# Linear vs. quadratic discriminant analysis classifier: a tutorial

Alaa Tharwat

Electrical Department,
Faculty of Engineering,
Suez Canal University,
Ismailia, Egypt
Email: emgalaatharwat@hotmail.com

**Abstract:** The aim of this paper is to collect in one place the basic background needed to understand the *discriminant analysis* (DA) classifier to make the reader of all levels be able to get a better understanding of the DA and to know how to apply this classifier in different applications. This paper starts with basic mathematical definitions of the DA steps with visual explanations of these steps. Moreover, in a step-by-step approach, a number of numerical examples were illustrated to show how to calculate the discriminant functions and decision boundaries when the covariance matrices of all classes were common or not. The singularity problem of DA was explained and some of the state-of-the-art solutions to this problem were highlighted with numerical illustrations. An experiment is conducted to compare between the linear and quadratic classifiers and to show how to solve the singularity problem when high-dimensional datasets are used.

**Keywords:** linear discriminant classifier; LDC; quadratic discriminant classifier QDC; classification; singularity problem; discriminant function; decision boundaries; subspace method; regularised linear discriminant analysis; RLDA.

**Reference** to this paper should be made as follows: Tharwat, A. (2016) 'Linear vs. quadratic discriminant analysis classifier: a tutorial', *Int. J. Applied Pattern Recognition*, Vol. 3, No. 2, pp.145–180.

**Biographical notes:** Alaa Tharwat received his BSc and MSc from the Faculty of Engineering, Computer and Control Systems Department, Mansoura University, Egypt in 2002 and 2008, respectively. He is an Assistant Lecturer in the Electrical Department, Faculty of Engineering at the Suez Canal University, Egypt. He was a researcher at Gent University, within the framework of the Welcome project – Erasmus Mundus Action 2 – with a title 'Novel approach of multi-modal biometrics for animal identification'. He is the author of many research studies published at national and international journals and conference proceedings. His major research interests include pattern recognition, machine learning, digital image processing, biometric authentication, and bio-inspired optimisation.

## 1 Introduction

Supervised learning used a labelled or known samples to build its model to predict or estimate the value for a new sample or estimate a future response. The *supervised* model tries to connect between the extracted features or attributes and the labels of each sample. There are two main techniques of supervised learning, namely, *regression* and *classification*. In the regression technique, the labelled data are real numbers that are used to build a model to estimate the value of the new sample, while in the classification technique, the labelled data are represented by a set of known values or classes, which are used to build a classification model. This classification model is then used to assign a class label to an unknown sample (Hastie et al., 2001; Duda et al., 2012).

Many classification and regression models have been proposed in the literature such as neural networks (NN) (Specht, 1990), linear regression (Montgomery et al., 2012; Seber and Lee, 2012), nonlinear regression (Motulsky and Christopoulos, 2004; Hahne et al., 2014), classification and regression trees (Loh, 2011), and discriminant analysis (DA) classifier (McLachlan, 2004). DA classifier was introduced by R. Fisher and it was used in many classification problems (Altman et al., 1994; Guo et al., 2007). DA classifier is one of the basic and simple classifiers. There are two types of DA classifier, namely, *linear discriminant analysis* (LDA) and *quadratic discriminant analysis* (QDA) classifiers. In LDA classifier, the decision surface is linear, while the decision boundary in QDA is nonlinear.

Although the DA classifier is considered one of the most well-known classifiers, it suffers from a singularity problem. In the singularity problem, DA fails to calculate the discriminant functions if the dimensions are much higher than the number of samples in each class. Thus, the covariance matrix is singular; hence, it cannot be inverted. There are different methods were proposed to solve this problem. The first method is the *regularised linear discriminant analysis* (RLDA). In this method, the identity matrix was scaled by multiplying it by a regularisation parameter and added to the covariance matrix (Friedman, 1989; Ye and Xiong, 2006). The *subspace* method was also used to solve the singularity problem by reducing the dimensions of a high dimensional data (Belhumeur et al., 1997; Gao and Davis, 2006).

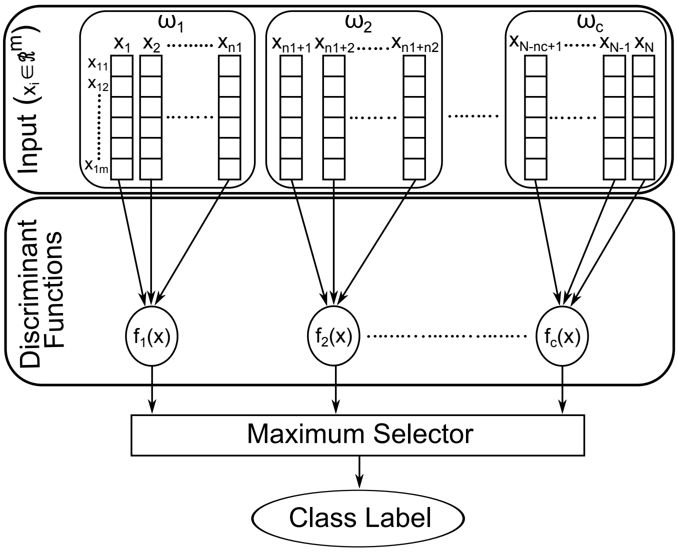
This paper gives a detailed tutorial about DA classifier and it is divided into four sections. Section 2 gives an overview about the definition of the main idea of the DA classifier and its background. This section begins by defining and explaining how to calculate, with visual explanations the discriminant functions and decision boundaries to build LDA and QDA classifiers. The training and testing algorithms of the discriminant analysis classifier, i.e., LDA and QDA, are then introduced. Section 3 illustrates numerical examples to show how to calculate the discriminant functions and decision boundaries for the DA classifier when the covariance matrices of all classes are common or not. Section 4 introduces the singularity problem of DA and two common methods to solve this problem. Moreover, numerical illustrations for the singularity problem and its solutions were explained. Finally, concluding remarks will be given in Section 6.

## 2 DA classifier

### *2.1 Background of DA classifier*

A pattern or sample is represented by a vector or a set of *m* features, which represent one point in m-dimensional space ($\mathcal{R}^m$) that is called pattern space. The goal of the pattern classification process is to train a model using the labelled patterns to assign a class label to an unknown pattern. The class labels represent the classes or categories of the labelled patterns that are used to calculate the *discriminant functions* for each class. The discriminant functions are then used to determine the *decision boundaries* and *decision regions* for each class (Duda et al., 2012; Vapnik, 2013).

**Figure 1** The structure of building a classifier, which includes *N* samples and *c* discriminant functions or classes


Input ($x_i \in \mathcal{R}^m$)
$\omega_1$
$x_1$ $x_2$ ......... $x_{n1}$
$x_{11}$
$x_{12}$
$x_{1m}$
$\omega_2$
$x_{n1+1}$ $x_{n1+2}$ ...... $x_{n1+n2}$
$\omega_c$
$x_{N-nc+1}$ ...... $x_{N-1}$ $x_N$
Discriminant Functions
$f_1(x)$
$f_2(x)$
$f_c(x)$
Maximum Selector
Class Label

#### *2.1.1 Discriminant functions*

The classifier is represented by c decisions or discriminant functions ($\{f_1, f_2, \ldots, f_c\}$), where *c* represents the number of classes as shown in Figure 1. The decision functions are used to determine the decision boundaries between classes and the region or area of each class as shown in Figure 2. Hence, the discriminant functions are used to determine the class label of the unknown pattern ($x$) based on comparing c different discriminant functions and assigns the class label of the maximum score to the unknown sample as shown in equation (1). Thus, within the region $\omega_i$, the $i^{\text{th}}$ discriminant function ($f_i$) will have the maximum value compared to all other discriminant functions (Hastie et al., 2001; Duda et al., 2012; Fukunaga, 2013). If the values of any two discriminant functions are equal ($f_i(x) = f_j(x)$), thus the unknown pattern ($x$) is on the boundary between the two classes.

$$f_i(x) > f_j(x), i, j = 1, 2, \ldots, c, i \neq j \tag{1}$$

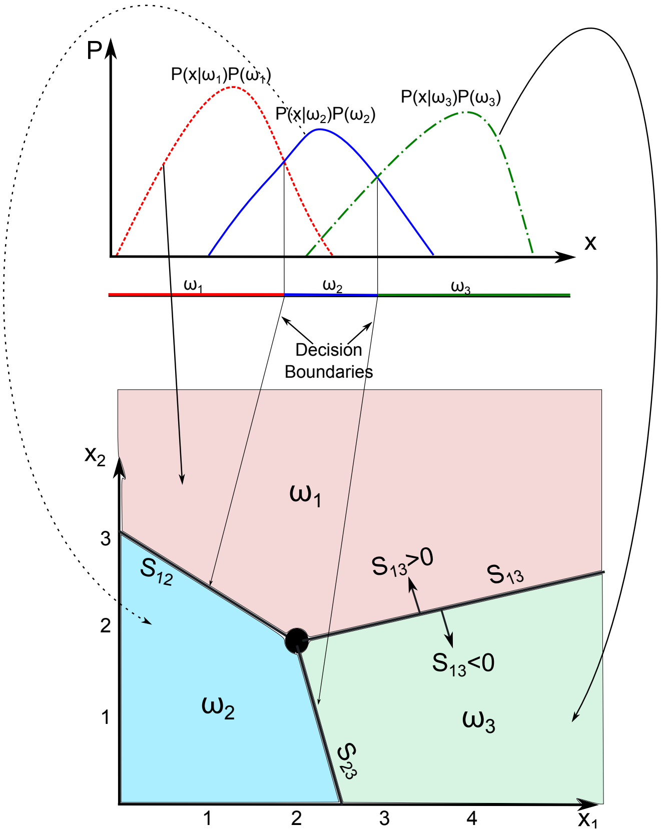
### 2.1.2 Decision boundaries

Discriminant functions are used to build the decision boundaries to discriminate between different classes into different regions ($\omega_i$, $i = 1, 2, \ldots, c$) as shown in Figure 2. Thus, the input space is divided into a collection of regions, each region is bounded by a number of decision boundaries.[1] In other words, each decision boundary ($\mathcal{S}_{ij}$) separates two different regions, i.e., two classes $\omega_i$ and $\omega_j$, and it consists of two discriminant functions, $f_i$ and $f_j$ (Hastie et al., 2001; Guo et al., 2007; Duda et al., 2012; Fukunaga, 2013).

Assume, we have two classes ($\omega_1$) and ($\omega_2$), thus there are two different discriminant functions ($f_1$ and $f_2$) and the decision boundary is calculated as follows, $\mathcal{S}_{12} = f_1 - f_2$. The decision region or class label of an unknown pattern $x$ is calculated as follows.

$$\operatorname{sgn}\left(\mathcal{S}_{12}(x)\right) = \operatorname{sgn}\left(f_1(x) - f_2(x)\right) = \begin{cases} \text{Class 1:} & \text{for } \mathcal{S}_{12}(x) \geq 0 \\ \text{Undefined:} & \text{for } \mathcal{S}_{12}(x) = 0 \\ \text{Class 2:} & \text{for } \mathcal{S}_{12}(x) < 0 \end{cases} \tag{2}$$

**Figure 2** Decision regions of three classes (see online version for colours)



## 2.2 Building a classifier model

### 2.2.1 Normal density

Let $\omega_1, \omega_2, \ldots, \omega_\text{c}$ be the set of $c$ classes, $P(x \mid \omega_i)$ represents the *likelihood* function or simply the likelihood or the conditional probability density function,[2] and $P(\omega_i)$ represents the priori probability. Priori probability or simply *priori* of each class reflects the prior knowledge about that class and it is simply equal to the ratio between the number of samples in that class and the total number of samples in all classes ($N$) as follows, $P(\omega_i) = \frac{n_i}{N}$, where $n_i$ represents the number of samples in the $i^{\text{th}}$ class, and $\sum_{i=1}^{c} P(\omega_i) = 1$ (Guo et al., 2007; Duda et al., 2012). Bayes formula calculates the posterior probability based on priori and likelihood as follows.

$$P\left(\omega = w_i \middle| x\right) = \frac{P\left(x \middle| \omega = \omega_i\right) P\left(\omega_i\right)}{P(x)} = \frac{\textit{likelihood} \times \textit{priori}}{\textit{evidence}} \tag{3}$$

where $P(\omega = \omega_i \mid x)$ represents the posterior probability or *a posteriori*,[3] $P(x)$ represents the evidence and it is calculated as follows, $P(x) = \sum_{i=1}^{c} P(x \mid \omega = \omega_i) P(\omega_i)$ (Duda et al., 2012). $P(x)$ is used only to scale the expressions in equation (3), thus the sum of the posterior probabilities is $1\left(\sum_{i=1}^{c} P(\omega_i \mid x) = 1\right)$. Generally, $P(\omega_i \mid x)$ is calculated using the likelihood ($P(x \mid \omega_i)$) and prior probability ($P(\omega_i)$).

