A Dual Approach to Linear Discriminant Analysis
Application to the Financial Sector

Imperial College London
Supervised by Dr Ioanna Papatsouma

Kyle Patel (01509564)
Kailun Peng (01575789)
Raquel Rogero Morón (01564308)
Alix Vermeulen (01564896)

June 2020

Acknowledgment

'We hereby declare this is our own work except where otherwise stated.'

Kyle Patel	Date
Kailun Peng	16/06/2020
Raquel Rogero Morón	
Alix Vermeulen	

Abstract

Linear discriminant analysis (LDA) has its roots in an approach known as Fisher's linear discriminant analysis, a method used in statistics, pattern recognition and machine learning to find a linear combination of features that characterises or separates two or more classes of objects or events. It focuses simultaneously on the maximization of the between-class variance and the minimization of the within-class variance and it has been used to solve various real-life problems, such as emotion recognition by classifying the emotions or face recognition by reducing the number of pixel values. The aim of this project is to study the LDA as a dimensionality reduction algorithm while retaining as much information as possible and as a classifier algorithm, apply both algorithms to real-life datasets and assess their performance. We will study an application of LDA as a classification method and as a dimensionality reduction technique on the financial sector.

Keywords: Linear Discriminant Analysis, Classification Algorithm, Dimensionality Reduction Techhnique.

List of Abbreviations

LD	Linear Discriminant
LDA	Linear Discriminant Analysis
LR	Logistic Regression
MDA	Multiple/Multi-class Discriminant Analysis
MFA	Marginal Fisher Analysis
RLDA	Robust Linear Discriminant Analysis
RMFA	Regularized Marginal Fisher Analysis
SCRDA	Shrunken Centroids Regularized Discriminant Analysis

Contents

Chapter 1

Introduction

1.1 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a classification method and dimensionality reduction technique based on R. A. Fisher's Linear Discriminant Analysis, a method he developed in 1936 traditionally only for binary classification problems involving two groups [1]. In 1948, C. R. Rao extended this theory of discriminant functions as Multiple Discriminant Analysis (MDA) for multi-class cases [2].

The main objective of dimensional reduction techniques is to project the group characteristics observed on higher dimensional spaces onto smaller subspaces without losing information. These techniques allow us to reduce the number of dimensions, the difficulty of the computations and also to avoid a curse of dimensionality problem [3].

The dimensional reduction techniques can be mainly divided into two different approaches, the supervised and the unsupervised approaches. In the first one, the correct output is known and therefore the technique can be corrected, whereas in the second one there is no known correct answer. Linear Discriminant Analysis is one of the most famous techniques of the supervised approach as it takes into consideration the class labels [4].

In order to transform the features into a lower dimensional space while maximizing the separability between the a priori defined groups, this method derives the linear combination of the continuous independent variables. There exists two different approaches to do this: the class-dependent transformation maximizes the ratio of between class variance to within class variance and transforms the data set depending on their class, whereas the class-independent transformation maximizes the ratio of overall variance to within class variance and then transforms the data uniformly [4, 5].

To be able to derive this linear relation, this technique requires four strong assumptions that are easily violated on a large number of real life situations. We need to assume that each independent variable follows a Gaussian distribution and that the group variance-covariance matrices are equal, also called homoscedasticity. We also need to assume there exists a linear relations among the independent variables and multicollinearity and singularity of these variables [6].

1.1.1 Motivation

LDA is a supervised method used commonly to classify data. It has become interestingly popular over time as we seek for patterns in data to extract meaningful conclusions to be able to solve problems. We will be studying a dual approach to this classification method: reducing the dimension of the problem as a pre-processing step for classification and simply the classification algorithm [4].

1.1.2 Advantages and Disadvantages of LDA

Advantages

LDA is a very simple and clear method that obtains strong results. It can also be expanded to a multi-class classifier unlike other methods such as Logistic Regression (LR). Also compared to LR, LDA still functions correctly even when the two classes are perfectly separable, unlike LR which fails in this case.

LDA overall is a method that is very simple to implement, and although it can have large matrix computations, these can still be calculated fast. It is also known to produce good results, although this is also due to the strong assumptions made as a precursor. LDA is also not complex to train.

Disadvantages

The strong assumptions of multivariate normality of the variables with equal variance is generally rare in practice. This implies, generally, that these are violated in order to obtain good results.

Generally, Fisher's linear discriminant is sensitive to outliers. This is because the ratio used to separate two distributions is based on the distance criterion using the L2-norm which is sensitive to outliers [7]. To solve this problem stronger techniques have been developed using Robust Linear Discriminant Analysis (RLDA) which are not as sensitive to outliers.

The small sample size problem is a well known issue within discriminant analysis. Due to the generally high number of dimensions of the data used, generally the sample space is much smaller i.e. when the total number of training samples is smaller than the number of dimensions of the feature vector. In multi-class Linear Discriminant Analysis, the within-class scatter matrix becomes singular, therefore its inverse cannot be calculated. To overcome this, the inverse is approximated which therefore allows the computation of the orientation matrix. In more developed methods, this can be solved by adding a regularisation term (Regularized MFA) or by taking the exponential MFA in Marginal Fisher Analysis [8].

Finally LDA utilizes large matrix computations which, if a large number of dimensions are being used, can be computationally very expensive. Logistic Regression which is also used in class classification is not as computationally expensive. This method is predominantly used in binary classification which makes use of a logistic function to classify the data. This method also does not require any assumptions on the in group variance unlike LDA.

1.2 Literature Review

LDA has a wide range of applications in the modern world.

As LDA is a classification method, it is of great use for pattern recognition. One of its most prominent applications is face recognition. LDA is used in reducing the features of data from images, which helps facial recognition systems identifying a person from an image by comparing its similarity with facial images in the database [9]. It can also be used to re-identify a person of interest in a large group of people [10].

Apart from this, LDA is also applied to emotion recognition, hand motion categorisation and the classification process of speech and music classification by creating a linear boundary between two different classes based on the centroids of the training data [11].

LDA can also be used to evaluate food quality. By performing different measurements on the product, it is possible to apply LDA as a classification method in order to determine if the product is safe for consumption [12].

The application of this classification method and dimensionality reduction technique is also very common in biostatistics. LDA can be used to classify genes with Shrunken Centroids Regularized Discriminant Analysis (SCRDA) [13].

Matrix discriminant analysis (MDA), a generalization of LDA for the data in matrix form, can also be used to classify the color changes of the chemo-responsive dyes in order to identify specific chemical toxicants [14].

In the medical field, LDA plays a significant role to help doctors with disease diagnostics. For instance, breast cancer has been one of the biggest risks that threatens women's health worldwide. LDA can be used to distinguish between benign and malignant lumps and classify the risk level of breast cancer of a patient with high accuracy [15, 16]. In addition to this, the condition of a patient with a certain disease can be categorised into 'mild', 'moderate' or 'severe' based on variables and the results provide information on what kind of treatments the patients should receive.

Furthermore, LDA could also be applied to other subjects such as social and behavioural sciences. LDA has been used as a classification method in generating prediction models of recidivism with high precision [17].

LDA is also proved to be useful in business. By analysing variables regarding the financial state of a company, the prediction of the likelihood of bankruptcy can be made. The model published by Edward Altman in 1968 is still widely applied nowadays [18]. We will study this more in depth as it is our chosen field of application.

1.3 Report Structure

The aim of this report is to assess the accuracy of two methods used to perform LDA on an example in the financial sector. The two methods are: simply classifying the data using a classification algorithm and secondly, reducing the dimension of the problem then classifying the data.

In the Methodology section, we cover a detailed explanation of LDA and the dual approach along with the important assumptions that a set of data must satisfy in order to be able to use LDA.

Part 3 focuses on an application of the two LDA methods discussed above to a real-world problem. We evaluate the accuracy of each method and discuss which one is more suitable in Part 4.

Chapter 2

Methodology

2.1 Assumptions

As already said in the introduction, LDA requires four strong assumptions that are not always respected in numerous cases. For example, in behavioural and social science, business, and biomedical sciences it is very difficult to find data that follow the required assumptions. Luckily, we will see that this method is quite robust when facing these problems [2, 19]. It has been shown that LDA as a dimensionality reduction technique is quite robust to the violation of these requirements whereas LDA as a classification method, its predictive power, is greatly affected in case these assumptions are not satisfied, but is still useful [3].

Independence

LDA has one categorical dependent variable corresponding to the group variable (e.g. bankruptcy or not) and then deals with continuous independent variables, therefore it is assumed that all the variables are independent excluding the group one.

Multicollinearity

To be able to derive the required linear relation, it is necessary to assume there exists a linear combination between the different independent variables. Multicollinearity between the independent variables can also cause a decrease on the predictive power of this classification method. This takes place when there exists one or more linear relation between some of the independent variables, to solve this it is necessary to reduce the number of independent variables [6, 20]. Some empirical studies have shown that multicollinearity can be a serious issue in discriminant analysis, especially in stepwise procedures, but it mostly weakens its prediction ability [21].

Gaussian distribution

LDA assumes the data of each different group follows a Gaussian distribution. This assumption is commonly violated by many measurements from most fields but is generally ignored as LDA is still a useful method but not as accurate.

Homoscedasticity

LDA requires the assumption that the variance-covariance matrices of each group are equal, what is called homoscedasticity. This assumption is also rarely satisfied by most data sets from most fields of applications.

2.2 Univariate and Multivariate Gaussian Distribution

Univariate Gaussian Distribution

From the assumption stated in section 2.1, the predictor $X = (X_1, X_2, ..., X_p)$ follows Gaussian distribution. Firstly, we start with a simple case, where $p = 1$. In this case, the observation Z depends on only one variable X.

