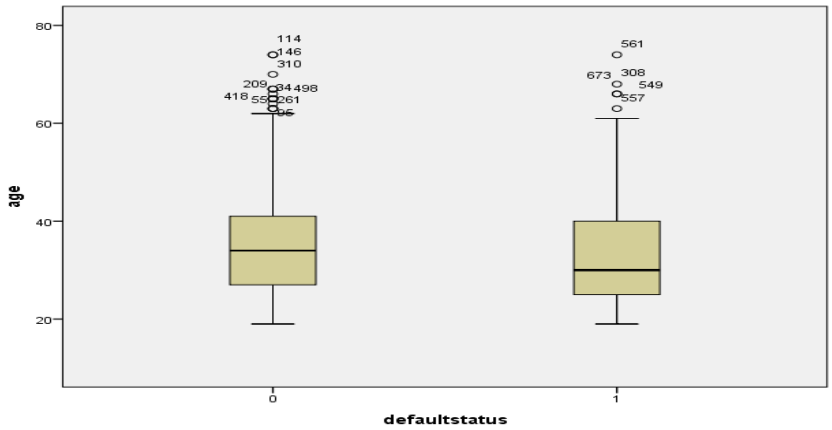


Figure 7.5: Age (in years) box plot

Notably, the box plots for loan amount and age have numerous outliers (especially loan amount in Figure 7.4) probably because there was a wide variation in the loan amount and the age of the applicants in our dataset or it signifies measurement errors.

Throughout the development of the scorecards in the following sections, the misclassifications costs are presumed to be constant and equal to one (1) (see, section (3.2.3) and section (5.2.5)). In addition, the prior probabilities are computed from the learning sample according to the formula given in equation (3.5), giving the results shown in Table 7.2.

Table 7.2: Prior Probabilities for Groups

| Class | Number of Cases | Prior Probability |
|---|---|---|
| 0 | 489 | 0.699 |
| 1 | 211 | 0.301 |
| Total | 700 | 1 |

### 7.3 Linear discriminant analysis

The development of scorecards in this section follows the theory that we have outlined in chapter three. One major assumption is that the observations in class $k \in (0,1)$ follow a 20-dimensional multivariate normal distribution with equal population covariance matrices.

#### 7.3.1 Bayesian approach

Using SPSS to implement the Bayesian approach that is outlined in section (3.2), the discriminant functions coefficients for a non-defaulter ($k = 0$) and a defaulter ($k = 1$) in Table 7.3 were produced.

Table 7.3: Linear discriminant function coefficients

| **Predictor variables** | | bankbal | durloan | prevdebtpay | purpcred | loanamt | savings | timeempl | instalmnt | marital.sex | guarantor | durinhse | valasset | age | curcred | hsetype | jobtype | dependents | tel | alien | (constant) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **coefficients** | $k = 0$ | 2.083 | 0.122 | 0.997 | 0.863 | -0.001 | 0.548 | 0.156 | 1.288 | 4.092 | 6.069 | 5.625 | 1.934 | 6.781 | 6.933 | 5.625 | 6.781 | 7.927 | 1.475 | 34.428 | -72.636 |
| | $k = 1$ | 1.451 | 0.149 | 0.578 | 0.817 | -0.001 | 0.371 | -0.024 | 1.614 | 3.857 | 5.777 | 1.988 | 1.338 | 0.050 | 6.656 | 5.143 | 6.612 | 8.133 | 1.251 | 33.579 | -68.736 |

The dimensions in Table 7.3 give rise to the following discriminant functions:

$$d_0(\mathbf{x}_i) = 2.083\text{bankbal} + 0.122\text{durloan} + \cdots + 34.428\text{alien} - 72.636 \qquad (7.1)$$

$$d_1(\mathbf{x}_i) = 1.451\text{bankbal} + 0.149\text{durloan} + \cdots + 33.579\text{alien} - 68.736 \qquad (7.2)$$

where, $\mathbf{x}_i = \{\text{bankloan}, \text{durloan}, \dots, \text{alien}\} \in R^{20}$ are the characteristics of the $i^{th}$ applicant as given in Table 7.1.

Therefore, a new credit applicant $\mathbf{x}_i^{new} \in R^{20}$ will be classified as a non-defaulter (0) if we have:

$$d_0(\mathbf{x}_i^{new}) > d_1(\mathbf{x}_i^{new}) \Longrightarrow d_0(\mathbf{x}_i^{new}) - d_1(\mathbf{x}_i^{new}) > 0$$

$$\Longrightarrow 0.632\text{bankbal} - 0.027\text{durloan} + \cdots + 0.849\text{alien} - 3.9 > 0$$

$$\Longrightarrow 0.632\text{bankbal} - 0.027\text{durloan} + \cdots + 0.849\text{alien} > 3.9 \quad (7.3)$$

Otherwise, the new applicant will be classified as a defaulter(1). An application of the classification rule (7.3) to applicants in our testing sample produced the classification matrix that is given in Table 7.4. The table shows that we have managed to achieve an error rate of $\frac{(21+50)}{300} = 0.237$, a sensitivity of 43.8% and a specificity of 90%.

Table 7.4: Testing sample classification matrix for the Bayesian LDA

| | | | Predicted Group Membership | | |
|---|---|---|---|---|---|
| | | defaultstatus | 0 | 1 | Total |
| Original | Count | 0 | 190 | 21 | 211 |
| | | 1 | 50 | 39 | 89 |
| | % | 0 | 90.0 | 10.0 | 100.0 |
| | | 1 | 56.2 | 43.8 | 100.0 |

### 7.3.2 Fisher's approach

Using SPSS to implement Fisher's LDA on the learning sample produced only one canonical discriminant function (because the response variable is binary) whose coefficients are given in Table 7.5.

Table 7.5: Unstandardized Canonical Discriminant Function Coefficients

| **Predictor variables** | bankbal | durloan | prevdebtpay | purpcred | loanamt | savings | timeempl | instalmnt | marital.sex | guarantor | durinhse | valasset | age | curcred | hsetype | jobtype | dependents | tel | alien | (constant) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Coefficients (unstandardized)** | 0.527 | -0.023 | 0.349 | 0.038 | 0.000 | 0.147 | 0.150 | -0.272 | 0.196 | 0.244 | -0.040 | -0.162 | -0.003 | 0.230 | 0.402 | 0.141 | -0.172 | 0.187 | 0.708 | - 4.191 |

The dimensions in Table 7.5 give rise to the following Fisher's canonical discriminant function:

$$d(\mathbf{x}_i) = 0.527bankbal - 0.23durloan + \cdots + 0.708alien - 4.191 \qquad (7.4)$$

where, $\mathbf{x}_i = \{\text{bankloan}, \text{durloan}, \ldots, \text{alien}\} \in R^{20}$ are the characteristics of the $i^{th}$ applicant.

The canonical discriminant function (7.4) can now be used to calculate a discriminant score $d(\mathbf{x}_i)$ for each applicant $\mathbf{x}_i$ in our learning sample. Table 7.6 show a mean value for these scores that we have obtained for the non-defaulters ($k = 0$) and defaulters ($k = 1$) in our learning sample.

Table 7.6: Class means scores

| Class | Mean score |
|---|---|
| $k = 0$ | 0.361 |
| $k = 1$ | -0.838 |

According to Table 7.6, if a new applicant's discriminant score $d(\mathbf{x}_i)$ lies close to $-0.838$, then he/she is more likely to be a defaulter than a non-defaulter. Conversely, if the score lies closer to 0.361, the applicant is more likely to be a non-defaulter than a defaulter. In practice, a cut-off point for distinguishing between these two classes is taken to be the average between the individual groups' mean scores:

$$c = \frac{(-0.838 + 0.361)}{2} = -0.2385 \qquad (7.6)$$

such that a new applicant $\mathbf{x}_i^{new} \in R^{20}$ will be classified as a non-defaulter (0) if:

$$0.527bankbal - 0.23durloan + \cdots + 0.708alien - 4.191 > -0.2385$$

$$\Leftrightarrow 0.527bankbal - 0.23durloan + \cdots + 0.708alien > 3.9525 \qquad (7.7)$$

Otherwise, the applicant will then be classified as a defaulter (1). Applying the classification rule (7.7) to applicants in our testing sample produced the classification matrix that is given in Table 7.7, which shows that we managed attain an error rate of $\frac{(25+57)}{300} = 0.273$, a sensitivity of 71.9% and a specificity of 73%.

Table 7.7: Testing sample classification matrix for Fisher's LDA

| | | | Predicted Group Membership | | |
|---|---|---|---|---|---|
| | | defaultstatus | 0 | 1 | Total |
| Original | Count | 0 | 64 | 25 | 89 |
| | | 1 | 57 | 154 | 211 |
| | % | 0 | 71.9 | 28.1 | 100.0 |
| | | 1 | 27.0 | 73.0 | 100.0 |

A link with the Bayesian approach can be made if we specify equal probabilities for our two classes. According to the cut-off value in equation (3.12) of section (3.2.2), if the prior probabilities were equal, the constant $c = -3.9$ in the decision rule (7.3) would decrease by a factor:

$$\ln\left(\frac{\hat{\pi}_0}{\hat{\pi}_1}\right) = \ln\left(\frac{0.699}{0.301}\right) \cong 0.84$$

since we will now have $\hat{\pi}_0 = \hat{\pi}_1 = 0.5$. This means that the constant in the decision rule (7.3) would change to: $c = -3.9 - 0.84 = -4.74$. Thus, if prior probabilities were presumed to be equal in the Bayesian classifier (7.3), one would classify a new applicant $\mathbf{x}_i^{new}$ as a non-defaulter(0) if:

$$0.632\text{bankbal} - 0.027\text{durloan} + \cdots + 0.849\text{alien} - 4.74 > 0$$

$$\Leftrightarrow 0.632\text{bankbal} - 0.027\text{durloan} + \cdots + 0.849\text{alien} > 4.74 \tag{7.8}$$

Multiplying equation (7.7) by the proportionality constant $\alpha \approx 1.199$ will result in the same decision rule as the one in equation (7.8). This verifies our more formal proof in section (3.3.3) which states that; the classification rule that has been derived under the Bayesian approach becomes equivalent to Fisher's class allocation rule when we assume equal prior probabilities.

Following on our discussion at the end of section (3.3.2), to make Fisher's classification function (7.7) the same with the Bayesian based classification function (7.3) that

assumes unequal prior probabilities, one can scale the cut-off point $c = 3.9$ in the Bayesian classifier (7.3) to,

$$c = \frac{3.9}{1.199} = 3.2527 \tag{7.9}$$

and use it for the classification function (7.7). Thus, a new applicant $\mathbf{x}_i^{new} \in R^{20}$ will now be classified as a non-defaulter (0) if

$$0.527bankbal - 0.23durloan + \cdots + 0.708alien > 3.2527 \tag{7.10}$$

Otherwise, $\mathbf{x}_i^{new}$ will be classified as a defaulter (1).

It is important to note that multiplying Fisher's classification function (7.10) by our proportionality constant $\alpha \approx 1.199$ gives the Bayesian based classification function (7.3). The modification done in (7.9) is often used to adjust the cut-off point computed in (7.6) in order to create a cut-off point that takes into account the fact that the two groups may be unequal in size. Because the two groups used in this study are unequal in size (i.e. proportion of non-defaulters greater than proportion of defaulters), for the remainder of the study we shall use the results in Table 7.4 as the working classification matrix of the LDA classifier when applied to the testing sample.

### 7.3.3 Optimal scoring approach

Following the procedure that has been outlined in Table 3.2 of section (3.5), we obtained only one optimal regression function coefficients (because the response variable is binary) that appear in Table 7.8.

Table 7.8: Optimal scoring based canonical discriminant function coefficients

| **Predictor variables** | bankbal | durloan | prevdebtpay | purpcred | loanamt | savings | timeempl | instalmnt | marital.sex | guarantor | durinhse | valasset | age | curcred | hsetype | jobtype | dependents | tel | alien | (constant) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Coefficients (unstandardized)** | 0.2231 | -0.0097 | 0.1478 | 0.0162 | 0.0000 | 0.0624 | 0.0636 | -0.1150 | 0.0828 | 0.1032 | -0.0169 | -0.0685 | -0.0012 | 0.0975 | 0.1702 | 0.0598 | -0.0728 | 0.0793 | 0.2999 | -1.7742 |

A comparison of Table 7.5 and Table 7.8 reveals that multiplying the optimal scoring based unstandardized coefficients in Table 7.8 by a proportionality constant $\alpha \approx 2.3657$ gives Fishers' LDA unstandardized coefficients in Table 7.5. This verifies the more formal proof in section (3.5.3) which shows that Fisher's and the optimal scoring approach to LDA produce the same first eigenvector (which is the vector containing the discriminant coefficients). Therefore, proceeding in a similar manner as outlined in the previous section for Fisher's LDA will produce the same classification results since according to the proof given in equation (3.37) it is only the direction of the vector that matters rather than its magnitude.

### 7.3.4 Judging variable importance

The information in Table 7.9 includes standardized canonical discriminant coefficients that can be used to determine the unique contribution that is being made by an individual predictor variable in the multivariate model.

Table 7.9: Standardized discriminant function coefficients

| **Predictor variables** | bankbal | durloan | prevdebtpay | purpcred | loanamt | savings | timeempl | instalmnt | marital.sex | guarantor | durinhse | valasset | age | curcred | hsetype | jobtype | dependents | tel | alien |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Coefficients (standardized)** | 0.620 | -0.279 | 0.368 | 0.104 | -0.128 | 0.231 | 0.177 | -0.298 | 0.133 | 0.110 | -0.044 | -0.167 | -0.031 | 0.160 | 0.209 | 0.092 | -0.063 | 0.092 | 0.134 |

A plot of the absolute values of the standardized discriminant coefficients, in decreasing order, is provided in Figure 7.6. The greater the absolute value of the standardized discriminant coefficients, the more important the predictor variable is being considered to include in one's classification algorithm.

Figure 7.6 reveals that an applicant's bank balance (bankbal) is considered as being a very important characteristic to include in the classification model. One the other hand, the age of an applicant is not considered as a very 'important' characteristics when attempting to distinguish defaulters from non-defaulters.

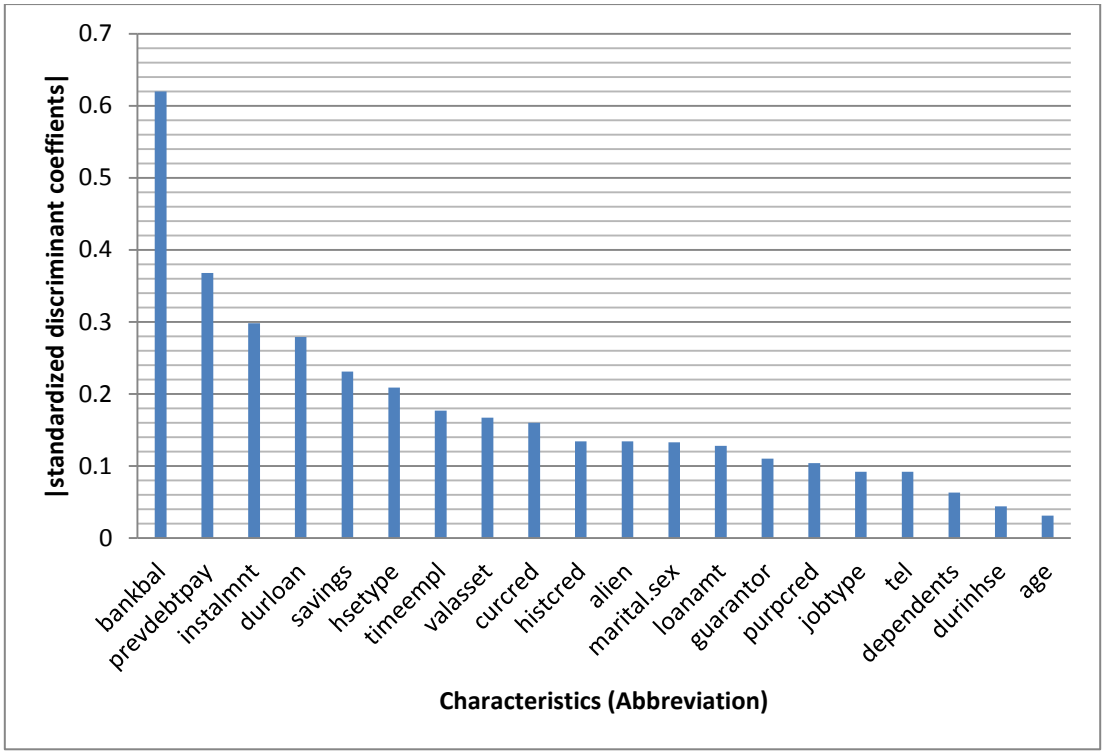


Figure 7.6: Ranking variable importance using absolute values of standardized canonical coefficients

### 7.4 Quadratic discriminant analysis

The 'MASS' package contained in R was used to develop the QDA based scorecard using the learning sample. Table 7.10 shows the classification matrix that result from applying the developed QDA model to our testing sample applicants. The table shows that we have managed to realize an error rate of $\frac{(38+40)}{300} = 0.260$, a sensitivity of 57.3% and a specificity of 81.0%.

Table 7.10: QDA classification matrix for the testing sample

| | | | Predicted Group Membership | | |
|---|---|---|---|---|---|
| | | defaultstatus | 0 | 1 | Total |
| Original | Count | 0 | 171 | 40 | 211 |
| | | 1 | 38 | 51 | 89 |
| | % | 0 | 81.0 | 19.0 | 100.0 |
| | | 1 | 42.6 | 57.3 | 100.0 |

### 7.5 Flexible discriminant analysis

Flexible discriminant analysis (FDA) was implemented on the learning sample using the 'mda' package contained in R. In using the MARS procedure, the forward process stops when the change in the RSS of the model caused by adding a term is less than 0.001 and a backward pruning procedure is employed using a penalty value of $c = 2$ for the GCV criterion. Depending on the parameter $B$ that is being used by the MARS model to govern the degree of interaction of the hinge functions allowed, five FDA models where created. On applying the created models to the testing sample applicants, the results in Figure 7.7 were obtained.