Assume that $P(x \mid \omega_i)$ is normally distributed $\left(P(x \mid \omega_i) \sim \mathcal{N}\left(\mu_i, \Sigma_i\right)\right)$ as follows.

$$P\left(x \middle| \omega_i\right) = \mathcal{N}\left(\mu_i, \Sigma_i\right) = \frac{1}{\sqrt{(2\pi)^m \left|\Sigma_i\right|}} \exp\left(-\frac{1}{2}\left(x - \mu_i\right)^T \Sigma_i^{-1} \left(x - \mu_i\right)\right) \tag{4}$$

where $\mu_i$ represents the mean of the $i^{\text{th}}$ class and it is calculated as in equation (5), $\Sigma_i$ is the covariance matrix of the $i^{\text{th}}$ class and it is calculated as in equation (6), $|\Sigma_i|$ and $\Sigma_i^{-1}$ represent the determinant and inverse of the covariance matrix, respectively, $m$ represents the number of features or the number of variables of the sample ($x$).[4]

$$\mu_i = \frac{1}{n_i} \sum_{i=1}^{n_i} x_i, x_i \in \omega_i, \forall i = 1, 2, \ldots, c \tag{5}$$

$$\Sigma_i = \frac{1}{n_i} \sum_{x \in \omega_i} \left(x - \mu_i\right)\left(x - \mu_i\right)^T, \forall i = 1, 2, \ldots, c \tag{6}$$

The covariance matrix is used when there are more than one variable (multivariate) and it is defined as follows, $\Sigma = E\{x_i x_j\} - E\{x_i\}E\{x_j\}$, where $E\{x_i\}$ represents the expected value or mean of the variable $x_i$. Covariance matrix ($\Sigma$) is symmetric matrix, i.e., $\Sigma = \Sigma^T$, and positive semi-definite matrix.[5] The diagonal values of the covariance matrix represent the variance of the variable with itself, while the off-diagonal elements represent the covariance between the $x_i$ and $x_j$ as shown in equation (7). In the covariance matrix, a positive value means a positive correlation between the two variables, while a negative value indicates negative correlation and zero value indicates that the two variables are uncorrelated or statistically independent (Whittaker, 2009).

$$\begin{pmatrix} \operatorname{var}(x_1,x_1) & \operatorname{cov}(x_1,x_2) & \dots & \operatorname{cov}(x_1,x_N) \\ \operatorname{cov}(x_2,x_1) & \operatorname{var}(x_2,x_2) & \dots & \operatorname{cov}(x_2,x_N) \\ \vdots & \vdots & \ddots & \vdots \\ \operatorname{cov}(x_N,x_1) & \operatorname{cov}(x_N,x_2) & & \operatorname{var}(x_N,x_N) \end{pmatrix} \tag{7}$$

### 2.2.2 *Discriminant functions for the normal density*

Assume, we have two classes $\omega_1$ and $\omega_2$ and each class has one discriminant function ($f_i$, $i = 1, 2$). If we have an unknown pattern ($x$) and $P(\omega_1 \mid x) > P(\omega_2 \mid x)$, thus the unknown pattern belongs to the first class ($\omega_1$). Similarly, if $P(\omega_2 \mid x) > P(\omega_1 \mid x)$; hence, $x$ belongs to $\omega_2$ (Hastie et al., 2001; Duda et al., 2012). Generally, the unknown sample will be classified to the class, which maximises the posterior probability or the likelihood, hence maximises the discriminant function as follows.

$$\begin{aligned} f_i(x) &= \ln P\left(\omega = \omega_i \middle| x\right) = P\left(x \middle| \omega = \omega_i\right) \\ P\left(\omega_i\right) &= \ln\left(P\left(x \middle| \omega = \omega_i\right)\right) + \ln\left(P\left(\omega_i\right)\right), \, i = 1, 2 \\ &= \ln \frac{1}{\sqrt{(2\pi)^m \, |\Sigma_i|}} \exp\left(-\frac{1}{2}\left(x - \mu_i\right)^T \Sigma_i^{-1}\left(x - \mu_i\right)\right) + \ln\left(P\left(\omega_i\right)\right) \\ &= -\frac{1}{2}\left(x - \mu_i\right)^T \Sigma_i^{-1}\left(x - \mu_i\right) - \frac{m}{2}\ln(2\pi) - \frac{\ln|\Sigma_i|}{2} + \ln\left(P\left(\omega_i\right)\right) \\ &= -\frac{\Sigma_i^{-1}}{2}\left(x^T x + \mu_i^T \mu_i - 2\mu_i^T x\right) - \frac{m}{2}\ln(2\pi) - \frac{\ln|\Sigma_i|}{2} + \ln\left(P\left(\omega_i\right)\right) \end{aligned} \tag{8}$$

where ln denotes the natural logarithm. As denoted in equation (8), the denominator of the posterior probability [see equation (3)] is removed because it is common for all discriminant functions.

The decision boundary or surface between the class $\omega_1$ and $\omega_2$ is represented by the difference between the two discriminant functions as follows.

$$\begin{aligned} \mathcal{S}_{12} = f_1 - f_2 &= \frac{\ln P\left(\omega = \omega_1 \middle| x\right)}{\ln P\left(\omega = \omega_2 \middle| x\right)} = \ln \frac{P\left(x \middle| \omega = \omega_1\right) P\left(\omega_1\right)}{P\left(x \middle| \omega = \omega_2\right) P\left(\omega_2\right)} \\ &= \ln \frac{P\left(x \middle| \omega = \omega_1\right)}{P\left(x \middle| \omega = \omega_2\right)} + \ln \frac{P\left(\omega_1\right)}{P\left(\omega_2\right)} = \ln P\left(x \middle| \omega = \omega_1\right) + \ln P\left(\omega_1\right) \\ &\quad - \ln P\left(x \middle| \omega = \omega_2\right) + \ln P\left(\omega_2\right) \end{aligned} \tag{9}$$

From equations (8) and (9) the decision boundary is

$$\begin{aligned}\mathcal{S}_{12} &= -\frac{1}{2}\left[\Sigma_1^{-1}\left(x^T x - 2\mu_1^T x + \mu_1^T \mu_1\right)\right] \\ &\quad - \Sigma_2^{-1}\left(x^T x - 2\mu_2^T x + \mu_2^T \mu_2\right) + \ln|\Sigma_1| - \ln|\Sigma_2| + \ln\frac{P(\omega_1)}{P(\omega_2)} \\ &= -\frac{1}{2}x^T\left(\Sigma_1^{-1} - \Sigma_2^{-1}\right)x + \left(\mu_1^T\Sigma_1^{-1} - \mu_2^T\Sigma_2^{-1}\right)x \\ &\quad -0.5\left(\mu_1^T\Sigma_1^{-1}\mu_1 - \Sigma_2^{-1}\mu_2 + \ln|\Sigma_1| - \ln|\Sigma_2|\right) + \ln\frac{P(\omega_1)}{P(\omega_2)} \\ &= x^T W x + w^T x + W_0\end{aligned} \tag{10}$$

where

$$W = -\frac{1}{2}\left(\Sigma_1^{-1} - \Sigma_2^{-1}\right) \tag{11}$$

$$w = \mu_1^T\Sigma_1^{-1} - \mu_2^T\Sigma_2^{-1} \tag{12}$$

$$W_0 = -0.5\left(\mu_1^T\Sigma_1^{-1}\mu_1 - \mu_2^T\Sigma_2^{-1}\mu_2 + \ln|\Sigma_1| - \ln|\Sigma_2|\right) + \ln\frac{P(\omega_1)}{P(\omega_2)} \tag{13}$$

where $W_0$ represents the threshold or *bias*, $w$ represents the slope of the line, and $W$ is the coefficient of the quadratic term $x^T$ $x$. Thus, the decision boundary is calculated by quadratic function or curve, which is called QDA as denoted in equation (10). The classification of an unknown pattern ($x$) will be as follows.

$$\operatorname{sgn}\left(\mathcal{S}_{12}(x)\right) = \begin{cases} +ve & \text{if } x^T W x + w^T x + W_0 > 0 \rightarrow x \in \omega_1 \\ 0 & \text{if } x^T W x + w^T x + W_0 = 0\text{; On the boundary} \\ -ve & \text{if } x^T W x + w^T x + W_0 < 0 \rightarrow x \in \omega_2 \end{cases} \tag{14}$$

### *2.2.3 Special case: common covariance matrices*

Assume the variance of all classes are equal ($\Sigma_1 = \Sigma_2 = \Sigma$), hence the term $W$ will be neglected [see equation (11)] and the decision boundary is calculated as in equation (15). Similarly, the term $\ln|\Sigma_1| - \ln|\Sigma_2|$ in equation (13) will be neglected and $W_0$ will be easier to calculate as in equation (17). Moreover, w will be easier to implement as shown in equation (16) (Hastie et al., 2001; Duda et al., 2012; Fukunaga, 2013). In other words, when each class has an individual covariance matrix, this leads to the so called QDA, hence the decision boundaries are quadratic curves. On the other hand, from the assumption of common covariance matrix, the discriminant function is simplified from quadratic to linear function, which is called LDA which is similar to neural network models as shown in equation (15). The classification of an unknown pattern using LDA will be calculated as in equation (18).

$$\mathcal{S}_{12} = w^T x + W_0 \tag{15}$$

where

$$w = \Sigma^{-1}\left(\mu_1^T - \mu_2^T\right) \tag{16}$$

and

$$W_0 = -0.5\Sigma^{-1}\left(\mu_1^T \mu_1 - \mu_2^T \mu_2\right) + \ln\frac{P(\omega_1)}{P(\omega_2)} \tag{17}$$

$$\operatorname{sgn}\left(\mathcal{S}_{12}\right) = \begin{cases} +ve & \text{if } w^T x + W_0 > 0 \rightarrow x \in \omega_1 \\ 0 & \text{if } w^T x + W_0 = 0\text{; on the boundary} \\ -ve & \text{if } w^T x + W_0 < 0 \rightarrow x \in \omega_2 \end{cases} \tag{18}$$

The decision boundary is the point where $\mathcal{S}_{12} = 0$ and this point will be calculated as follows.

$$\mathcal{S}_{12} = 0 \rightarrow \Sigma^{-1}\left(\mu_1^T - \mu_2^T\right)x - 0.5\Sigma^{-1}\left(\mu_1^T \mu_1 - \mu_2^T \mu_2\right) + \ln\frac{P(\omega_1)}{P(\omega_2)} = 0 \tag{19}$$

The decision boundary $x_{DB}$ is

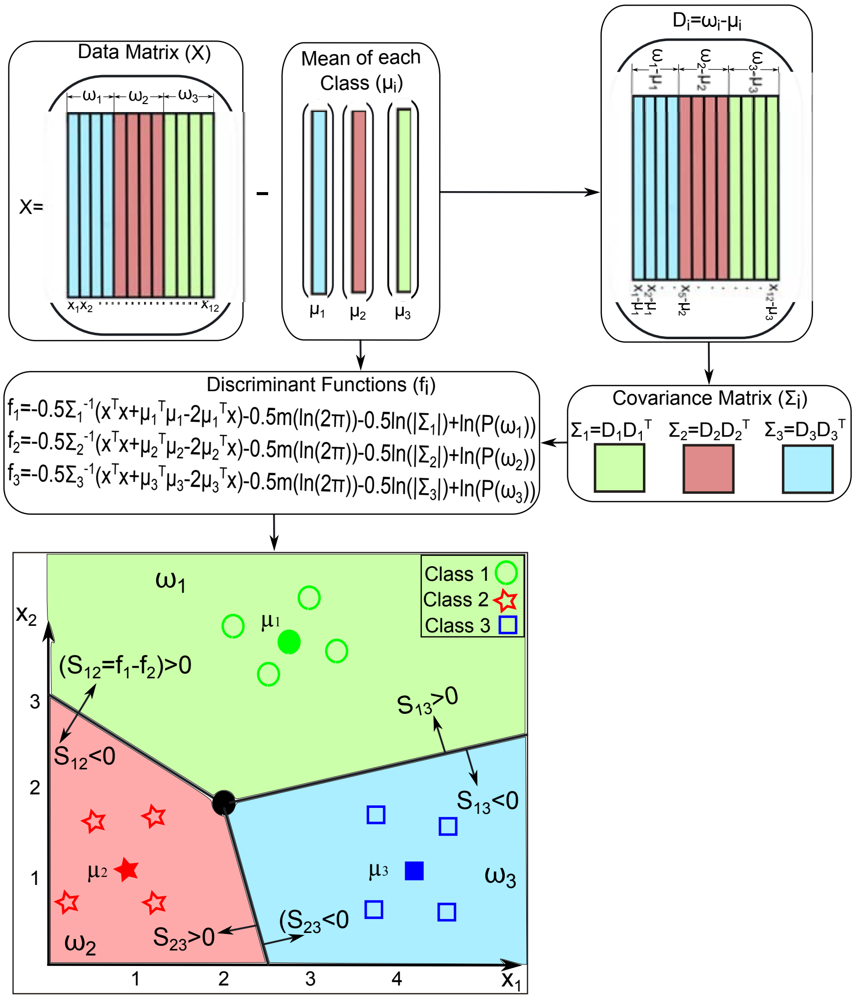
$$x_{DB} = \frac{\mu_1 + \mu_2}{2} + \frac{\Sigma}{\mu_2 - \mu_1}\ln\frac{P(\omega_1)}{P(\omega_2)} \tag{20}$$

When the two classes are equiprobable, then the second term in equation (20) will be neglected and the decision boundary is the point in the middle of the class centres as follows, $x_{DB} = \frac{\mu_1 + \mu_2}{2}$. In other words, the decision boundary will be closer to the class that has lower prior probability. For example, if $P(\omega_i) > P(\omega_j)$, then $|\mu_j - x_{DB}| < |\mu_i - x_{DB}|$.