For a continuous variable X which is drawn from Gaussian distribution, we write $X \sim N(\mu, \sigma^2)$, where $\mu = E(X)$ and $\sigma^2 = Var(X)$. Then the probability density function (PDF) is the following [25]:

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{(x-\mu)^2}{2\sigma^2} \right\}, -\infty < x < \infty \tag{2.1}$$

Multivariate Gaussian Distribution

Now suppose $p \geq 2$. In this case, the observation Z depends on p variables X_1, X_2,...,X_p. Again, $\boldsymbol{X} = (X_1, X_2, ..., X_p)$ has a multivariate Gaussian distribution, as each of the components is assumed to follow a univariate Gaussian distribution. The multivariate Gaussian density is defined as

$$f(\boldsymbol{X}) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{ -\frac{1}{2}(\boldsymbol{X}-\mu)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{X}-\mu)\right\} \tag{2.2}$$

We write $X \sim N(\mu, \boldsymbol{\Sigma})$.

Similarly, $\mu = E(\boldsymbol{X}) = (\mu_1, \mu_2, ..., \mu_p)^T$, where $\mu_k = E(X_k)$ for all $k = 1, 2, ..., p$. $\boldsymbol{\Sigma} = Cov(\boldsymbol{X})$, which is a $p \times p$ covariance matrix of $\boldsymbol{X}$.[24]

$$\boldsymbol{\Sigma} = \begin{pmatrix} Var(X_1) & Cov(X_1, X_2) & \cdots & Cov(X_1, X_p) \\ Cov(X_2, X_1) & Var(X_2) & \cdots & Cov(X_2, X_p) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(X_p, X_1) & Cov(X_p, X_2) & \cdots & Var(X_p) \end{pmatrix} \tag{2.3}$$

When X_i and X_j are uncorrelated for all $i \neq j$, by the properties of covariance, $Cov(X_i, X_j)$=0. Hence in this case, $\boldsymbol{\Sigma}$=Diag(Var(X_1),Var(X_2),...,Var(X_p)). Figure 2.1 shows the bivariate Gaussian distribution. The graph on the left represents the distribution of 2 uncorrelated variables X_1 and X_2, whereas on the right hand side, $Cov(X_1, X_2)$=0.7.

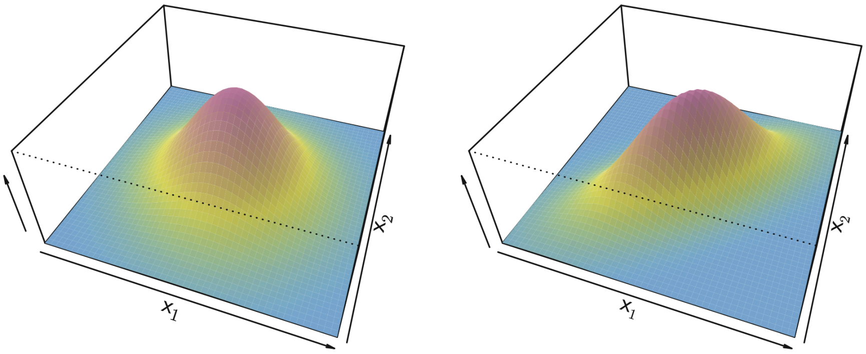


Figure 2.1: Bivariate Gaussian distribution with left: X_1 and X_2 uncorrelated, right: X_1 and X_2 have a correlation of 0.7 [24].

2.3 Classification

Given an input x, our goal is to find which class k it belongs to, this is achieved using a decision boundary: $G(x) = arg\max_k P(Z = k \mid X = x)$

We must therefore estimate $P(Z = k \mid X = x)$ and maximise it.

By Bayes' Theorem and the law of total probability we obtain:

$$
\begin{aligned}
P(Z = k \mid X = x) &= \frac{P(X = x \mid Z = k)P(Z = k)}{P(X = x)} && (2.4)\\
&= \frac{P(X = x \mid Z = k)P(Z = k)}{\sum_{i=1}^{n} P(X = x \mid Z = i)P(Z = i)} && (2.5)
\end{aligned}
$$

Simplifying notation, we can write the prior distribution of class k as: $P(Z = k) = \pi_k$.

By assumption, the data is Gaussian distributed so the likelihood can be written as:

$$
P(X = x \mid Z = k) = f_k(x) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma_k}|^{1/2}} \exp\left\{ -\frac{1}{2}(x - \mu_k)^T \boldsymbol{\Sigma_k}^{-1}(x - \mu_k)\right\} \qquad (2.6)
$$

where μ_k is the mean of the inputs from class k and $\boldsymbol{\Sigma_k} = \boldsymbol{\Sigma}$ is the variance-covariance matrix that is equal for all classes, again by assumption. $|\boldsymbol{\Sigma_k}|$ is the determinant of the variance-covariance matrix.

The denominator of Equation 2.5 does not depend on the class k hence we can disregard it:

$$
\begin{aligned}
P(Z = k \mid X = x) &\propto f_k(x)\pi_k && (2.7)\\
&\propto \frac{\pi_k}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{ -\frac{1}{2}(x - \mu_k)^T \boldsymbol{\Sigma}^{-1}(x - \mu_k)\right\} && (2.8)\\
&\propto \pi_k \exp\left\{ -\frac{1}{2}(x - \mu_k)^T \boldsymbol{\Sigma}^{-1}(x - \mu_k)\right\} && (2.9)\\
&= C\pi_k \exp\left\{ -\frac{1}{2}(x - \mu_k)^T \boldsymbol{\Sigma}^{-1}(x - \mu_k)\right\} && (2.10)
\end{aligned}
$$

where C is a constant of proportionality.

To maximise the posterior probability, we take the log of both sides:

$$
\log(P(Z = k \mid X = x)) = \log(C) + \log(\pi_k) - \frac{1}{2}(x - \mu_k)^T \boldsymbol{\Sigma}^{-1}(x - \mu_k) \qquad (2.11)
$$

$$
\begin{aligned}
\log(\pi_k) - \frac{1}{2}(x - \mu_k)^T \boldsymbol{\Sigma}^{-1}(x - \mu_k) &= \log(\pi_k) - \frac{1}{2}[x^T \boldsymbol{\Sigma}^{-1} x + \mu_k^T \boldsymbol{\Sigma}^{-1}\mu_k] + x^T \boldsymbol{\Sigma}^{-1}\mu_k && (2.12)\\
&= D + \log(\pi_k) - \frac{1}{2}\mu_k^T \boldsymbol{\Sigma}^{-1}\mu_k + x^T \boldsymbol{\Sigma}^{-1}\mu_k && (2.13)
\end{aligned}
$$

where $D = -\frac{1}{2}x^T\mathbf{\Sigma}^{-1}x$

and hence we obtain the linear discriminant functions for each class k:

$$\delta_k(x) = \log(\pi_k) - \frac{1}{2}\mu_k^T\mathbf{\Sigma}^{-1}\mu_k + x^T\mathbf{\Sigma}^{-1}\mu_k \tag{2.14}$$

In most cases, we do not have the values π_k, μ_k and $\mathbf{\Sigma}$, hence we need to estimate them using our training data. These are given by:

$$\hat{\pi}_k = \frac{N_k}{N} \quad \text{where } N_k \text{ is the number of observations in class } k \tag{2.15}$$

$$\hat{\mu}_k = \frac{1}{N_k}\sum_{z_i=k} x_i \tag{2.16}$$

$$\hat{\mathbf{\Sigma}} = \frac{1}{N-K}\sum_{k=1}^{K}\sum_{z_i=k}(x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T \tag{2.17}$$

$$\tag{2.18}$$

We can therefore re-write our decision boundary as: $G(x) = arg\max_k \delta_k(x)$

We assign x to the class k that yields the greatest discriminant function (largest posterior probability) [22].

2.4 Reduction of Dimensions and Classification

LDA can be a very effective method to reduce the number of dimensions in the data whilst still retaining as much of the relevant information required for classification. In this method we will first examine the ideas Fisher introduced in 1936 and how it can be extended for more classes.

Suppose first for the binary class case we take a P-dimensional input vector x. We are going to project this down to a one dimensional space by applying a linear combination on all of the features.

$$y = w^T x \tag{2.19}$$

Then if we impose a condition on y, for example y $\geq$ c as class Z_1 and y $<$ c as Z_2. An important concept to understand is that LDA must select a projection that ensures no information is lost.

The projection we choose is important to show good separability.

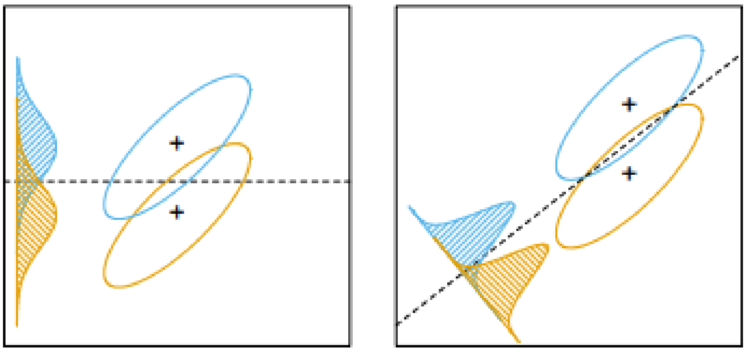


Figure 2.2: Difference in separability between a projection along the class means and along Fisher's Linear Discriminant [22]

Figure 2.2 illustrates two different projections that can be used. It is clearly more beneficial to use the projection on the right as this provides significantly less overlap than the projection on the left.