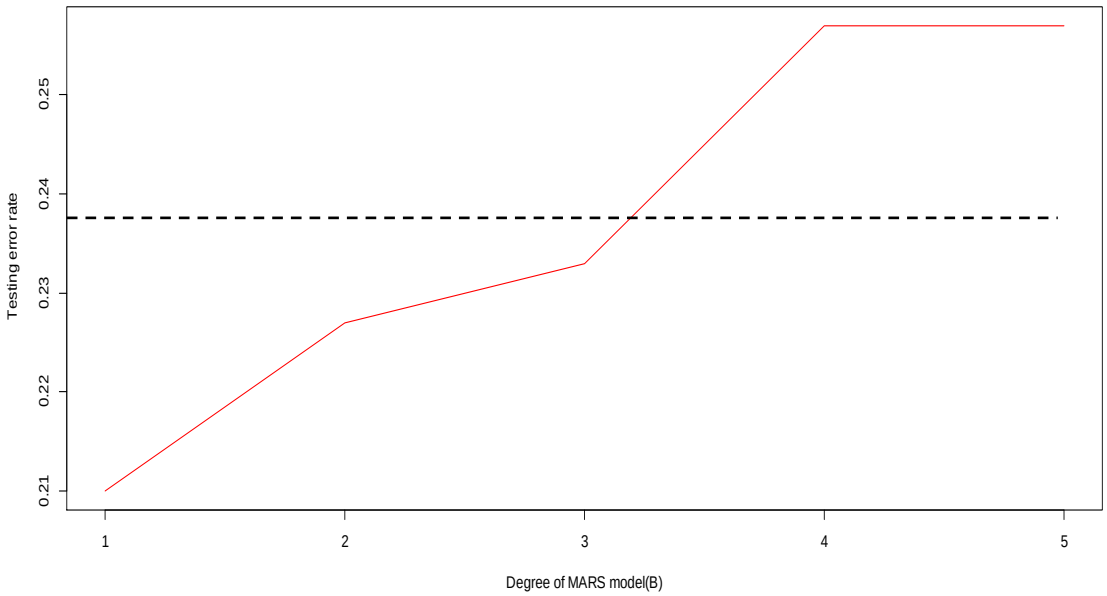


Figure 7.7: FDA models testing error rates

According to Figure 7.7, the FDA model with MARS functions of degree one (additive model) is the best. As the complexity of the MARS functions increases, the accuracy of the FDA model decreases and then remains constant for values B greater than or equal to four. The dashed reference line in Figure 7.7 at a value of 0.237 is the testing error rate of the FDA model that uses multivariate linear regression functions, which is equivalent to the optimal scoring approach to LDA. Figure 7.7 suggests that the FDA models with MARS functions one, two and three are the most appropriate because they improve upon the testing error rate of the LDA classifier we have developed.

Table 7.11 contains a summary of the error rate, sensitivity and specificity of these appropriate models when applied to testing sample applicants.

Table 7.11: FDA models performance

| Model | Error rate | Sensitivity | Specificity |
|---|---|---|---|
| FDA (MARS, B=1) | 0.210 | 46.1% | 92.9%. |
| FDA (MARS, B=2) | 0.227 | 51.7% | 88.2%. |
| FDA (MARS, B=3) | 0.233 | 49.4% | 88.2% |

### 7.6 Mixture discriminant analysis

Mixture discriminant analysis (MDA) was implemented on the learning sample using the 'mda' package contained in R. Following the procedure that has been outlined in Algorithm 4.3, we obtained six mixture discriminant analysis (MDA) models by varying the number of latent subclasses per class from one to six. On applying these models to our testing sample applicants, we obtained the results in Figure 7.8 depending on the number of latent subclasses per class specified in the model.

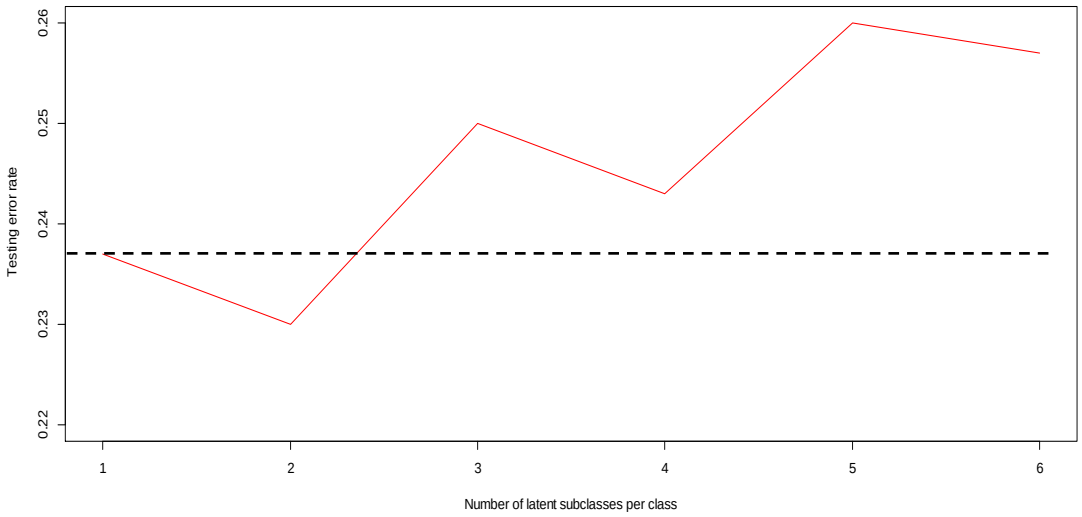


Figure 7.8: MDA using Optimal Scoring (multivariate linear regression functions)

It is important to note that the MDA model with a single group centroid (one latent subclass per class) is the LDA model produced by optimal scoring, which corresponds to the dashed reference line in Figure 7.8 at a testing error rate value of 0.237. We select the MDA model with two latent subclasses per class as the most appropriate model, because, it improves the testing error rate of the LDA classifier we have developed.

Table 7.12 shows the classification matrix when the testing sample applicants are classified using the MDA (2 subclasses, linear regression) model we have chosen to be the most appropriate. The table reveals that we have managed to get a testing error rate of $\frac{(46+23)}{300} = 0.23$, a sensitivity of 48.3% and specificity of 89.1%.

Table 7.12: Testing sample classification matrix for the MDA (2 subclasses, linear regression) model

| | | | Predicted Group Membership | | |
|---|---|---|---|---|---|
| | | defaultstatus | 0 | 1 | Total |
| Original | Count | 0 | 188 | 23 | 211 |
| | | 1 | 46 | 43 | 89 |
| | % | 0 | 89.1 | 10.9 | 100.0 |
| | | 1 | 51.7 | 48.3 | 100.0 |

Figure 7.9 is a plot of observations in the learning sample against the first two canonical discriminant functions (since they account for most of the variation in the data) from the MDA (2 subclasses, linear regression) model we have chosen to be most appropriate. Non-defaulters (0) are plotted in red and defaulters (1) in green. The latent subclass centroids are circled.

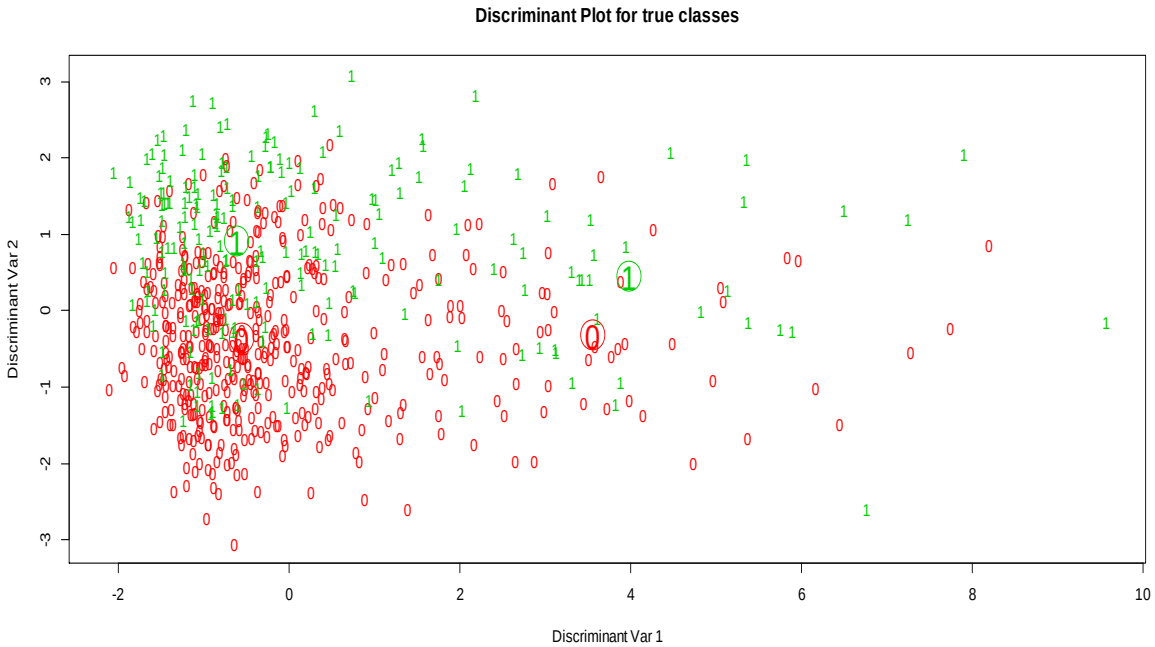


Figure 7.9: Plot of the learning sample using MDA coordinates

## 7.7 Classification and regression trees

The stages in the development of a classification tree for credit scoring using a real life credit-related dataset is illustrated in this section, as outlined in chapter five. The Classification and Regression Tree (CART) procedure is implemented on the learning sample using the 'Rpart' package contained in R.

### 7.7.1 Growing the tree

The unpruned classification tree in Figure 7.10 was constructed from the learning sample using the method outlined in section (5.2).

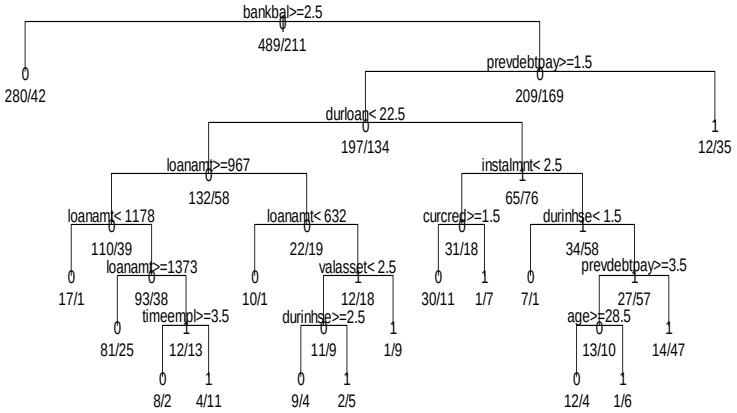


Figure 7.10: Unpruned credit scoring classification tree

The Gini index was used as the impurity function. The standard set of questions used to split each non-terminal node is shown. If the response to the splitting rule is affirmative, the case is assigned to the left child node. If otherwise, a case is assigned to the right child node. Since it is not known which node will become terminal after pruning, each node is assigned a class label according to which group is predominant. For example the root node contains 489 non-defaulters (0) and 211 defaulters (1); therefore, it is assigned the class label '0'.

A node becomes terminal if the change in the Gini index at node *t* caused by the making the split $s^*$ is less than a factor, *cp=0.01*, where *cp* is the cost-complexity parameter. In addition, a node is not split further if it contains less than 20 applicants. Before we use the tree to classify new observations, it is advisable to prune it first in order to remove unimportant branches.

### 7.7.2 Pruning the Tree

Table 7.13 shows the cost-complexity parameter (CP) value, number of splits and the relative error for each of the subtrees $T_0 > T_1 > T_2 > \cdots > T_6$ obtained from cost-complexity pruning.

Table 7.13: Cost-Complexity pruning

| Subtree | CP-value | Number of splits | Relative error |
|---|---|---|---|
| 6 | >0.054502 | 0 | 1 |
| 5 | 0.054502 | 4 | 0.77725 |
| 4 | 0.028436 | 6 | 0.72038 |
| 3 | 0.018957 | 8 | 0.68246 |
| 2 | 0.014218 | 10 | 0.65403 |
| 1 | 0.011848 | 12 | 0.65033 |
| 0 | 0.01 | 15 | 0.59716 |

The number of leaves (or terminal nodes) is obtained by adding one to the number of splits. The relative error is the standardized re-substitution error estimate of each of the subtrees such that the root node has an error rate of one. Since the root node makes 211 out of 700 misclassifications, we multiply the relative errors by 211 to get the total number of misclassifications for a particular subtree.

The CP table is printed from the smallest tree with no splits (root node) to the largest tree with 15 splits (16 terminal nodes). The pruned subtrees associated with the CP values in Table 7.13 are shown in Figures 7.11 to 7.15. Note that subtree six($cp > 0.054502$) is just a root node and subtree zero ($cp = 0.01$) is the original tree in Figure 7.10. All the subtrees are nested around the original tree.

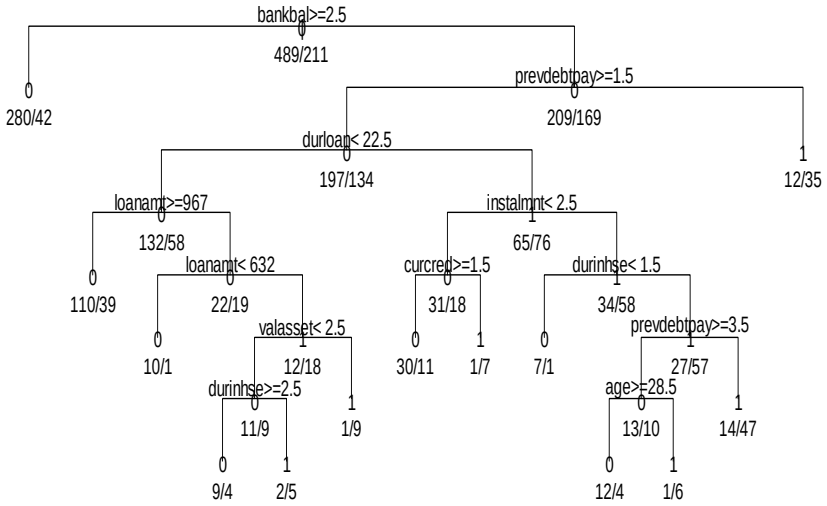


Figure 7.11: Subtree 1 with CP = 0.011848

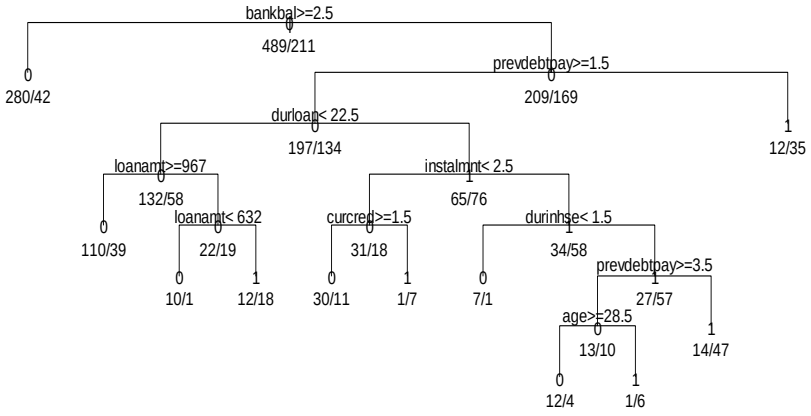


Figure 7.12: Subtree 2 with CP = 0.014218


Figure 7.13: Subtree 3 with CP = 0.018957

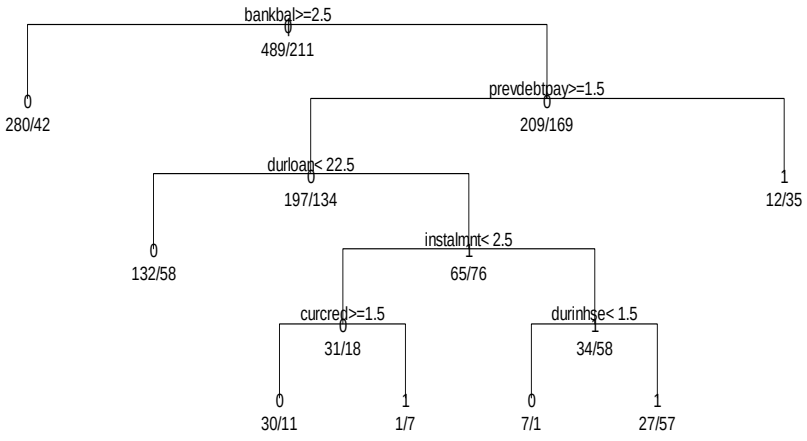


Figure 7.14: Subtree 4 with CP = 0.028436

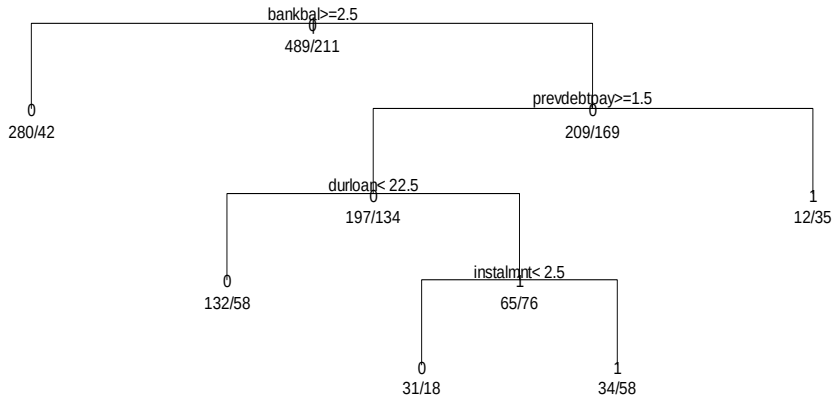


Figure 7.15: Subtree 5 with CP = 0.054502

### 7.7.3 Selecting the Optimal Tree

Testing sample validation was used to determine the optimal size of the tree. A plot of the change in the resubstitution error estimate as the number of terminal nodes (size of the tree) increases is shown in Figure 7.16.