**Table 1** Notation

| *Notation* | *Description* | *Notation* | *Description* |
|---|---|---|---|
| $m$ | Dimension of the samples | $P(\omega_i)$ | Priori probability of the $i^{th}$ class |
| $\sigma^2$ | Variance | $\omega_i$ | the i$^{th}$ class |
| $\mu_i$ | the mean of the $i^{th}$ class | $x_i$ | the $i^{th}$ sample |
| $D_i$ | Mean-centring data for the class $\omega_i$ | $c$ | Total number of classes |
| $n_i$ | Number of samples in class $\omega_i$ | $\Sigma_i$ | Covariance matrix of the $i^{th}$ class |
| $f_i$ | Discriminant function of $\omega_i$ | $S_{ij}$ | The decision boundary between class $\omega_i$ and $\omega_j$ |
| $W$ | Coefficient of the quadratic term $x^T x$ | $w$ | The slope of the line |
| $W_0$ | Threshold or bias | $N$ | Total number of samples |

**Figure 3** Steps of calculating DA classifier given three classes, each class has four samples (see online version for colours)


Data Matrix (X)
X=
$\omega_1$ $\omega_2$ $\omega_3$
$x_1 x_2$ ............... $x_{12}$
-
Mean of each Class ($\mu_i$)
$\mu_1$ $\mu_2$ $\mu_3$
$D_i=\omega_i-\mu_i$
$\omega_1-\mu_1$ $\omega_2-\mu_2$ $\omega_3-\mu_3$
$x_1-\mu_1$ $x_2-\mu_1$ $x_5-\mu_2$ $x_{12}-\mu_3$
Discriminant Functions ($f_i$)
$f_1=-0.5\Sigma_1^{-1}(x^Tx+\mu_1^T\mu_1-2\mu_1^Tx)-0.5m(\ln(2\pi))-0.5\ln(|\Sigma_1|)+\ln(P(\omega_1))$
$f_2=-0.5\Sigma_2^{-1}(x^Tx+\mu_2^T\mu_2-2\mu_2^Tx)-0.5m(\ln(2\pi))-0.5\ln(|\Sigma_2|)+\ln(P(\omega_2))$
$f_3=-0.5\Sigma_3^{-1}(x^Tx+\mu_3^T\mu_3-2\mu_3^Tx)-0.5m(\ln(2\pi))-0.5\ln(|\Sigma_3|)+\ln(P(\omega_3))$
Covariance Matrix ($\Sigma_i$)
$\Sigma_1=D_1D_1^T$ $\Sigma_2=D_2D_2^T$ $\Sigma_3=D_3D_3^T$
$\omega_1$
Class 1 ○
Class 2 ☆
Class 3 □
$x_2$
$\mu_1$
$(S_{12}=f_1-f_2)>0$
$S_{13}>0$
$S_{12}<0$
$S_{13}<0$
$\mu_2$
$\mu_3$
$\omega_3$
$\omega_2$
$S_{23}>0$
$(S_{23}<0$
3
2
1
1 2 3 4
$x_1$

## 2.3 LDA algorithm

In this section, the detailed steps of building the DA classifier are explained. The first step in the algorithm is to construct a data or feature matrix (*X*), where each sample is represented as one column and the number of rows represents the dimension (i.e., the number of features) of the samples as shown in Figure 3. The mean of each class, mean-centring data, and covariance matrices for each class are then calculated to calculate the discriminant functions. As shown from Figure 3, the discriminant functions are used to construct the decision boundaries, which are used to separate different classes into different regions. The detailed steps of building the classifier model are summarised in Algorithm (1). Algorithm (2) summarises the steps of classifying an unknown sample using the discriminant functions which are calculated in Algorithm (1).

**Algorithm 1** DA classifier (building model)

```
Input: data matrix $X$, which consists of $N$ samples $[x_i]_{i=1}^{N}$, each of which is represented as a
column of length $m$ as in Figure 3 and Table 1, where $x_i$ represents the $i^{th}$ sample.
Compute the mean of each class $\mu_i(m \times 1)$ as in equation (5).
Calculate the priori probability of each class $P(\omega_i) = \frac{n_i}{N}$.
Compute the covariance matrix for each class ($\Sigma_i$) as in equation (6).
for all (class $\omega_i$, $i = 1, 2, \dots, c$) do
    Calculate the discriminant function ($f_i$) as in equation (8).
end for
```

**Algorithm 2** DA classifier (classify an unknown sample)

```
Input: An unknown sample ($T(m \times 1)$).
Output: Class label ($\omega_i$).
for all (Discriminant functions ($f_i$), which are calculated in Algorithm 1) do
    Substitute the value of the unknown sample ($T$) in the discriminant function ($f_i$).
end for
Assign the class label ($\omega_{\max}$) to the unknown sample ($T$), where ($\omega_{\max}$) represents the class
that has the maximum discriminant function.
```

## 3 Numerical examples

In this section, three numerical examples were presented to show how to build the DA classifier. In the first example, all features were statistically independent and have the same variance. In the second example, all the covariance matrices of all classes were equal but arbitrary. In the third example, the covariance matrices of all classes were different.

### 3.1 Example 1: Equal variance ($\Sigma_i = \sigma^2 I$)

In this example, the features were statistically independent, i.e., all off-diagonal elements of the covariance matrices were zeros, and had the same variance ($\sigma^2$). Thus, the covariance matrices were diagonal and its diagonal elements were $\sigma^2$. Geometrical interpretation for this case is that each class is centred around its mean, the distance from the mean to all samples of the same class are equal as shown in Figure 4, and the distributions of all classes are spherical in an m-dimensional space. As mentioned in Section 2.2.3, the term $W$ will be neglected and the decision boundaries were linear and it will be calculated as in equation (15). The next two sections explain how to build the DA classifier and how to classify an unknown sample. MATLAB code for this experiment is introduced in Appendix.

### *3.1.1 Model training (building classifier)*

Given three different classes denoted by, $\omega_1$, $\omega_2$, $\omega_3$ as shown in Figure 4. Each class ($\omega_i$) consists of four samples, i.e., $n_i = 4$, and each sample was represented by two features, i.e., $x \in R^2$). The values of all samples in each class are shown below.

$$\omega_1 = \begin{bmatrix} 3.00\,4.00 \\ 3.00\,5.00 \\ 4.00\,4.00 \\ 4.00\,5.00 \end{bmatrix}, \omega_2 = \begin{bmatrix} 3.00\,2.00 \\ 3.00\,3.00 \\ 4.00\,2.00 \\ 4.00\,3.00 \end{bmatrix}, \text{and } \omega_3 = \begin{bmatrix} 6.00\,2.00 \\ 6.00\,3.00 \\ 7.00\,2.00 \\ 7.00\,3.00 \end{bmatrix} \tag{21}$$

The mean of each class can be calculated as in equation (5) and the values of means are shown below.

$$\mu_1 = [3.50\;4.50], \mu_2 = [3.50\;2.50], \text{and } \mu_3 = [6.50\;2.50] \tag{22}$$

Subtract the mean of each class from each sample in that class, i.e., mean-centring data, as follows, $D_i = \omega_i - \mu_i$, where $D_i$ represents the samples of the $i^{\text{th}}$ class ($\omega_i$) minus the mean of that class. The values of $D_1$, $D_2$ and $D_3$ are as follows.

$$D_1 = \begin{bmatrix} -0.50\,-0.50 \\ -0.50\,-0.50 \\ 0.50\,0.50 \\ 0.50\,0.50 \end{bmatrix}, D_2 = \begin{bmatrix} -0.50\,-0.50 \\ -0.50\,0.50 \\ 0.50\,-0.50 \\ 0.50\,0.50 \end{bmatrix}, \text{and } D_3 = \begin{bmatrix} -0.50\,-0.50 \\ -0.50\,0.50 \\ 0.50\,-0.50 \\ 0.50\,0.50 \end{bmatrix} \tag{23}$$

The covariance matrix for each class ($\Sigma_i$) was then calculated as follows, $\Sigma_i = D_i * D_i^T$. The values of the covariance matrices and their corresponding inverse matrices are as follows.

$$\Sigma_1 = \Sigma_2 = \Sigma_3 = \begin{bmatrix} 1.00\,0.00 \\ 0.00\,1.00 \end{bmatrix} \tag{24}$$

$$\Sigma_1^{-1} = \Sigma_2^{-1} = \Sigma_3^{-1} = \begin{bmatrix} 1.00\,0.00 \\ 0.00\,1.00 \end{bmatrix} \tag{25}$$

As mentioned in Section 2.2, priori probability for each class represents the ratio between the number of samples in that class to the total number of samples as follows: $P(\omega_1) = P(\omega_2) = P(\omega_3) = \frac{4}{12}$.

The discriminated functions for each class were then calculated as in equation (8) and the three functions will be as follows.

$$\begin{aligned} f_1 &= -0.5x_1^2 - 0.5x_2^2 + 3.50x_1 + 4.50x_2 - 17.35 \\ f_2 &= -0.5x_1^2 - 0.5x_2^2 + 3.50x_1 + 2.50x_2 - 10.35 \\ f_3 &= -0.5x_1^2 - 0.5x_2^2 + 6.50x_1 + 2.50x_2 - 25.35 \end{aligned} \tag{26}$$

The decision boundaries between each two classes were then calculated as in equation (15) and the decision boundaries are as follows:

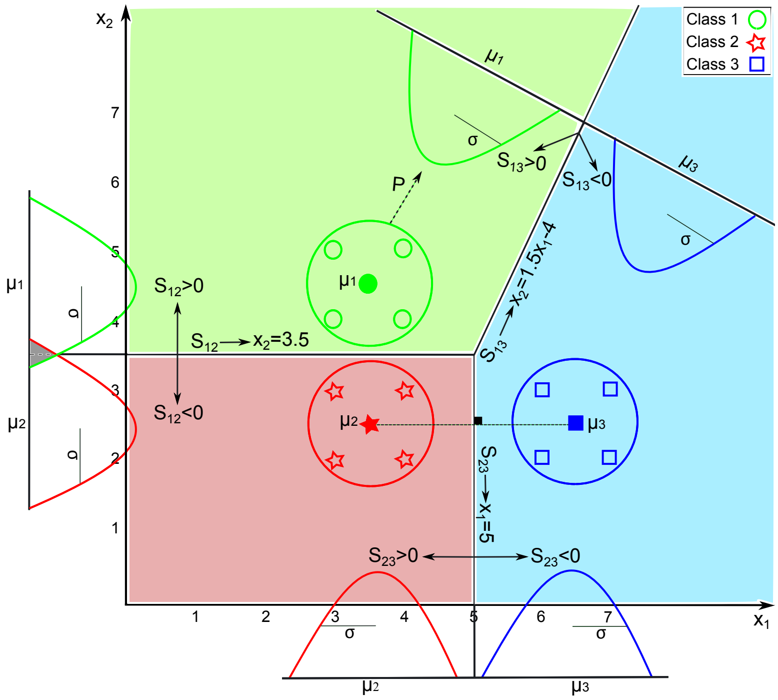
$$\begin{aligned}\mathcal{S}_{12} &= f_1 - f_2 \rightarrow x_2 = 3.50\\ \mathcal{S}_{13} &= f_1 - f_3 \rightarrow x_2 = 1.5x_1 - 4.00\\ \mathcal{S}_{23} &= f_2 - f_3 \rightarrow x_1 = 5.00\end{aligned} \tag{27}$$

Figure 4 shows graphically the decision boundaries between all classes, region of each class, and the distributions of all classes. In Figure 4, the original data, which consists of three classes as in our example are plotted. As shown from Figure 4, according to the data of our example, the covariance matrices were equal and proportional to the identity matrix; hence, the distributions of all classes are represented by circles.[6] Another important finding from Figure 4 that the decision boundary divides the space into positive and negative half spaces. For example, $\mathcal{S}_{12} = 2.00x_2 - 7.00$ divides the space into two spaces, namely, positive half space (where samples from class $\omega_1$ are located) and negative half space (where samples from class $\omega_2$ are located). Three-dimensional visualisation for the decision and functions and decision boundaries are illustrated in Figure 5.

From equation (27), the decision boundary $\mathcal{S}_{12}$ depends only on $x_2$. Thus, for all samples belonging to class $\omega_1$, the value of $x_2$ is greater than 3.5 to be positive. On the other hand, the values of all samples of class $\omega_2$ are lower than 3.5 to be negative.

From equation (27), it can be noticed that w is orthogonal to the decision boundary and it is oriented towards the positive half space. For example, the decision boundary $\mathcal{S}_{12} = 2x_2 - 7$, hence $w = [x_1\ x_2] = [0\ 2]$, which is orthogonal to $\mathcal{S}_{12}$ and it is oriented towards the positive half space or $\omega_1$.

**Figure 4** The calculated decision boundaries for three different classes where the features or variables are statistically independent and have the same variance (our example in Section 3.1) (see online version for colours)



### 3.1.2 Distance between samples and decision boundaries

In this section, the shortest distances between the samples and its decision boundaries were calculated.