For this binary case based methodology lets consider there being N_1 points in class Z_1 and N_2 points in class Z_2. The 2 classes have mean vectors:

$$m_i = \frac{1}{N_i} \sum_{z \in Z_i} x_z \qquad \text{for i = 1,2 in this case.} \tag{2.20}$$

As discussed before, it would be incorrect to seek out the projection onto the plane connecting the class means. Instead Fisher proposed we seek a projection that gives a large between class variance whilst maintaining a small variance within each class. The within class variance for a projected class is:

$$s_i^2 = \sum_{z \in Z_i} (y_z - m_i)^2 \qquad \text{where } y_z = w^T x_z \tag{2.21}$$

By the definition of variance, we can define the total within class variance for the entire data set to be $s_1^2 + s_2^2$. We have now developed the Fisher criterion which can be expressed as a ratio of the between class variance to the within class variance:

$$J(w) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \tag{2.22}$$

Therefore we need to maximise this with respect to w. First we rearrange this ratio to a better form. Explicitly in terms of w we have so far:

$$J(w) = \frac{w^T (m_2 - m_1)(m_2 - m_1)^T w}{w^T (C_1 + C_2) w} \qquad \text{where } C_i \text{ is the covariance matrix of } x_i \tag{2.23}$$

Now we introduce the concept of the scatter matrix in order to massage this equation a bit more. The scatter matrix for a set of points $x_1, x_2, \ldots, x_n$ is defined as $S = \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T$. So we therefore have $S_i = N_i C_i$. Instead of covariance in Fisher's criterion we will use the scatter matrices we have just defined so we have now:

$$J(w) = \frac{w^T (m_2 - m_1)(m_2 - m_1)^T w}{w^T (S_1 + S_2) w} \tag{2.24}$$

Now we introduce:

$$S_B = (m_2 - m_1)(m_2 - m_1)^T, \tag{2.25}$$

$$S_W = S_1 + S_2 \tag{2.26}$$

This therefore allows us to define Fisher's criterion as:

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \tag{2.27}$$

It is worth to note that the RHS of the equation 2.27 is also known as the generalised Rayleigh Quotient which is an eigenvalue problem. Differentiating with respect to w and setting equal to 0, we obtain the equation:

$$\begin{aligned} S_W w &= \frac{w^T S_W w}{w_T S_B w} S_B w \qquad (2.28) \\ &= \frac{w^T S_W w}{w^T S_B w} (m_2 - m_1)(m_2 - m_1)^T w \qquad (2.29) \\ &= \frac{w^T S_W w}{w^T S_B w} (m_2 - m_1)^T w (m_2 - m_1) \qquad (2.30) \end{aligned}$$

Since $\frac{w^T S_W w}{w^T S_B w} (m_2 - m_1)^T w$ is a scalar value we omit this. This means that

$$w \propto S_W^{-1}(m_2 - m_1) \tag{2.31}$$

This can then be normalized to obtain the optimal w.

Now we will examine how this method can be extended for the case of multiple classes. For this problem let us assume we have K classes. We define:

$$X_k = \{x_i | y_i = k\}, \qquad \text{the subset of input } x \text{ that once transformed belongs to the kth class} \tag{2.32}$$

$$m_k = \frac{1}{N_k} \sum_{x \in X_k} x \tag{2.33}$$

$$S_k = N_k C_k = \sum_{x \in X_k} (x - m_k)(x - m_k)^T \tag{2.34}$$

The overall within class scatter matrix is therefore the sum of the scatter matrices of the K classes. Therefore

$$S_W = \sum_{k=1}^{K} S_k \tag{2.35}$$

However it is not the same story for the S_B matrix which cannot be as easily extended. First we must introduce the scatter for the entire data set.

$$S_T = \sum_{i=1}^{N} (x_i - m)(x_i - m)^T \quad \text{where in this case } m = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{2.36}$$

We define this matrix to be the total scatter matrix. Now we subtract the within class scatter matrix from the total scatter matrix. This gives us, after massaging the equations our definition of the between class scatter matrix.

$$S_T - S_W = \sum_{k=1}^{K} N_k (m_k - m)(m_k - m)^T = S_B \tag{2.37}$$

So again we attempt to maximise

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \tag{2.38}$$

This as mentioned before is the Rayleigh Quotient. This equation has solutions determined by the eigenvectors of $S_W^{-1}S_B$, when S_W is invertible. The optimal w is given by the corresponding eigenvector of the largest eigenvalue.

It is also worth to note that for a K class problem we can only extract up to $(K-1)$ features. This is because S_B is composed of the sum of K matrices, of which only $(K-1)$ of these matrices are independant. This is a direct consequence of $m = \frac{1}{N}\sum_{i=1}^{N} x_i = \frac{1}{N}\sum_{k=1}^{K} N_k m_k$. Therefore S_B has at most a rank of $(K-1)$ and so therefore there at most $(K-1)$ non-zero eigenvalues. This means that the projection onto a $(K-1)$ dimensional subspace spanned by the eigenvectors of S_B does not effect the value of $J(w)$ [23].

From this we have can obtain the reduced data by applying the w obtained. Then with this data we can classify as discussed earlier.

Chapter 3

Application

3.1 Methodology in R

For our investigation, we will determine the accuracy of the two classification techniques on the prediction of Bankruptcy. We collected our data from UCI Machine Learning Repository. Our data, [26], exhibits 64 statistics based around Bankruptcy which can be found at 5.1. Here, we make use of them along with our assumptions to model using LDA. We investigate here the accuracy of the 2 classification methods from LDA, to predict the status of Polish companies in reference to Bankruptcy for up to 5 years.

As seen later in 5.2, we make use of several libraries in R. Notably, the MASS library which contains the lda function. As can be seen in 5.2, during the reduction of dimensions, we found our within scatter matrix to be singular. This is a consequence of the small sample size problem as discussed earlier. In our training data, there was less than 64 data points for some of the classes resulting in a singular within scatter matrix. To overcome this for our investigation, we instead took the pseudo-inverse of the within scatter matrix in order to obtain the projection.

Both methods are trained on the same data and predict the outcome of the same data. In total we had 19,967 data points after the data was prepared for the investigation. We split this into a 4:1 ratio between training and testing respectively. This resulted in 15,972 data points being used for training and 3,995 data points being used for testing.

The data set we obtained are divided into 5 year groups. In the n^{th} year group, the observations are categorized either as 0 (will keep operating in the next $5-n$ years) or 1 (will collapse in $5-n$

years). Companies of two financial states were analysed in two different time periods: bankrupt companies in 2000-2012, while others are analysed in 2007-2013. For simplicity, we assumed all companies in all year group that belong to class 0 will not bankrupt in the next 5 years.

3.2 Classification

First of all, we use our LDA code only for classification. To visualise our data, we will be using scatter plots (biplots) and stacked histograms with linear discriminant function values.

The confusion matrix we obtain from R is:

$$\begin{array}{ccccccc} & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 3854 & 16 & 20 & 18 & 18 & 7 \\ 1 & 13 & 0 & 0 & 0 & 1 & 0 \\ 2 & 7 & 1 & 0 & 0 & 0 & 0 \\ 3 & 14 & 0 & 0 & 0 & 0 & 0 \\ 4 & 4 & 0 & 0 & 0 & 0 & 0 \\ 5 & 19 & 0 & 1 & 0 & 0 & 0 \end{array}$$

We observe that only the values on the diagonal are the ones that have been correctly classified, here only 3854 in class 0. Anything in the off-diagonal is a misclassification. For example, our classification method established that 16 businesses were in class 1 when they were actually in class 0, and it said 13 businesses were in class 0 when they actually were in class 1.

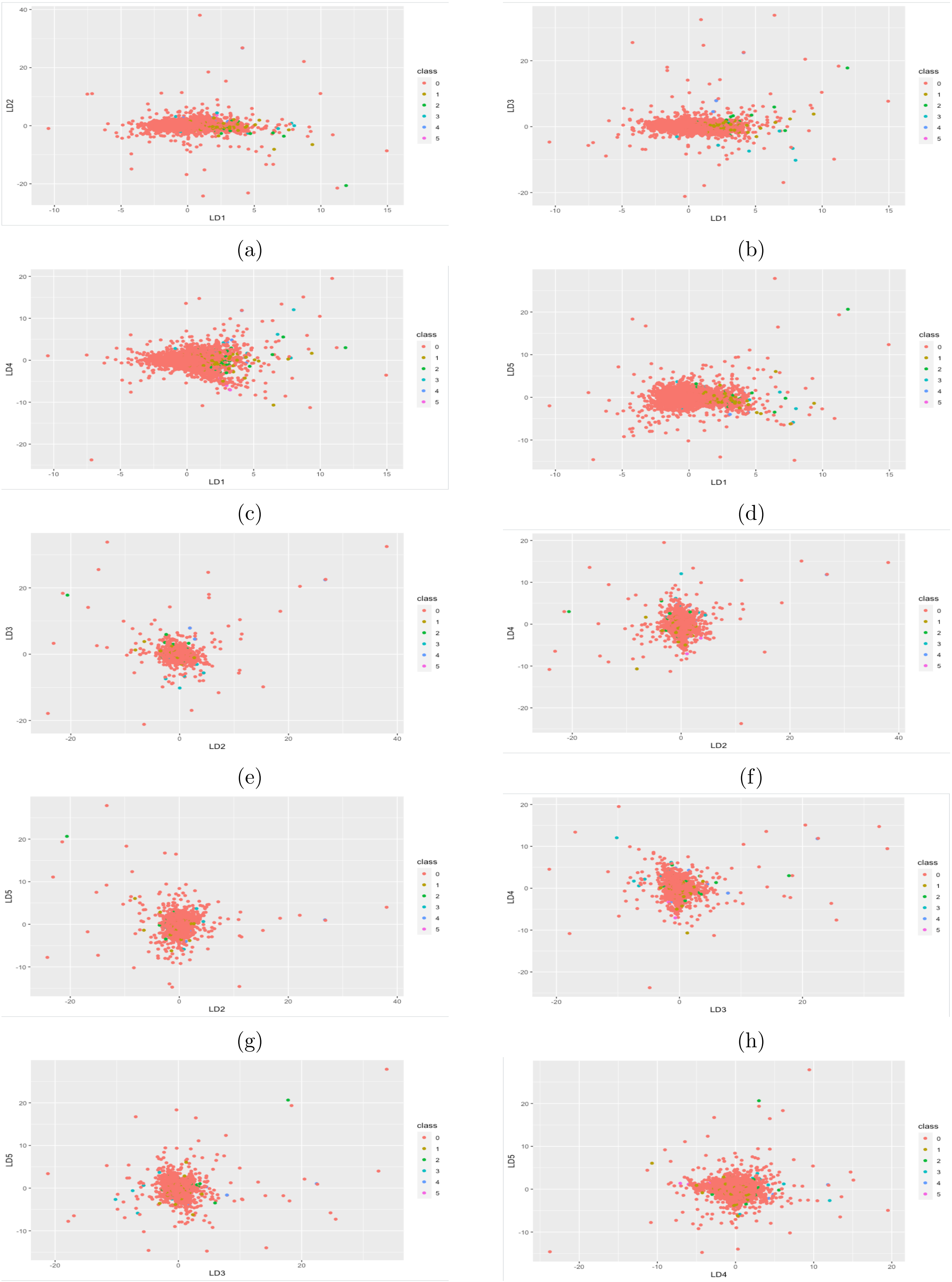
Let A be the confusion matrix obtained. If we do the ratio of the trace of A (sum of the elements in the diagonal) over the sum of all the elements of A (sum A), we obtain: $\frac{trace(A)}{sumA} = 0.9652$. This number corresponds to the accuracy of the classification method.

