Privacy-Preserving Classification Methods

Submitted by

Dan Wu, B.Sc.

A thesis submitted to
The Faculty of Graduate Studies and Research
In partial fulfillment of
The requirements for the degree of

Master of Science

School of Mathematics and Statistics

Ottawa-Carleton Institute of Mathematics and Statistics
Carleton University
Ottawa, Ontario, Canada
March 2008

Your file Votre référence
ISBN: 978-0-494-40664-9
Our file Notre référence
ISBN: 978-0-494-40664-9

Abstract

The development of privacy-preserving statistical methods has become increasingly necessary to protect sensitive personal information . In this thesis we combine Fisher's classification methods [1] with a perturbation method to protect privacy presented by Du et al [2] . The result is a methodology for classification of an observation into one of two populations. The methodology is incorporated into R code and applied for illustrative purposes to a medical dataset. We also evaluate the misclassification rate associated with this technique.

Table of Contents

1. Introduction

We consider a situation where two parties hold separate parts of the data we wish to use in our analysis and they are not permitted to share their data directly. The goal is to perform classification of anobservation into one of two populations using Fisher's discrimination analysis[1]. In this thesis, we incorporate protocols presented in [2] and illustrate how these may be used to perform discriminant analysis on vertically or horizontally partitioned datasets. For illustrataive purposes we consider a situation in medical research. Subjects taking part in a study are guaranteed privacy of their information. To this end, the research department of a hospital has just part of the information on the patients, while the hospital administration has the other part. To conduct a detailed statistical analysis involving building a classifier, the researchers need information stored in the data owned by the hospital administration. But to protect patients' privacy, the hospital administration can only release perturbated data to the research group. The above is an example of a vertically partitioned dataset case. An example of a horizontally partitioned dataset is the following. Several hospitals are involved in a joint research project where they have the same variables measured but on different patients and they need information stored by all the hoospitals taking part in the project to improve the accuracy of the classification result. This thesis deals with both situations, developing the mathematical results associated with the methodology proposed, developing R code to implement the methodology, applying the methodology for illustrative purposes, and evaluating the misclassification rate of the proposed procedure.

2. Fisher's Linear Discriminant Function and Quadratic Discriminant Analysis(QDA)

2.1 Fisher's Linear Discriminant Function

We first consider using Fisher's linear discriminant function [1] as a classifier. A statistic, restricted to be a linear combination of the data, is computed from the original data and this statistic is used to do the classification or discrimination In general we assume that there are g multivariate normal populations represented as $\pi_1, \pi_2, ..., \pi_g$, although the assumption of multivariate normality is not necessary However we do require that all the $p \times p$ population covariance matrices are equal and of full rank (ie: $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \cdots = \boldsymbol{\Sigma}_g = \boldsymbol{\Sigma}$). We let $\boldsymbol{\mu}_i$ represent the mean vector for the i^{th} population. We take a training set of random size n_i from $\pi_i, i = 1,2,...,g$. Let $\bar{\mathbf{x}}_i$ denote the sample mean vector and S_i the sample covariance matrix of the sample from π_i, $\bar{\mathbf{x}}$ the overall average vector for the training sets, and B_{SS} the sample between-groups sums of cross products. Then

$$B = \sum_{i=1}^{g} n_i(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})'$$

where

$$\bar{\mathbf{x}} = \frac{\sum_{i=1}^{g} n_i \bar{\mathbf{x}}_i}{\sum_{i=1}^{g} n_i} = \frac{\sum_{i=1}^{g}\sum_{j=1}^{n_i} \mathbf{x}_{ij}}{\sum_{i=1}^{g} n_i} \qquad [2.1]$$

An estimate of $\boldsymbol{\Sigma}$ is given by $\mathbf{S}_{pooled} = \dfrac{\mathbf{W}}{\sum_{i=1}^{g}(n_i - 1)}$, where $\mathbf{W}$ is the sample within-groups matrix:

$$\mathbf{W} = \sum_{i=1}^{g}(n_i - 1)S_i$$
$$= \sum_{i=1}^{g}\sum_{j=1}^{n_i}(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)' \qquad [2.2]$$

Let $\lambda_1, \lambda_2, ..., \lambda_s$ be the $s \leq \min(g-1, p)$ nonzero eigenvalues of $\mathbf{W}^{-1}\mathbf{B}_{SS}$ and $\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_s$ be the corresponding eigenvectors.scaled so that $\mathbf{e}'\mathbf{S}_{pooled}\mathbf{e} = 1$ Then the vector of coefficients $\mathbf{a}$ that maximizes the ratio of the between-group sum of squares to the within-group sum of squares

$$\frac{\mathbf{a}'\mathbf{B}_{SS}\mathbf{a}}{\mathbf{a}'\mathbf{W}\mathbf{a}}$$

for the transformed data $\mathbf{Y} = \mathbf{a}'\mathbf{X}$ is given by $\hat{\mathbf{a}} = \mathbf{e}$. We note that $\text{Cov}(y_i, y_j) = 0$ for all $i \neq j$.

The linear combination $y_k = \mathbf{e}_k'\mathbf{x}$ is called the *k^{th} linear sample discriminant*, $k \leq s$. Thus we obtain:

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_s \end{pmatrix}$$

with mean vector

$$\boldsymbol{\mu}_{iY} = \begin{pmatrix} \mu_{iy_1} \\ \mu_{iy_2} \\ \vdots \\ \mu_{iy_s} \end{pmatrix} = \begin{pmatrix} \mathbf{e}_1'\boldsymbol{\mu}_i \\ \mathbf{e}_2'\boldsymbol{\mu}_i \\ \vdots \\ \mathbf{e}_s'\boldsymbol{\mu}_i \end{pmatrix}$$

If only $r(< s)$ of the discriminants are used for the classification, we use the rule:

Allocate $\mathbf{x}$ to the π_k if

$$\sum_{j=1}^{r}(y_j - \mu_{ky_1})^2 = \sum_{j=1}^{r}[e_j'(\mathbf{x} - \boldsymbol{\mu}_k)]^2 \leq \sum_{j=1}^{r}[e_j'(\mathbf{x} - \boldsymbol{\mu}_i)]^2 \qquad \text{for all } i \neq k \qquad [2.3]$$

or more generally,

$$\sum_{j=1}^{r}(y_j - \bar{y}_{ky_1})^2 = \sum_{j=1}^{r}[e_j'(\mathbf{x} - \bar{\mathbf{x}}_k)]^2 \leq \sum_{j=1}^{r}[e_j'(\mathbf{x} - \bar{\mathbf{x}}_i)]^2 \qquad \text{for all } i \neq k \qquad [2.4]$$

where $\bar{y}_{ky_1} = \hat{\mathbf{a}}'\bar{\mathbf{x}}_k$ and $\hat{\mathbf{a}} = \mathbf{e}$.