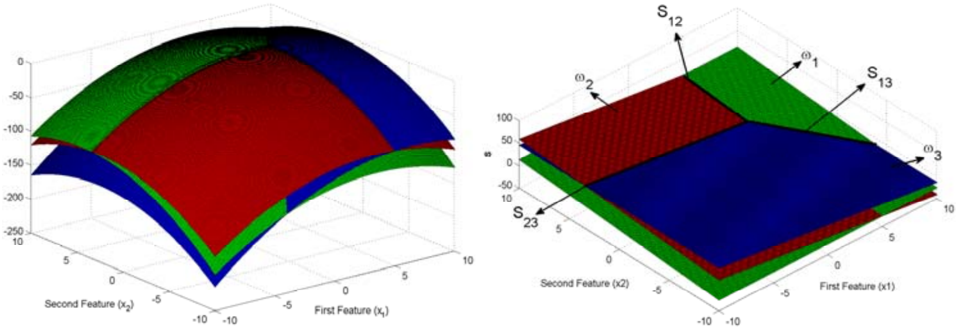
The shortest distance from each sample to the decision boundary ($S$) can be calculated as follows.

$$d_n = \frac{S(x)}{\| w \|} = \frac{w^T x + W_0}{\| w \|} \tag{28}$$

The distances between the decision boundary $S_{12}$ and the samples of the first and second class are denoted by $Dist_1$ and $Dist_2$, respectively. The values of $Dist_1$ and $Dist_2$ are as follows.

$$Dist_1 = \begin{bmatrix} 0.50 \\ 1.50 \\ 0.50 \\ 1.50 \end{bmatrix}, Dist_2 = \begin{bmatrix} -0.50 \\ -1.50 \\ -0.50 \\ -1.50 \end{bmatrix} \tag{29}$$

**Figure 5** Classification of three Gaussian classes with the same covariance matrix ($\Sigma_1 = \Sigma_2 = \Sigma_3 = \sigma^2 I$) (our first example), (a) the green, red, and blue surfaces represent the discriminant functions, $f_1$, $f_2$, and $f_3$, respectively (b) decision boundaries (separation curves) $S_{12} = f_1 - f_2$, $S_{13} = f_1 - f_3$, and $S_{23} = f_2 - f_3$ (see online version for colours)



(a)

(b)

As shown from the above results, the magnitude values for $Dist_1$ and $Dist_2$ are the same, because the two classes, i.e., $\omega_1$ and $\omega_2$, have the same prior probability and the same covariance matrix; hence, the decision boundary $S_{12}$ intersects the distance between $\omega_1$ and $\omega_2$. Moreover, the sign of the two classes is different because the samples of the first class are located in the positive half plane, while the samples of the second class are located in the negative half plane.

### 3.1.3 Matching an unknown sample

In this section, we need to show how an unknown sample was classified by assigning the class that has the maximum discriminant functions to the unknown sample.

Given an unknown or test sample ($T$[2 2]), which has the same dimension of all the training samples, i.e., all samples of $\omega_1$, $\omega_2$, and $\omega_3$. To classify this sample, we need only to substitute the values of the unknown sample into the calculated discriminant functions in equation (26), and assigns the class which has the maximum value, i.e., the largest value of the discriminant function, to the unknown sample as shown in Algorithms (1) and (2). From equation (30), $f_2$ achieved the maximum value, thus the unknown sample belongs to the second class.

$$\begin{aligned} f_1 &= -5.35 \\ f_2 &= -2.35 \\ f_3 &= -11.35 \end{aligned} \tag{30}$$

### 3.1.4 Changing priori probability

In this section, the influence of changing the priori probability was tested on the discriminant functions, decision boundaries, and the final decision of the classifier. From equation (16), the slope of the discriminant function will not be affected by changing the priori probability. On the other hand, the bias of each discriminant function changes according to the prior probability. Hence, the decision boundary will be changed according to the new values of biases. Moreover, if the prior probability of the two classes is equal as in our example, thus the term $\ln \frac{P(\omega_i)}{P(\omega_j)}$ will be zero.

Assume, the priori probability of the three classes in our example were changed to be as follows, $P(\omega_1) = \frac{8}{12}$, $P(\omega_2) = \frac{2}{12}$, and $P(\omega_3) = \frac{2}{12}$, thus the discriminant functions will be as follows.

$$\begin{aligned} f_1 &= -0.5x_1^2 - 0.5x_2^2 + 3.50x_1 + 4.50x_2 - 16.94 \\ f_2 &= -0.5x_1^2 - 0.5x_2^2 + 3.50x_1 + 2.50x_2 - 10.64 \\ f_3 &= -0.5x_1^2 - 0.5x_2^2 + 6.50x_1 + 2.50x_2 - 25.64 \end{aligned} \tag{31}$$

From the above results, it can be seen that the biases of all decision functions were deviated with a little value, while the slope still constant. Consequently, the decision boundaries will be changed as follows.

$$\begin{aligned} S_{12} &= f_1 - f_2 \rightarrow x_2 = 3.15 \\ S_{13} &= f_1 - f_3 \rightarrow x_2 = 1.5x_1 - 4.35 = 0 \\ S_{23} &= f_2 - f_3 \rightarrow x_1 = 5.00 \end{aligned} \tag{32}$$

As shown from the above results, it is apparent that the decision boundary between $\omega_2$ and $\omega_3$ remains constant because the priori probabilities of the two classes are still equal. Thus, the decision boundary between the two classes represents the perpendicular bisector of the line segments joining the centroids as shown in Figure 4. On the other hand, the area of the first class increased because it has priori probability more than the other two classes. In other words, the decision boundaries between the first class and other classes are moved to increase the area of the first class. To conclude, the decision boundary is closer to the mean of the less probable class (Kecman, 2001).

### *3.2 Example 2: Equal variance ($\Sigma_i = \Sigma$)*

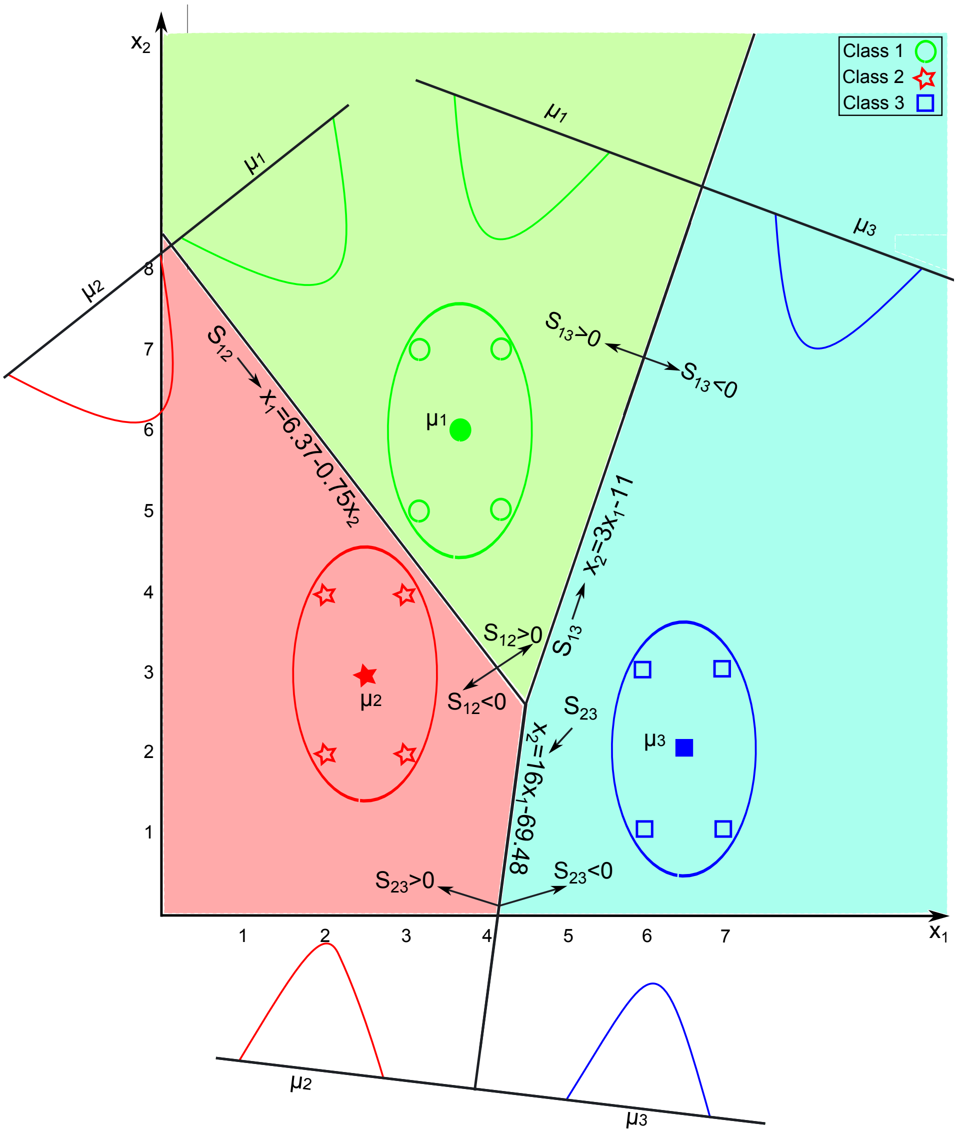
In this example, the covariance matrices of all classes were equal but arbitrary. This case was quite similar to the first case, but the variance of the variables or features were not equal. Geometrical interpretation for this case is that the distributions of all classes were elliptical in m-dimensions space. Due to equal covariance matrices, the discriminant functions of this case also were linear. In this example, the same steps of the first example are followed to calculate the discriminant functions and decision boundaries. MATLAB code for this experiment is introduced in Appendix.

**Table 2** Feature values, mean, mean-centring data, and covariance matrices for all classes of the example in Section 3.2

| *Pattern no.* | *Features* | | *Class* | *Mean* | | *D* | | *Covariance matrix ($\Sigma_i$)* |
|---|---|---|---|---|---|---|---|---|
| | $x_1$ | $x_2$ | | $x_1$ | $x_2$ | $x_1$ | $x_2$ | |
| 1 | 3.00 | 5.00 | $\omega_1$ | 3.50 | 6.00 | –0.50 | –1.00 | $\Sigma_1 = \begin{bmatrix} 1.00 & 0.00 \\ 0.00 & 4.00 \end{bmatrix}$ |
| 2 | 3.00 | 7.00 | | | | –0.50 | 1.00 | |
| 3 | 4.00 | 5.00 | | | | 0.50 | –1.00 | |
| 4 | 4.00 | 7.00 | | | | 0.50 | 1.00 | |
| 5 | 2.00 | 2.00 | $\omega_2$ | 2.50 | 3.00 | –0.50 | –1.00 | $\Sigma_2 = \begin{bmatrix} 1.00 & 0.00 \\ 0.00 & 4.00 \end{bmatrix}$ |
| 6 | 2.00 | 4.00 | | | | –0.50 | 1.00 | |
| 7 | 3.00 | 2.00 | | | | 0.50 | –1.00 | |
| 8 | 3.00 | 4.00 | | | | 0.50 | 1.00 | |
| 9 | 6.00 | 1.00 | $\omega_3$ | 6.50 | 2.00 | –0.50 | –1.00 | $\Sigma_3 = \begin{bmatrix} 1.00 & 0.00 \\ 0.00 & 4.00 \end{bmatrix}$ |
| 10 | 6.00 | 3.00 | | | | –0.50 | 1.00 | |
| 11 | 7.00 | 1.00 | | | | 0.50 | –1.00 | |
| 12 | 7.00 | 3.00 | | | | 0.50 | 1.00 | |

Given three different classes denoted by, $\omega_1$, $\omega_2$, $\omega_3$ as shown in Figure 6. Each class consists of four samples, and each sample was represented by two features, $x_1$ and $x_2$ as shown in Table 2. Values of the mean of each class, mean-centring data, and the covariance matrices are summarised in Table 2.

**Figure 6** The calculated decision boundaries for three different classes where their covariance matrices were equal but arbitrary (our example in Section 3.2) (see online version for colours)



Values of the inverse of the covariance matrices are as follows

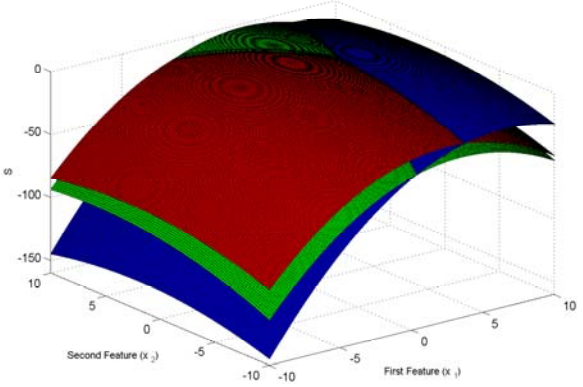
$$\Sigma_1^{-1} = \Sigma_2^{-1} = \Sigma_3^{-1} = \begin{bmatrix} 1.00 \; 0.00 \\ 0.00 \; 0.25 \end{bmatrix} \tag{33}$$

As shown in Table 2, each class is represented by four samples, thus the priori probability of each class is as follows, $P(\omega_1) = P(\omega_2) = P(\omega_3) = \frac{4}{12}$.