It is possible to observe the overall statistics of our classification thanks to a function in R. As we can see in the screenshot 5.1 of section 5.2 of the Appendix, the accuracy of our classification is 96.47%.

It can be seen on the screenshot 5.2 of section 5.2 of the Appendix that the sensitivity of our classification for class 0 is very close to 1 (0.9854), therefore 98.54% of actual businesses in class 0 are correctly classified by our LDA code. For the other classes the sensitivity is equal to 0 as none of the businesses in those classes are correctly classified as we can see in the confusion matrix (all the diagonal entries are 0 apart from the first one).

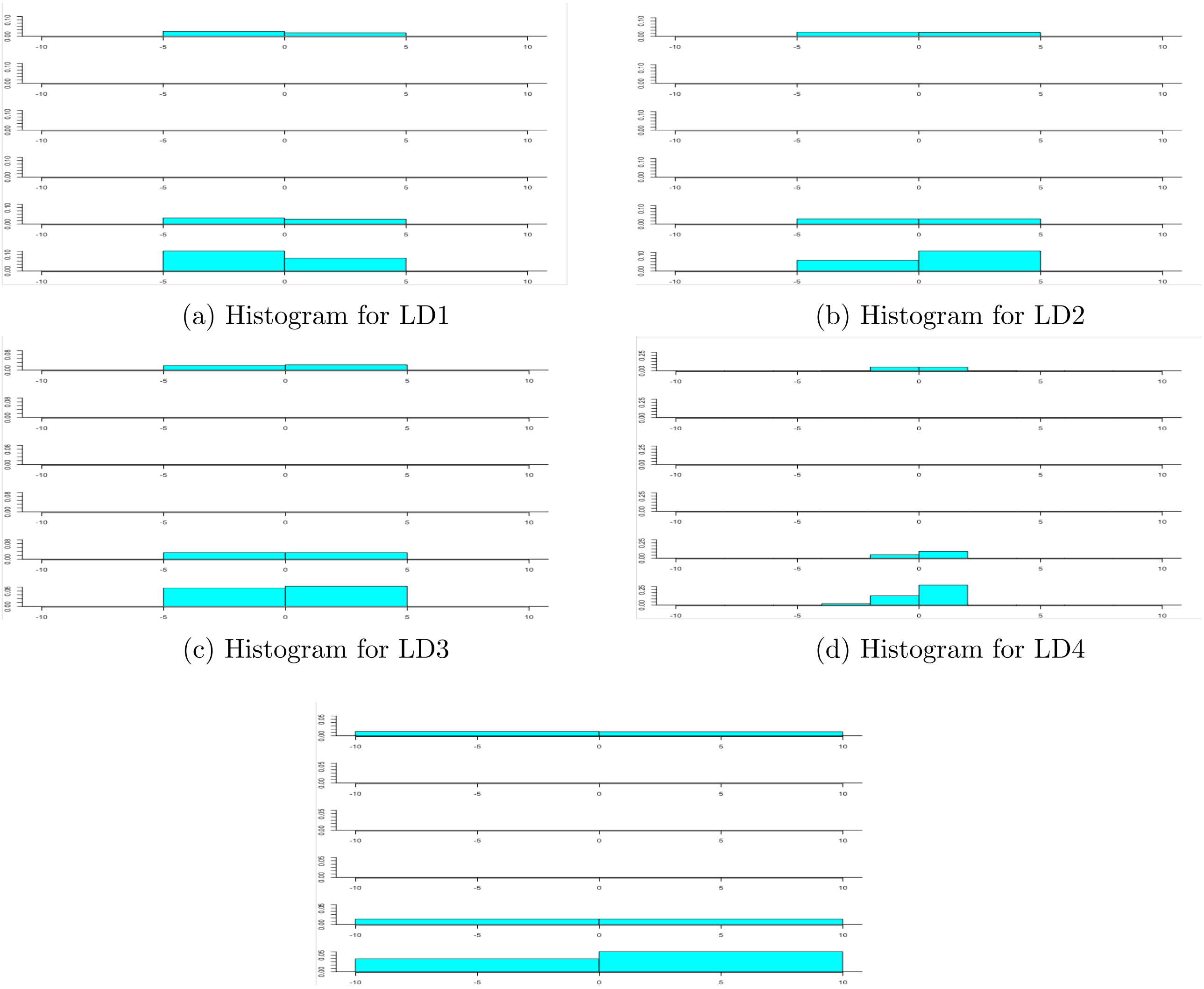
The specificity of class 0 shows us that only 5.95% of actual businesses that are not in class 0 are accurately classified as such, whereas more than 99.4% of any other businesses that is not in classes 1,2,3,4 or 5 is correctly classified as such.

By using the data points from the training data it is possible to determine the coefficients of 5 different linear discriminant functions (denoted by LD1 to LD5). Figure 3.1 illustrates all the possible combinations of one Linear Discriminant Function against each other, each colour associated with a different class. Even if it is possible to see that some of the data points in the same class keep close to each other (e.g. data points in class 1 in some of the graphs), ideally it would be possible to identify 6 different groups with each different colour. It is even harder to visualize the separation of the groups due to the large difference of the number of data points in group 0 (red) compared to the number of data points of the other groups.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

Figure 3.1: All possible combinations of one linear discriminant function against another



(a) Histogram for LD1

(b) Histogram for LD2

(c) Histogram for LD3

(d) Histogram for LD4

(e) Histogram for LD5

Figure 3.2: Stacked histograms for the 5 linear discriminant function, the 6 levels correspond to the 6 classes starting with the upper most as class 0

Each stack of Figure 3.2 is based only on the coefficients of one of the Linear Discriminant functions LD1 to LD5. Each level of the histograms represents one of the classes starting with the upper most as class 0. Due to the limitations of our data, there is a lack of separation between the different histograms of this Figure. Ultimately, with an ideal set of data, we would observe very little or no overlap between the different histograms of the 6 classes.

Our first classification technique (without reduction of dimensions) on the prediction of Bankruptcy has a very high accuracy of 96.47% but due to the limitations of our data, in fact, more than 95% of our data points are in class 0, the sensitivity of our technique is very high for class 0 and 0 for any other class. By similar reasons, the specificity of class 0 is very low compared to a specificity greater than 99% for any other class. Our limited data also affects the visualization of the different classes in representations like Figure 3.1.

3.3 Reduction before Classification

First of all, we use our LDA code to reduce the dimensions and then for classification.