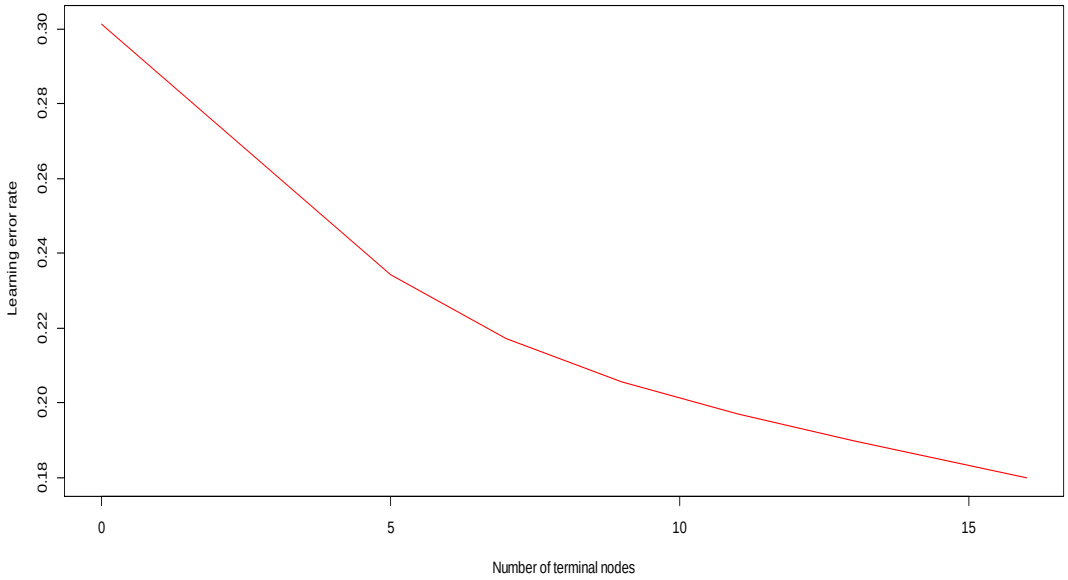


Figure 7.16: Evolution of the learning error rate against number of terminal nodes

Figure 7.16 indicates that as the size of the tree increases, the classification error rate decreases monotonically for the learning sample. This corresponds to the proof of the theorem in section (5.2.4) which shows that a large tree will always give the best fit to the learning/training dataset.

In contrast, a plot of the change in the classification error rate when classifying applicants in the testing sample as the size of the tree increases that is given Figure 7.17 shows that the classification error rate decreases sharply. It then starts oscillating in a zigzag manner. We select the tree with five terminal nodes, subtree five in Figure 7.15, as the optimal size of the tree because it gives the lowest testing error rate.

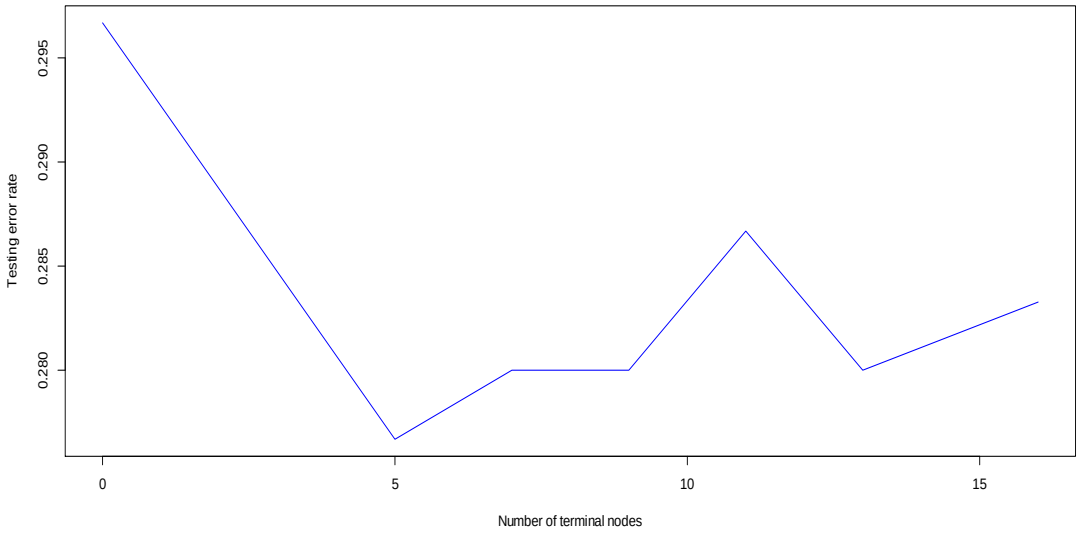


Figure 7. 17: Evolution of the testing error rate against number of terminal nodes

### 7.7.4 Scoring new credit applicants

The chosen optimal tree is shown in Figure 7.18 and below can be used to classify new applicants as follows:

1. If a new applicant is assigned a ‘bank balance’ rank that is greater than or equal to 2.5, immediately classify the applicant as a non-defaulter (0);

2. If a new applicant is assigned a ‘bank balance’ rank that is less than 2.5 and a ‘repayment of previous debts’ rank that is less than 1.5, classify the applicant as a defaulter (1);

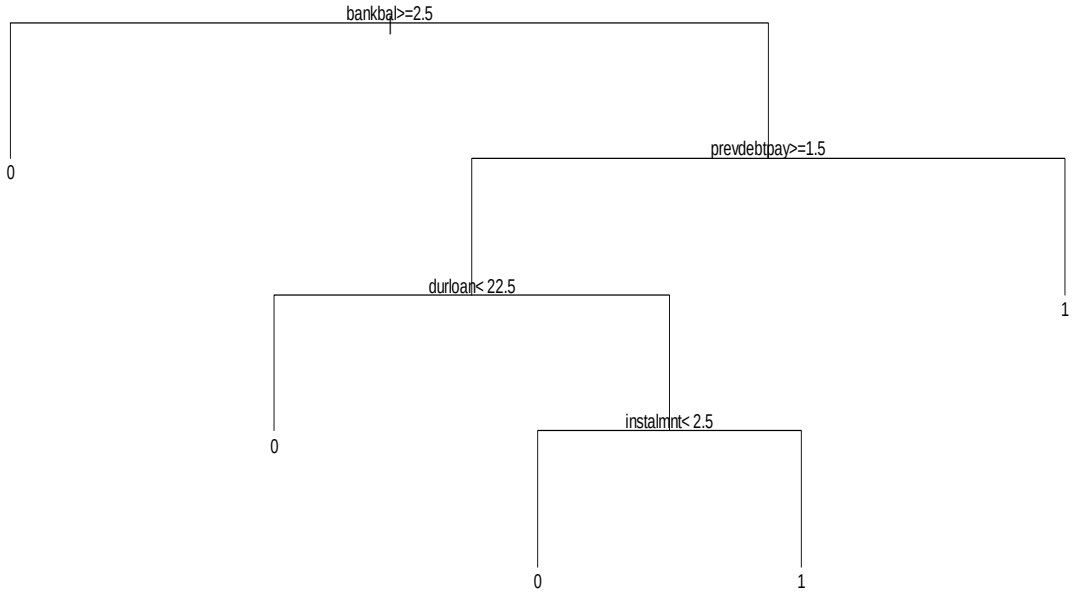


Figure 7.18: Optimal classification tree for scoring new credit applicants

3. Otherwise, if a new applicant is assigned a ‘bank balance’ rank less than 2.5, a rank greater than or equal to 1.5 for ‘payment of previous debts’ and the ‘duration of the loan’ the applicant require is less than 22.5 months, classify the applicant as a non-defaulter (0). However, if the ‘duration of the loan’ the same applicant requires is more than or equal to 22.5 months and the ‘instalment’ rank is less than 2.5, classify the applicant as a non-defaulter (0). Otherwise, if the ‘instalment’ rank is greater than or equal to 2.5, the applicant is classified as a defaulter (1).

One obvious advantage of the CART based scorecard is its simplicity. Furthermore, the optimal decision tree model in Figure 7.18 makes decisions based on only four out of the twenty-predictor variables. This is a huge dimension reduction, which results in decisions being reached quickly.

Applying the testing sample to the optimal tree above produced the classification matrix in Table 7.14.

Table 7.14: Classification matrix for classifying testing sample applicants using the optimal classification tree

| | | | Predicted Group Membership | | |
|---|---|---|---|---|---|
| | | defaultstatus | 0 | 1 | Total |
| Original | Count | 0 | 186 | 25 | 211 |
| | | 1 | 58 | 31 | 89 |
| | % | 0 | 88.2 | 11.8 | 100.0 |
| | | 1 | 65.2 | 34.8 | 100.0 |

The table above reveals that we have managed to accomplish a testing error rate of $\frac{(25+58)}{300} = 0.277$, a sensitivity of 34.8% and a specificity of 88.2%.

### 7.7.5 Judging variable importance

The overall improvement to the impurity of the optimal tree in Figure 7.18 that is attributed to each predictor variable in the learning algorithm is shown in Figure 7.19 below.

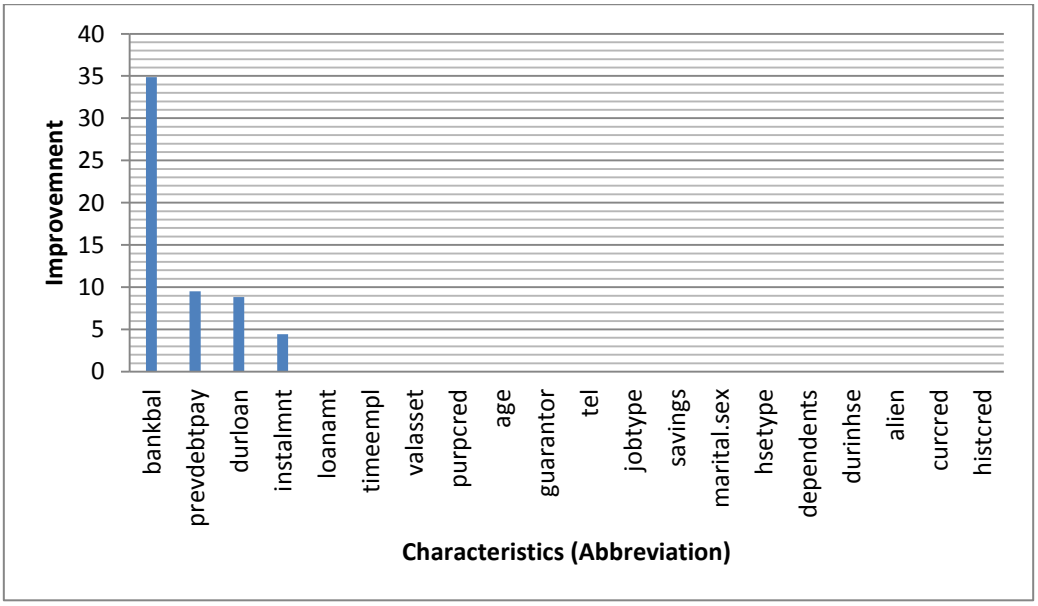


Figure 7.19: Ranking variable importance using CART

An applicant's bank balance is by far the most important predictor variable in the optimal classification tree. If the tree is allowed to grow bigger, then more predictor variables have a chance to play a role in the tree construction process and not receive zero improvement values.

## 7.8 Bagging

In this section, we use the 'adabag' package contained in R to implement the bagging procedure on the learning sample, using CART as the base classifier. In Figure 7.20, applicants in the testing sample are being classified using the bagging estimate given in equation (6.2) as more trees B are combined. The dashed reference line at a value of 0.283 is the testing error rate of the single unpruned classification tree in Figure 7.10.

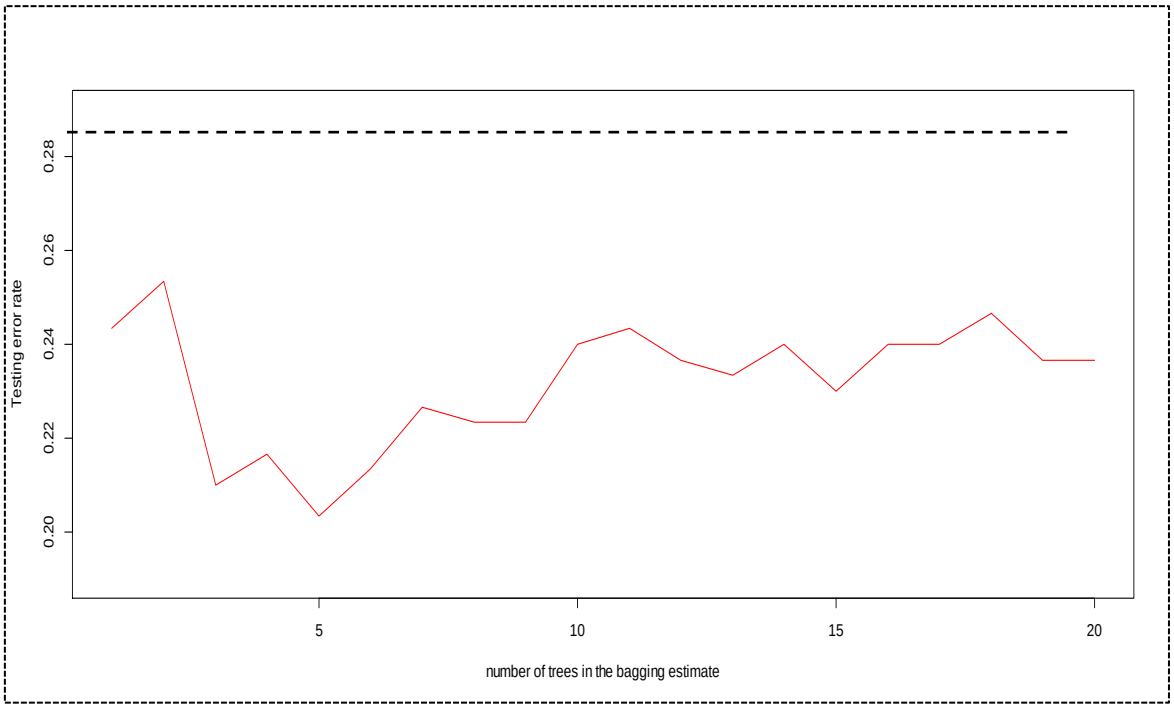


Figure 7.20: Evolution of the Testing error against number of trees

A visual inspection of Figure 7.20 shows that the bagging procedure significantly improved the accuracy of a single unpruned tree. It is suggested in Figure 7.20 that over-fitting occurs when more than five trees are used for the bagging estimate. The zigzag pattern that one observes in the evolution of the error rate can be attributed to the random nature in which the bootstrap samples are being generated. We select five trees as the optimal number of trees to 'bag' because that is when the testing error rate is lowest.

The classification matrix is shown in Table 7.15 when this optimal bagging estimate composed of five trees is used to classify applicants in the testing sample.

Table 7.15: Classification matrix of running the testing sample down the optimal bagged estimate

| | | | Predicted Group Membership | | |
|---|---|---|---|---|---|
| | | defaultstatus | 0 | 1 | Total |
| Original | Count | 0 | 198 | 13 | 211 |
| | | 1 | 48 | 41 | 89 |
| | % | 0 | 93.8 | 6.2 | 100.0 |
| | | 1 | 53.9 | 46.1 | 100.0 |

The classification matrix above shows that we managed to attain an error rate of $\frac{(13+48)}{300} = 0.203$, a sensitivity of 46.1% and a specificity of 93.8 %.

The figure below shows the average improvement, in descending order of importance, attributed to each of the predictor variables in the optimal bagging estimate.

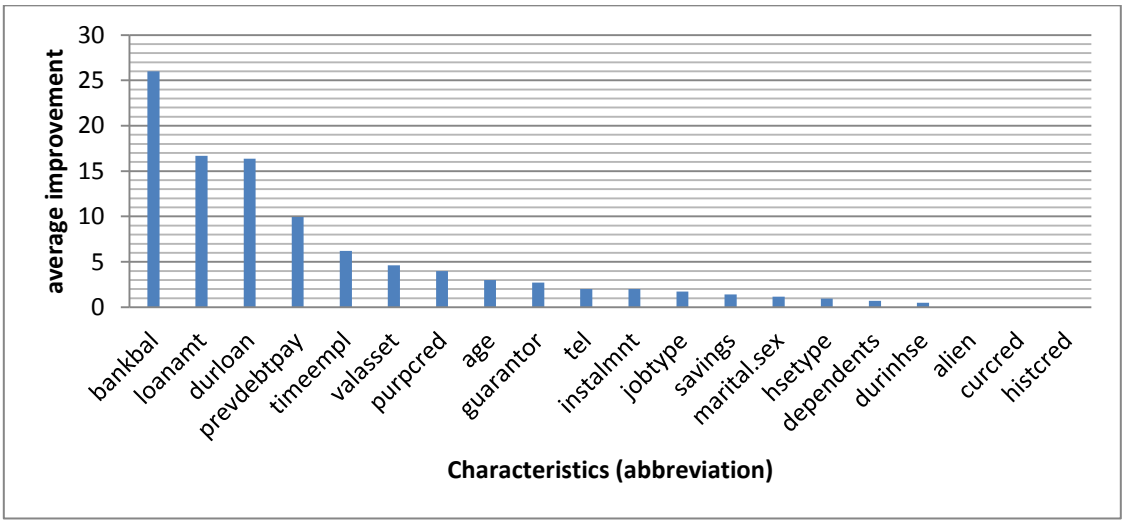


Figure 7.21: Ranking variable importance in the bagging estimate

### 7.9 Random Forests

The random forests procedure was implemented on the learning sample using the 'randomForest' package that is contained in R. The main tuning parameters are the size of the forest B and the number of predictor variables to consider at each split, *m*. Initially, default values of $B = 500$ trees and $m = \text{integer part}\left[\sqrt[2]{20}\right] = 4$ predictor variables are used to produce the error evolution in Figure 7.22.