2.2 Quadratic Discriminant Analysis(QDA)

If the covariance matrices of the different populations are unequal, classification becomes more complicated, resulting in a quadratic discriminant function given by QDA[1]. We assume each of the g populations follows a multivariate normal distribution with mean vector $\boldsymbol{\mu}_i(i = 1, 2, \cdots g)$ and covariance matrix $\boldsymbol{\Sigma}_i(i = 1, 2, \cdots g)$. If we let p_i represent the prior probability of population π_i and

adopt the criteria of minimum expected cost of misclassification method, then we obtain the following rule

Allocate $\mathbf{x}$ to π_k, if

$$d_k^Q = \max\{d_i^Q, i = 1,2,\cdots g\}$$

where

$$d_i^Q(\mathbf{x}) = -\frac{1}{2}\ln|\mathbf{\Sigma}_i| - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)'\mathbf{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i) + \ln p_i \qquad i = 1,2,\cdots g$$

represents the *quadratic discrimination score* for the i^{th} population.

Alternatively, when $\boldsymbol{\mu}_i$ and Σ_i are unknown we use the rule:

Allocate $\mathbf{x}$ to π_k, if

$$\hat{d}_k^Q = \max\{\hat{d}_i^Q, i = 1,2,\cdots g\} \qquad [2.4]$$

where

$$\hat{d}_i^Q(\mathbf{x}) = -\frac{1}{2}\ln|S_i| - \frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}}_i)'\mathbf{\Sigma}_i^{-1}(\mathbf{x}-\bar{\mathbf{x}}_i) + \ln p_i \qquad i = 1,2,\cdots g.$$

3. Privacy-Preserving Protocols

Suppose there are two parties H and R where H has the subset $\mathbf{A}_{n\times N}$, and R has the subset $\mathbf{B}_{N\times m}$. To protect privacy we present protocols from [2].

3.1 Matrix Product (AB) Protocol

First we consider some matrix product protocols [2] which let H and R get **AB** but do not allow one party to derive the other party's significant information, when $N \gg n$ and $N \gg m$.

3.1.1 Protocol 1 – Commodity Server

The Commodity Server generates a random $n \times N$ matrix $\mathbf{R}_a$, and another random $N \times m$ matrix $\mathbf{R}_b$ and lets $\mathbf{r}_a + \mathbf{r}_b = \mathbf{R}_a \bullet \mathbf{R}_b$, where $\mathbf{r}_a$ (or $\mathbf{r}_b$) is a randomly generated $n \times m$ matrix. Then the server sends $(\mathbf{R}_a, \mathbf{r}_a)$ to H and $(\mathbf{R}_b, \mathbf{r}_b)$ to R.

1. H sends $\widehat{\mathbf{A}} = \mathbf{A} + \mathbf{R}_a$ to R, and R sends $\widehat{\mathbf{B}} = \mathbf{B} + \mathbf{R}_b$ to H.
2. R generates a random $n \times m$ matrix $\mathbf{V}_b$, then computes $\mathbf{T} = \widehat{\mathbf{A}}\mathbf{B} + (\mathbf{r}_b - \mathbf{V}_b)$, and sends the result **T** to H.
3. H computes $\mathbf{V}_a = \mathbf{T} + \mathbf{r}_a - \left(\mathbf{R}_a\widehat{\mathbf{B}}\right)$

It is easy to verify that

$$\begin{aligned}\mathbf{V}_a + \mathbf{V}_b &= \left[\left(\widehat{\mathbf{A}}\mathbf{B} + (\mathbf{r}_b - \mathbf{V}_b)\right) + \mathbf{r}_a - \left(\mathbf{R}_a\widehat{\mathbf{B}}\right)\right] + \mathbf{V}_b \\ &= \mathbf{AB} + \mathbf{r}_a + \mathbf{r}_b - \mathbf{R}_a\mathbf{R}_b \\ &= \mathbf{AB}\end{aligned}$$

In [2], the security of the protocol is also proved.

3.1.2 Protocol 1 – Two Party

1. H and R jointly generate a random invertible $N \times N$ matrix

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_{left\, N \times \frac{N}{2}} & \mathbf{M}_{right\, N \times \frac{N}{2}} \end{pmatrix}$$

$$\mathbf{M}^{-1} = \begin{pmatrix} \mathbf{M}_{inv-top\, \frac{N}{2} \times N} \\ \mathbf{M}_{inv-bottom\, \frac{N}{2} \times N} \end{pmatrix}.$$

2. H computes $\mathbf{A}_1 = \mathbf{A}'\mathbf{M}_{left}$, and $\mathbf{A}_2 = \mathbf{A}'\mathbf{M}_{right}$, and sends $\mathbf{A}_1$ to R.

3. R computes $\mathbf{B}_1 = \mathbf{M}_{inv-top}\mathbf{B}$ and $\mathbf{B}_2 = \mathbf{M}_{inv-bottom}\mathbf{B}$, and sends $\mathbf{B}_2$ to H.