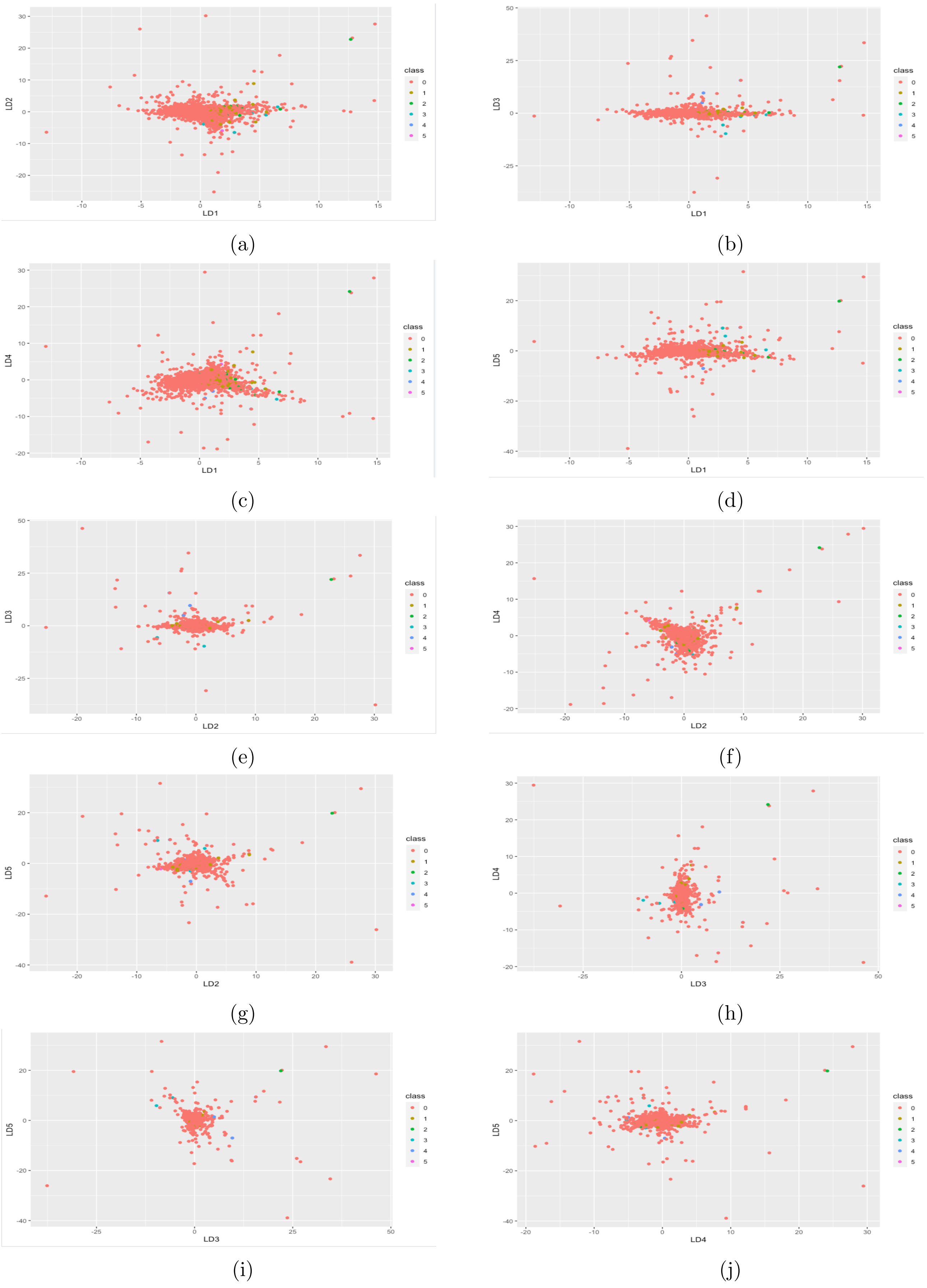
Similarly, the confusion matrix we obtain is:

$$\begin{array}{ccccccc} & 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 3881 & 16 & 22 & 18 & 18 & 7 \\ 1 & 17 & 0 & 0 & 0 & 1 & 0 \\ 2 & 2 & 1 & 0 & 0 & 0 & 0 \\ 3 & 5 & 0 & 0 & 0 & 0 & 0 \\ 4 & 2 & 0 & 0 & 0 & 0 & 0 \\ 5 & 4 & 0 & 1 & 0 & 0 & 0 \end{array}$$

Again, only the observed values on the diagonal are the ones that have been correctly classified. Let B be the new confusion matrix obtained. If we do the ratio of the trace of B (the sum of the elements in the diagonal) over the sum of all the elements of B (sum B), we obtain: $\frac{trace(B)}{sumB} = 0.9715$. In Figure screenshot 5.3 of section 5.3 of the Appendix, it is possible to observe the overall statistics of our classification. As we can see, the accuracy of our classification after reducing the dimensions is 97.51%. Hence, our second classification is more accurate than without reducing the dimensions, however our first technique is still very accurate (It has an accuracy of 96.47% as it can be seen in section 3.2).

It can be seen on the screenshot 5.4 of section 5.3 of the Appendix that the sensitivity of our classification for class 0 is even closer to 1 (0.9923 compared to 0.9854 on our first technique), therefore 99.23% of actual businesses in class 0 are correctly classified by our LDA code with reduction of dimension. For the other classes the sensitivity is equal to 0 as in the confusion matrix all the diagonal entries are 0 apart from the first one, exactly as in our previous section.

The specificity of class 0 shows us that only 3.57% of actual businesses that are not in class 0 are accurately classified as such, this is an even smaller proportion than in our first method (Only 5.95%). The proportion of any other businesses that is not in classes 1,2,3,4 or 5 being correctly classified as such is also bigger than in our previous case.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

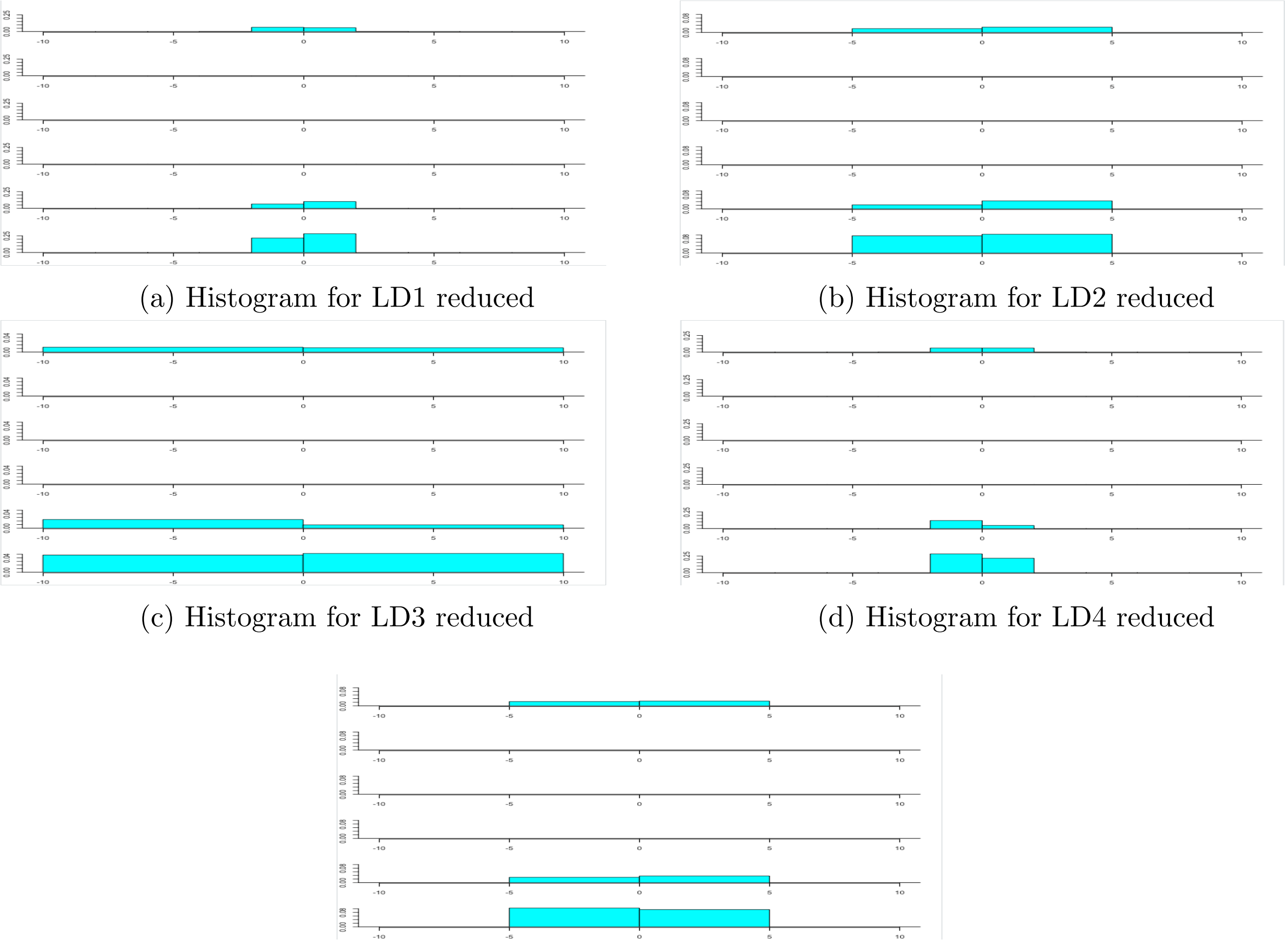
(h)

(i)

(j)

Figure 3.3: All possible combinations of one linear discriminant function against another after reduction of dimension

Figure 3.3 illustrates again all the possible combinations of one linear discriminant function against each other after reduction of dimension, each colour associated with a different class.



(a) Histogram for LD1 reduced

(b) Histogram for LD2 reduced

(c) Histogram for LD3 reduced

(d) Histogram for LD4 reduced

(e) Histogram for LD5 reduced

Figure 3.4: Stacked histograms for the 5 linear discriminant function in the reduction of dimension method, the 6 levels correspond to the 6 classes starting with the upper most as class 0

Similarly to Figure 3.2, we observe that Figure 3.4 lacks separation between the classes. This is again due to the limitations of our data.

Our second classification technique (done after reduction of dimension) on the prediction of Bankruptcy has an even higher accuracy of 97.51% than our first technique. Even if unfortunately our data is not ideal, the sensitivity of this method after reducing the dimensions is higher than without reducing the dimensions for class 0, but is still equal to 0 for all the other classes. On the opposite, specificity is even lower than in our first case for class 0 but keeps equally high for the rest of the classes. In any way, it is still difficult to separate the different groups as we can see in the two last Figures.