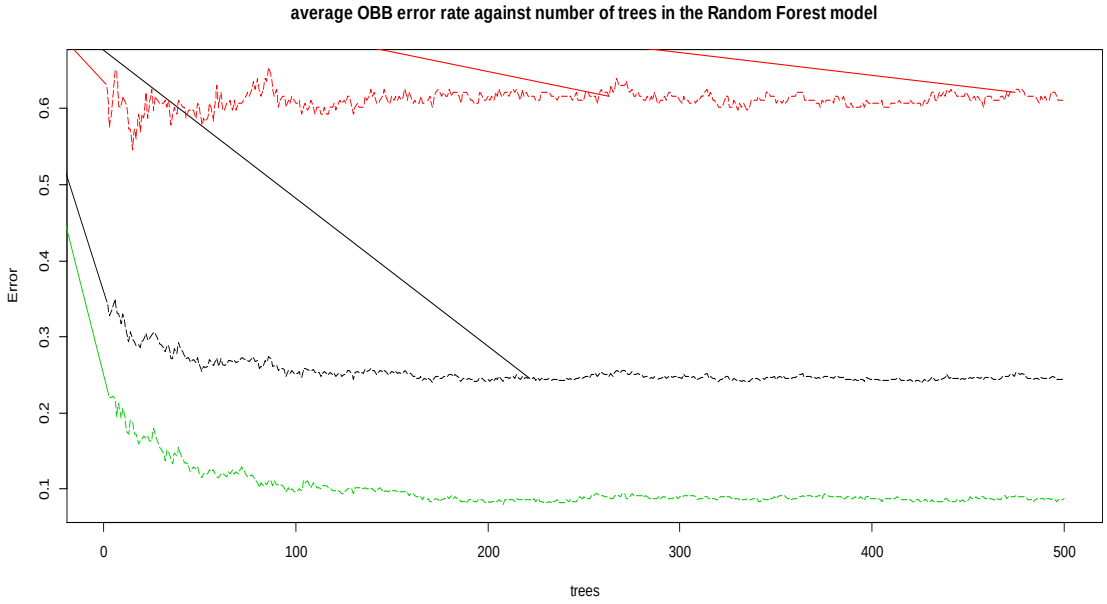


Figure 7.22: Bootstrap error rates against number of trees

The average 'out of bag' error rate (OOB-ER), traced by the middle black line, stabilizes at a value of approximately 0.2443 after about 200 trees are used in the random forests procedure. Consequently, we fix the number of trees for the random forests procedure at *B=200*. The top red line in Figure 7.22 traces the fraction of defaulters incorrectly classified as non-defaulters (a miss) which converges at an error rate of approximately 0.6066. The bottom green line traces the fraction of non-defaulters incorrectly classified as defaulters (a false alarm), which converges at an error rate of approximately 0.0879.

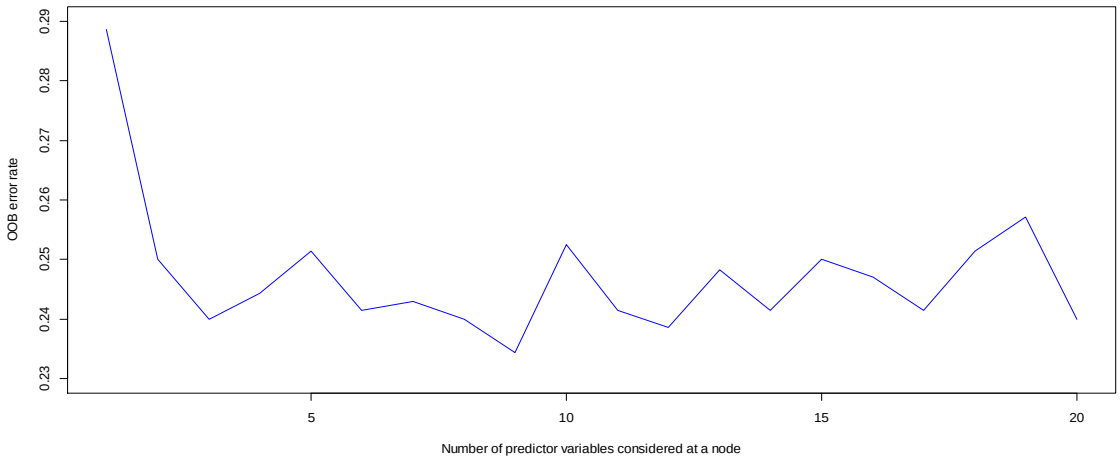


Figure 7.23: Change in average OOB error rate as the number of predictor variables selected at each node varies

Figure 7.23 shows that varying the values of $m$ from 1 to 20 with the size of the forest fixed at $B = 200$ reaches a minimum average OOB-ER estimate at a value $m = 9$. Consequently, we chose as appropriate the random forest model with 200 trees and 9 predictor variables at each node being randomly selected out the possible $p = 20$.

The classification of the applicants in the testing sample using a random forests model with parameter values $B = 200$ and $m = 9$ produced the classification matrix in Table 7.16. The table shows that we have managed to realize a testing error rate of $\frac{(26+48)}{300} = 0.2467$, a sensitivity of 46.1% and a specificity of 87.7%.

Table 7.16: Classification matrix of running the testing sample through the appropriate random forest model

| | | | Predicted Group Membership | | |
|---|---|---|---|---|---|
| | | defaultstatus | 0 | 1 | Total |
| Original | Count | 0 | 185 | 26 | 211 |
| | | 1 | 48 | 41 | 89 |
| | % | 0 | 87.7 | 12.3 | 100.0 |
| | | 1 | 53.9 | 46.1 | 100.0 |

A plot of variable importance is shown in Figure 7.24 below

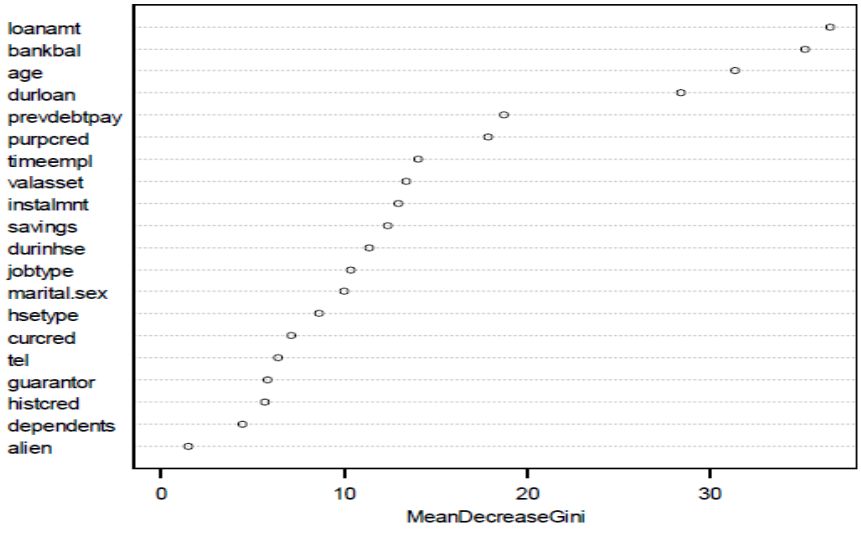


Figure 7.24: Mean decrease in Gini

The higher the value associated with the 'mean decrease in Gini', the more important the predictor variable is. There is a significant break between the top four-important predictor variables and the other variables.

### 7.10 Boosting

The boosting procedure was implemented on the learning sample using the 'adabag' package that is contained in R. Figure 7.25 shows the evolution of the testing error rate of the combined classifier, as the number of iterations increases. The dashed reference line indicates the error of the single unprunned tree in Figure 7.10.

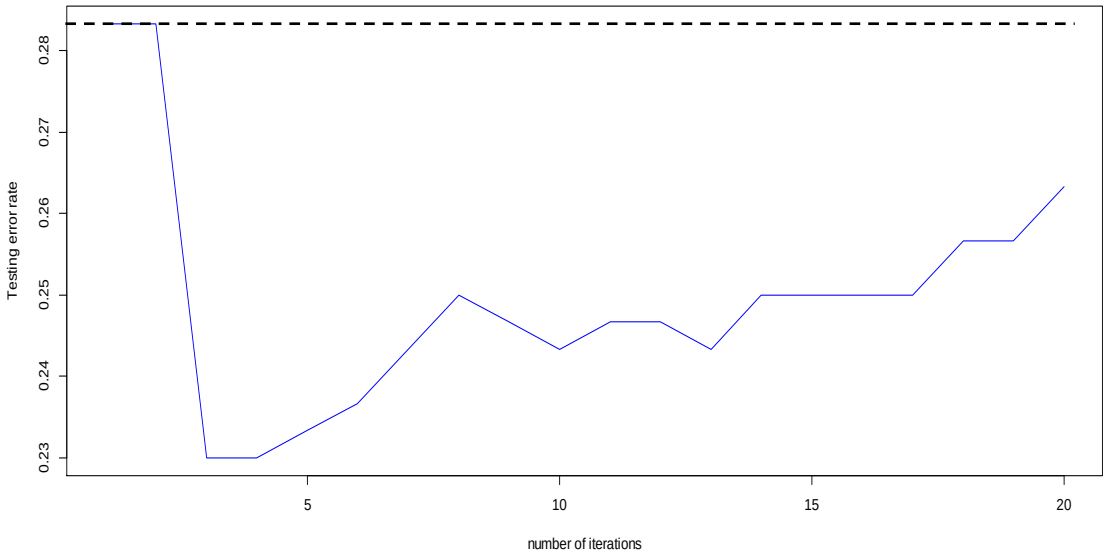


Figure 7.25: Evolution of testing error rate against number of trees

A plot of the change in the testing error rate as the number of iterations increases that is given in Figure 7.25 shows that the testing error rate decreases sharply, reaching its lowest point after only three iterations. Thereafter, it starts to oscillate in a zigzag fashion. Consequently, we have selected three iterations as an optimal for boosting the CART based scorecards that we have been developing.

Table 7.17 shows the classification matrix obtained when the chosen optimal 'boosted' CART model is used to classify applicants in the testing sample. The table shows that we have managed to achieve a testing error rate of $\frac{(23+46)}{300} = 0.23$, sensitivity of 48.3% and specificity of 89.1%.

Table 7.17: Classification matrix for the optimal boosted CART model for the testing sample

| | | | Predicted Group Membership | | |
|---|---|---|---|---|---|
| | | defaultstatus | 0 | 1 | Total |
| Original | Count | 0 | 188 | 23 | 211 |
| | | 1 | 46 | 43 | 89 |
| | % | 0 | 89.1 | 10.9 | 100.0 |
| | | 1 | 51.7 | 48.3 | 100.0 |

Figure 7.26 shows the weighted average improvement, in descending order, of the each of the predictor variables in the optimal boosting estimate. The greater the weighted average improvement value, the more the predictor variable is considered as being an important variable to include in one's classification algorithm.

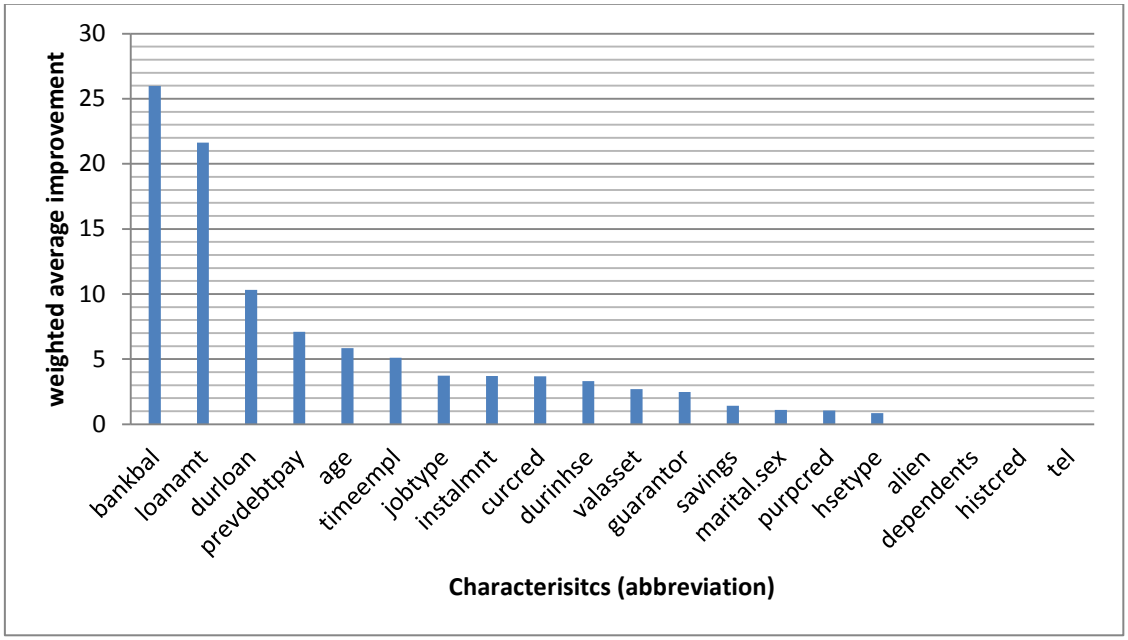


Figure 7.26: Ranking variable importance in the boosting estimate

### 7.11 Summary and comparison of results

In this section, we summarize and compare the performance of the scorecards we developed in section (7.3) to (7.10) in terms of classification error rates, sensitivity and specificity. In addition, the overall discriminatory power of the developed scorecards is compared using the area under the ROC curve (AUC). The concept behind these four model performance measures is discussed in section (2.5).

### 7.11.1 Classification error rates

A summary of the classification error rates that we obtained when the developed scorecards were used to classify the 'new' applicants contained in the testing sample is displayed in Figure 7.27. In the figure, the best performing scorecard (one with the lowest classification error rate) is at the top and the worst performing scorecard (one with the highest classification error rate) is at the bottom.

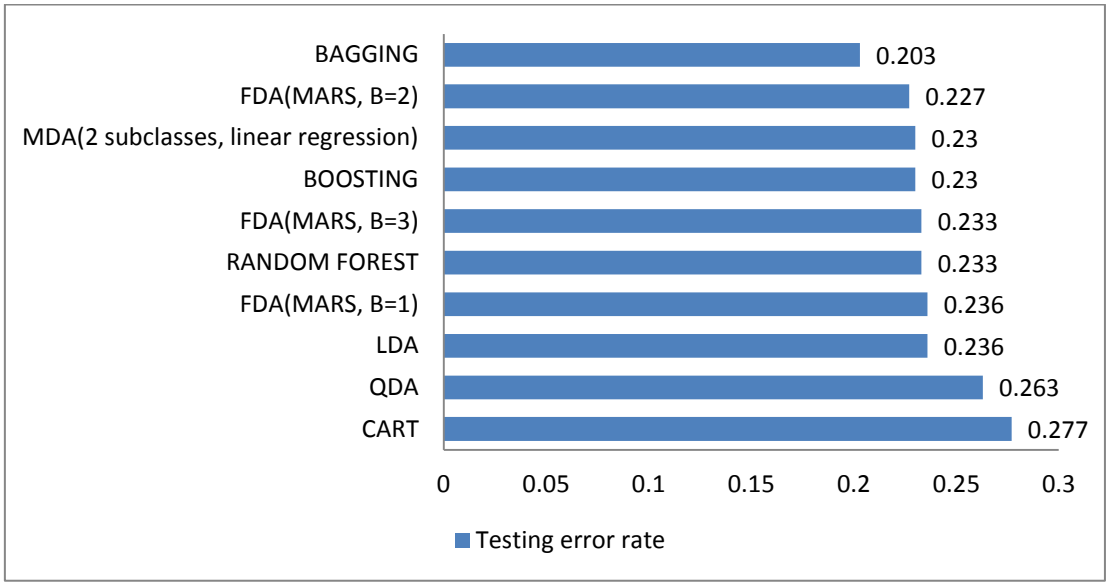


Figure 7.27: Comparison of the classification error rates of all the scorecards when classifying testing sample applicants

The bagging procedure, with the lowest testing error rate of 0.203, is the best scorecard. The other scorecards all seemed to perform equally as well except QDA and CART.

### 7.11.2 Sensitivity

A summary of the sensitivity (in descending order) of the scorecards used in this study when used to classify the new applicants in the testing sample is shown in Figure 7.28. In credit scoring, a lender is more interested in how well the scorecard can correctly identify defaulters (sensitivity) since they pose more risk to the firm.

Therefore, the best method to use to identify defaulters is QDA with a sensitivity of 57.3%.

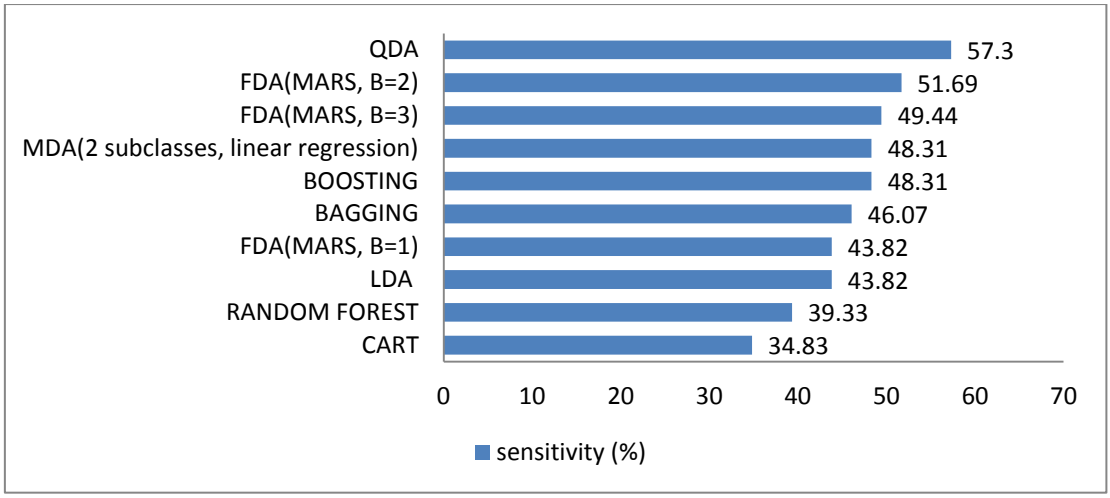


Figure 7.28: Comparison of the sensitivity of all the scorecards when classifying testing sample applicants

### 7.11.3 Specificity

Figure 7.29 shows the specificity (in descending order) of the scorecards used in this study when classifying the new applicants in the testing sample. The higher the specificity, the greater the percentage of non-defaulters that are being correctly identified is. In credit scoring however, specificity is not as serious a problem as sensitivity because most lenders prefer to develop a scorecard that is good at detecting defaulters rather than one that is good at detecting non-defaulters. The most appropriate method to use to identify non-defaulters is bagging with a very high specificity of 93.84%.