4. H computes $\mathbf{V}_a = \mathbf{A}_2\mathbf{B}_2$.

5. R computes $\mathbf{V}_b = \mathbf{A}_1\mathbf{B}_1$

It is easy to see that the above protocol achieves the following:

$$\begin{aligned} \mathbf{A}'\mathbf{B} &= \mathbf{A}\mathbf{M}\mathbf{M}^{-1}\mathbf{B} \\ &= \begin{pmatrix} \mathbf{A}_1 & \mathbf{A}_2 \end{pmatrix} \begin{pmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{pmatrix} \\ &= \mathbf{V}_a + \mathbf{V}_b \end{aligned}$$

3.2 Matrix Inverse Protocol

Suppose H has the subset **A** and R has the subset **B**. where both **A** and **B** are $n \times n$ matrices and $\mathbf{A} + \mathbf{B}$ is invertible. To get $(\mathbf{A} + \mathbf{B})^{-1}$ without disclosing their own original data to the other party, the solution is:

Firstly, H and R jointly convert matrix $(\mathbf{A} + \mathbf{B})$ to $\mathbf{P}(\mathbf{A} + \mathbf{B})\mathbf{Q}$ using two random matrices **P** and **Q** that are only known to R. The results of

$\mathbf{P}(\mathbf{A}+\mathbf{B})\mathbf{Q}$ will only be known by H who can conduct the inverse computation and get $\mathbf{Q}^{-1}(\mathbf{A}+\mathbf{B})^{-1}P^{-1}$. In the second step, H and R jointly remove $\mathbf{Q}^{-1}$ and $\mathbf{P}^{-1}$ and gets $\mathbf{V}_a+\mathbf{V}_b=(\mathbf{A}+\mathbf{B})^{-1}$.

Both steps can be achieved using the (**AB**) protocol.

4. Privacy-Preserving Medical Problem

We consider the case where two parties, hospital (H) and research group (R), have different exclusive data. Suppose H has data matrix **A** and R has data matrix **B** and the response or final classification is public data known to both H and R. According to the response, the data can be divided into g populations. In this thesis, we consider only the case of $g = 2$. (For example, one group might be those who responded favourably to a medical procedure, while the other group might be those who responded negatively to the same medical procedure.)

4.1 Vertically Partitioned Data

In the case of vertically partitioned data, H and R have different exclusive variables. That is **A** contains m variables and **B** has n variables and they do not overlap. We consider the situation where the variables of **A are independent of those of B**. We now consider a sample of size n_1 from population π_1 This is represented by

$$\mathbf{X}_1=\begin{pmatrix} \mathbf{A}_1 & \mathbf{B}_1 \end{pmatrix}_{n_1 x(m+n)}$$

where H holds $\mathbf{A}_1$ and R holds $\mathbf{B}_1$. Similarly we have a sample of of size n_2 from population π_2, represented by:

$$\mathbf{X}_2=\begin{pmatrix} \mathbf{A}_2 & \mathbf{B}_2 \end{pmatrix}_{n_2 x(m+n)}$$

where H holds $\mathbf{A}_2$ and R holds $\mathbf{B}_2$. Thus H has $\mathbf{A}_{(n_1+n_2)xm}$, partitioned into $\mathbf{A}_{1n_1\times m}$ and $\mathbf{A}_{2n_2\times m}$; R has $\mathbf{B}_{(n_1+n_2)xn}$, partitioned as $\mathbf{B}_{1n_1\times n}$ and $\mathbf{B}_{2n_2\times n}$.

The sample mean vector for the sample data from population i is:

$$\bar{\mathbf{x}}_i = \begin{pmatrix} \bar{\mathbf{x}}_{A_i} & \bar{\mathbf{x}}_{B_i} \end{pmatrix} \text{ where } i = 1,2$$

where $\bar{\mathbf{x}}_{A_i}$ ($\bar{\mathbf{x}}_{|B_i}$) is the sample mean vector of $\mathbf{A}_i$ ($\mathbf{B}_i$). Using the assumption given above that the variables of **A** are independent of those of **B**, we know that $\bar{\mathbf{x}}_{A_i}$ and $\bar{\mathbf{x}}_{|B_i}$, $i = 1,2$, are independent.

The sample mean vector of the sample data from the combined two populations is:

$$\bar{\mathbf{x}} = \begin{pmatrix} \bar{\mathbf{x}}_A & \bar{\mathbf{x}}_B \end{pmatrix}$$

where

$$\bar{\mathbf{x}}_A = \frac{n_1\bar{\mathbf{x}}_{A_1} + n_2\bar{\mathbf{x}}_{A_2}}{n_1 + n_2}$$

and

$$\bar{\mathbf{x}}_B = \frac{n_1\bar{\mathbf{x}}_{B_1} + n_2\bar{\mathbf{x}}_{B_2}}{n_1 + n_2}$$

and $\bar{x}_A$ ($\bar{x}_B$) is the sample mean vector of A (B).

In summary we have

H(hospital administration computes and keeps):	R(Research group computes and keeps):
$\bar{\mathbf{x}}_{A_i}(i = 1,2)$	$\bar{\mathbf{x}}_{B_i}(i = 1,2)$
$\bar{\mathbf{x}}_A c$	$\bar{\mathbf{x}}_B$

We now consider special cases concerning the population covariance matrices.

4.1.1 The $(m+n)\times(m+n)$ population covariance matrices of different populations are equal

Here we assume both population covariance matrices ($\boldsymbol{\Sigma}_i, i = 1,2$) are equal. Thus we have:

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} \boldsymbol{\Sigma}_{A_1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{B_1} \end{pmatrix} = \boldsymbol{\Sigma}_2 = \begin{pmatrix} \boldsymbol{\Sigma}_{A_2} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{B_2} \end{pmatrix} = \boldsymbol{\Sigma}$$

where $\boldsymbol{\Sigma}$ is the common covariance matrix.