3.4 Summary of Results

From the confusion matrices obtained in Chapter 3, LDA has an overall accuracy of 96.47% (classification method) and 97.15% (reduction-classification method) respectively. This implies that LDA has relatively good performances as both methods in terms of accuracy. However, due to our unbalanced class distribution of the data points (see limits of the data in Section 3.5) the sensitivity and specificity of the classes changes brutally depending on if it is class 0 or any other class. In fact, both our classification methods rarely classify an observation in classes other than 0 and the proportion of correctly classified observations from class 1 to 5 is 0. This implies the low separability of LDA in our case.

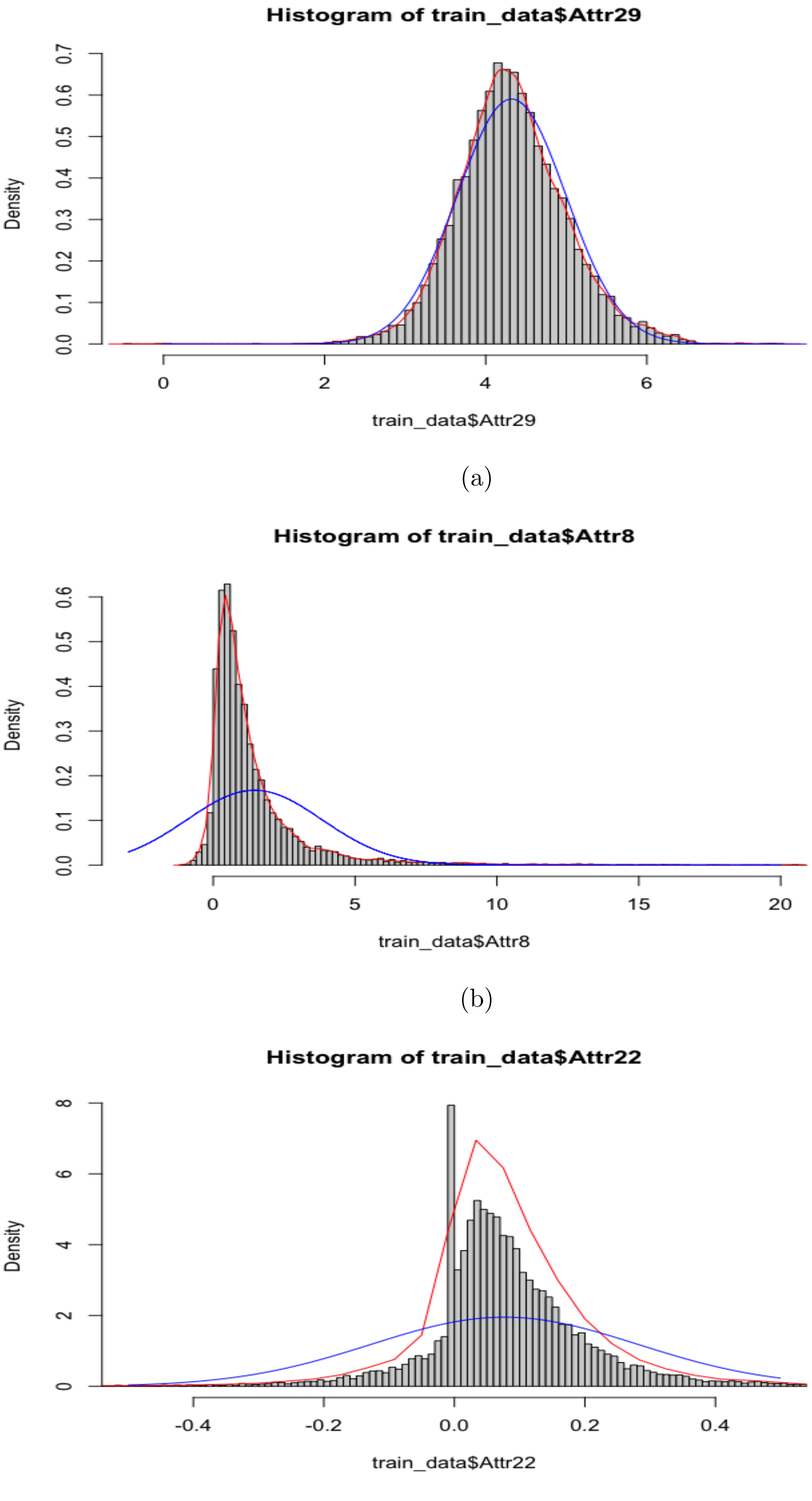
3.5 Discussions

3.5.1 Discussions on data

Assumptions and the actual data

As in Chapter 2.1, we have 4 assumptions: (1) independence, (2) multicollinearity of variables (3) multivariate normal distribution of variables and (4) homoscedasticity. When running LDA function to our data using R, it is shown in the program that our variables are multicollinear, which is exactly what we assumed - multicollinearity is satisfied.

By our third assumption, all of the variables should follow a Gaussian distribution. Some of our variables indeed follow a Gaussian distribution, however generally speaking, variables in our dataset are slightly skewed. For example, Figure 3.5 are plots of the distribution of three variables: X8, X22 and X29. Red curve represents empirical density, whereas the blue curve represents the normal distribution. It is clear that the distribution of X29 is close to a Gaussian distribution, however, this is not the case for X8 and X22.


Histogram of train_data$Attr29

(a)

Histogram of train_data$Attr8

(b)

Histogram of train_data$Attr22

(c)

Figure 3.5: Distribution of three variables: (a).X8, (b).X22 and (c).X29

In our fourth assumption, the variance-covariance matrix of each class is the same. However this is not true in our data set of Polish companies, which has a negative impact on the accuracy of LDA. Although our results imply that LDA has a high accuracy as both methods, this is due to the fact that the skewness of data which leads to high accuracy outweighs the violation of the assumption. Nevertheless, problems related to this could be improved by scaling the data so they follow a standard Gaussian distribution before applying LDA.

Limitations of the data

As already mentioned, LDA can be used for multiclass problems. However, as the prior distributions are estimated from the training set, if the number of data points belongs to a particular group is small, observations will rarely be classified in that group. This can explain why, even though both our two classification techniques (with our without dimensionality reduction) on the financial sector are highly accurate, data points outside of class 0(will not go bankrupt) are generally misclassified (see section 3.2 and 3.3).

Usually, there are more existing companies than bankrupt companies in real life. In fact, out of the total number of 15972 observations we used to train LDA, approximately 97.82% of them (15624 observations) are from class 0. It is clear that the class distribution is very unbalanced [27]. Having a large amount of observations in class 0 results in the within scatter matrix being singular. Although we solved this problem by taking pseudo-inverse of this matrix, LDA has a poor performance on separating classes, which is opposed to what we expected. Therefore, the properties of the actual data set used has a strong impact on the performances of LDA.

Chapter 4

Conclusion

Linear Discriminant Analysis is both a classification method and a dimensionality reduction technique based on a method developed by Fisher in 1936. It has become interestingly popular over time as it is a very simple and clear method to obtain strong results. Even though the assumptions can be easily violated, it is very widely used in the modern world with applications in various fields. In this report, we assessed the performances of the two possible techniques of application of LDA: simply using LDA as a classification algorithm and reducing dimensions before classification. We found out that reducing the dimension of the data set and then performing classification proved to be slightly more accurate than simply classifying the data in our application. However, the separability of LDA appeared to be low in both cases, which is different from we expected. This is mainly due to the imbalance of our data set. Hence we can conclude that the performance of LDA relies on the nature of the actual data set.

In order to eliminate problems involving unbalanced data sets, other methods such as Extreme Gradient Boosting model is proposed and proved efficient [26]. As seen in the introduction, even though LDA is one of the most famous classification techniques, it has numerous disadvantages. It would be very interesting to study one of the many other techniques that have been developed based on LDA when classifying other data sets. For instance, we have already seen that LDA requires four strong assumptions that are not generally met in practice. In order to solve this problem, stronger techniques such as Robust Linear Discriminant Analysis (RLDA) have been created.

Chapter 5

Appendix

5.1 Statistics Measured

X1 net profit / total assets

X2 total liabilities / total assets

X3 working capital / total assets

X4 current assets / short-term liabilities

X5 [(cash + short-term securities + receivables - short-term liabilities) / (operating expenses - depreciation)] * 365

X6 retained earnings / total assets

X7 EBIT / total assets

X8 book value of equity / total liabilities

X9 sales / total assets

X10 equity / total assets

X11 (gross profit + extraordinary items + financial expenses) / total assets

X12 gross profit / short-term liabilities

X13 (gross profit + depreciation) / sales

X14 (gross profit + interest) / total assets

X15 (total liabilities * 365) / (gross profit + depreciation)

X16 (gross profit + depreciation) / total liabilities

X17 total assets / total liabilities

X18 gross profit / total assets

X19 gross profit / sales

X20 (inventory * 365) / sales

X21 sales (n) / sales (n-1)

X22 profit on operating activities / total assets

X23 net profit / sales

X24 gross profit (in 3 years) / total assets

X25 (equity - share capital) / total assets

X26 (net profit + depreciation) / total liabilities

X27 profit on operating activities / financial expenses

X28 working capital / fixed assets

X29 logarithm of total assets

X30 (total liabilities - cash) / sales

X31 (gross profit + interest) / sales

X32 (current liabilities * 365) / cost of products sold

X33 operating expenses / short-term liabilities

X34 operating expenses / total liabilities

X35 profit on sales / total assets

X36 total sales / total assets

X37 (current assets - inventories) / long-term liabilities

X38 constant capital / total assets

X39 profit on sales / sales

X40 (current assets - inventory - receivables) / short-term liabilities

X41 total liabilities / ((profit on operating activities + depreciation) * (12/365))