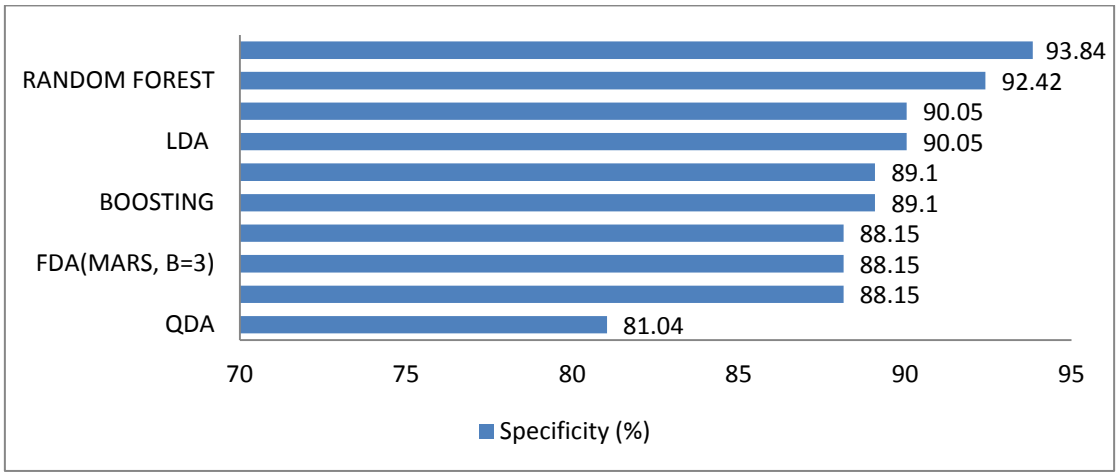


Figure 7.29: Comparison of the specificity of all the scorecards when classifying testing sample applicants

### 7.11.4 Discriminatory power

Figure 7.30 shows the ROC curves that result when the testing sample is passed through the CART based scorecards (bagging, random forests and boosting) as the parameters of the classification rule varies.

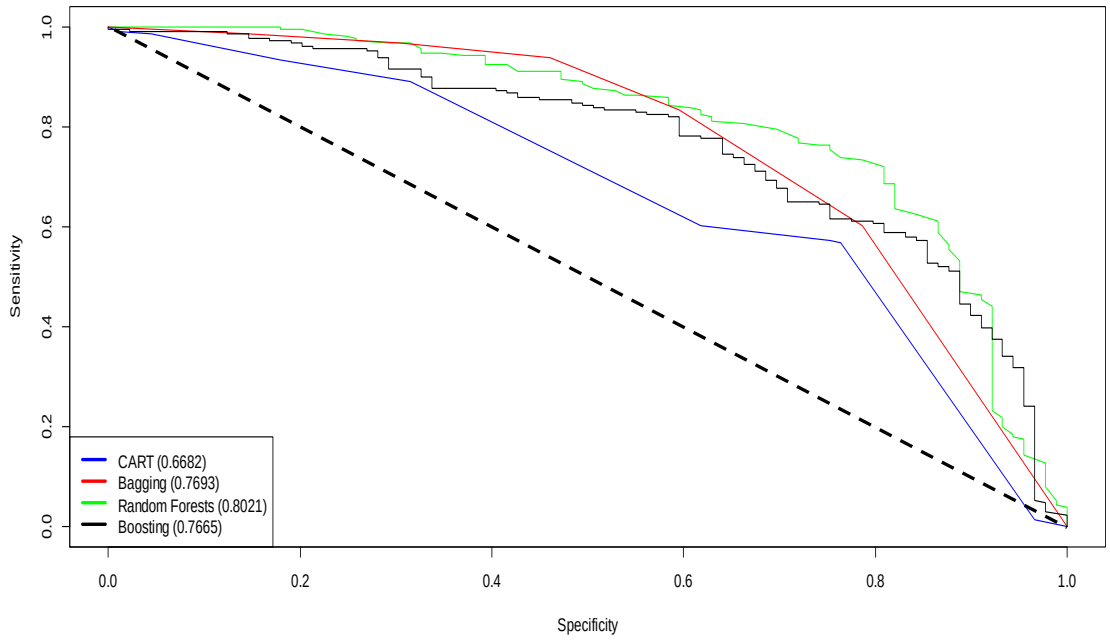


Figure 7.30: ROC curves for CART, bagging, random forests and boosting when classifying testing sample applicants

Curves closer to the top-right corner represent better scorecards since they imply a higher AUC value. The diagonal broken line represents a model that is as good as classification by chance. All the CART based scorecards are better than random guessing. The numbers in brackets on the bottom left corner in Figure 7.30 are AUC values of the corresponding scorecards. The random forest technique has the greatest discriminatory power (AUC=0.8021). Evidently, all the techniques discussed in chapter six (bagging, random forests and boosting) have quite significantly improved the discriminatory power of CART.

Figure 7.31 below shows the ROC curves that result when the testing sample is passed through the LDA based scorecards (QDA, FDA and MDA) as the parameters of the classification rule varies.

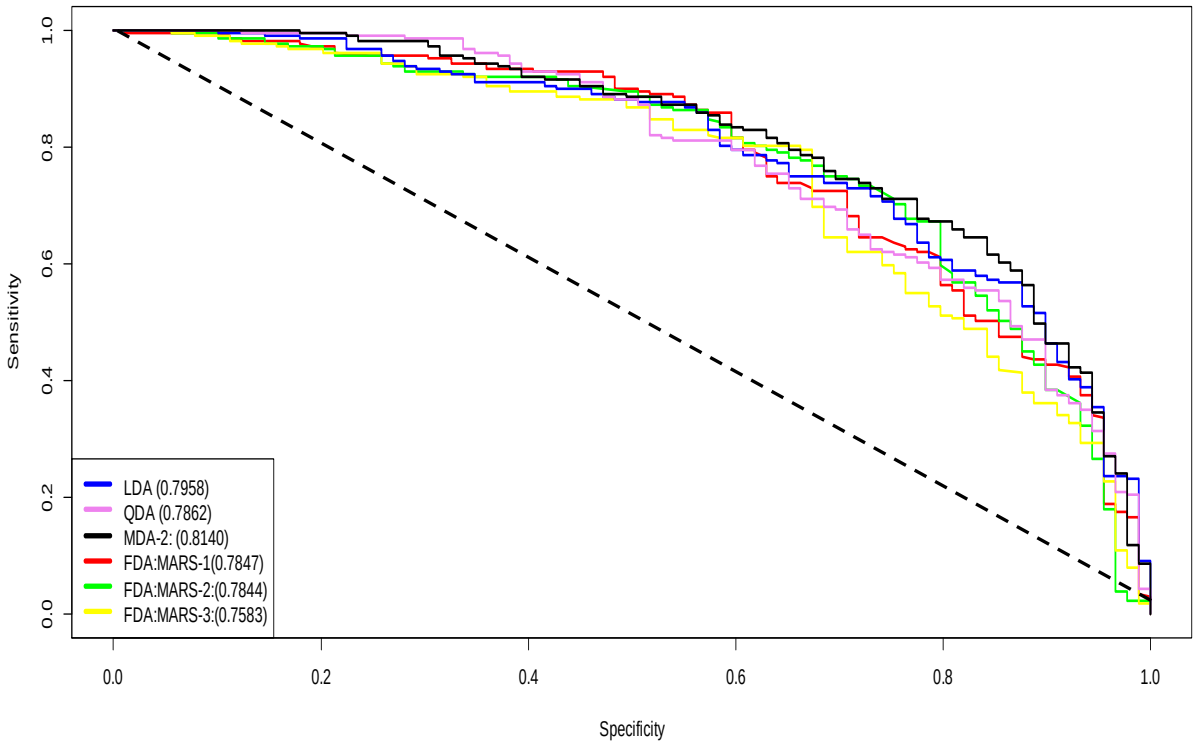


Figure 7.31: ROC curves for discriminant analysis when classifying testing sample applicants

All the discriminant analysis based scorecards we have developed in this study perform better than classification by chance (as represented by the diagonal broken line). MDA (2 subclasses, linear regression) with an AUC of 0.8140 is the best scorecard among all the models developed as extensions of LDA in chapter four. Interestingly, LDA has the second highest AUC value of 0.7958, suggesting that it may be a competitive classifier as compared to some its extensions.

A summary plot of the AUC values (in descending order) of all the models when classifying testing sample applicants is shown in Figure 7.32. The MDA (2 subclasses, linear regression) model with an AUC of 0.814 has the greatest discriminatory power among all the scorecards that we have developed in this study.

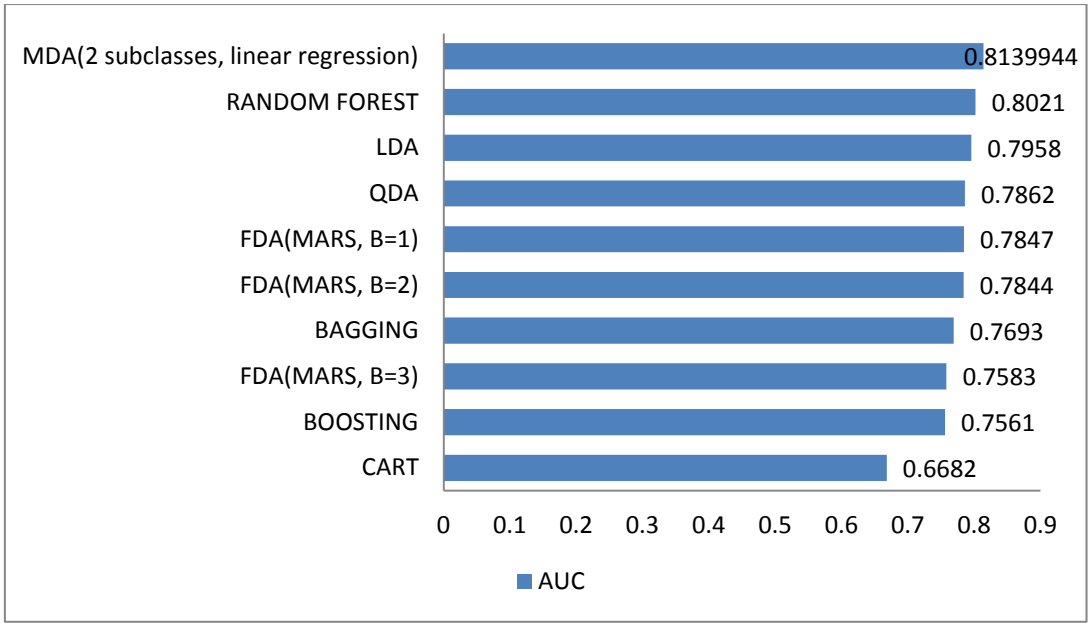


Figure 7.32: AUC for all the scorecards when classifying testing sample applicants

### 7.12 Conclusion

In respect of the research objectives of this study set in the introductory chapter, it is imperative that we address a set of key questions arising therein. Such questions will be addressed based on the empirical results of this study. The questions in issue include; what the best credit-scoring model is and what the effects of techniques that improves upon the performance of LDA and CART are. It is also of paramount importance to consider what the most important predictor variables are. As such, the exercise will serve as a barometer that tests the effectiveness of the techniques that we have been developing in theory, thus satisfying the objectives of this study.

#### 7.12.1 The best credit scoring model

In order to determine the ideal credit-scoring model from among the 10 different scorecards we have managed to create in this study, a score of 10 will be assigned to that scorecard that has the lowest error rate. The same method of scoring will be used for the other categories of predictive performance that the researcher is interested in, namely, the model's sensitivity, specificity and discriminatory power when applied to the testing sample. The points allocated to each scorecard using this criterion are shown in Table 7.18.

Table 7.18: Point system for ranking overall performance of the scorecards

| Scorecard | Testing sample validation model performance measure ranking | | | |
|---|---|---|---|---|
| | Error rate | Sensitivity | Specificity | Discriminatory power |
| CART | 1 | 1 | 4 | 1 |
| Bagging | 10 | 5 | 10 | 4 |
| Random forests | 5 | 2 | 9 | 9 |
| Boosting | 7 | 6 | 5 | 2 |
| LDA | 3 | 3 | 7 | 8 |
| QDA | 2 | 10 | 1 | 7 |
| FDA(MARS, B =1) | 4 | 4 | 8 | 6 |
| FDA (MARS, B=2) | 9 | 9 | 2 | 5 |
| FDA(MARS, B =3) | 6 | 8 | 3 | 3 |
| MDA(2 subclasses, linear regression) | 8 | 7 | 6 | 10 |

According to the results that have been given in the above table:

- bagging had the lowest error rate and highest specificity ranking,
- QDA had the highest sensitivity ranking and
- MDA (2 subclasses, linear regression) had the greatest discriminatory power ranking.

One could proceed further by computing a row sum for all the points that have been assigned to the scorecards in Table 7.18, and then choose that method that produces the highest rank sum as being the best method to use. This idea is illustrated in Figure 7.33, where MDA (2 subclasses, linear regression) delivers the best credit scoring model overall. 'Bagged' classification trees also performed equally well according to this criterion.

However, one needs to note that this method of arriving at a best scorecard does not necessarily take into account the risk appetite of a lender. For example, a risk averse lender would prefer a scorecard that has high sensitivity because such a method would be good at detecting defaulters. Thus, a risk averse lender would choose QDA as the most appropriate credit-scoring model among the scorecards that we have developed.

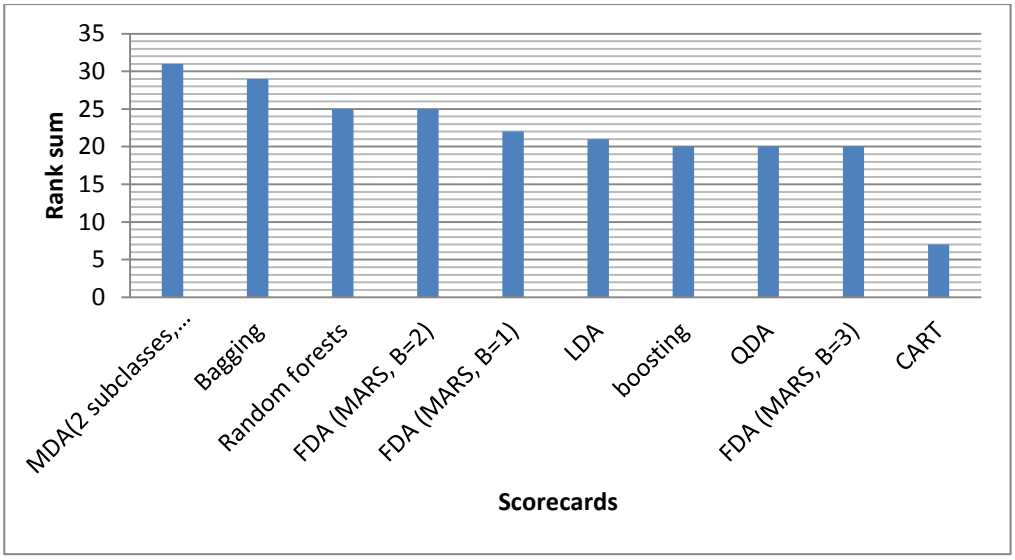


Figure 7.33: Comparison of overall performance of the scorecards

Similarly, a lender who wants to maximize income would prefer a scorecard that makes use of 'bagged' classification trees since the technique has a high specificity value and therefore performs well when attempting to identify a non-defaulter.

Because the structure and logic behind CART is easy to understand, a credit analyst may want to choose this method for classifying a new applicant. The results in this section however, indicate that CART is the worst performing scorecard amongst all the credit-scoring models that we have developed in this chapter.

### 7.12.2 The effect of techniques for improving the performance of LDA and CART

The accuracy of CART, as assessed using four measures: a classification error rate, a sensitivity measure, a specificity measure and the area under the ROC curve (AUC), improved quite significantly on applying the bagging, random forests and boosting procedures in chapter six.

It is also important to note that LDA could still be a competitive classifier, without having to use some of its extensions discussed in chapter four (QDA, FDA and MDA). This is evidenced by its high discriminatory power as measured by its AUC value (see Figure 7.32).

### 7.12.3 The most important predictor variables

Using the same method of ranking that we have used in section (7.12.1) for choosing the best scorecard, 20 points will be assigned to the best predictor, 19 points to the next best predictor and so on. If this sequence of assignment is followed, 1 point will be assigned to the worst predictor variable. Consequently, if a predictor variable does not contribute anything to the model it will receive zero points. The points that have been allocated to all the applicants' characteristics based on this scoring system are shown in Table 7.19, together with the total number of points they received (in descending order).

Table 7.19: Point system for ranking overall variable importance

| Characteristic | CART | Bagging | Boosting | Random Forests | LDA | Total points |
|---|---|---|---|---|---|---|
| Balance of current account | 20 | 20 | 20 | 19 | 20 | 99 |
| duration of loan | 19 | 18 | 18 | 17 | 17 | 89 |
| payment of previous credits | 18 | 17 | 17 | 16 | 19 | 87 |
| Loan amount | 16 | 19 | 19 | 20 | 8 | 82 |
| Value of asset | 15 | 15 | 10 | 13 | 13 | 66 |
| Installment in percentage of available income | 12 | 10 | 13 | 12 | 18 | 65 |
| Values of savings or stock | 17 | 8 | 8 | 11 | 16 | 60 |
| Time employed | 0 | 16 | 15 | 14 | 14 | 59 |
| Purpose of credit | 11 | 14 | 6 | 15 | 6 | 52 |
| Age | 0 | 13 | 16 | 18 | 1 | 48 |
| House ownership | 13 | 6 | 5 | 7 | 15 | 46 |
| Sex/marital status | 14 | 7 | 7 | 8 | 9 | 45 |
| Duration in current house | 10 | 4 | 11 | 10 | 2 | 37 |
| Occupation | 0 | 9 | 14 | 9 | 5 | 37 |
| Guarantor | 0 | 12 | 9 | 4 | 7 | 32 |
| Further running credits | 0 | 0 | 12 | 6 | 12 | 30 |
| Telephone ownership | 0 | 11 | 0 | 5 | 4 | 20 |
| Number of previous credits at the bank | 0 | 0 | 0 | 3 | 11 | 14 |
| Foreign worker | 0 | 0 | 0 | 1 | 10 | 11 |
| Number of dependents | 0 | 5 | 0 | 2 | 3 | 10 |

From the results in the table above, one can observe that the balance on an applicant's current account is undoubtedly the most important predictor variable to include in one's credit scoring model. The duration of the loan, payment of previous credits and loan amount are also very important predictor variables. The following characteristics may not be important enough to include in one's scorecard; telephone ownership, being a foreign worker, number of previous credits at the bank and number of dependents.