The discriminant functions were then calculated and their values will be as follows.

$$\begin{aligned} f_1 &= -0.5x_1^2 - 0.125x_2^2 + 3.50x_1 + 1.5x_2 - 11.72 \\ f_2 &= -0.5x_1^2 - 0.125x_2^2 + 2.50x_1 + 0.75x_2 - 5.35 \\ f_3 &= -0.5x_1^2 - 0.125x_2^2 + 6.50x_1 + 0.50x_2 - 22.72 \end{aligned} \tag{34}$$

**Figure 7** Classification of three Gaussian classes with the same covariance matrix ($\Sigma_1 = \Sigma_2 = \Sigma_3$) (our second example)



Note: Green, red, and blue surfaces represent $f_1, f_2$, and $f_3$, respectively.

The decision boundaries between each two classes were then calculated as follows:

$$\begin{aligned} S_{12} &= f_1 - f_2 \rightarrow x_1 = 6.37 - 0.75x_2 \\ S_{13} &= f_1 - f_3 \rightarrow x_2 = 3.00x_1 - 11.00 \\ S_{23} &= f_2 - f_3 \rightarrow x_2 = 16x_1 - 69.48 \end{aligned} \tag{35}$$

Figure 6 shows graphically the original data of this example, which consists of three classes, the decision boundaries between all classes, the region for each class, and the distributions of all classes. Moreover, Figure 7 shows the decision functions in threedimensional space.

### *3.3 Example 3: Different covariance matrices ($\Sigma_i$ = arbitrary)*

In this example, the covariance matrices were different for all classes and we can consider this case represents the common case. MATLAB code for this experiment is introduced in Appendix.

In this case, due to the arbitrary covariance matrices, the distributions of all classes were different. Moreover, the coefficient of the quadratic term was calculated as in equation (11). In addition, the discriminant functions and decision boundaries were calculated as denoted in equations (8), (10), respectively.

Given three different classes denoted by, $\omega_1$, $\omega_2$, $\omega_3$ as shown in Figure 8. Each class had four samples, and each sample was represented by two features, x1 and x2 as shown in Table 3. Values of the mean of each class, mean-centring data, and the covariance matrices are summarised in Table 3.

**Table 3** Feature values, mean, mean-centring data, and covariance matrices for all classes of the example in Section 3.3

| *Pattern no.* | *Features* | | *Class* | *Mean* | | *D* | | *Covariance matrix* ($\Sigma_i$) |
|---|---|---|---|---|---|---|---|---|
| | $x_1$ | $x_2$ | | $x_1$ | $x_2$ | $x_1$ | $x_2$ | |
| 1 | 7.00 | 3.00 | $\omega_1$ | 7.50 | 3.50 | –0.50 | –0.50 | $\Sigma_1 = \begin{bmatrix} 1.00 & 0.00 \\ 0.00 & 1.00 \end{bmatrix}$ |
| 2 | 8.00 | 3.00 | | | | 0.50 | –0.50 | |
| 3 | 7.00 | 4.00 | | | | –0.50 | 0.50 | |
| 4 | 8.00 | 4.00 | | | | 0.50 | 0.50 | |
| 5 | 2.00 | 2.00 | $\omega_2$ | 3.50 | 2.50 | –1.50 | –0.50 | $\Sigma_2 = \begin{bmatrix} 9.00 & 0.00 \\ 0.00 & 1.00 \end{bmatrix}$ |
| 6 | 5.00 | 2.00 | | | | 1.50 | –0.50 | |
| 7 | 2.00 | 3.00 | | | | –1.50 | 0.50 | |
| 8 | 5.00 | 3.00 | | | | 1.50 | 0.50 | |
| 9 | 1.00 | 6.00 | $\omega_3$ | 3.00 | 6.50 | –2.00 | –0.50 | $\Sigma_3 = \begin{bmatrix} 16.00 & 0.00 \\ 0.00 & 1.00 \end{bmatrix}$ |
| 10 | 5.00 | 6.00 | | | | 2.00 | –0.50 | |
| 11 | 1.00 | 7.00 | | | | –2.00 | 0.50 | |
| 12 | 5.00 | 7.00 | | | | 2.00 | 0.50 | |

The values of the inverse of the covariance matrices are as follows.

$$\Sigma_1^{-1} = \begin{bmatrix} 1.00 & 0.00 \\ 0.00 & 1.00 \end{bmatrix} \Sigma_2^{-1} = \begin{bmatrix} 0.11 & 0.00 \\ 0.00 & 1.00 \end{bmatrix} \Sigma_3^{-1} = \begin{bmatrix} 0.06 & 0.00 \\ 0.00 & 1.00 \end{bmatrix} \tag{36}$$

From Table 3, the priori probability for each class is as follows, $P(\omega_1) = P(\omega_2) = P(\omega_3) = \frac{4}{12}$.

The discriminated functions were then calculated and its values will be as follows:

$$\begin{aligned} f_1 &= -0.50x_1^2 - 0.50x_2^2 + 7.50x_1 + 3.50x_2 - 35.35 \\ f_2 &= -0.06x_1^2 - 0.50x_2^2 + 0.39x_1 + 2.50x_2 - 6.00 \\ f_3 &= -0.03x_1^2 - 0.50x_2^2 + 0.19x_1 + 6.50x_2 - 23.89 \end{aligned} \tag{37}$$

The decision boundaries between each two classes were then calculated as follows:

$$\begin{aligned} \mathcal{S}_{12} &= f_1 - f_2 \rightarrow x_2 = 0.44x_1^2 - 7.11x_1 + 29.35 \\ \mathcal{S}_{13} &= f_1 - f_3 \rightarrow x_2 = 0.16x_1^2 + 2.44x_1 - 3.82 \\ \mathcal{S}_{23} &= f_2 - f_3 \rightarrow x_2 = -0.01x_1^2 + 0.05x_1 + 4.47 \end{aligned} \tag{38}$$

From the above results it can be seen that all the decision boundaries are represented by parabola, i.e., nonlinear. The vertex of the parabola is calculated as follows, $\frac{-b}{2a}$, where a and b in our example represent the coefficients of $x_1^2$ and $x_1$, respectively. Thus, the vertex of $\mathcal{S}_{12}$ on equation (38) is calculated as follows, $\left( x_1 = \frac{7.11}{2 * 0.44} \approx, 8.08, x_2 \approx 0.63 \right)$.

Similarly, the vertices of the $\mathcal{S}_{13}$ and $\mathcal{S}_{23}$ equations will be follows, (7.630, 5.48), and (2.50, 4.53).

**Figure 8** The calculated decision boundaries for three different classes where their covariance matrices are different (our example in Section 3.3) (see online version for colours)

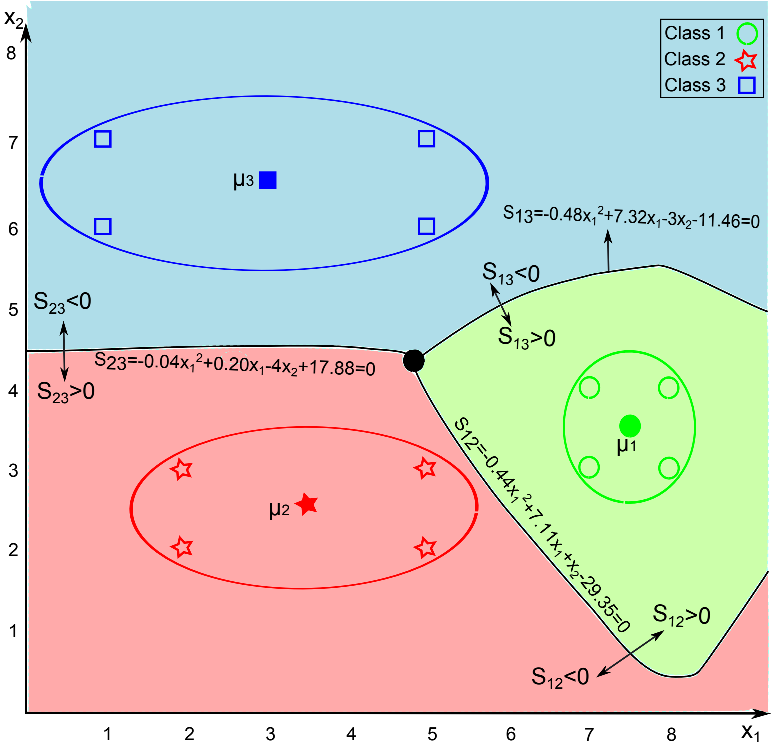
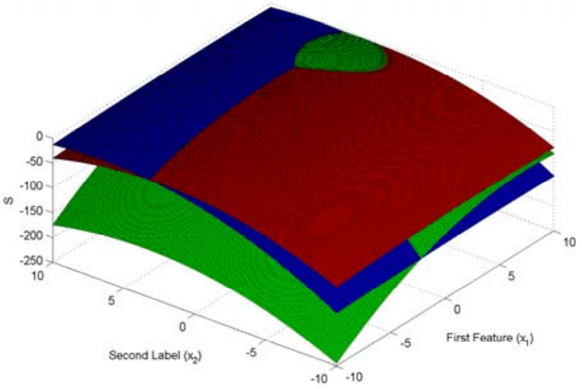


**Figure 9** Classification of three Gaussian classes with different covariance matrix (our third example)



Note: Green, red, and blue surfaces represent the discriminant functions, $f_1$, $f_2$, and $f_3$, respectively.

Figure 8 shows graphically the original data of this example, which consists of three classes, the decision boundaries between all classes, the region for each class, and the distributions of all classes. In addition, Figure 9 shows the decision functions in three-dimensional space.

## 4 Singularity problem

One of the DA problems is the singularity[7] of the covariance matrix, which is called singularity, small sample size, or under-sampled problem. This problem results from a high-dimensional pattern or sample and a lower number of samples in each class (Lu et al., 2005; Ye and Xiong, 2006; Guo et al., 2007). In other words, the upper bound of the rank[8] of the covariance matrix ($\Sigma_i$) is $n_i - 1$, while the dimension of $\Sigma_i$ is $M \times M$ (Lu et al., 2005) and in most cases $M \gg n_i - 1$, which leads to the singularity problem. For example, in face recognition applications, the sizes of the face image my reach to 100 × 100 = 10,000 pixels, which represent high-dimensional features and to make the covariance matrix non-singular, we need at least 10,000 samples, which are not available in all datasets. In the next section, a numerical example is introduced to explain the singularity problem and how to solve it.

**Table 4** The feature values, mean, mean-centring data of all samples of the example in Section 4.1

| *Pattern no.* | *Features* | | | | *Class* | *Mean* | | | | *D* | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $x_1$ | $x_2$ | $x_3$ | $x_4$ | | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ |
| 1 | 3 | 4 | 3 | 5 | $\omega_1$ | 3.33 | 4.33 | 4.67 | 5.33 | –0.33 | –0.33 | –1.67 | –0.33 |
| 2 | 3 | 5 | 6 | 4 | | | | | | –0.33 | 0.67 | 1.33 | –1.33 |
| 3 | 4 | 4 | 5 | 7 | | | | | | 0.67 | –0.33 | 0.33 | 1.67 |
| 4 | 3 | 2 | 5 | 2 | | | | | | –0.33 | –0.33 | 0.67 | –1.33 |
| 5 | 3 | 3 | 5 | 3 | $\omega_2$ | 3.33 | 2.33 | 4.33 | 3.33 | –0.33 | 0.67 | 0.67 | –0.33 |
| 6 | 4 | 2 | 3 | 5 | | | | | | 0.67 | –0.33 | –1.33 | 1.67 |
| 7 | 6 | 2 | 5 | 6 | | | | | | –0.33 | –0.33 | –0.33 | –0.67 |
| 8 | 6 | 3 | 6 | 7 | $\omega_3$ | 6.33 | 2.33 | 5.33 | 6.67 | –0.33 | 0.67 | 0.67 | 0.33 |
| 9 | 7 | 2 | 5 | 7 | | | | | | 0.67 | –0.33 | –0.33 | 0.33 |

### 4.1 Numerical illustration of singularity problem

Given three different classes denoted by, $\omega_1$, $\omega_2$, $\omega_3$, each class ($\omega_i$) had three samples, and each sample was represented by four features, $x_1$, $x_2$, $x_3$, and $x_4$, i.e., $M > n_i$, $i = 1, 2, 3$. Table 4 shows the values of all samples in each class. Assume the priori probability of the three classes were equal $\left(P(\omega_1) = P(\omega_2) = P(\omega_3) = \frac{1}{3}\right)$. The values of the mean of each class and the mean-centring data ($D_i$, $i = 1, 2, 3$) are also shown in Table 4.

The covariance matrices for all classes are as follows.

$$\Sigma_1 = \begin{bmatrix} 0.66 & -0.33 & 0.33 & 1.67 \\ -0.33 & 0.67 & 1.33 & -1.33 \\ 0.33 & 1.33 & 4.67 & -0.67 \\ 1.67 & -1.33 & -0.67 & 4.67 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 0.67 & -0.33 & -1.33 & 1.67 \\ -0.33 & 0.67 & 0.67 & -0.33 \\ -1.33 & 0.67 & 2.67 & -3.33 \\ 1.67 & -0.33 & -3.33 & 4.67 \end{bmatrix}$$
$$\Sigma_3 = \begin{bmatrix} 0.67 & -0.33 & -0.33 & 0.33 \\ -0.33 & 0.67 & 0.67 & 0.33 \\ -0.33 & 0.67 & 0.67 & 0.33 \\ 0.33 & 0.33 & 0.33 & 0.67 \end{bmatrix} \tag{39}$$

As shown from the above results, the rank of all covariance matrices was two, i.e., the rank = $n_i$ – 1. Thus, the covariance matrices were singular; hence, the discriminant functions cannot be calculated. There are many studies proposed different solutions for this problem, each has its advantages and drawbacks. In this research, two common state-of-the-art methods were used to solve the singularity problem of the covariance matrix, namely, RLDA and subspace methods. In the next two subsections, the two methods are explained with numerical illustrations.