The estimate of the covariance matrices of $\mathbf{X}_i$ is the sample covariance matrix $\mathbf{S}_i$ of $\mathbf{X}_i, i = 1,2$ and is given by

$$\mathbf{S}_i = \begin{pmatrix} \mathbf{S}_{A_i} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{B_i} \end{pmatrix} \quad , i = 1,2$$

and $S_{A_i}(S_{B_i})$ is the sample covariance matrix of $A_i(B_i)$.Then the estimate of the common population covariance matrix Σ is given by

$$\mathbf{W} = (n_1 - 1)\mathbf{S}_1 + (n_2 - 1)\mathbf{S}_2.$$

Thus

$$\mathbf{W} = \begin{pmatrix} (n_1 - 1)\mathbf{S}_{A_1} + (n_2 - 1)\mathbf{S}_{A_2} & \mathbf{0} \\ \mathbf{0} & (n_1 - 1)\mathbf{S}_{B_1} + (n_2 - 1)\mathbf{S}_{B_2} \end{pmatrix}$$
$$= \begin{pmatrix} \mathbf{W}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_B \end{pmatrix}$$

where

$$\mathbf{W}_A = (n_1 - 1)\mathbf{S}_{A_1} + (n_2 - 1)\mathbf{S}_{A_2}$$

and

$$\mathbf{W}_B = (n_1 - 1)\mathbf{S}_{B_1} + (n_2 - 1)\mathbf{S}_{B_2}.$$

At this point we have

H(hospital administration computes and keeps):	R(Research group computes and keeps):
$\mathbf{S}_{A_i}(i = 1,2)$	$\mathbf{S}_{B_i}(i = 1,2)$
$\mathbf{W}_A$	$\mathbf{W}_B$

4.1.1.1 The $(m + n) \times (m + n)$ covariance matrices are of full rank as well.

Since the covariance matrices are of full rank we know

$$\mathbf{W}^{-1} = \begin{pmatrix} \mathbf{W}_A^{-1} & 0 \\ 0 & \mathbf{W}_B^{-1} \end{pmatrix}$$

and the between sum of square is

$$\mathbf{B}_{SS} = n_1 \begin{pmatrix} \bar{\mathbf{x}}_{A_1} - \bar{x}_A \\ \bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_B \end{pmatrix} \begin{pmatrix} \bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_A \\ \bar{\mathbf{x}} - \bar{\mathbf{x}}_B \end{pmatrix}' + n_2 \begin{pmatrix} \bar{\mathbf{x}}_{A_2} - \bar{\mathbf{x}}_A \\ \bar{\mathbf{x}}_{B_2} - \bar{\mathbf{x}}_B \end{pmatrix} \begin{pmatrix} x_{A_2} - \bar{\mathbf{x}}_A \\ x_{B_2} - \bar{\mathbf{x}}_B \end{pmatrix}'$$

where $\bar{\mathbf{x}}_{A_i}(\bar{\mathbf{x}}_{B_i})$ is the sample mean vector of $\mathbf{A}_i$ $(\mathbf{B}_i)$, $i = 1,2$, $\bar{\mathbf{x}}_A$ is the sample mean vector of $\mathbf{A}$

$$\bar{\mathbf{x}}_A = \frac{n_1\bar{\mathbf{x}}_{A_1} + n_2\bar{\mathbf{x}}_{A_2}}{n_1 + n_2}$$

and

$$\bar{\mathbf{x}}_B = \frac{n_1\bar{\mathbf{x}}_{B_1} + n_2\bar{\mathbf{x}}_{B_2}}{n_1 + n_2}$$

is the sample mean vector of $\mathbf{B}$ Now the difference of $\bar{\mathbf{x}}_{A_i}$ and $\bar{\mathbf{x}}_A$ is:

$$\bar{\mathbf{x}}_{A_i} - \bar{\mathbf{x}}_A = \frac{n_j}{n_i + n_j}(\bar{\mathbf{x}}_{A_i} - \bar{\mathbf{x}}_{A_j}), i \neq j, i,j = 1,2$$

and the difference of $\bar{\mathbf{x}}_{B_i}$ and $\bar{\mathbf{x}}_B$ is:

$$\bar{\mathbf{x}}_{B_i} - \bar{\mathbf{x}}_B = \frac{n_j}{n_i + n_j}(\bar{\mathbf{x}}_{B_i} - \bar{\mathbf{x}}_{B_j}), i \neq j, i,j = 1,2$$

We now have:

H(hospital administration computes and keeps):	R(Research group computes and keeps):
$\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_A$	$\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_B$
$\bar{\mathbf{x}}_{A_2} - \bar{\mathbf{x}}_A$	$\bar{\mathbf{x}}_{B_2} - \bar{\mathbf{x}}_B$

Therefore the between sum of squares and cross products can be obtained from the following calculation:

$$\begin{aligned}
\mathbf{B}_{SS} &= n_1 \bullet \frac{n_2^2}{(n_1+n_2)^2}\begin{pmatrix} \bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2} \\ \bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2} \end{pmatrix}\begin{pmatrix} \bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2} \\ \bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2} \end{pmatrix}' \\
&\quad + n_2 \bullet \frac{n_1^2}{(n_1+n_2)^2}\begin{pmatrix} \bar{\mathbf{x}}_{A_2} - \bar{\mathbf{x}}_{A_1} \\ \bar{\mathbf{x}}_{B_2} - \bar{\mathbf{x}}_{B_1} \end{pmatrix}\begin{pmatrix} \bar{\mathbf{x}}_{A_2} - \bar{\mathbf{x}}_{A_1} \\ \bar{\mathbf{x}}_{B_2} - \bar{\mathbf{x}}_{B_1} \end{pmatrix}' \\
&= \frac{n_1 n_2}{n_1+n_2}\begin{pmatrix} \bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2} \\ \bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2} \end{pmatrix}\begin{pmatrix} \bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2} \\ \bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2} \end{pmatrix}' \\
&= \frac{n_1 n_2}{n_1+n_2}\begin{pmatrix} (\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})(\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})' & (\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})(\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})' \\ (\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})(\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})' & (\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})(\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})' \end{pmatrix}
\end{aligned}$$

Thus

$$\begin{aligned}
|\mathbf{W}^{-1}\mathbf{B}_{SS} - \lambda\mathbf{I}| &= |\mathbf{W}^{-1}||\mathbf{B}_{SS} - \lambda\mathbf{W}| \\
&= |\mathbf{W}^{-1}|\left|\begin{pmatrix} \frac{n_1 n_2}{n_1+n_2}(\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})(\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})' - \lambda W_A & \frac{n_1 n_2}{n_1+n_2}(\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})(\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})' \\ \frac{n_1 n_2}{n_1+n_2}(\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})(\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})' & \frac{n_1 n_2}{n_1+n_2}(\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})(\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})' - \lambda W_B \end{pmatrix}\right| \\
&= |\mathbf{W}^{-1}|\left|\begin{pmatrix} \mathbf{Q}_A & \mathbf{Q}_{AB} \\ \mathbf{Q}'_{AB} & \mathbf{Q}_B \end{pmatrix}\right| \\
&= 0
\end{aligned}$$

where

$$\mathbf{Q}_A = \frac{n_1 n_2}{n_1+n_2}(\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})(\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})' - \lambda\mathbf{W}_A$$

$$\mathbf{Q}_{AB} = \frac{n_1 n_2}{n_1 + n_2}(\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})(\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})'$$

$$\mathbf{Q}_B = \frac{n_1 n_2}{n_1 + n_2}(\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})(\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})' - \lambda W_B$$

Now $\mathbf{Q}_A$ ($\mathbf{Q}_B$) can be computed by H (R) independently.