X42 profit on operating activities / sales

X43 rotation receivables + inventory turnover in days

X44 (receivables * 365) / sales

X45 net profit / inventory

X46 (current assets - inventory) / short-term liabilities

X47 (inventory * 365) / cost of products sold

X48 EBITDA (profit on operating activities - depreciation) / total assets

X49 EBITDA (profit on operating activities - depreciation) / sales

X50 current assets / total liabilities

X51 short-term liabilities / total assets

X52 (short-term liabilities * 365) / cost of products sold)

X53 equity / fixed assets

X54 constant capital / fixed assets

X55 working capital

X56 (sales - cost of products sold) / sales

X57 (current assets - inventory - short-term liabilities) / (sales - gross profit - depreciation)

X58 total costs /total sales

X59 long-term liabilities / equity

X60 sales / inventory

X61 sales / receivables

X62 (short-term liabilities *365) / sales

X63 sales / short-term liabilities

X64 sales / fixed assets

5.2 Code in R

```
library(RWeka)

library(dplyr)

library(MASS)

library(DiscriMiner)

library(pracma)

library(caret)


# Loading the data

Year1 = read.arff(’1year.arff’)
```

```r
Year2 = read.arff(’2year.arff’)
Year3 = read.arff(’3year.arff’)
Year4 = read.arff(’4year.arff’)
Year5 = read.arff(’5year.arff’)

# Removing any entries that have a NA anywhere in the data
Year1 <- Year1[complete.cases(Year1),]
Year2 <- Year2[complete.cases(Year2),]
Year3 <- Year3[complete.cases(Year3),]
Year4 <- Year4[complete.cases(Year4),]
Year5 <- Year5[complete.cases(Year5),]

# Changing class values to be 5 for Year1, 4 for Year2 etc, and 0 for all non-bankrupt.
Year1$class <- as.numeric(ifelse(as.numeric(as.character(Year1$class)) == 1,5,0))
Year2$class <- as.numeric(ifelse(as.numeric(as.character(Year2$class)) == 1,4,0))
Year3$class <- as.numeric(ifelse(as.numeric(as.character(Year3$class)) == 1,3,0))
Year4$class <- as.numeric(ifelse(as.numeric(as.character(Year4$class)) == 1,2,0))
Year5$class <- as.numeric(as.character(Year5$class))

# Randomly Shuffling the rows
set.seed(42)
Year1 <- Year1[sample(nrow(Year1)), ]
Year2 <- Year2[sample(nrow(Year2)), ]
Year3 <- Year3[sample(nrow(Year3)), ]
Year4 <- Year4[sample(nrow(Year4)), ]
Year5 <- Year5[sample(nrow(Year5)), ]

Year1Train_index <- sample(1:nrow(Year1), 0.8 * nrow(Year1))
Year1Test_index <- setdiff(1:nrow(Year1), Year1Train_index)
```

```
Year1_train <- Year1[Year1Train_index, ]
Year1_Test <- Year1[Year1Test_index, ]

Year2Train_index <- sample(1:nrow(Year2), 0.8 * nrow(Year2))
Year2Test_index <- setdiff(1:nrow(Year2), Year2Train_index)
Year2_train <- Year2[Year2Train_index, ]
Year2_Test <- Year2[Year2Test_index, ]

Year3Train_index <- sample(1:nrow(Year3), 0.8 * nrow(Year3))
Year3Test_index <- setdiff(1:nrow(Year3), Year3Train_index)
Year3_train <- Year3[Year3Train_index, ]
Year3_Test <- Year3[Year3Test_index, ]

Year4Train_index <- sample(1:nrow(Year4), 0.8 * nrow(Year4))
Year4Test_index <- setdiff(1:nrow(Year4), Year4Train_index)
Year4_train <- Year4[Year4Train_index, ]
Year4_Test <- Year4[Year4Test_index, ]

Year5Train_index <- sample(1:nrow(Year5), 0.8 * nrow(Year5))
Year5Test_index <- setdiff(1:nrow(Year5), Year5Train_index)
Year5_train <- Year5[Year5Train_index, ]
Year5_Test <- Year5[Year5Test_index, ]

#producing final training and test data
train_data <- rbind(Year1_train, Year2_train, Year3_train, Year4_train, Year5_
   ↪ train)
test_data <- rbind(Year1_Test, Year2_Test, Year3_Test, Year4_Test, Year5_Test)

#lda on data
model.lda <- lda(class ~ ., data=train_data)
model.results = predict(model.lda, test_data[1:64])

#results printing
t = table(model.results$class, test_data$class)
print(confusionMatrix(t))

# stacked histograms
# histogram for LDA1, as [,1] is the 1st column
hist1 = ldahist(data = model.results$x[,1], g = train_data$class, xlim=c(-10,10))
# histogram for LDA2
hist2 = ldahist(data = model.results$x[,2], g = train_data$class, xlim=c(-10,10))
# histogram for LDA3
hist3 = ldahist(data = model.results$x[,3], g = train_data$class, xlim=c(-10,10))
# histogram for LDA4
hist4 = ldahist(data = model.results$x[,4], g = train_data$class, xlim=c(-10,10))
# histogram for LDA5
hist5 = ldahist(data = model.results$x[,5], g = train_data$class, xlim=c(-10,10))

# scatterplot of two Discriminant Functions (all combinations)
library(ggplot2)
lda.data <- cbind(train_data, predict(model.lda)$x)
ggplot(lda.data, aes(LD1, LD2, color = class)) + geom_point(aes(color = as.
   ↪ character(class)))
ggplot(lda.data, aes(LD1, LD3, color = class)) + geom_point(aes(color = as.
   ↪ character(class)))
ggplot(lda.data, aes(LD1, LD4, color = class)) + geom_point(aes(color = as.
   ↪ character(class)))
ggplot(lda.data, aes(LD1, LD5, color = class)) + geom_point(aes(color = as.
   ↪ character(class)))
```

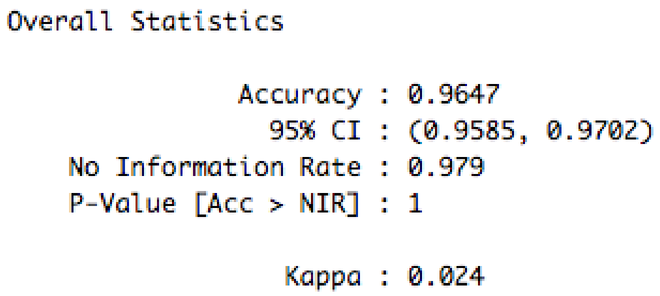
```
ggplot(lda.data, aes(LD2, LD3, color = class)) + geom_point(aes(color = as.
   ↪ character(class)))
ggplot(lda.data, aes(LD2, LD4, color = class)) + geom_point(aes(color = as.
   ↪ character(class)))
ggplot(lda.data, aes(LD2, LD5, color = class)) + geom_point(aes(color = as.
   ↪ character(class)))
ggplot(lda.data, aes(LD3, LD4, color = class)) + geom_point(aes(color = as.
   ↪ character(class)))
ggplot(lda.data, aes(LD3, LD5, color = class)) + geom_point(aes(color = as.
   ↪ character(class)))
ggplot(lda.data, aes(LD4, LD5, color = class)) + geom_point(aes(color = as.
   ↪ character(class)))

#data to reduce and its class membership
to_reduce <- train_data[, 1:64]
class_membership <- train_data[, 65]

#scatter matrices
within_scatter <- withinSS(to_reduce, class_membership)
between_scatter <- betweenSS(to_reduce, class_membership)
within_scatter_inv <- pinv(within_scatter)

#projection to 5 dimensions
decomp <- eigen(within_scatter_inv %*% between_scatter)
projection <- Re(as.matrix(decomp$vectors[, 1:5]))

#projected training and test data
train_reduced <- as.data.frame(t(t(projection) %*% t(as.matrix(to_reduce))))
train_reduced$class = class_membership
```

```
test_reduced <- as.data.frame(t(t(projection) %*% t(as.matrix(test_data[, 1:64]))
   ↪ ))
test_reduced$class = test_data[, 65]


#lda on reduced data
model_reduced.lda <- lda(class ~ ., data=train_reduced)
model_reduced.results = predict(model_reduced.lda, test_reduced[1:5])


#results printing
t_reduced = table(model_reduced.results$class, test_reduced$class)
print(confusionMatrix(t_reduced))


# histogram for LDA1
par(mar=rep(2,4))
hist1_reduced = ldahist(data = model_reduced.results$x[,1], g = class_membership,
   ↪  xlim=c(-10,10))
# histogram for LDA2
hist2_reduced = ldahist(data = model_reduced.results$x[,2], g = class_membership,
   ↪  xlim=c(-10,10))
# histogram for LDA3
hist3_reduced = ldahist(data = model_reduced.results$x[,3], g = class_membership,
   ↪  xlim=c(-10,10))
# histogram for LDA4
hist4_reduced = ldahist(data = model_reduced.results$x[,4], g = class_membership,
   ↪  xlim=c(-10,10))
# histogram for LDA5
hist5_reduced = ldahist(data = model_reduced.results$x[,5], g = class_membership,
   ↪  xlim=c(-10,10))


# scatterplot of two Discriminant Functions
```

```
library(ggplot2)
lda_reduced.data <- cbind(train_reduced, predict(model_reduced.lda)$x)
ggplot(lda_reduced.data , aes(LD1, LD2, color = class)) + geom_point(aes(color =
   ↪ as.character(class)))
ggplot(lda_reduced.data, aes(LD1, LD3, color = class)) + geom_point(aes(color =
   ↪ as.character(class)))
ggplot(lda_reduced.data, aes(LD1, LD4, color = class)) + geom_point(aes(color =
   ↪ as.character(class)))
ggplot(lda_reduced.data, aes(LD1, LD5, color = class)) + geom_point(aes(color =
   ↪ as.character(class)))
ggplot(lda_reduced.data, aes(LD2, LD3, color = class)) + geom_point(aes(color =
   ↪ as.character(class)))
ggplot(lda_reduced.data, aes(LD2, LD4, color = class)) + geom_point(aes(color =
   ↪ as.character(class)))
ggplot(lda_reduced.data, aes(LD2, LD5, color = class)) + geom_point(aes(color =
   ↪ as.character(class)))
ggplot(lda_reduced.data, aes(LD3, LD4, color = class)) + geom_point(aes(color =
   ↪ as.character(class)))
ggplot(lda_reduced.data, aes(LD3, LD5, color = class)) + geom_point(aes(color =
   ↪ as.character(class)))
ggplot(lda_reduced.data, aes(LD4, LD5, color = class)) + geom_point(aes(color =
   ↪ as.character(class)))
```