## 4.2 RLDA method

In this method, the identity matrix is scaled by multiplying it by a regularisation parameter ($1 > \eta > 0$) and adding it to the covariance matrix to make it non-singular (Friedman, 1989; Lu et al., 2005; Ye and Xiong, 2006). Thus, the diagonal elements of the covariance matrix are biased as follows, $\hat{\Sigma} = \Sigma + \eta I$. However, choosing the value of the regularisation parameter requires more tuning and a poor choice for this parameter can degrade the performance of the method (Lu et al., 2005). Another problem of this method is that the parameter $\eta$ is just added to perform the inverse of $\Sigma$ and has no clear mathematical interpretation (Lu et al., 2005; Sharma and Paliwal, 2014).

### 4.2.1 Numerical illustration of the RLDA method

The aim of this section was to numerically illustrate the RLDA solution to solve the singularity problem of the covariance matrices. In this example, $\eta$ = 0.05 and the covariance matrices were calculated as follows, $\hat{\Sigma}_i = \Sigma_i + \eta\Sigma_i$ and the values of $\hat{\Sigma}_i$ are as follows.

$$\hat{\Sigma}_1 = \begin{bmatrix} 0.72 & -0.33 & 0.33 & 1.67 \\ -0.33 & 0.72 & 1.33 & -1.33 \\ 0.33 & 1.33 & 4.72 & -0.67 \\ 1.67 & -1.33 & -0.67 & 4.72 \end{bmatrix} \quad \hat{\Sigma}_2 = \begin{bmatrix} 0.72 & -0.33 & -1.33 & 1.67 \\ -0.33 & 0.72 & 0.67 & -0.33 \\ -1.33 & 0.67 & 2.72 & -3.33 \\ 1.67 & -0.33 & -3.33 & 4.72 \end{bmatrix}$$
$$\hat{\Sigma}_3 = \begin{bmatrix} 0.72 & -0.33 & -0.33 & 0.33 \\ -0.33 & 0.72 & 0.67 & 0.33 \\ -0.33 & 0.67 & 0.72 & 0.33 \\ 0.33 & 0.33 & 0.33 & 0.72 \end{bmatrix} \tag{40}$$

The inverse of $\hat{\Sigma}_i$ are as follows.

$$\hat{\Sigma}_1^{-1} = \begin{bmatrix} 17.38 & 0.96 & -2.38 & -6.21 \\ 0.96 & 17.85 & -4.54 & 4.07 \\ -2.38 & -4.54 & 1.63 & -0.21 \\ -6.21 & 4.07 & -0.21 & 3.52 \end{bmatrix} \quad \hat{\Sigma}_2^{-1} = \begin{bmatrix} 17.93 & 3.00 & 4.14 & 3.20 \\ 3.00 & 6.11 & -6.01 & -4.88 \\ 4.14 & -6.00 & 11.73 & 6.40 \\ -3.20 & -4.88 & 6.40 & 5.52 \end{bmatrix}$$
$$\hat{\Sigma}_3^{-1} = \begin{bmatrix} 8.53 & 3.88 & 3.88 & -7.58 \\ 3.88 & 12.23 & -7.77 & -3.88 \\ 3.88 & -7.77 & 12.23 & -3.88 \\ -7.58 & -3.88 & -3.88 & 8.53 \end{bmatrix} \tag{41}$$

As shown from the above results, the singularity problem of the covariance matrices is solved and the discriminant functions are then calculated as in equation (8). However, the discriminant functions and decision boundaries are changed according to different regularisation parameters.

### 4.3 Subspace method

In this method, a non-singular intermediate space is obtained to reduce the dimension of the original data to be equal to the rank of the covariance matrix, hence $\Sigma_i$ becomes full-rank.[9] In other words, a dimensionality reduction method is used to remove the null-space of the covariance matrices (Yu and Yang, 2001). Principal component analysis (PCA) is one of the most common dimensionality reduction methods (Tharwat et al., 2012, 2015; Tharwat, 2016).

However, losing some discriminant information is a common drawback associated with the use of this method (Gao and James, 2006).

**Table 5** The feature values, mean, mean-centring data, covariance matrices, and the inverse of the covariance matrices of all classes of the example in Section 4.1 after projecting it onto the PCA space to reduce the dimension of the original data

| *Pattern no.* | *Features* | | *Class* | *Mean* | | *D* | | *Covariance matrix* ($\Sigma_i$) | *Inverse covariance matrix* ($\Sigma_i^{-1}$) |
|---|---|---|---|---|---|---|---|---|---|
| | $x_1$ | $x_2$ | | $x_1$ | $x_2$ | $x_1$ | $x_2$ | | |
| 1 | 1.38 | 6.18 | $\omega_1$ | 0.67 | 7.80 | 0.71 | −1.61 | $\Sigma_1 = \begin{bmatrix} 6.22 & 0.00 \\ 0.00 & 4.45 \end{bmatrix}$ | $\Sigma_1^{-1} = \begin{bmatrix} 0.16 & 0.00 \\ 0.00 & 0.23 \end{bmatrix}$ |
| 2 | −1.34 | 8.08 | $\omega_1$ | | | −2.01 | 0.23 | | |
| 3 | 1.96 | 9.13 | $\omega_1$ | | | 1.30 | 1.33 | | |
| 4 | −0.69 | 3.56 | $\omega_1$ | 7.64 | 4.13 | −1.46 | −0.57 | $\Sigma_2 = \begin{bmatrix} 7.86 & 0.00 \\ 0.00 & 0.81 \end{bmatrix}$ | $\Sigma_2^{-1} = \begin{bmatrix} 0.13 & 0.00 \\ 0.00 & 1.24 \end{bmatrix}$ |
| 5 | −0.04 | 4.81 | $\omega_1$ | | | −0.80 | 0.69 | | |
| 6 | 3.02 | 4.00 | $\omega_1$ | | | 2.26 | −0.12 | | |
| 7 | 4.43 | 8.49 | $\omega_1$ | 4.95 | 9.19 | −0.53 | −0.71 | $\Sigma_3 = \begin{bmatrix} 1.67 & 0.00 \\ 0.00 & 1.00 \end{bmatrix}$ | $\Sigma_3^{-1} = \begin{bmatrix} 0.60 & 0.00 \\ 0.00 & 1.00 \end{bmatrix}$ |
| 8 | 6.01 | 9.19 | $\omega_1$ | | | 1.05 | 0 | | |
| 9 | 4.43 | 9.90 | $\omega_1$ | | | −0.53 | 0.71 | | |

### 4.3.1 Numerical illustration of the subspace method

In this section, the dimensions of the original data, i.e., the data of each class, were reduced using PCA technique to be equal to the rank of the covariance matrix. The main idea of the PCA technique is to calculate the eigenvalues and eigenvectors of the data matrix and neglect the eigenvectors, which have lower eigenvalues.

The eigenvalues ($\lambda$) and eigenvectors ($V$) of all classes are as follows.

$$\begin{aligned}
\lambda_1 &= \begin{bmatrix} 6.22 \\ 4.45 \\ 0.00 \\ 0.00 \end{bmatrix}, V_1 = \begin{bmatrix} 0.21 & 0.30 & 0.43 & -0.82 \\ -0.32 & 0.06 & -0.81 & -0.49 \\ -0.55 & 0.79 & 0.15 & 0.22 \\ 0.74 & 0.53 & -0.37 & 0.18 \end{bmatrix} \\
\lambda_2 &= \begin{bmatrix} 7.86 \\ 0.81 \\ 0.00 \\ 0.00 \end{bmatrix}, V_2 = \begin{bmatrix} 0.29 & -0.15 & -0.60 & -0.73 \\ -0.10 & 0.85 & -0.48 & 0.18 \\ -0.57 & 0.30 & 0.42 & -0.64 \\ 0.76 & 0.40 & 0.48 & 0.18 \end{bmatrix} \\
\lambda_3 &= \begin{bmatrix} 7.86 \\ 0.81 \\ 0.00 \\ 0.00 \end{bmatrix}, V_3 = \begin{bmatrix} -0.32 & 0.71 & -0.19 & -0.61 \\ 0.63 & 0.00 & -0.58 & -0.51 \\ 0.63 & 0.00 & -0.77 & -0.09 \\ 0.32 & 0.71 & 0.19 & 0.60 \end{bmatrix}
\end{aligned} \tag{42}$$

where $\lambda_i$ and $V_i$ represent the eigenvalues and eigenvectors of the $i^{\text{th}}$ class ($\omega_i$).

As shown from the above results, the first two eigenvalues represent 100% of the total variance, which reflects that the other, i.e., last, two eigenvectors are less important and it can be removed. In other words, the first two eigenvectors ($V_i^{1,2}$) were used to construct a lower dimensional space to project the original data onto as follows, $\omega_i V_i^{1,2}$. Thus, the projected data are represented by only two features. Table 5 illustrates the values of the samples of all classes after projection. Moreover, Table 5 summarises the mean of each class, mean-centring data, covariance matrices, and the inverse of the covariance matrices. Hence, the discriminant functions were then calculated as denoted in equation (8). The values of the discriminant functions and decision boundaries were changed according to the dimensionality reduction method and the number of features after reduction.

## 5 Experimental results and discussion

In this section, an experiment was conducted to compare between the linear and quadratic discriminant analysis using different numbers of training samples. Moreover, different datasets with different dimensions were used to show how to solve the singularity problem of the higher-dimensional datasets.

**Table 6** Datasets descriptions

| *Dataset* | *Number of classes* | *Number of features (dimension)* | *Number of samples* |
|---|---|---|---|
| Iris | 3 | 4 | 150 (3 × 50) |
| Sonar | 2 | 60 | 208 (97 + 111) |
| Wine | 3 | 13 | 178 (59 + 71 + 48) |
| Liver | 2 | 6 | 345 (145 + 200) |
| Diabetes | 2 | 8 | 768 (500 + 268) |
| Breast cancer | 2 | 10 | 683 (444 + 239) |
| Ovarian | 2 | 4,000 | 216 (121 + 95) |
| $\text{ORL}_{32\times32}$ | 10 | 1,024 | 400 (40 × 10) |
| $\text{Ear}_{32\times32}$ | 17 | 1,024 | 112 (17 × 6) |
| $\text{Yale}_{32\times32}$ | 15 | 1,024 | 165 (15 × 11) |

## *5.1 Experimental setup*

In this experiment, ten standard datasets with different dimensions were used (see Table 6). Each dataset has different numbers of attributes, classes, and samples. Table 6 shows the number of classes was in the range of [2, 17], the dimensionality was in the range of [4, 4000], and the number of samples was in the range of [112, 768]. The iris dataset consists of three different types of flowers, namely, Setosa, Versicolor, and Versicolor. Sonar dataset consists of two classes, rock (if the detected object is a rock) and metal (if the detected object is metal). Wine dataset represents a chemical analysis of three types of wines. The liver dataset represents the liver disorders of male individuals that might arise from excessive alcohol consumption. The diabetes dataset records the diabetes patients that obtained from:

1 automatic recording device

2 paper records.