# CHAPTER 8

# 8. Summary and Conclusion

## 8.1 Summary

The core of this research was to examine the use of classification techniques to model credit risk. This study was broken down into eight chapters. At the beginning of this study, we set out to model credit risk using one parametric and one non-parametric classification technique, as our main objective. After developing these credit-scoring models, also called scorecards, the second objective was to improve their predictive capabilities. In the process of creating these scorecards, determination of baseline demographic characteristics that are considered as being important variables to include in one's classification algorithm became the thrust. To achieve the aforementioned objectives, the first phase was to explore the theoretical framework of various classification techniques in order to be able to develop and apply them appropriately. The second phase saw the creation of the credit scoring models discussed in theory using a real life credit-related dataset and thus, their performance assessed.

## 8.2 Results and Conclusion

The study revealed that there is no single ideal scorecard for modelling credit risk. Therefore, choosing the most appropriate credit-scoring model is dependent on the aims and objectives of the lender, the details of the problem and the data structure. Techniques for improving the accuracy of classifiers discussed in this study were effective as well. Since the goal of credit scoring is to improve the quality of the decisions when issuing loans, any slight increase in the accuracy of a scorecard will translate into huge profits considering that many of these loans are usually issued. In addition, the variable importance measures generally produced consistent results. The knowledge of important characteristics is essential for the development of better scorecards and for policy implementation.

More so, we gathered that there are some limitations and challenges associated with credit scoring. Firstly, because the scorecards developed in this study are based on a sample drawn from a particular population. These may not perform well if used to score a different population. As a case in point, the sample used to develop the scorecards in

this study contains pre-screened then accepted credit applicants, who later turned out to be either defaulters or non-defaulters. This suggests that such a sample would contain more non-defaulters than defaulters since those apparent defaulters would have been rejected during the pre-screening stages. Resultantly, a credit-scoring model developed on such a dataset may not perform well when applied to the general population (which includes those excluded in the pre-screening process). However, the scorecard will still perform well when used to score pre-screened credit applicants.

The other limitation that came to the researcher's attention in relation to the credit scoring models developed in this study is that we are using historic data to predict the future. The trends and patterns in the general population are susceptible to change over time which consequently affects the accuracy of scorecards developed based on a sample of past credit applicants. Yet again, it poses another challenge in that there is a possibility of prospective borrowers manipulating the system in a bid to improve characteristics considered as being important in determining creditworthiness. Some companies have since been created to help borrowers improve their credit scores.

This study also revealed that credit scores and/ or posterior probabilities could be used in other quantitative analysis of credit risk. For example, the posterior probability of default (PD) is a key parameter in the estimation of econometric capital under the BASEL II regulations for banks (Engelmann & Rauhmeier, 2006). This result is a recommendable approach that can be adopted by financial institutions.

Furthermore, they can be used `to determine a fair price to charge prospective borrowers where an applicant with a low credit score is charged a higher interest rate compared to one with a high credit score. Lending institutions can also use credit scores to set credit limits. In this case, a person with a high credit score is eligible to borrow more money as compared to the one with a lower credit score. Furthermore, the loans can be divided into different portfolios based on their risk levels (for example high, medium and low risk loans), as measured by default probabilities or credit scores and thus managed separately. Such risk-based credit management and pricing techniques should shield lenders from huge losses in the event of defaults.

Having considered all the limitations recorded, the premise of this research, which acclaims credit scoring as an undoubtedly essential and efficient tool for good credit risk management, stands out. This in turn is crucial for the survival, competitiveness and profitability of any lending institution thus the growth of the financial markets at large.

### 8.3 Challenges and Recommendations

The major limitation was acquiring a relevant and current credit-related dataset to work with. The reason for this impediment was that most lending institutions were very reluctant to disclose such information, apart from the collection of a credit-related dataset itself, being expensive. The rationale for this non-disclosure is that such a dataset may contain sensitive information. Moreover, because a dataset containing default patterns and trends of borrowers is the key ingredient to constructing a good scorecard, which in turn gives the lender a competitive advantage over other lenders, lenders are very protective of such datasets. To overcome the dataset challenge, a publicly available dataset containing 1000 past credit applicants was used. This dataset included past credit applicants who were granted loans and later turned out to be either defaulters or non-defaulters. Albeit the encountered impediment, the researcher sought for an alternative.

Even though there is no single ideal credit-scoring model, we recommend that lending institutions consider various scorecards that can handle simple to complex data structures. These range from simple and conventional classification techniques such as LDA, QDA and CART to advanced, exotic and computer intensive techniques such as FDA, MDA, random forests, boosting and bagging. CART may be a better tool when the lender's goal is simply to create an easy to understand and interpret credit-scoring model. For predictive purposes, the researcher recommends that CART be strengthened by the bagging, random forests and/or boosting procedures.

### 8.4 Future Research

Future studies on credit scoring could focus on an automatic method of updating the credit scoring models, which takes into account current information on the performance and behavior of existing loan holders. Future research could also focus on using more than two class outcomes where the following outcomes are being considered: default,

partially default or fully default. One could also include those applicants denied credit during the pre-screening process in the development of the credit scoring models. Prospective research should also aim to incorporate misclassification costs into the modelling approach to reflect the risk appetite of the lenders. Such prospective researches could effectively overcome some of the limitations and challenges in the discipline and hopefully contribute meaningfully to a healthy financial credit market.

# References


1. Abdou, H., Pointon, J., & El-Masry, A. (2008). Neural nets versus conventional techniques in credit scoring in Egyptian banking. *Expert Systems with Applications, 35*(3), 1275-1292. doi: 10.1016/j.eswa.2007.08.030

2. Akkoç, S. (2012). An empirical comparison of conventional techniques, neural networks and the three stage hybrid Adaptive Neuro Fuzzy Inference System (ANFIS) model for credit scoring analysis: The case of Turkish credit card data. *European Journal of Operational Research, 222*(1), 168-178. doi: 10.1016/j.ejor.2012.04.009

3. Altman, E. I. (2005). An emerging market credit scoring system for corporate bonds. *Emerging Markets Review, 6*(4), 311-323. doi: 10.1016/j.ememar.2005.09.007

4. Anderson, R. (2007). *The credit scoring toolkit : theory and practice for retail credit risk management and decision automation*. Oxford: Oxford University Press.

5. Anderson, T. W. (1958). *An introduction to multivariate statistical analysis*. New York: John Wiley.

6. Baesens, B., Gestel, T. V., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking State-of-the-Art Classification Algorithms for Credit Scoring. *Journal of the Operational Research Society, 54*(6), 627-635.

7. Basel Committee on Banking Supervision. (2000). *Principles for the Management of Credit Risk.* Retrieved from http://www.bis.org/publ/bcbs75.htm

8. Ben-David, A., & Frank, E. (2009). Accuracy of machine learning models versus "hand crafted" expert systems – a credit scoring case study. *Expert Systems with Applications, 36*((3/1)), 5264-5271.

9. Bellotti, T., & Crook, J. (2009). Support vector machines for credit scoring and discovery of significant features. *Expert Systems with Applications, 36*(2/2), 3302-3308.

10. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer.

11. Bolton, C. (2009). *Logistic regression and its application in credit scoring*. Thesis. University of Pretoria. Pretoria.

12. Breimam, L. (1998). Arcing classifiers (with discussion). *Annals of Statistics*, 26, 801–849.

13. Breiman, L. (1993). Hinging hyperplanes for regression, classification, and function approximation. *Information Theory, IEEE Transactions on, 39*(3), 999-1013. doi: 10.1109/18.256506

14. Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*, 123-140.

15. Breiman, L. (2001). Random Forests. *Machine Learning, 45*, 5-32.

16. Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. H. (1984). *classification and regression trees*. london: chapman&hall.

17. Clemmensen, L., Hastie, T., Witten, D., & Ersboll, B. (2001). *Sparse Discriminant Analysis.* Retrieved from http://www.standford.edu

18. Crook, J. N., Edelman, D. B., & Thomas, L. C. (2007). Recent developments in consumer credit risk assessment. *European Journal of Operational Research, 183*(3), 1447-1465. doi: http://dx.doi.org/10.1016/j.ejor.2006.09.100

19. DeYoung, R., Frame, W. S., Glennon, D., McMillen, D. P., & Nigro, P. (2008). Commercial lending distance and historically underserved areas. *Journal of Economics and Business, 60*(1–2), 149-164. doi: 10.1016/j.jeconbus.2007.08.004

20. Durand, D. (1941). *Risk elements in consumer instalment financing* ((Technical edition) ed.). New York: National bureau of economic research.

21. Efron, B. (1979). Bootstrap methods: another look at the jackknife. *Annals of Statistics*, 7(1),1-26.

22. Engelmann, B., & Rauhmeier, R. (Eds.). (2006). *The Basel II Risk Parameters*. Berlin: Springer.

23. Feldman, D., & Gross, S. (2005). Mortgage Default: Classification Trees Analysis. *The Journal of Real Estate Finance and Economics, 30*(4), 369-396. doi: 10.1007/s11146-005-7013-7

24. Fisher. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics, 7*, 179-188.

25. Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences, 55(51),* 119-139.

26. Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and Computation, 121(120),* 256-285.

27. Freund, Y., & Schapire, R. E. (1999). A Short Introduction to Boosting. *Journal of Japanese Society for Artificial Intelligence, 14(15),* 771-780.

28. Friedman, J. (1991). Multivariate Adaptive Regression Splines. *The Annals of Statistics, 19,* 11-141.

29. Friedman, J., & Stuetzle, W. (1981). Projection Persuit Regression. *Journal of American Statistical Association,76*, 817-823.

30. Fukunaga. (1990). *Introduction to Statistical Pattern Recognition*. San Diego: Academic Press.

31. Geisser, S. (1964). Posterior Odds for Multivariate Normal Classifications. *Journal of the Royal Statistical Society. Series B (Methodological), 26(1),* 69-76.

32. Hamdi, M., & Karaa, A. (2012). Predicting Financial Distress of Tunisian Firms: A Comparative Study Between Financial Analysis and Neuronal Analysis. *School of Doctoral Studies (European Union) Journal*, 145-153.

33. Hand, D. J. (1997). *Construction and Assessment of Classification Rules*. New York: John Wiley & Sons Ltd.

34. Hand, D. J., & Henley, W. E. (1997). Statistical Classification Methods in Consumer Credit Scoring: a Review. *Journal of Royal Statistical Society, Series A, 160(3),* 523-541.

35. Hand, D. J., & Jacka, S. D. (1998). *Statistics in finance*. New York: John Wiley & Sons Ltd.

36. Hansen, P. (2011). *Approximating the Binomial Distribution by the Normal Distribution – Error and Accuracy*. Retrieved from http://uu.diva-portal.org

37. Hastie, T., & Tibshirani, R. (1996). Discriminant Analysis by Gaussian Mixtures. *Journal of the Royal Statistical Society,Series B (Methodological), 58*, 155-176.

38. Hastie, T., Tibshirani, R., & Buja, A. (1994). Flexible Discriminant Analysis by Optimal Scoring. *Journal of the American Statistical Association 89*(428), 1255-1270.

39. Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The Elements of Statistical Learning:Data Mining,Inference and Prediction*. California: Springer.

40. Henley, W. E., & Hand, D. J. (1996). A k-nearest-neighbour approach for assessing consumer credit risk. *The Stastician, 45(1),* 77-95.

41. Hofmann, H. (1994). *Datasets at the Department of Statistics,University of Munich, and the SFB386*. Retrieved from http://www.stat.uni-muenchen.de/service/datenarchiv/kredit/kredit_e.html

42. Hosmer, D. W., & Lemeshow, S. (1989). *Applied Logistic Regression*. New York: John Wiley & Sons.

43. Kocenda, E., & Vojtek, M. (2009). Default Predictors and Credit Scoring Models for Retail Banking. Retrived from http://www.SSRN.com

44. Koh, H. C., Tan, W. C., & Goh, C. P. (2006). A two-step method to construct credit scoring models with data mining techniques. *International Journal of Business and Information, 1*(1), 96-118.

45. Kumra, R., Stein, R., & Assersohn, I. (2006). Assessing a knowledge-based approach to commercial loan underwriting. *Expert Systems with Applications, 30*(3), 507-518.

46. Lee, T.S., & Chen, I. F. (2005). A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines. *Expert Systems with Applications, 28*(4), 743-752. doi: 10.1016/j.eswa.2004.12.031Lee,

47. Lee,T. S., Chiu, C. C., Lu, C. J., & Chen, I. F. (2002). Credit scoring using the hybrid neural discriminant technique. *Expert Systems with Applications, 23*(3), 245-254. doi: 10.1016/s0957-4174(02)00044-1

48. Lee, T. H., & Jung, S. (2000). Forecasting credit worthiness: Logistic regression vs. artificial neural net. *The Journal of Business Forecasting Methods and Systems, 10*(4), 28-30.

49. Lensberg, T., Eilifsen, A., & McKee, T. (2006). Bankruptcy theory development and classification via genetic programming. *European Journal of Operational Research, 169*(2), 766-697.

50. Li, S.T., Shiue, W., & Huang, M.H. (2006). The evaluation of consumer loans using support vector machines. *Expert Systems with Applications, 30*(4), 772-782. doi: 10.1016/j.eswa.2005.07.041

51. Lippman, R. (1989). Pattern Classification Using Neural Networks. *IEEE Communications Magazine,11*, 47-64.

52. Nanni, L., & Lumini, A. (2009). An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring. *Expert Systems with Applications, 36*(2, Part 2), 3028-3033. doi: 10.1016/j.eswa.2008.01.018

53. Ong, C., Huang, J., & Tzeng, G. (2005). Building Credit Scoring Models Using Genetic Programming. *Expert Systems with Applications, 29*(1), 41-47.

54. Orgler, Y. E. (1970). A credit scoring model for commercial loans. *Journal of Money, Credit and Banking II, 4*, 435-445.

55. Quah, J. T. S., & Sriganesh, M. (2008). Real-time credit card fraud detection using computational intelligence. *Expert Systems with Applications, 35*(4), 1721-1732. doi: 10.1016/j.eswa.2007.08.093

56. Quah, T. S., & Srinivasan, B. (1999). Improving returns on stock investment through neural network selection. *Expert Systems with Applications, 17*(4), 295-301. doi: 10.1016/s0957-4174(99)00041-x

57. Ripley. (1996). *Pattern Recognition and Neural Networks*. Cambridge: Cambridge University Press.

58. Schapire, R. (1990). The strength of weak learnability. *Machine Learning, 5*(2), 197-227.

59. Shahrokhi, M. (2011). The Global Financial Crises of 2007–2010 and the future of capitalism. *Global Finance Journal, 22*(3), 193-210. doi: 10.1016/j.gfj.2011.10.010

60. Surowiecki, J. (2000). *The Wisdom of Crowds: Why the Many are Smarter than the Few and How Collective Wisdom Shapes Business, Economics*. United States: Doubleday;Anchor.

61. Sustersic, M., Mramor, D., & Zupan, J. (2009). Consumer credit scoring models with limited data. *Expert Systems with Applications, 36*(3), 4736-4744. doi: 10.1016/j.eswa.2008.06.016

62. Thomas, L. C., Edelman, D. B., & Crook, J. N. (2002). *Credit scoring and its applications*. Philadelphia, PA: Society for Industrial and Applied Mathematics.

63. TransUnion. (2007). *The Importance of Credit Scoring for Economic Growth*. Retrieved from TransUnion website: https://www.transunion.com/docs/interstitial/scoringWhitepaper.pdf

64. Tsai, C. F., & Wu, J. W. (2008). Using neural network ensembles for bankruptcy prediction and credit scoring. *Expert Systems with Applications, 34*(4), 2639-2649. doi: 10.1016/j.eswa.2007.05.019

65. West, D., Dellana, S., & Qian, J. (2005). Neural network ensemble strategies for financial decision applications. *Computers & Operations Research, 32*(10), 2543-2559.

66. Xia, Y., Liu, B., Wang, S., & Lai, K. K. (2000). A model for portfolio selection with order of expected returns. *Computers & Operations Research, 27*(5), 409-422.

67. Zhu, J., Zou, H., Rosset, S. and Hastie, T. (2009): "Multi-class AdaBoost". Statistics and Its Interface, 2, 349–360.

# Appendix A: The EM algorithm

## A.1 Introduction

In section 4.4 (page 55) we introduced a technique called the the Expectation-Maximization (EM)-algorithm as a means of estimating the parameters $\boldsymbol{\Phi} = \{\omega_{kr}, \boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}_{kr} = \boldsymbol{\Sigma}) \}$ of the following Gaussian Mixture Model (GMM) (where we assumed equal covariance matrices i.e. $\boldsymbol{\Sigma}_{kr} = \boldsymbol{\Sigma}$):

$$P(\mathbf{x}_i|y_i = k) = \sum_{r=1}^{R_k} \omega_{kr} p(\mathbf{x}_i, \boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}_{kr}) \qquad (A.1)$$

The aim of this section is to; describe the maximum-likelihood parameter estimation problem for GMMs, find a solution to this problem using the EM algorithm and show how this procedure give rise to the well-known parameter estimates of the mean and covariance matrix used for the parametric classification procedures in chapter three and four, viz:

$$\bar{\mathbf{x}}_k = \frac{1}{N_k} \sum_{y_i=k} \mathbf{x}_i \qquad \forall k = 1,2, \dots, K$$

$$\boldsymbol{S}_k = \frac{1}{N_k - 1} \sum_{y_i=k} (\mathbf{x}_i - \bar{\mathbf{x}}_k)(\mathbf{x}_i - \bar{\mathbf{x}}_k)^T$$

## A.2 The maximum likelihood estimation problem

The parameter estimates $\boldsymbol{\Phi} = \{\omega_{kr}, \boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}_{kr}\}$ of the GMM in (A.1) are computed by maximizing its likelihood function,

$$L(\mathbf{x}_i, \boldsymbol{\Phi}) = \prod_{i=1}^{N} P(\mathbf{x}_i|y_i = k) = \prod_{i=1}^{N} \left[ \sum_{r=1}^{R_k} \omega_{kr} p(\mathbf{x}_i, \boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}_{kr}) \right] \qquad (A.2)$$

with respect to $\boldsymbol{\Phi}$.