But because $(\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})(\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})'$ is the product of $(\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})$ and $(\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})'$, $\mathbf{Q}_{AB}$ can not be obtained by only H or only R. And because of the privacy protection, H or R can not send the original data to the other to allow its computation. So to solve this problem, we use the following procedure:

Since $g = 2, s \le \min(g-1, m+n) = \min(1, m+n) = 1$ therefore $|\mathbf{W}^{-1}\mathbf{B} - \lambda\mathbf{I}| = 0$ must have one and only one solution $\hat{\lambda}$. Therefore at least one of $|\mathbf{Q}_A|$ and $|\mathbf{Q}_B|$ is not equal to 0. Assume here $|\mathbf{Q}_A| \neq 0$ (if $|\mathbf{Q}_B| \neq 0$, the procedures are similar to the following), then

$$\begin{aligned}
|\mathbf{W}^{-1}\mathbf{B}_{SS} - \lambda\mathbf{I}| &= |\mathbf{W}^{-1}| \left| \begin{pmatrix} \mathbf{Q}_A & \mathbf{Q}_{AB} \\ \mathbf{Q}'_{AB} & \mathbf{Q}_B \end{pmatrix} \right| \\
&= |\mathbf{W}^{-1}| \\
&\quad \cdot \left| \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{Q}'_{AB}\mathbf{Q}_A^{-1} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{Q}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_B - \mathbf{Q}'_{AB}\mathbf{Q}_A^{-1}\mathbf{Q}_{AB} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{Q}_A^{-1}\mathbf{Q}_{AB} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \right| \\
&= |\mathbf{W}^{-1}||\mathbf{Q}_A||\mathbf{Q}_B - \mathbf{Q}'_{AB}\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}| \\
&= |\mathbf{W}^{-1}||\mathbf{Q}_A| \\
&\quad \cdot \left| \mathbf{Q}_B - \left(\frac{n_1 n_2}{n_1 + n_2}\right)^2 (\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})(\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})'\mathbf{Q}_A^{-1}(\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})(\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})' \right| \\
&= |\mathbf{W}^{-1}||\mathbf{Q}_A| \left| \mathbf{Q}_B - \left(\frac{n_1 n_2}{n_1 + n_2}\right)^2 (\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2}) Z_A (\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})' \right| \\
&= 0
\end{aligned}$$

where

$$Z_A = (\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})'\mathbf{Q}_A^{-1}(\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})$$

Now Z_A does not disclose any information of the original data of **A** and can be obtained by H alone . $\mathbf{Q}_A$ ($\mathbf{Q}_B$) belongs to H (R). Now since $|\mathbf{W}^{-1}| \neq 0$ and $|\mathbf{Q}_A| \neq 0$ therefore the solution of

$$\left|\mathbf{Q}_B - (\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})Z_A(\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})'\right| = 0$$

is just the one of

$$|\mathbf{W}^{-1}\mathbf{B}_{SS} - \lambda\mathbf{I}| = 0.$$

So now H calculates

$$Z_A = (\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})'\mathbf{Q}_A^{-1}(\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})$$

and then sends it to R. R uses it to solve

$$\left|\mathbf{Q}_B - \left(\frac{n_1 n_2}{n_1 + n_2}\right)^2(\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})Z_A(\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})'\right| = 0$$

and get $\hat{\lambda}$.

Then R solves

$$\left(\mathbf{Q}_B - \left(\frac{n_1 n_2}{n_1 + n_2}\right)^2(\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})Z_A(\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})'\right)\tilde{\mathbf{e}}_B' = 0$$

and gets $\tilde{\mathbf{e}}_B'$.

Now let $\tilde{\mathbf{e}}=\begin{pmatrix} \tilde{\mathbf{e}}_A \\ \tilde{\mathbf{e}}_B \end{pmatrix}$ and we then solve

$$\begin{pmatrix} \mathbf{Q}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_B - \mathbf{Q}_{AB}'\mathbf{Q}_A^{-1}\mathbf{Q}_{AB} \end{pmatrix}\begin{pmatrix} \tilde{\bar{\mathbf{x}}}_A \\ \tilde{\bar{\mathbf{x}}}_B \end{pmatrix} = \mathbf{0}_{(m+n)\times 1}$$

Since $|\mathbf{Q}_A| \neq 0$ then $\tilde{\bar{\mathbf{x}}}_A = \mathbf{0}_{m\times 1}$.

Let $\hat{\mathbf{e}} = \begin{pmatrix} \hat{\mathbf{e}}_A \\ \hat{\mathbf{e}}_B \end{pmatrix}$ be the corresponding eigenvector of $\mathbf{W}^{-1}\mathbf{B}_{SS}$. Then