The ovarian dataset was collected at the Pacific Northwestern National Lab (PNNL) and Johns Hopkins University. This dataset was considered one of the large-scale or high-dimensional datasets used to investigate tumours through deep proteomic analysis and it consists of two classes, namely Normal and Cancer. Olivetti Research Laboratory, Cambridge (ORL)[10] dataset (Samaria and Harter, 1994), which consists of 40 individuals, each has ten grey scale images. The size of each image was 92 × 112 and it resized to be 323 × 32 to reduce the computational time. Ear dataset images,[11] consists of 17 individuals, each has six grey scale images (Carreira-Perpinan, 1995). The images have different dimensions, thus, all images were resized to be 32 × 32. Yale[12] face dataset images contain 165 grey scale images in GIF format of 15 individuals (Yang et al., 2004). Each individual has 11 images in different expressions and configuration: centre-light, happy, left-light, with glasses, normal, right-light, sad, sleepy, surprised, and wink. The size of each image was 320 × 243 and it resized to be 32 × 32. Figure 10 shows samples from the face and ear datasets. The results of this experiment were evaluated using the accuracy or recognition rate which represents the percentage of the total number of predictions that were correct as denoted in equation (43). Due to the random selection of the training and testing samples, each classifier run ten times and the mean and standard deviation of the results were calculated. Finally, the experimental environment includes Window XP operating system, Intel(R) Core(TM) i5-2400 CPU @ 3.10 GHz, 4.00 GB RAM, and MATLAB (R2013b).

$$ACC = \frac{\text{Number of correctly classified samples}}{\text{Total number of testing samples}} \tag{43}$$

**Table 7** A comparison between linear and QDA classifiers in terms of accuracy

| *Dataset* | *Linear discriminant classifier* | | | |
|---|---|---|---|---|
| | *20%* | *40%* | *60%* | *80%* |
| Iris | 60.5 ± 0.5 | 75.6 ± 0.2 | 85.6 ± 0.1 | 97.33 ± 0.05 |
| Sonar | 52.6 ± 0 | 58.6 ± 0.3 | 64.5 ± 0.2 | 73.05 ± 0.02 |
| Liver | 47.2 ± 0.1 | 51.3 ± 0.1 | 56.2 ± 0.5 | 59.0 ± 0.08 |
| Wine | 88.5 ± 0.1 | 90.5 ± 0.1 | 94.2 ± 0.3 | 99.41 ± 0.09 |
| Diabetes | 61.5 ± 0.3 | 66 ± 0.3 | 68.5 ± 0.4 | 72.24 ± 0.06 |
| Breast cancer | 82.5 ± 0.1 | 87.5 ± 0.3 | 91.2 ± 0.4 | 95.025 ± 0.25 |
| $\text{ORL}_{32\times32}$ | 75.4 ± 0.2 | 82.6 ± 0.9 | 86.3 ± 0.4 | 90.4 ± 0.02 |
| $\text{Yale}_{32\times32}$ | 69.4 ± 0.5 | 73.1 ± 0.4 | 78.6 ± 0.4 | 81.5 ± 0.02 |
| $\text{Ear}_{32\times32}$ | 68.4 ± 0.5 | 72.0 ± 0.3 | 79.4 ± 0.5 | 88.45 ± 0.25 |
| *Dataset* | *Quadratic discriminant classifier* | | | |
| | *20%* | *40%* | *60%* | *80%* |
| Iris | 65.2 ± 0.2 | 78.4 ± 0.1 | 89.5 ± 0.2 | 99.33 ± 0.01 |
| Sonar | 53.4 ± 0.1 | 59.4 ± 0.5 | 63.5 ± 0 | 74.5 ± 0.06 |
| Liver | 51.2 ± 0.6 | 56.1 ± 0.1 | 59.4 ± 0.7 | 65.3 ± 0.1 |
| Wine | 89.2 ± 0.4 | 92.3 ± 0.7 | 95.3 ± 0.5 | 100 ± 0 |
| Diabetes | 61.2 ± 0.3 | 64.5 ± 0.4 | 67.4 ± 0.5 | 74.6 ± 0.05 |
| Breast cancer | 81.5 ± 0.2 | 86.3 ± 0.5 | 92.5 ± 0.7 | 95.3 ± 0.02 |
| $\text{ORL}_{32\times32}$ | 76.8 ± 0.4 | 80.5 ± 0.4 | 86.3 ± 0.2 | 91.54 ± 0.23 |
| $\text{Yale}_{32\times32}$ | 70.5 ± 0.4 | 72.3 ± 0.8 | 77.8 ± 0.8 | 82.6 ± 0.12 |
| $\text{Ear}_{32\times32}$ | 73.2 ± 0.1 | 75.4 ± 0.8 | 79.2 ± 0.3 | 89.2 ± 0.04 |

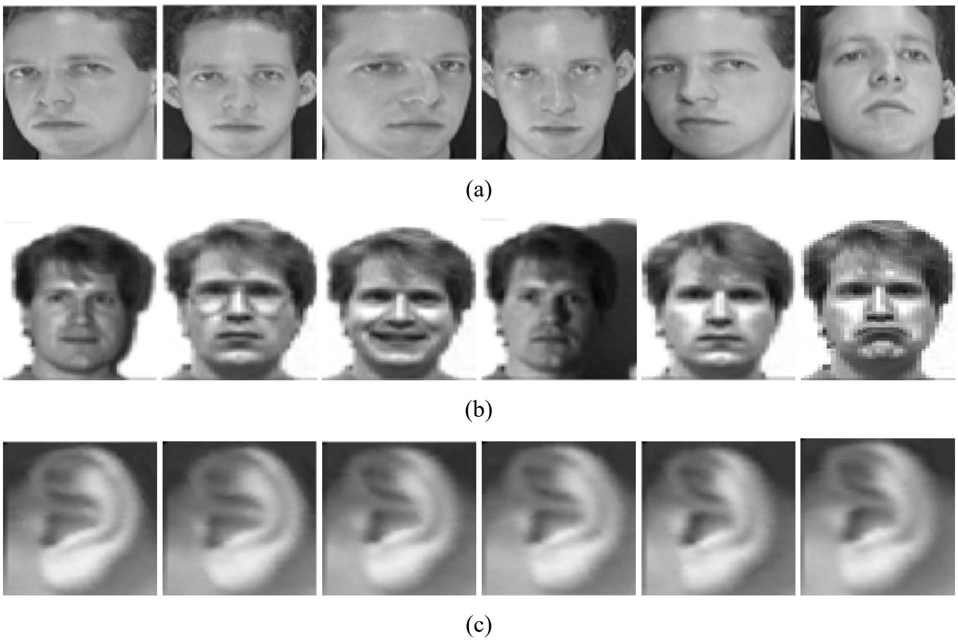
### *5.2 Experimental scenario*

This experiment is divided into two sub-experiments. The first sub-experiment used the datasets which have low dimensions, while the second sub-experiment used the high-dimensional datasets. Due to the low-dimensionality in the first sub-experiment, the linear and quadratic classifiers can be calculated directly. On the other hand, the high-dimensional datasets led to singularity problem. This problem, i.e., singularity problem, was solved by using PCA to reduce the number of features to make the covariance matrix full rank. In both sub-experiments, different percentages of the training samples were used. The results of this experiment are summarised in Table 7. The steps of this experiment are summarised in Algorithm 3 and the MATLAB code for this experiment is presented in Appendix.

### *5.2.1 Discussion*

The results in Table 7 show that the accuracy of linear and quadratic classifiers using different datasets.

**Figure 10** Samples of the first individual in (a) ORL face dataset, (b) Yale face dataset and (c) ear dataset



(a)

(b)

(c)

**Algorithm 3** Steps of the linear and quadratic classifiers

```
Divide the given dataset into training and testing sets, where each set has labelled samples.
if (if the dataset has high dimensions [see Section 4)] then
    Use the PCA technique to reduce the number of features to the extent which makes the
    covariance matrix is full rank.
end if
Train the linear and quadratic models using the training set.
Predict or estimate the class of the testing or unseen data using the trained model.
```

As shown in Table 7, the accuracy of both linear and quadratic classifiers was proportional to the number of training samples. As shown, the minimum accuracy achieved when only 20% of the training samples was used, while the accuracy increased when the number of training samples increased. Moreover, quadratic discriminant classifier achieved accuracy better than linear classifier.

## 6 Conclusions

Calculating the discriminant functions and constructing the decision boundaries are two important goals in different classifiers. This paper focuses on the DA classifier, which is one of the most common classifiers. This paper followed the approach of not only explaining the steps of calculating the discriminant functions but also visualising these steps with figures and diagrams to make it easy to understand. In addition, three numerical examples were given and graphically illustrated to explain how to calculate the discriminant functions when the covariance matrices of all classes were common, i.e., the decision boundaries are linear or quadratic. In all numerical examples, the discriminant functions and decision boundaries were calculated with a graphical illustration. Moreover, the singularity problem was mathematically explained using numerical examples, then two state-of-the-art solutions were highlighted with numerical illustrations.

## References

Altman, E.I., Marco, G. and Varetto, F. (1994) 'Corporate distress diagnosis: comparisons using linear discriminant analysis and neural networks (the Italian experience)', *Journal of Banking and Finance*, Vol. 18, No. 3, pp.505–529.

Belhumeur, P.N., Hespanha, J.P. and Kriegman, D.J. (1997) 'Eigenfaces vs. fisherfaces: Recognition using class specific linear projection', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp.711–720.

Carreira-Perpinan, M.A. (1995) *Compression Neural Networks for Feature Extraction: Application to Human Recognition from Ear Images*, MS thesis, Faculty of Informatics, Technical University of Madrid, Spain.

Duda, R.O., Hart, P.E. and Stork, D.G. (2012) *Pattern Classification*, 2nd ed., John Wiley & Sons, New York, NY, ISBN: 978-0-471-05669-0.

Friedman, J.H. (1989) 'Regularized discriminant analysis', *Journal of the American statistical Association*, Vol. 84, No. 405, pp.165–175.

Fukunaga, K. (2013) *Introduction to Statistical Pattern Recognition*, 2nd ed., Academic Press, San Diego, CA, USA, ISBN: 0-12-269851-7.

Gao, H., and Davis, J.W. (2006) 'Why direct LDA is not equivalent to LDA', *Pattern Recognition*, Vol. 39, No. 5, pp.1002–1006.

Golub, G.H. and Van Loan, C.F. (2012) *Matrix Computations*, 3rd ed., John Hopkins University Press, Baltimore, ISBN: 978-0801854149.

Guo, Y., Hastie, T. and Tibshirani, R. (2007) 'Regularized linear discriminant analysis and its application in microarrays', *Biostatistics*, Vol. 8, No. 1, pp.86–100.

Hahne, J.M., BieBmann, F., Jiang, N., Rehbaum, H., Farina, D., Meinecke, F.C. and Parra, L.C. (2014) 'Linear and nonlinear regression techniques for simultaneous and proportional myoelectric control', *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 22, No. 2, pp.269–279.

Hastie, T., Tibshirani, R., Friedman, J. and Franklin, J. (2005) 'The elements of statistical learning: data mining, inference and prediction', *The Mathematical Intelligence*, Vol. 27, No. 2, pp.83–85.

Kecman, V. (2001) *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*, 1st ed., MIT Press.

Loh, W.Y. (2011) 'Classification and regression trees', *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Vol. 1, No. 1, pp.14–23.

Lu, J., Plataniotis, K.N. and Venetsanopoulos, A.N. (2005) '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.

McLachlan, G. (2004) *Discriminant analysis and Statistical Pattern Recognition*, 2nd ed., John Wiley & Sons, Hoboken, New Jersey, USA, ISBN: 978-0-471-69115-0.

Montgomery, D.C., Peck, E.A. and Vining, G.G. (2012) *Introduction to Linear Regression Analysis*, 4th ed., John Wiley & Sons, Hoboken, New Jersey, USA, ISBN: 978-0-470-54281-1.

Motulsky, H. and Christopoulos, A. (2004) *Fitting Models to Biological Data Using Linear and Nonlinear Regression: A Practical Guide to Curve Fitting*, 1st ed., Oxford University Press.

Samaria, F.S. and Harter, A.C. (1994) 'Parameterisation of a stochastic model for human face identification', *Proceedings of the Second IEEE Workshop on in Applications of Computer Vision*, December, pp.138–142.

Seber, G.A. and Lee, A.J. (2012) *Linear Regression Analysis*, 2nd ed., John Wiley & Sons, Hoboken, New Jersey, USA, ISBN: 978-0-471-41540-4.

Sharma, A. and Paliwal, K.K. (2014) 'Linear discriminant analysis for the small sample size problem: an overview', *International Journal of Machine Learning and Cybernetics*, Vol. 6, No. 3, pp.443–454.

Specht, D.F. (1990) 'Probabilistic neural networks', *Neural Networks*, Vol. 3, No. 1, pp.109–118.

Strang, G. and Aarikka, K. (1986) *Introduction to Applied Mathematics*, 4th ed., Vol. 16, Wellesley-Cambridge Press, Massachusetts.

Tharwat, A. (2016) 'Principal component analysis – a tutorial', *International Journal of Applied Pattern Recognition*, Inderscience, in press.

Tharwat, A., Ibrahim, A. and Ali, H.A. (2012) 'Personal identification using ear images based on fast and accurate principal component analysis', *8th International Conference on Informatics and Systems (INFOS)*, Vol. 56, p.59.

Tharwat, A., Ibrahim, A., Hassanien, A.E. and Schaefer, G. (2015) 'Ear recognition using block-based principal component analysis and decision fusion', *Proceedings of Pattern Recognition and Machine Intelligence*, pp.246–254.

Vapnik, V. (2013) *The Nature of Statistical Learning Theory*, 2nd ed., Springer Science & Business Media, New York, USA, ISBN: 978-0-387-98780-4.

Whittaker, J. (2009) *Graphical Models in Applied Multivariate Statistics*, 1st ed., Wiley Publishing, New York, USA, ISBN: 978-0-471-91750-2.

Yang, J., Zhang, D., Frangi, A.F. and Yang, J.Y. (2004) 'Two-dimensional PCA: a new approach to appearance-based face representation and recognition', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No. 1, pp.131–137.

Ye, J. and Xiong, T. (2006) 'Computational and theoretical analysis of null space and orthogonal linear discriminant analysis', *The Journal of Machine Learning Research*, July, Vol. 7, pp.1183–1204.

Yu, H. and Yang, J. (2001) 'A direct LDA algorithm for high-dimensional data – with application to face recognition', *Pattern Recognition*, Vol. 34, No. 10, pp.2067–2070.

## Notes

[1] These decision boundaries are the points, lines or curves, planes or surfaces, hyperplanes or hypersurfaces in the case of one, two, three, and higher dimensional feature spaces, respectively.

[2] $P(x \mid \omega_i)$ means the likelihood or probability of the pattern or sample $x$ belongs to the class $\omega_i$.