Because the natural logarithm function is monotonic, we need only find values of $\boldsymbol{\Phi}$ that maximizes:

$$l(\mathbf{x}_i, \boldsymbol{\Phi}) = \log L(\mathbf{x}_i, \boldsymbol{\Phi}) = \sum_{i=1}^{N} \log \left[ \sum_{r=1}^{R_k} \omega_{kr} p(\mathbf{x}_i, \boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}) \right] \quad (A.3)$$

However, the summation of the log terms makes (A.3) difficult to maximize directly.

**A.3 The maximum likelihood estimation solution using the EM algorithm**

The conventional and appropriate method for computing the *maximum-likelihood estimates* (MLEs) for mixture distributions is the EM-algorithm (Bilmes, 1998; Dempster *et al*., 1977).

The EM-algorithm is a technique used to find the MLE of parameters of incomplete datasets or datasets with missing variables such that by assuming that our dataset has additional variables that are missing, we can use the technique to estimate parameters of likelihood functions that are complex.

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N\}$ be the set of independent and identically (i.i.d) observations, which we shall call the 'incomplete dataset', where

$$p(\mathbf{x}_i|\boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}_{kr}) = p(\mathbf{x}_i, \boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}_{kr}) \sim N(\boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}_{kr})$$

Similarly, let $Y = \{y_1, y_2, \ldots y_N\}$ be a set of i.i.d missing observations such that our 'complete dataset' is $g = \{X, Y\}$. In addition, let the joint distribution function of the complete dataset be $p(Y, X|\boldsymbol{\Phi})$.

Consequently, the *complete-data likelihood function* would be given by,

$$L(\boldsymbol{\Phi}|X, Y) = p(X, Y|\boldsymbol{\Phi}) = \prod_{i=1}^{N} p(\mathbf{x}_i, y_i|\boldsymbol{\Phi}) \quad (A.4)$$

and the *incomplete-data likelihood function* would be given by

$$L(\boldsymbol{\Phi}|X) = p(X|\boldsymbol{\Phi}) = \prod_{i=1}^{N} P(\mathbf{x}_i|\boldsymbol{\Phi}) = \prod_{i=1}^{N} \left[ \sum_{r=1}^{R_k} \omega_{kr} p(\mathbf{x}_i|\boldsymbol{\mu}_{kr}, \boldsymbol{\Sigma}_{kr}) \right] \quad (A.5)$$

Because $Y$ is unknown, the complete-data log-likelihood is a random variable. Thus, the first step in the EM algorithm is to find the expectation of the complete-data log-likelihood with respect to $Y$, given the observed data $X$ and the current parameter estimates $\boldsymbol{\Phi}^t = (\omega_{kr}^t, \boldsymbol{\mu}_{kr}^t, \boldsymbol{\Sigma}_{kr}^t)$ at the $t^{th}$ iteration, i.e.

$$\Longrightarrow E_Y[\log p(\mathrm{X}, \mathrm{Y}|\boldsymbol{\Phi}^t, \mathrm{X})] = \sum_Y [\log p(\mathrm{X}, \mathrm{Y}|\boldsymbol{\Phi})] p(Y|X, \boldsymbol{\Phi}^t) = \boldsymbol{Q}(\boldsymbol{\Phi}, \boldsymbol{\Phi}^t) \quad (\mathrm{A}.6)$$

where,$\boldsymbol{\Phi}$ is the new set of parameters that will maximize $\boldsymbol{Q}(\boldsymbol{\Phi}, \boldsymbol{\Phi}^t)$. The computation of the expectation of the complete-data log-likelihood is known as the *E-step*.

The next step is to find the parameters $\boldsymbol{\Phi}$ that maximize the expectation of the complete-data log-likelihood (A.6), i.e.

$$\boldsymbol{\Phi} = arg\max_{\boldsymbol{\Phi}} [\boldsymbol{Q}(\boldsymbol{\Phi}, \boldsymbol{\Phi}^t)] \quad (\mathrm{A}.7)$$

This implies that, we need new parameter estimates $\boldsymbol{\Phi}$ such that,

$$\boldsymbol{Q}(\boldsymbol{\Phi}, \boldsymbol{\Phi}^t) \geq \boldsymbol{Q}(\boldsymbol{\Phi}^t, \boldsymbol{\Phi}^t) \quad (\mathrm{A}.8)$$

for all, $t = 1,2, \dots, T$ iterations. The computation of new parameters $\boldsymbol{\Phi}$ that maximize the expectation of the complete-data log-likelihood is known as the *M-step*.

The EM-algorithm finds the optimal parameter estimates $\boldsymbol{\Phi}$ that maximizes incomplete-data likelihood function (A.5) by oscillating between the E and M steps until $\boldsymbol{\Phi}$ converges.

### A.3.1 Convergence property of the EM algorithm

Finding the parameter estimates $\boldsymbol{\Phi}$ that satisfy (A.8) will result in the log-likelihood function being maximized, viz:

$$\boldsymbol{Q}(\boldsymbol{\Phi}, \boldsymbol{\Phi}^t) \geq \boldsymbol{Q}(\boldsymbol{\Phi}^t, \boldsymbol{\Phi}^t) \Longrightarrow l(X|\boldsymbol{\Phi}) \geq l(X|\boldsymbol{\Phi}^t) \quad \forall t = 1,2, \dots, T \quad (\mathrm{A}.9)$$

**Proof**

From probability theory,

$$p(Y, X|\boldsymbol{\Phi}) = \frac{p(X, Y, \boldsymbol{\Phi})}{p(\boldsymbol{\Phi})} = \frac{p(X, Y, \boldsymbol{\Phi})}{p(X, \boldsymbol{\Phi})} * \frac{p(X, \boldsymbol{\Phi})}{p(\boldsymbol{\Phi})} \quad (\mathrm{A}.10)$$

$$\Longrightarrow p(Y,X|\boldsymbol{\Phi}) = p(Y|\mathrm{X},\boldsymbol{\Phi}) * p(X|\boldsymbol{\Phi}) = p(X|\mathrm{Y},\boldsymbol{\Phi}) * p(Y|\boldsymbol{\Phi}) \quad \text{(A. 11)}$$

$$\Longrightarrow p(X|\boldsymbol{\Phi}) = \frac{p(Y,X|\boldsymbol{\Phi})}{p(Y|\mathrm{X},\boldsymbol{\Phi})} \quad \text{(A. 12)}$$

The incomplete-data log-likelihood given current parameter estimates $\boldsymbol{\Phi}^t$ can now be expressed as:

$$l(X|\boldsymbol{\Phi}^t) = \log p(X|\boldsymbol{\Phi}^t)$$

$$= [\log p(X|\boldsymbol{\Phi}^t)] * \sum_Y p(Y|\mathrm{X},\boldsymbol{\Phi}^t) \qquad \left\{\text{since } \sum_Y p(Y|\mathrm{X},\boldsymbol{\Phi}^t) = 1\right\}$$

$$= \sum_Y p(Y|\mathrm{X},\boldsymbol{\Phi}^t) \log p(X|\boldsymbol{\Phi}^t)$$

$$= \sum_Y p(Y|\mathrm{X},\boldsymbol{\Phi}^t) \log \frac{p(Y,X|\boldsymbol{\Phi}^t)}{p(Y|\mathrm{X},\boldsymbol{\Phi}^t)} \qquad \{\text{using the result(A. 12)}\}$$

$$= \sum_Y [\log p(Y,X|\boldsymbol{\Phi}^t)] p(Y|\mathrm{X},\boldsymbol{\Phi}^t) - \sum_Y [\log p(Y|\mathrm{X},\boldsymbol{\Phi}^t)] p(Y|\mathrm{X},\boldsymbol{\Phi}^t)$$

$$= \boldsymbol{Q}(\boldsymbol{\Phi}^t,\boldsymbol{\Phi}^t) - \boldsymbol{R}(\boldsymbol{\Phi}^t,\boldsymbol{\Phi}^t) \quad \text{(A. 13)}$$

where,

$$\boldsymbol{R}(\boldsymbol{\Phi}^t,\boldsymbol{\Phi}^t) = \sum_Y [\log p(Y|\mathrm{X},\boldsymbol{\Phi}^t)] p(Y|\mathrm{X},\boldsymbol{\Phi}^t) \quad \text{(A. 14)}$$

and $\boldsymbol{Q}(\boldsymbol{\Phi}^t,\boldsymbol{\Phi}^t)$ follows from equation (A. 6).

On the other hand, the incomplete-data log-likelihood given the new parameter estimates $\boldsymbol{\Phi}$ is:

$$l(X|\boldsymbol{\Phi}) = \log p(X|\boldsymbol{\Phi})$$

$$= \log\left\{\sum_Y p(Y|\mathrm{X},\boldsymbol{\Phi}^t) * \frac{p(Y,X|\boldsymbol{\Phi})}{p(Y|\mathrm{X},\boldsymbol{\Phi}^t)}\right\}$$

$$= \log\left\{E_y\left(\frac{p(Y,X|\boldsymbol{\Phi})}{p(Y|\mathrm{X},\boldsymbol{\Phi}^t)}\middle|\mathrm{X},\boldsymbol{\Phi}^t\right)\right\} \qquad \{\text{using definition of expectation}\}$$

By Jensen (1906)'s inequality we have the following result for a convex function $f$ :

$$f[E(x)] \geq E[f(x)]$$

such that,

$$\log\left\{E_y\left(\frac{p(Y,X|\boldsymbol{\Phi})}{p(Y|\mathrm{X},\boldsymbol{\Phi}^t)}\middle|\mathrm{X},\boldsymbol{\Phi}^t\right)\right\} \geq E_y\left\{\log\left(\frac{p(Y,X|\boldsymbol{\Phi})}{p(Y|\mathrm{X},\boldsymbol{\Phi}^t)}\middle|\mathrm{X},\boldsymbol{\Phi}^t\right)\right\}$$

$$= \left\{\sum_Y\left[\log\left\{\frac{p(Y,X|\boldsymbol{\Phi})}{p(Y|\mathrm{X},\boldsymbol{\Phi}^t)}\right\}\right] * p(Y|\mathrm{X},\boldsymbol{\Phi}^t)\right\}$$

$$= \sum_Y [\log p(Y,X|\boldsymbol{\Phi})]p(Y|\mathrm{X},\boldsymbol{\Phi}^t)$$

$$-\sum_Y [\log p(Y|\mathrm{X},\boldsymbol{\Phi}^t)]p(Y|\mathrm{X},\boldsymbol{\Phi}^t)$$

$$= \boldsymbol{Q}(\boldsymbol{\Phi},\boldsymbol{\Phi}^t) - \boldsymbol{R}(\boldsymbol{\Phi}^t,\boldsymbol{\Phi}^t) \qquad (\mathrm{A.15})$$

Combining, the equations(A. 8), (A. 13)and (A. 15) we have,

$$l(X|\boldsymbol{\Phi}) \geq \boldsymbol{Q}(\boldsymbol{\Phi},\boldsymbol{\Phi}^t) - \boldsymbol{R}(\boldsymbol{\Phi}^t,\boldsymbol{\Phi}^t) \qquad \{\text{by equation (A. 15)}\}$$

$$\geq \boldsymbol{Q}(\boldsymbol{\Phi}^t,\boldsymbol{\Phi}^t) - \boldsymbol{R}(\boldsymbol{\Phi}^t,\boldsymbol{\Phi}^t) \qquad \{\text{using equation (A. 8)}\}$$

$$= l(X|\boldsymbol{\Phi}^t) \qquad \{\text{from equation (A. 13)}\}$$

$$\Longrightarrow l(X|\boldsymbol{\Phi}) \geq l(X|\boldsymbol{\Phi}^t)$$

### A.3.2 Computing the parameter estimates for Gaussian Mixture Models

Dropping the subscript $k$ on the GMM in (A.1) that label the $k^{th}$ class for which the parameter estimates for the $r = 1,2,\ldots,R$ latent subclasses are being computed, consider a problem where we want to find the parameter estimates of the following GMM

$$p(\mathbf{x}_i|\boldsymbol{\Phi}_r) = \sum_{r=1}^{R} \omega_r p(\mathbf{x}_i|\boldsymbol{\theta}_r) \qquad (\mathrm{A.16})$$

where $\boldsymbol{\theta}_r = (\bar{\mathbf{x}}_r, \boldsymbol{S}_r)$ are the parameter estimates of the $r^{th}$ component Gaussian distribution. The main challenge in estimating the parameter estimates $\boldsymbol{\Phi} = (\omega_r, \boldsymbol{\theta}_r)$ is that the $r^{th}$ component generating the observation $\mathbf{x}_i \in R^p$ is unknown.

Let,

$$\mathrm{y}_i = \begin{cases} r & if\ \mathbf{x}_i \quad \text{is generated by the } r^{th} \text{ mixture component} \\ 0 & \text{otherwise} \end{cases}$$

such that,

$$\delta_{r,\mathrm{y}_i} = \begin{cases} 1 & if\ \ \mathrm{y}_i = r \\ 0 & \text{otherwise} \end{cases} \tag{A.17}$$

If the $r^{th}$ component generating $\mathbf{x}_i$ is known (i.e. $\delta_{r,\mathrm{y}_i} = 1$), the complete-data log-likelihood (A.4) would be,

$$l(\boldsymbol{\Phi}|X,Y) = \log p(Y,X|\boldsymbol{\Phi}) = \log \prod_{i=1}^{N} p(\mathbf{x}_i, \mathrm{y}_i|\boldsymbol{\Phi})$$

$$= \sum_{i=1}^{N} \log[\, p(\mathbf{x}_i, \mathrm{y}_i|\boldsymbol{\Phi})]$$

$$= \sum_{i=1}^{N} \log[p(\mathbf{x}_i|y_i, \boldsymbol{\Phi}) * p(y_i|\boldsymbol{\Phi})]$$

$$= \sum_{i=1}^{N} \log[\omega_{y_i} p_{y_i}(\mathbf{x}_i|\boldsymbol{\theta}_{y_i})] \tag{A.18}$$

where, $\omega_{y_i} = p(y_i|\boldsymbol{\Phi})$ can be thought of as the prior probability (if $\boldsymbol{\Phi}$ is known) of the $i^{th}$ case being generated by the $r^{th}$ mixture component (i.e. $y_i = r$). In addition, $p_{y_i}(\mathbf{x}_i|\boldsymbol{\theta}_{y_i}) = p(\mathbf{x}_i|y_i, \boldsymbol{\Phi})$ is the probability density function of the observation $\mathbf{x}_i$ when $y_i = r$ and the parameter estimates $\boldsymbol{\Phi}$ are known. Bayes' theorem allows one to write

$$p(y_i|\mathbf{x}_i, \mathbf{\Phi}^t) = \frac{p(\mathbf{x}_i|y_i, \mathbf{\Phi}^t) * p(y_i|\mathbf{\Phi}^t)}{p(\mathbf{x}_i|\mathbf{\Phi}^t)} = \frac{\omega_r^t p_{y_i}(\mathbf{x}_i|\boldsymbol{\theta}_{y_i})}{\sum_{l=1}^{R} \omega_l^t p_{y_i}(\mathbf{x}_i|\boldsymbol{\theta}_{y_i})} \quad (A.19)$$

as the probability density function of the latent observations $Y = \{\mathrm{y}_1, \mathrm{y}_2, \ldots \mathrm{y}_\mathrm{N}\}$ given the observations $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_\mathrm{N}\}$ and the parameter estimates $\mathbf{\Phi}^t$ at the $t^{th}$ iteration. Consequently,

$$\Longrightarrow p(Y|X, \mathbf{\Phi}^t) = \prod_{i=1}^{N} p(y_i|\mathbf{x}_i, \mathbf{\Phi}^t) \quad (A.20)$$

**1. E-STEP:**

Inserting equations (A.18) and (A.20) into the expectation of the complete-data log-likelihood (A.6) gives,

$$\boldsymbol{Q}(\mathbf{\Phi}, \mathbf{\Phi}^t) = E_Y[\log p(\mathrm{X}, \mathrm{Y}|\mathbf{\Phi}^t, \mathrm{X})]$$

$$= \sum_{Y} [\log p(\mathrm{X}, \mathrm{Y}|\mathbf{\Phi})] p(Y|X, \mathbf{\Phi}^t)$$

$$= \sum_{Y} \left[ \sum_{i=1}^{N} \log[\omega_{y_i} p_{y_i}(\mathbf{x}_i|\boldsymbol{\theta}_{y_i})] \right] \prod_{j=1}^{N} p(y_j|\mathbf{x}_j, \mathbf{\Phi}^t)$$

$$= \sum_{y_1=1}^{R} \sum_{y_2=1}^{R} \cdots \sum_{y_{N=1}}^{R} \left[ \sum_{i=1}^{N} \log[\omega_{y_i} p_{y_i}(\mathbf{x}_i|\boldsymbol{\theta}_{y_i})] \right] \prod_{j=1}^{N} p(y_j|\mathbf{x}_j, \mathbf{\Phi}^t)$$

$$= \sum_{y_1=1}^{R} \sum_{y_2=1}^{R} \cdots \sum_{y_{N=1}}^{R} \sum_{i=1}^{N} \sum_{r=1}^{R} \delta_{r,\mathrm{y}_i} \log[\omega_{y_i} p_{y_i}(\mathbf{x}_i|\boldsymbol{\theta}_{y_i})] \prod_{j=1}^{N} p(y_j|\mathbf{x}_j, \mathbf{\Phi}^t)$$

$$= \sum_{r=1}^{R} \sum_{i=1}^{N} \log[\omega_{y_i} p_{y_i}(\mathbf{x}_i|\boldsymbol{\theta}_{y_i})] \sum_{y_1=1}^{R} \sum_{y_2=1}^{R} \cdots \sum_{y_{N=1}}^{R} \delta_{r,\mathrm{y}_i} \prod_{j=1}^{N} p(y_j|\mathbf{x}_j, \mathbf{\Phi}^t) \quad (A.21)$$

Because $\delta_{r,\mathrm{y}_i} = 1$ if $y_i = r$ and zero otherwise, we can rewrite (A.21) as,

$$\boldsymbol{Q}(\mathbf{\Phi}, \mathbf{\Phi}^t)$$

$$= \sum_{r=1}^{R} \sum_{i=1}^{N} \log[\omega_r p_r(\mathbf{x}_i|\boldsymbol{\theta}_r)] \left\{ \sum_{y_1=1}^{R} \sum_{y_2=1}^{R} \cdots \sum_{y_{N=1}}^{R} \prod_{j=1, j\neq k}^{N} p(y_j|\mathbf{x}_j, \mathbf{\Phi}^t) \right\} p(y_k = r|\mathbf{x}_i, \mathbf{\Phi}^t)$$

$$= \sum_{r=1}^{R}\sum_{i=1}^{N} \log[\omega_r p_r(\mathbf{x}_i|\boldsymbol{\theta}_r)] \left\{ \prod_{j=1, j\neq k}^{N} \sum_{y_j=1}^{R} p\left(y_j \middle| \mathbf{x}_j, \boldsymbol{\Phi}^t\right) \right\} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t)$$

$$= \sum_{r=1}^{R}\sum_{i=1}^{N} \log[\omega_r p_r(\mathbf{x}_i|\boldsymbol{\theta}_r)]\, p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t) \qquad \left\{ \text{since} \sum_{y_j=1}^{R} p\left(y_j \middle| \mathbf{x}_j, \boldsymbol{\Phi}^t\right) = 1 \right\}$$

$$= \sum_{r=1}^{R}\sum_{i=1}^{N} \log \omega_r\, p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t) + \sum_{r=1}^{R}\sum_{i=1}^{N} \log p_r(\mathbf{x}_i|\boldsymbol{\theta}_r)\, p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t) \qquad (\text{A}.22)$$

**2. M-STEP:-**

To maximize the expectation of the complete-data log-likelihood in (A. 22), we separately consider its right hand side terms, viz:

$$\sum_{r=1}^{R}\sum_{i=1}^{N} \log \omega_r\, p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t) \qquad (\text{A}.23)$$

and,

$$\sum_{r=1}^{R}\sum_{i=1}^{N} \log p_r(\mathbf{x}_i|\boldsymbol{\theta}_r)\, p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t) \qquad (\text{A}.24)$$

This is because the term (A.23) is independent of the unknown $r^{th}$ component parameters $\boldsymbol{\theta}_r = (\bar{\mathbf{x}}_r, \boldsymbol{S}_r)$ and the term (A.24) is independent of the unknown $r^{th}$ component weight $\omega_r$.