$$
\begin{aligned}
\left(\mathbf{W}^{-1}\mathbf{B}_{SS} - \hat{\lambda}\mathbf{I}\right)\begin{pmatrix} \hat{\mathbf{e}}_A \\ \hat{\mathbf{e}}_B \end{pmatrix} &= \mathbf{W}^{-1}\left(\mathbf{B}_{SS} - \hat{\lambda}\mathbf{W}\right)\begin{pmatrix} \hat{\mathbf{e}}_A \\ \hat{\mathbf{e}}_B \end{pmatrix} \\
&= \mathbf{W}^{-1}\begin{pmatrix} \mathbf{Q}_A & \mathbf{Q}_{AB} \\ \mathbf{Q}'_{AB} & \mathbf{Q}_B \end{pmatrix}\begin{pmatrix} \hat{\mathbf{e}}_A \\ \hat{\mathbf{e}}_B \end{pmatrix} \\
&= \mathbf{W}^{-1}\begin{pmatrix} \mathbf{I}_m & \mathbf{0} \\ \mathbf{Q}'_{AB}\mathbf{Q}_A^{-1} & \mathbf{I}_n \end{pmatrix} \\
&\quad \cdot \begin{pmatrix} \mathbf{Q}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_B - \mathbf{Q}'_{AB}\mathbf{Q}_A^{-1}\mathbf{Q}_{AB} \end{pmatrix}\begin{pmatrix} \mathbf{I}_m & \mathbf{Q}_A^{-1}\mathbf{Q}_{AB} \\ \mathbf{0} & \mathbf{I}_n \end{pmatrix}\begin{pmatrix} \hat{\mathbf{e}}_A \\ \hat{\mathbf{e}}_B \end{pmatrix} \\
&= \mathbf{0}_{(m+n)\times 1} \\
&= \begin{pmatrix} \mathbf{Q}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_B - \mathbf{Q}'_{AB}\mathbf{Q}_A^{-1}\mathbf{Q}_{AB} \end{pmatrix}\begin{pmatrix} \mathbf{0}_{m\times 1} \\ \tilde{\mathbf{e}}_B \end{pmatrix}
\end{aligned}
$$

If $\left(\mathbf{Q}_B-\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\right)^{-1}$ is called **QsInv**, then

$$\begin{pmatrix} \hat{\mathbf{e}}_A \\ \hat{\mathbf{e}}_B \end{pmatrix} = \begin{pmatrix} \mathbf{I} & -\mathbf{Q}_A^{-1}\mathbf{Q}_{AB} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}\begin{pmatrix} \mathbf{Q}_A^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{QsInv} \end{pmatrix}\begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1} & \mathbf{I} \end{pmatrix}$$
$$\bullet\,\mathbf{W}\begin{pmatrix} \mathbf{Q}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_B-\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1}\mathbf{Q}_{AB} \end{pmatrix}\begin{pmatrix} \mathbf{0}_{m\times 1} \\ \tilde{\mathbf{e}}_B \end{pmatrix}$$
$$= \begin{pmatrix} \mathbf{Q}_A^{-1} & -\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\,\mathbf{QsInv} \\ \mathbf{0} & \mathbf{QsInv} \end{pmatrix}\begin{pmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1} & \mathbf{I} \end{pmatrix}$$
$$\bullet\,\mathbf{W}\begin{pmatrix} \mathbf{Q}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_B-\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1}\mathbf{Q}_{AB} \end{pmatrix}\begin{pmatrix} \mathbf{0}_{m\times 1} \\ \tilde{\mathbf{e}}_B \end{pmatrix}$$
$$= \begin{pmatrix} \mathbf{Q}_A^{-1}+\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\,\mathbf{QsInv}\,\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1} & -\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\,\mathbf{QsInv} \\ -\mathbf{QsInv}\,\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1} & \left(\mathbf{Q}_B-\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\right)^{-1} \end{pmatrix}$$
$$\bullet\begin{pmatrix} \mathbf{W}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_B \end{pmatrix}\begin{pmatrix} \mathbf{Q}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_B-\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1}\mathbf{Q}_{AB} \end{pmatrix}\begin{pmatrix} \mathbf{0}_{m\times 1} \\ \tilde{e}_B \end{pmatrix}$$
$$= \begin{pmatrix} \mathbf{Q}_A^{-1}+\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\,\mathbf{QsInv}\,\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1} & -\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\,\mathbf{QsInv} \\ -\mathbf{QsInv}\,\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1} & \mathbf{QsInv} \end{pmatrix}$$
$$\bullet\begin{pmatrix} \mathbf{W}_A\mathbf{Q}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{W}_B\left(\mathbf{Q}_B-\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\right) \end{pmatrix}\begin{pmatrix} \mathbf{0}_{m\times 1} \\ \tilde{\mathbf{e}}_B \end{pmatrix}$$

so

$$\begin{pmatrix} \hat{\mathbf{e}}_A \\ \hat{\mathbf{e}}_B \end{pmatrix} = \begin{pmatrix} \left[\mathbf{Q}_A^{-1}+\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\,\mathbf{QsInv}\,\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1}\right]\mathbf{W}_A\mathbf{Q}_A & -\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\,\mathbf{QsInv}\,\mathbf{W}_B\left(\mathbf{Q}_B-\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\right) \\ -\mathbf{QsInv}\,\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1}\mathbf{W}_A\mathbf{Q}_A & \mathbf{QsInv}\,\mathbf{W}_B\left(\mathbf{Q}_B-\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\right) \end{pmatrix}$$
$$\bullet\begin{pmatrix} \mathbf{0}_{m\times 1} \\ \tilde{\mathbf{e}}_B \end{pmatrix}$$
$$= \begin{pmatrix} -\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\left(\mathbf{Q}_B-\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\right)^{-1}\mathbf{W}_B\left(\mathbf{Q}_B-\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\right)\tilde{\mathbf{e}}_B \\ \left(\mathbf{Q}_B-\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\right)^{-1}\mathbf{W}_B\left(\mathbf{Q}_B-\mathbf{Q}'_{AB}\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\right)\tilde{\mathbf{e}}_B \end{pmatrix}$$

Therefore

$$\begin{aligned}\hat{\mathbf{e}}_A &= -\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\left(\mathbf{Q}_B - \mathbf{Q}_{AB}'\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\right)^{-1}\mathbf{W}_B\left(\mathbf{Q}_B - \mathbf{Q}_{AB}'\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\right)\tilde{\mathbf{e}}_B \\ &= -\mathbf{Q}_A^{-1} \cdot \frac{n_1 n_2}{n_1 + n_2}(\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_{A_2})(\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})' \\ &\quad \cdot \left(\mathbf{Q}_B - \mathbf{Q}_{AB}'\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\right)^{-1}\mathbf{W}_B\left(\mathbf{Q}_B - \mathbf{Q}_{AB}'\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\right)\tilde{\mathbf{e}}_B\end{aligned}$$

and

$$\hat{\mathbf{e}}_B = \left(\mathbf{Q}_B - \mathbf{Q}_{AB}'\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\right)^{-1}\mathbf{W}_B\left(\mathbf{Q}_B - \mathbf{Q}_{AB}'\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\right)\tilde{\mathbf{e}}_B.$$

Now R calculates

$$\tilde{\tilde{\mathbf{e}}}_B = (\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_{B_2})'\left(\mathbf{Q}_B - \mathbf{Q}_{AB}'\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\right)^{-1}\mathbf{W}_B\left(\mathbf{Q}_B - \mathbf{Q}_{AB}'\mathbf{Q}_A^{-1}\mathbf{Q}_{AB}\right)\tilde{\mathbf{e}}_B$$

and $\hat{\mathbf{e}}_B$, then sends $\tilde{\tilde{e}}_B$ and λ to H. H uses it to obtain $\hat{e}_A$.