[3] $P(\omega_i \mid x)$ indicates the probability of finding the unknown pattern $x$ in the class $\omega_i$.

[4] If $m = 1$ then the density is univariate; otherwise the density is *multivariate*.

[5] $X$ is positive semi-definite if $v^TXv \geq 0$ for all $v \neq 0$, in other words all eigenvalues of $X$ are $\geq 0$.

[6] The distribution of all classes are represented by circles because the samples are represented by only two features, i.e., $m = 2$. If $m > 2$, the distribution of all classes are spherical.

[7] A matrix is singular if it is square, does not have a matrix inverse, its determinant is zero, thus not all columns and rows are independent.

[8] The rank of the matrix represents the number of linearly independent rows or columns.

[9] $A$ is a full-rank matrix if all columns and rows of the matrix are independent, i.e., rank ($A$)= #rows = #cols, thus, $\Sigma_i$ can be inverted (Strang and Aarikka, 1986; Golub and Van Loan, 2012).

[10] http://www.cam-orl.co.uk.

[11] http://faculty.ucmerced.edu/mcarreira-perpinan/software.html.

[12] http://vision.ucsd.edu/content/yale-face-database.

## Appendix

In this section, a MATLAB codes for the numerical examples and the experiment are introduced.

### *MATLAB codes for our numerical examples*

In this section, the MATLAB codes the numerical examples in Sections 3 and 4 are introduced.

### *Example 1*

This code follows the steps of the numerical example in Section 3.1.

(see online version for colours)

```
% This is a MATLAB code for the first numerical example ...
(Section 3.1)
clc
clear all;
%% The data of the three classes
C1 = [3 4; 3 5; 4 4; 4 5]
C2 =[3 2;3 3;4 2;4 3]
C3 =[6 2;6 3;7 2;7 3]
%% Calculate the number of samples in each class (ni)
n1 = size (C1 ,1);
n2 = size (C2 ,1);
n3 = size (C3 ,1);
%% Calculate the total number of samples (N = n1 + n2 + n3)
N = n1 + n2 + n3
%% Calculate the mean of each class (mui)
```

```
mu1 = mean (C1)
mu2 = mean (C2)
mu3 = mean (C3)
%% Subtract the mean of each class from the class ...
samples (mean - centring data ) (Di=Ci - mui )
D1 = C1 - repmat (mu1, n1, 1)
D2 = C2 - repmat (mu2, n2, 1)
D3 = C3 - repmat (mu3, n3, 1)
%% Calculate the covariance matrix of each class (Covi)
Cov1 = D1 ‘* D1
Cov2 = D2 ‘* D2
Cov3 = D3 ‘* D3
%% Calculate the inverse of the covariance matrices
invCov1 = inv(Cov1);
invCov2 = inv(Cov2);
invCov3 = inv(Cov3);
%% Calculate the prior probability (Pi)
P1 = n1 / N
P2 = n2 / N
P3 = n3 / N
%% Calculate the discriminant functions of all classes
%% In this case ( Case 1): Wi0 represents the bias term, ...
Wi is the coefficients of the linear term , and WWi ... is
the qudaratic term
%% Note: the inverse of all covariance matrices are ...
equal, hence the values of the quadratic ... coefficients
are the same
W10 = log(P1) -0.5 * mu1 *((invCov1) * mu1’)
W1 = (invCov1) * mu1’
WW1 = -0.5 * inv(Cov1)

W20 = log(P2) -0.5 * mu2 *((invCov2) * mu2’)
W2 = (invCov2) * mu2’
WW2 = -0.5 * inv(Cov2)

W30 = log(P3) -0.5 * mu3 *((invCov3) * mu3’)
W3 = (invCov3) * mu3’
WW3 = -0.5 * inv(Cov3)
```

*Example 2*

This code follows the steps of the numerical example in Section 3.2.

(see online version for colours)

```
% This is a MATLAB code for the second numerical example ...
(Section 3.2)

clc
clear all

%% The data of the three classes
C1 = [3 5;3 7;4 5;4 7]
C2 = [2 2;2 4;3 2;3 4]
C3 = [6 1;6 3;7 1;7 3]

%% Calculate the number of samples in each class (ni)
n1 = size (C1, 1);
n2 = size (C2, 1);
n3 = size (C3, 1);

%% Calculate the total number of samples (N = n1 + n2 + n3)
N = n1 + n2 + n3

%% Calculate the mean of each class (mui)
mu1 = mean (C1)
mu2 = mean (C2)
mu3 = mean (C3)

%% Subtract the mean of each class from the class ...
samples (mean – centring data) (Di = Ci – mui)
D1 = C1 – repmat (mu1, n1, 1)
D2 = C2 – repmat (mu2, n2, 1)
D3 = C3 – repmat (mu3, n3, 1)

%% Calculate the covariance matrix of each class (Covi)
Cov1 = D1 '* D1
Cov2 = D2 '* D2
Cov3 = D3 '* D3

%% Calculate the inverse of the covariance matrices
```

```
invCov1 = inv(Cov1)
invCov2 = inv(Cov2)
invCov3 = inv(Cov3)

%% Calculate the prior probability (Pi)
P1 = n1 / N
P2 = n2 / N
P3 = n3 / N

%% Calculate the discriminant functions of all classes
%% In this case ( Case 2): Wi0 represents the bias term ...
and Wi is the coefficients of the linear term
%% Note : the inverse of all covariance matrices are ...
equal , hence the values of the quadratic ... coefficients
are the same

W10 = log(P1) -0.5 * mu1 * (invCov1 * mu1')
W1 = (invCov1) * mu1'

W20 = log(P2) -0.5 * mu2 * ((invCov2) * mu2')
W2 = (invCov2) * mu2'

W30 = log(P3) -0.5 * mu3 * ((invCov3) * mu3')
W3 = (invCov3) * mu3'
```

*Example 3*

This code follows the steps of the numerical example in Section 3.3.

(see online version for colours)

```
% This is a MATLAB code for the second numerical example ...
(Section 3.3)
clc
clear all
%% The data of the three classes
C1 = [7 3;8 3;7 4;8 4]
C2 = [2 2;5 2;2 3;5 3]
C3 =[1 6;5 6;1 7;5 7]
%% Calculate the number of samples in each class (ni)
n1 = size (C1, 1);
n2 = size (C2, 1);
n3 = size (C3, 1);
```

```
%% Calculate the total number of samples (N = n1 + n2 + n3)
N = n1 + n2 + n3
%% Calculate the mean of each class (mui)
mu1 = mean (C1)
mu2 = mean (C2)
mu3 = mean (C3)
%% Subtract the mean of each class from the class ...
samples (mean - centring data ) (Di = Ci - mui)
D1 = C1 - repmat (mu1, n1, 1)
D2 = C2 - repmat (mu2, n2, 1)
D3 = C3 - repmat (mu3, n3, 1)
%% Calculate the covariance matrix of each class (Covi)
Cov1 = D1’ * D1
Cov2 = D2’ * D2
Cov3 = D3’ * D3
%% Calculate the inverse of the covariance matrices
invCov1 = inv(Cov1)
invCov2 = inv(Cov2)
invCov3 =inv(Cov3)
%% Calculate the prior probability (Pi)
P1 = n1 / N
P2 = n2 / N
P3 = n3 / N
%% Calculate the discriminant functions of all classes
%% In this case (Case 3): Wi0 represents the bias term, ...
Wi is the coefficients of the linear term, and WWi ... is
the qudaratic term
W10 = log(P1) -0.5 * mu1 * ((inv(Cov1)) * mu1’) -0.5 *
log(det(Cov1))
W1 = (inv(Cov1)) * mu1’
WW1 = -0.5 * inv(Cov1)

W20 = log(P2) -0.5 * mu2 *((inv(Cov2)) * mu2’) -0.5 *
log(det(Cov2))
W2 = (inv(Cov2)) * mu2’
WW2 = -0.5 * inv(Cov2)

W30 = log(P3) -0.5 * mu3 * ((inv(Cov3)) * mu3’) -0.5 *
log(det(Cov3))
W3 = (inv(Cov3)) * mu3’
WW3 = -0.5 * inv(Cov3)

48
```

```
WW21(1, 1) - WW31(1, 1)
WW21(2,2) - WW31(2, 2)

W21(1, 1) - W31(1, 1)
W21 (2, 1) - W31(2, 1)
W20 - W30
```

*RLDA numerical example*

This code follows the steps of the numerical example in Section 4.2.

(see online version for colours)

```
%% This example to explain the regularised - LDA (RLDA)
%% The data of three classes
C1 = [3 4 3 5; 3 5 6 4; 4 4 5 7]
C2 = [3 2 5 2; 3 3 5 3; 4 2 3 5]
C3 = [6 2 5 6; 6 3 6 7; 7 2 5 7]
%% Calculate the number of samples in each class (ni)
n1 = size (C1, 1);
n2 = size (C2, 1);
n3 = size (C3, 1);
n = n1 + n2 + n3
%% Calculate the mean of each class (mui)
mu1 = mean (C1)
mu2 = mean (C2)
mu3 = mean (C3)
%% Subtract the mean of each class from the class ...
samples (mean - centring data ) (Di = Ci - mui)
D1 = C1 - repmat (mu1, n1, 1)
D2 = C2 - repmat (mu2, n2, 1)
D3 = C3 - repmat (mu3, n3, 1)
%% Calculate the covariance matrix of each class (Covi)
Cov1 = D1’ * D1
Cov2 = D2’ * D2
Cov3 = D3’ * D3
%% Calculate the new covariance matrices of each class ...
 (Covi = Covi +eta * Covi)
eta = 0.05;
Rcov1 = Cov1 + eta * eye(size (Cov1, 1), size (Cov1, 2))
Rcov2 = Cov2 + eta * eye(size (Cov2, 1), size (Cov2, 2))
Rcov3 = Cov3 + eta * eye(size (Cov3, 1), size (Cov3, 2))
%% Calculate the inverse of the covariance matrices
```

```
invCov1 = inv(Rcov1);
invCov2 = inv(Rcov2);
invCov3 = inv(Rcov3);
```

*Subspace numerical example*

This code follows the steps of the numerical example in Section 4.3.

(see online version for colours)

```
%% This example to explain the regularised - LDA (RLDA)
%% The data of three classes
C1 = [3 4 3 5; 3 5 6 4; 4 4 5 7]
C2 = [3 2 5 2; 3 3 5 3; 4 2 3 5]
C3 =[6 2 5 6; 6 3 6 7; 7 2 5 7]
%% Calculate the number of samples in each class (ni)
n1 = size (C1, 1);
n2 = size (C2, 1);
n3 = size (C3, 1);
%% Reduce the dimension of all samples in all classes to ...
be only two
%% features
[vec1, eval1] = pca_new_final (C1’)
NewC_1 = C1 * vec1(:, 1:2)

[vec2, eval2 ] = pca_new_final (C2’)
NewC_2 = C2 * vec2(:, 1:2)

[vec3, eval3] = pca_new_final (C3’)
NewC_3 = C3 * vec3(:, 1:2)
%% Calculate the mean of each class (mui)
mu1 = mean (NewC_1)
mu2 = mean (NewC_2)
mu3 = mean (NewC_3)
%% Subtract the mean of each class from the class ...
samples (mean - centring data) (Di = Ci - mui)
D1 = NewC_1 - repmat (mu1, n1, 1)
D2 = NewC_2 - repmat (mu2, n2, 1)
D3 = NewC_3 - repmat (mu3, n3, 1)
%% Calculate the covariance matrix of each class (Covi)
Cov1 = D1’ * D1
Cov2 = D2’ * D2
```

```
Cov3 = D3' * D3
%% Calculate the inverse of the covariance matrices
invCov1 = inv(Cov1);
invCov2 = inv(Cov2);
invCov3 = inv(Cov3);
```

### *MATLAB code for our experiment*

In this section, the codes for our experiment in Section 5 are introduced.

(see online version for colours)

```
%% Load the labelled dataset (iris dataset)
%% data (N x m) where N is the number of samples and m is
... the dimension of each sample
%% Y is the labels of the samples
load iris
%% Sort the samples randomly
y = randperm (size (X, 1));
data = data (y, :);
Labels = Y(y);
%% Per is the percentage of the training samples and ...
1- Per is the percentage of the testing samples
Per = 0.9;
Tr = data (1: ceil (Per * size (data, 1)), :);
Test = data (1 + ceil (Per * size (data, 1)): end, :);
TrL = Labels (1: ceil (Per * size (data, 1)), :);
TestL = Labels (1 + ceil (Per * size (data, 1)): end, :);
%% Train the model using linear and quadratic ...
discriminant classifiers
%% Linera Discriminant Analysis (ldaclass is the ...
prediction of the testing samples)
 [ldaclass, err, p,logp, coeff] = classify (Test, Tr, TrL,
'linear');
disp (['Linear accuracy is' ... int2str (100 * sum(ldaclass
== TestL) / size (TestL, 1))])
%% Quadratic Discriminant Analysis (ldaclass is the ...
prediction of the testing samples)
[ldaclassQ, err, p,logp, coeff] = classify (Test, Tr, TrL,
... .. 'Quadratic');

disp (['Quadratic accuracy is' ... int2str (100* sum(
ldaclassQ == TestL )/ size (TestL ,1) )])
```