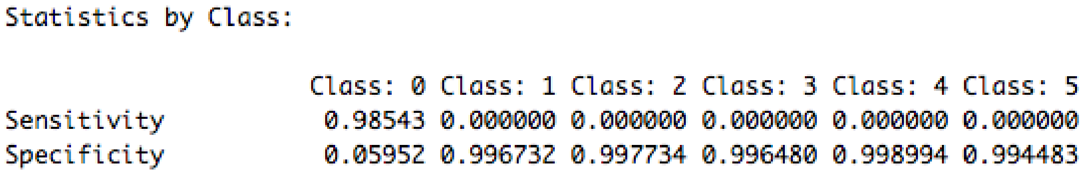
5.3 Application

5.3.1 Screenshots of section 3.2 - Classification



```
Overall Statistics

               Accuracy : 0.9647
                 95% CI : (0.9585, 0.9702)
    No Information Rate : 0.979
    P-Value [Acc > NIR] : 1

                  Kappa : 0.024
```

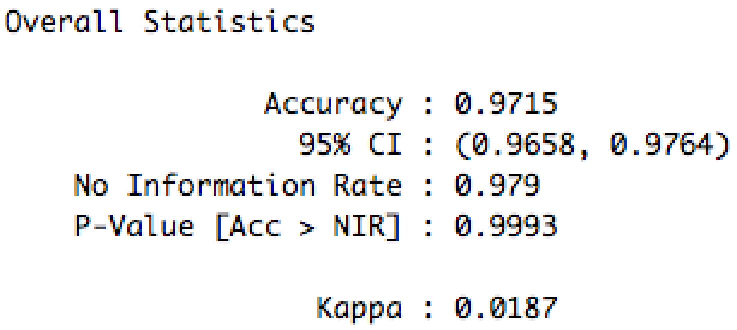
Figure 5.1: Overall Statistics of our first classification method



```
Statistics by Class:

                     Class: 0 Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
Sensitivity           0.98543 0.000000 0.000000 0.000000 0.000000 0.000000
Specificity           0.05952 0.996732 0.997734 0.996480 0.998994 0.994483
```

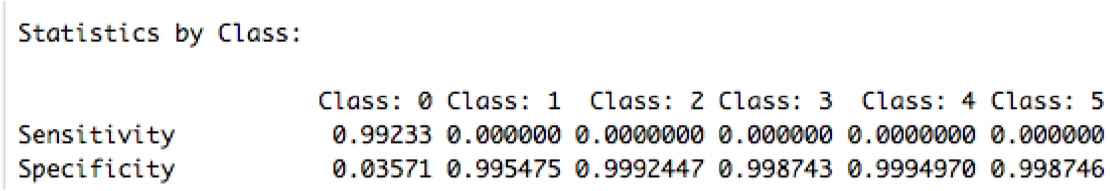
Figure 5.2: Statistics by class of our first classification method

5.3.2 Screenshots of section 3.3 - Reduction and Classification



```
Overall Statistics

               Accuracy : 0.9715
                 95% CI : (0.9658, 0.9764)
    No Information Rate : 0.979
    P-Value [Acc > NIR] : 0.9993

                  Kappa : 0.0187
```

Figure 5.3: Overall Statistics of our second classification method after reduction



```
Statistics by Class:

                     Class: 0 Class: 1  Class: 2 Class: 3  Class: 4 Class: 5
Sensitivity           0.99233 0.000000 0.0000000 0.000000 0.0000000 0.000000
Specificity           0.03571 0.995475 0.9992447 0.998743 0.9994970 0.998746
```

Figure 5.4: Statistics by class of our second classification method after reduction

Bibliography

[1] Fischer RA. The use of multiple measurements in taxonomic problems. *Annals of eugenics*. 1936; 7(2): 179–188.

[2] Rao CR. The Utilization of Multiple Measurements in Problems of Biological Classification. *Journal of the Royal Statistical Society. Series B (Methodological)*. 1948; 10(2): 159-203.

[3] Raschka S. *Linear Discriminant Analysis - Bit by Bit*, Viewed 31 May 2020, `http://sebastianraschka.com/Articles/2014_python_lda.html`

[4] Tharwat A, Gaber T, Ibrahim A, Hassanien AE. Linear discriminant analysis: A detailed tutorial. *AI Communications*. 2017; 30(2): 169–190.

[5] Balakrishnama S, Ganapathiraju A. Linear discriminant analysis - a brief tutorial. *Institute for Signal and Information Processing*. 1998; 18 1-8.

[6] McLachlan GJ. Discriminant analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*. 2012; 4(5): 421-431.

[7] Nie F, Wang H, Wang Z, Huang H. Robust Linear Discriminant Analysis Using Ratio Minimization of L1,2-Norms. *arXiv preprint arXiv:1907.00211*. 2019; .

[8] M. Kamel, A. Campilho. Image Analysis and Recognition 10th International Conference, ICIAR 2013, Póvoa do Varzim, Portugal, June 26-28, 2013.

[9] J. Lu, K.N. Plataniotis, A.N. Venetsanopoulos. Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition. *Pattern Recognition Letters*, Vol. 26, No. 2, pp. 181-191, 2005.

[10] L. Wu, C. Shen, A. v.d. Hengel. Deep linear discriminant analysis on fisher networks: A hybrid architecture for person re-identification. *Pattern Recognition*, Vol. 65, pp. 238-250, 2017.

[11] E. Alexandre-Cortizo, M. Rosa-Zurera, F. Lopez-Ferreras. Application of Fisher Linear Discriminant Analysis to Speech/Music Classification. *EUROCON 2005 - The International Conference on "Computer as a Tool"* : IEEE, pp. 1666-1669, 2005.

[12] D. Sun. *Computer vision technology for food quality evaluation.* 1. ed. ed. Burlington [u.a.]: Academic Press; 2008.

[13] Y. Guo, T. Hastie, R. Tibshirani, Regularized linear discriminant analysis and its application in microarrays. *Biostatistics*, Vol. 8, No. 1, pp. 86–100, January 2007.

[14] W. Zhong, K.S. Suslick, Matrix Discriminant Analysis With Application to Colorimetric Sensor Array Data. *Technometrics*, Vol. 57, No. 4, pp. 524-534, 2015.

[15] A. J. Izenman. *Mordern Multivariate Statistical Techniques*, 2008.

[16] H. Rajaguru, S. K. Prabhakar. *Bayesian Linear Discriminant Analysis for Breast Cancer Classification*, 2017 2nd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, pp. 266-269, 2017, doi: 10.1109/CESYS.2017.8321279.

[17] N. Tollenaar, P. G. M. van der Heijden. Which method predicts recidivism best?: a comparison of statistical, machine learning and data mining predictive models. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, Vol. 176, No. 2 , pp. 565-584, February 2013.

[18] E. I. Altman. Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. *The Journal of Finance*, Vol. 23, No. 4 , pp. 589-609, September 1968.

[19] Y. Yoon, G. Swales, Jr. and Thomas M. Margavio. A Comparison of Discriminant Analysis versus Artificial Neural Networks. *The Journal of the Operational Research Society*, Vol. 44, No. 1 , pp.51-60, January 1993.

[20] A. Alin. Multicollinearity. *Wiley Interdisciplinary Reviews: Computational Statistics.*, Vol. 2, No. 3, pp. 370-374, May/June 2010.

[21] B. Back, T. Laitinen, K. Sere, M. van Wezel. Choosing Bankruptcy Predictors Using Discriminant Analysis, Logit Analysis, and Genetic Algorithms. *Turku Centre for Computer Science Technical Report*, Vol. 40, No. 2, pp. 1-18, September 1996.

[22] T. Hastie, R. Tibshirani, J. Friedman. *The Elements of Statistical Learning Data Mining, Inference, and Prediction (2nd edition).* Springer Series in Statistics, pp. 113-119, 2009.

[23] C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, Cambridge, pp.181-192, 2006.

[24] G. James, D. Witten, T. Hastie, R. Tibshirani. *An Introduction to Statistical Learning with Applications in R*, Springer, New York, pp.142-143, 2013.

[25] J. A. Rice. *Mathematical Statistics and Data Analysis*, pp.54-55, 2007.

[26] M. Zikeba, S. K.Tomczak, J. M.Tomczak. *Ensemble Boosted Trees with Synthetic Features Generation in Application to Bankruptcy Prediction*, Expert Systems with Applications, Elsevier, 2016.

[27] M. Döring. *Linear, Quadratic, and Regularized Discriminant Analysis*, Published 20 November 2018, Viewed 12 June 2020.
https://www.datascienceblog.net/post/machine-learning/linear-discriminant-analysis/