In the above procedures:

H: (Hospital administration computes)	R: (Research group computes)	H and R : compute jointly	H sends to R:	R sends to H:
$\lvert\mathbf{Q}_A\rvert$	$\lvert\mathbf{Q}_B\rvert$	$\mathbf{Q}_{AB}$	$\mathbf{Z}_A$	$\tilde{\tilde{\mathbf{e}}}_B$
$\mathbf{Z}_A$	$\tilde{\mathbf{e}}_B$	λ		λ
$\hat{\mathbf{e}}_A$	$\tilde{\tilde{e}}_B$			
	$\hat{\mathbf{e}}_B$			

According to Fisher's classification procedure based on sample discriminants, we allocate $\mathbf{x}$ to populaltion group π_k if:

$$[\hat{\mathbf{e}}'(\mathbf{x} - \bar{\mathbf{x}}_k)]^2 \leq [\hat{\mathbf{e}}'(\mathbf{x} - \bar{x}_i)]^2 \qquad \text{for all } i \neq k$$

Since

$$\hat{\mathbf{e}} = \begin{pmatrix} \hat{\mathbf{e}}_A \\ \hat{\mathbf{e}}_B \end{pmatrix}$$

therefore discriminant:

$$\begin{aligned} z_j &= \hat{\mathbf{e}}'(\mathbf{x} - \bar{x}_j) \\ &= \begin{pmatrix} \hat{\mathbf{e}}_A \\ \hat{\mathbf{e}}_B \end{pmatrix}' \begin{pmatrix} \mathbf{x}_A - \bar{\mathbf{x}}_{jA} \\ \mathbf{x}_B - \bar{\mathbf{x}}_{jB} \end{pmatrix} \\ &= \hat{\mathbf{e}}_A'(\mathbf{x}_A - \bar{\mathbf{x}}_{jA}) + \hat{\mathbf{e}}_B'(\mathbf{x}_B - \bar{\mathbf{x}}_{jB}) \\ &= z_{jA} + z_{jB} \qquad j = 1,2 \end{aligned}$$

where

$$z_{jA} = \hat{\mathbf{e}}_A'(\mathbf{x}_A - \bar{\mathbf{x}}_{jA})$$

and

$$z_{jB} = \hat{\mathbf{e}}_B'(\mathbf{x}_B - \bar{\mathbf{x}}_{jB})$$

Now H calculates all the z_{jA} and sends them to R; R also calculates all the z_{jB} and obtains the linear discriminants z_1 and z_2. We use these values to do the classification according to the above rule.

Thus in Section 4.1.1.1 we have:

H(Hospital computes):	R(Research group computes):	H and R compute jointly:	H → R:	R → H:
$\bar{\mathbf{x}}_{Ai}(i = 1,2)$	$\bar{\mathbf{x}}_{B_i}(i = 1,2)$	$\mathbf{Q}_{AB}$	$\mathbf{Z}_A$	$\tilde{\tilde{\mathbf{e}}}_B$
$\bar{\mathbf{x}}_A$	$\bar{\mathbf{x}}_B$	λ	$z_{jA}(j = 1,2)$	λ
$\mathbf{S}_{A_i}(i = 1,2)$	$\mathbf{S}_{B_i}(i = 1,2)$	$z_j(j = 1,2)$		$z_{jB}(j = 1,2)$
$\mathbf{W}_A$	$\mathbf{W}_B$			
$\bar{\mathbf{x}}_{A_1} - \bar{\mathbf{x}}_A$	$\bar{\mathbf{x}}_{B_1} - \bar{\mathbf{x}}_B$			
$\bar{\mathbf{x}}_{A_2} - \bar{\mathbf{x}}_A$	$\bar{\mathbf{x}}_{B_2} - \bar{\mathbf{x}}_B$			
$\lvert\mathbf{Q}_A\rvert$	$\lvert\mathbf{Q}_B\rvert$			
Z_A	$\tilde{\mathbf{e}}_B$			
$\hat{\mathbf{e}}_A$	$\tilde{\tilde{\mathbf{e}}}_B$			
$z_{jA}(j = 1,2)$	$\hat{e}_B$			
	$z_{jB}(j = 1,2)$			

4.1.1.2 The $(m+n)\times(m+n)$ population covariance matrices are equal but not of full rank

We now consider the case where the two covariance matrices are equal butr not of full rank.

$$\boldsymbol{\Sigma}_1 = \begin{pmatrix} \boldsymbol{\Sigma}_{A_1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{B_1} \end{pmatrix} = \boldsymbol{\Sigma}_2 = \begin{pmatrix} \boldsymbol{\Sigma}_{A_2} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{B_2} \end{pmatrix} = \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_A & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_B \end{pmatrix}.$$

Let $\mathbf{P}_A = \begin{pmatrix} \mathbf{e}_{A1} & \mathbf{e}_{A2} & \cdots & \mathbf{e}_{Aq} \end{pmatrix}$ ($\mathbf{P}_B = \begin{pmatrix} \mathbf{e}_{B1} & \mathbf{e}_{B2} & \cdots & \mathbf{e}_{Bp} \end{pmatrix}$) be the eigenvectors of $\boldsymbol{\Sigma}_A$ ($\boldsymbol{\Sigma}_B$) corresponding to nonzero eigenvalues $\begin{pmatrix} \lambda_{A1} & \lambda_{A2} & \cdots & \lambda_{Aq} \end{pmatrix}$ ($\begin{pmatrix} \lambda_{B1} & \lambda_{B2} & \cdots & \lambda_{Bp} \end{pmatrix}$).Let $\mathbf{P} = \begin{pmatrix} \mathbf{P}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_B \end{pmatrix}$, We replace $\mathbf{X}$ by

$$\begin{aligned} \widetilde{\mathbf{X}} &= \mathbf{X}\mathbf{P} \\ &= \begin{pmatrix} \mathbf{X}_A & \mathbf{X}_B \end{pmatrix}\begin{pmatrix} \mathbf{P}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_B \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{X}_A\mathbf{P}_A & \mathbf{X}_B\mathbf{P}_B \end{pmatrix}. \end{aligned}$$