To find the MLE of $\omega_r$ under the constrain, $\sum_{r=1}^{R} \omega_r = 1 \Longrightarrow \sum_{r=1}^{R} \omega_r - 1 = 0$, we use the method of Lagrange multipliers to attach the constrain to the term (A.23) and solve,

$$\Longrightarrow \frac{\partial}{\partial \omega_r}\left( \sum_{r=1}^{R}\sum_{i=1}^{N} \log \omega_r\, p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t) + \lambda\left( \sum_{r=1}^{R} \omega_r - 1 \right) \right) = 0$$

where $\lambda \in \mathbb{R}$ is the Lagrange multiplier, to get

$$\sum_{i=1}^{N} \frac{1}{\omega_r} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t) + \lambda = 0$$

$$\Longrightarrow \sum_{i=1}^{N} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t) = -\lambda\omega_r \qquad (A.25)$$

Summing both sides of (A.25) over the *r* components gives:-

$$\sum_{i=1}^{N}\sum_{r=1}^{R} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t) = -\lambda \sum_{r=1}^{R} \omega_r$$

$$\Longrightarrow N = -\lambda$$

since, $\sum_{r=1}^{R} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t) = 1$ and $\sum_{r=1}^{R} \omega_r = 1$.

Inserting, $N = -\lambda$ into (A. 25) we get the MLE of $\omega_r$ as,

$$\omega_r = \frac{1}{N}\sum_{i=1}^{N} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t) \qquad \forall r = 1,2, \dots, R \qquad (A.26)$$

Turning our attention to the problem where we want to find the parameter estimates for $\boldsymbol{\theta}_r = (\bar{\mathbf{x}}_r, \boldsymbol{S}_r)$ that govern each *r* component Gaussian probability density function,

$$p_r(\mathbf{x}_i|\bar{\mathbf{x}}_r, \boldsymbol{S}_r) \sim N(\bar{\mathbf{x}}_r, \boldsymbol{S}_r)$$

we insert,

$$\log p_r(\mathbf{x}_i|\bar{\mathbf{x}}_r, \boldsymbol{S}_r) = -\frac{p}{2}\log 2\pi - \frac{1}{2}\log \boldsymbol{S}_r - \frac{1}{2}(\mathbf{x}_i - \bar{\mathbf{x}}_r)^T \mathbf{S}_r^{-1}(\mathbf{x}_i - \bar{\mathbf{x}}_r) \qquad (A.27)$$

into the term (A.24) to get

$$\sum_{r=1}^{R}\sum_{i=1}^{N} \log p_r(\mathbf{x}_i|\boldsymbol{\theta}_r)\, p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t)$$

$$= \sum_{r=1}^{R}\sum_{i=1}^{N} \left(-\frac{p}{2}\log 2\pi - \frac{1}{2}\log \mathbf{S}_r - \frac{1}{2}(\mathbf{x}_i - \bar{\mathbf{x}}_r)^T \mathbf{S}_r^{-\mathbf{1}}(\mathbf{x}_i - \bar{\mathbf{x}}_r)\right) p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t) \quad (A.28)$$

Differentiating equation (A.28) with respect to $\bar{\mathbf{x}}_r$ and equating to the zero vector gives,

$$\sum_{i=1}^{N} (\boldsymbol{\Sigma}_r^{-\mathbf{1}}(\mathbf{x}_i - \bar{\mathbf{x}}_r))\, p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t) = \mathbf{0}$$

$$\Rightarrow \sum_{i=1}^{N} \mathbf{x}_i p(r|\mathbf{x}_i, \mathbf{\Phi}^t) = \bar{\mathbf{x}}_r \sum_{i=1}^{N} p(r|\mathbf{x}_i, \mathbf{\Phi}^t)$$

$$\Rightarrow \bar{\mathbf{x}}_r = \frac{\sum_{i=1}^{N} \mathbf{x}_i p(r|\mathbf{x}_i, \mathbf{\Phi}^t)}{\sum_{i=1}^{N} p(r|\mathbf{x}_i, \mathbf{\Phi}^t)} \qquad \forall r = 1,2,\dots,R \qquad \text{(A. 29)}$$

Finally, to find the MLE of $\mathbf{S}_r$we shall use the following results given square matrices $\boldsymbol{A}$,$\boldsymbol{B}$ and a vector $\mathbf{x}_i$ (see Fukunaga (1990:564-571)):

1. $\sum_i \mathbf{x}_i^T \boldsymbol{A} \mathbf{x}_i, = \sum_i \boldsymbol{A}\mathbf{x}_i, \mathbf{x}_i^T = tr(\boldsymbol{AB}) \quad where\ \boldsymbol{B} = \sum_i \mathbf{x}_i \mathbf{x}_i^T$

2. $\frac{\partial \mathbf{x}_i^T \mathbf{A} \mathbf{x}_i}{\partial \mathbf{x}_i,} = (\mathbf{A} + \mathbf{A}^T)\mathbf{x}_i$

3. $\frac{\partial \log|\boldsymbol{A}|}{\partial \boldsymbol{A}} = 2\boldsymbol{A}^{-1} - diag(\boldsymbol{A}^{-1})$

4. $\frac{\partial tr(\boldsymbol{AB})}{\partial \boldsymbol{A}} = \boldsymbol{B} + \boldsymbol{B}^T - diag(\boldsymbol{B})$

where, diag ($\boldsymbol{A}$) is a diagonal matrix $\boldsymbol{A}$.

Letting $\boldsymbol{B}_{ir} = (\mathbf{x}_i - \bar{\mathbf{x}}_r)(\mathbf{x}_i - \bar{\mathbf{x}}_r)^T$ **,** (A. 28) becomes

$$\Rightarrow \sum_{r=1}^{R} \sum_{i=1}^{N} \log p_r(\mathbf{x}_i|\boldsymbol{\theta}_r)\, p(r|\mathbf{x}_i, \mathbf{\Phi}^t)$$

$$= \sum_{r=1}^{R} \left( -\frac{p}{2} \log 2\pi + \frac{1}{2} \log\left|\mathbf{S}_r^{-1}\right| - \frac{1}{2} \sum_{i=1}^{N} tr\,(\mathbf{S}_r^{-1} \boldsymbol{B}_{ir}) \right) p(r|\mathbf{x}_i, \mathbf{\Phi}^t)$$

which on differentiating with respect to $\mathbf{\Sigma}_r^{-1}$ and equating to the zero matrix gives,

$$\frac{1}{2} \sum_{i=1}^{N} p(r|\mathbf{x}_i, \mathbf{\Phi}^t)[2\mathbf{S}_r - diag(\mathbf{S}_r)] - \frac{1}{2} \sum_{i=1}^{N} p(r|\mathbf{x}_i, \Phi^t)[\boldsymbol{B}_{ir} + \boldsymbol{B}_{ir}^T - diag(\boldsymbol{B}_{ir})] = \mathbf{0}$$

Since $\boldsymbol{B}_{ir}$ is asymmetric matrix ($\boldsymbol{B}_{ir} = \boldsymbol{B}_{ir}^T$),

$$\Rightarrow \frac{1}{2}\sum_{i=1}^{N} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t)[2\mathbf{S}_r - diag(\mathbf{S}_r)] - \frac{1}{2}\sum_{i=1}^{N} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t)[2\boldsymbol{B}_{ir} - diag(\boldsymbol{B}_{ir})] = \mathbf{0}$$

$$\Rightarrow \frac{1}{2}\sum_{i=1}^{N} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t)[\{2\mathbf{S}_r - diag(\mathbf{S}_r)\} - \{2\boldsymbol{B}_{ir} - diag(\boldsymbol{B}_{ir})\}] = \mathbf{0}$$

$$\Rightarrow \frac{1}{2}\sum_{i=1}^{N} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t)[2\{\mathbf{S}_r - \boldsymbol{B}_{ir}\} - diag(\mathbf{S}_r - \boldsymbol{B}_{ir})] = \mathbf{0}$$

Let $\boldsymbol{C}_{ir} = \mathbf{S}_r - \boldsymbol{B}_{ir}$

$$\Rightarrow \frac{1}{2}\sum_{i=1}^{N} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t)[2\boldsymbol{C}_{ir} - diag(\boldsymbol{C}_{ir})\}] = \mathbf{0}$$

$$\Rightarrow \sum_{i=1}^{N} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t)\boldsymbol{C}_{ir} - diag\left(\sum_{i=1}^{N} \frac{1}{2} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t)\boldsymbol{C}_{ir}\right) = \mathbf{0}$$

Let $\boldsymbol{D}_{ir} = \frac{1}{2}\sum_{i=1}^{N} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t)\boldsymbol{C}_{ir}$

$$\Rightarrow 2\boldsymbol{D}_{ir} - diag(\boldsymbol{D}_{ir}) = \mathbf{0}$$

$$\Rightarrow 2\boldsymbol{D}_{ir} = diag(\boldsymbol{D}_{ir}) \qquad (\text{A}.30)$$

The only solution to $(\text{A}.30)$ above is $\boldsymbol{D}_{ir} = \mathbf{0}$.

$$\Rightarrow \frac{1}{2}\sum_{i=1}^{N} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t)\boldsymbol{C}_{ir} = \frac{1}{2}\sum_{i=1}^{N} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t)(\mathbf{S}_r - \boldsymbol{B}_{ir}) = \mathbf{0}$$

$$\Rightarrow \mathbf{S}_r = \frac{\sum_{i=1}^{N} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t)(\boldsymbol{B}_{ir})}{\sum_{i=1}^{N} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t)} = \frac{\sum_{i=1}^{N} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t)(\mathbf{x}_i - \bar{\mathbf{x}}_r)(\mathbf{x}_i - \bar{\mathbf{x}}_r)^T}{\sum_{i=1}^{N} p(r|\mathbf{x}_i, \boldsymbol{\Phi}^t)} \qquad (\text{A}.31)$$

$$\forall r = 1,2,\dots,R$$

In summary, the parameter estimates $\boldsymbol{\Phi}_r = (\omega_r, \bar{\mathbf{x}}_r, \mathbf{S}_r)$ for the Gaussian mixture model,

$$p(\mathbf{x}_i|\boldsymbol{\Phi}_r) = \sum_{r=1}^{R} \omega_r \, p(\mathbf{x}_i|\bar{\mathbf{x}}_r, \mathbf{S}_r)$$

are,

$$\omega_r = \frac{1}{N}\sum_{i=1}^{N} p(r|\mathbf{x}_i, \mathbf{\Phi}_r) \qquad \forall r = 1,2,\dots,R$$

$$\bar{\mathbf{x}}_r = \frac{\sum_{i=1}^{N} \mathbf{x}_i p(r|\mathbf{x}_i, \mathbf{\Phi}_r)}{\sum_{i=1}^{N} p(r|\mathbf{x}_i, \mathbf{\Phi}_r)} \qquad \forall r = 1,2,\dots,R$$

$$\mathbf{S}_r = \frac{\sum_{i=1}^{N} p(r|\mathbf{x}_i, \mathbf{\Phi}_r)(\mathbf{x}_i - \bar{\mathbf{x}}_r)(\mathbf{x}_i - \bar{\mathbf{x}}_r)^T}{\sum_{i=1}^{N} p(r|\mathbf{x}_i, \mathbf{\Phi}_r)} \qquad \forall r = 1,2,\dots,R$$

where,

$$p(r|\mathbf{x}_i, \mathbf{\Phi}_r) = \frac{\omega_r p_r(\mathbf{x}_i|\bar{\mathbf{x}}_r, \mathbf{S}_r)}{\sum_{r=1}^{R} \omega_r p_r(\mathbf{x}_i|\bar{\mathbf{x}}_r, \mathbf{S}_r)} \qquad \forall r = 1,2,\dots,R$$

### A.4 Computing parameter estimates for a Gaussian density function

For the usual Gaussian distribution, there are no hidden subclasses with a particular class (i.e. $r = 1$ and $\omega_r = 1$) such that the parameter estimates for the mean and covariance matrix are:

$$\bar{\mathbf{x}} = \frac{\sum_{i=1}^{N} \mathbf{x}_i}{N} \qquad (A.32)$$

$$\mathbf{S} = \frac{\sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T}{N} \qquad (A.33)$$

since $p(r|\mathbf{x}_i, \mathbf{\Phi}^t) = 1$ in (A.29) and (A.31). The sample based MLE of the covariance matrix above is usually corrected to its unbiased estimate:

$$\mathbf{S} = \frac{\sum_{i=1}^{N} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T}{N - 1} \qquad (A.34)$$

# References


1. Bilmes, J. A. (1998). *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models.* Retrieved from http://www.icsi.berkeley.edu

2. Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B, 39,* 31-38.

3. Fukunaga. (1990). *Introduction to Statistical Pattern Recognition*. San Diego: Academic Press.

4. Jensen, J. L. W. V. (1906). Sur les fonctions convexes et les inegalites entre les valeurs moyennes. *Acta Mathematica 30*(1), 175-193. doi:10.1007/BF02418571

# Appendix B: Variable Coding

The variable coding used for the dataset in this analysis is presented in this section as follows (Hofmann, 1994):

DEFAULTSTATUS: binary outcome variable

- 0: non- defaulters
- 1: defaulters

BALACC: status of existing bank account in German currency: deutsche marks (DM)

- 1: no running account
- 2: no balance or debit
- 3: $0 \leq bank\ balance < 200DM$
- 4: $bank\ balance \geq 200DM$

DURLOAN: duration of loan in months

PAYPREVCRED: Payment of previous credits

- 0: hesitant payment of previous credits
- 1: problematic running account / there are further credits running but at other banks
- 2: no previous credits / paid back all previous credits
- 3: no problems with current credits at this bank
- 4: paid back previous credits at this bank

LOANAMT: The loan amount borrowed in German Duetsche Marks (DM)

SAVINGS: Value of savings or stocks

- 1: not available / no savings
- 2: $value\ of\ savings\ or\ stock < 200DM$
- 3: $100 \leq value\ of\ savings\ or\ stock < 500DM$
- 4: $500 \leq value\ of\ savings\ or\ stock < 1000DM$
- 5: $value\ of\ savings\ or\ stock \geq 1000DM$

TIMEEMPL: Time applicant been employed by current employer

- 1: Unemployed
- 2: $year \leq 1$
- 3: $1 \leq years < 4$

- 4: $4 \leq years < 7$
- 5: $years \geq 7$

AGE: Age of applicant in years

INSTALMNT: Instalment rate as a percentage of available or disposable income

- 1: $installment\ rate \geq 35$
- 2: $25 \leq installment\ rate < 35$
- 3: $20 \leq installment\ rate < 25$
- 4: $installment\ rate < 20$

MARITAL.SEX:- Marital status or sex

- 1: a male who is divorced or separated
- 2: a female who is divorced or separated or married
- 2: a single male applicant
- 3: a male applicant who is married or widowed
- 4: a single female applicant

GUARANTOR: Guarantors of the loan

- 1: none
- 2: co-applicant
- 3: guarantor

DURHSE: Period applicant been living in current house

- 1: $year \leq 1$
- 2: $1 \leq years < 4$
- 3: $4 \leq years < 7$
- 4: $years \geq 7$

VALASSET: Most valuable available assets

- 1: no assets
- 2: Car/other
- 3: Savings contract with a building society / Life insurance
- 4: Ownership of house or land

CURCRED: Current or further running credits

- 1: at other banks
- 2: at department store or mail order house
- 3: no further running credits

HSETYPE: Type of house or apartment

- 1: free apartment
- 2: rented flat
- 3: owner-occupied flat

HISTCRED: Number of previous credits at this bank (including the running one)

- 1: one
- 2: two or three
- 3: four or five
- 4: six or more

JOBTYPE: Type of job or occupation of applicant

- 1: unemployed / unskilled with no permanent residence
- 2: unskilled with permanent residence
- 3: skilled worker / skilled employee / minor civil servant
- 4: executive / self-employed / higher civil servant

DEPENDENTS: Number of people dependent on the applicant

- 1: more than 3
- 2: 0 to 2

TEL: Telephone ownership

- 1: no
- 2: yes

ALIEN: Foreign worker

- 1: yes
- 2: no

# References

1. Hofmann, H. (1994). *Datasets at the Department of Statistics and the SFB386*. Retrieved from http://www.stat.uni-muenchen.de/service/datenarchiv/kredit/kreditvar_e.html