Since $\mathbf{X}_A\mathbf{P}_A$ ($\mathbf{X}_B\mathbf{P}_B$) has a full rank covariance matrix $\widetilde{\boldsymbol{\Sigma}}_A = \mathbf{P}_A'\boldsymbol{\Sigma}_A\mathbf{P}_A$ ($\widetilde{\boldsymbol{\Sigma}}_B = \mathbf{P}_B'\boldsymbol{\Sigma}_B\mathbf{P}_B$), the covariance matrix of $\widetilde{\mathbf{X}}$ is given by:

$$\begin{aligned} \widetilde{\boldsymbol{\Sigma}} &= \mathbf{P}'\boldsymbol{\Sigma}\mathbf{P} \\ &= \begin{pmatrix} \mathbf{P}_A' & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_B' \end{pmatrix}\begin{pmatrix} \boldsymbol{\Sigma}_A & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_B \end{pmatrix}\begin{pmatrix} \mathbf{P}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_B \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{P}_A'\boldsymbol{\Sigma}_A\mathbf{P}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_B'\boldsymbol{\Sigma}_B\mathbf{P}_B \end{pmatrix} \\ &= \begin{pmatrix} \widetilde{\boldsymbol{\Sigma}}_A & \mathbf{0} \\ \mathbf{0} & \widetilde{\boldsymbol{\Sigma}}_B \end{pmatrix}. \end{aligned}$$

Now since $\widetilde{\Sigma}$ is of full rank then H and R can just calculate the Fisher's discriminants of $\widetilde{\mathbf{X}}$ and get the classification results as described in Section 4.1.1.1.

In this part,

H(Hospital administration computes):	R(Research group computes):
$\mathbf{P}_A$	$\mathbf{P}_B$
$\mathbf{X}_A\mathbf{P}_A$	$\mathbf{X}_B\mathbf{P}_B$

4.1.2 The $(m+n)\times(m+n)$ population covariance matrices of different populations are unequal

We now consider the more general case where the two covariance matrices $(\mathbf{\Sigma}_i, i = 1,2)$ are unequal

$$\mathbf{\Sigma}_1 = \begin{pmatrix} \mathbf{\Sigma}_{A_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{B_1} \end{pmatrix} \neq \mathbf{\Sigma}_2 = \begin{pmatrix} \mathbf{\Sigma}_{A_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{B_2} \end{pmatrix}.$$

Here, we need more information to solve such a problem. In this thesis we assume $\mathbf{X}$ follows a multivariate normal density and we set the estimate of the prior probability (the probability $\mathbf{x}$ comes from π_i) to p_i where:

$$\widehat{p_i} = \frac{n_i}{\sum_{k=1}^{g} n_k}.$$

Then according to Fisher's quadratic discriminant analysis(QDA)[1], the quadratic discrimination score for the i^{th} population is:

$$d_i^Q = -\frac{1}{2}\ln|\mathbf{\Sigma}_i| - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)'\mathbf{\Sigma}_i^{-1}(\mathbf{x}-\mu_i) + \ln p_i$$

The covariance matrix $\mathbf{\Sigma}_i$ of population i, $i = 1,2$ can be written as

$$\Sigma_i = \begin{pmatrix} \Sigma_{A_i} & \mathbf{0} \\ \mathbf{0} & \Sigma_{B_i} \end{pmatrix} \qquad i = 1,2,\cdots g$$

and the mean vector μ_i of population $i, i = 1,2)$ can be partitioned also as

$$\mu_i = \begin{pmatrix} \mu_{iA} \\ \mu_{iB} \end{pmatrix} \qquad i = 1,2,\cdots g$$

Partitioning the test observation data similarly we have:

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_A \\ \mathbf{x}_B \end{pmatrix}$$

therefore the quadratic discrimination score for the i^{th} population is given by:

$$\begin{aligned} d_i^Q &= -\frac{1}{2}(\ln|\Sigma_{A_i}| + \ln|\Sigma_{B_i}|) - \frac{1}{2}\begin{pmatrix} (\mathbf{x}_A-\mu_{iA})' & (\mathbf{x}_B-\mu_{iB})' \end{pmatrix} \\ &\quad \cdot \begin{pmatrix} \Sigma_{A_i}^{-1} & \mathbf{0} \\ \mathbf{0} & \Sigma_{B_i}^{-1} \end{pmatrix}\begin{pmatrix} \mathbf{x}_A-\mu_{iA} \\ \mathbf{x}_B-\mu_{iB} \end{pmatrix} + \ln p_i \\ &= -\frac{1}{2}(\ln|\Sigma_{A_i}| + \ln|\Sigma_{B_i}|) \\ &\quad - \frac{1}{2}\left[(\mathbf{x}_A-\mu_{iA})'\Sigma_{A_i}^{-1}(\mathbf{x}_A-\mu_{iA}) + (\mathbf{x}_B-\mu_{iB})'\Sigma_{B_i}^{-1}(\mathbf{x}_B-\mu_{iB})\right] + \ln p_i \\ &= -\frac{1}{2}\left[\ln|\Sigma_{A_i}| + (\mathbf{x}_A-\mu_{iA})'\Sigma_{A_i}^{-1}(\mathbf{x}_A-\mu_{iA})\right] \\ &\quad - \frac{1}{2}\left[\ln|\Sigma_{B_i}| + (\mathbf{x}_B-\mu_{iB})'\Sigma_{B_i}^{-1}(\mathbf{x}_B-\mu_{iB})\right] + \ln p_i \\ &= z_{iA}^Q + z_{iB}^Q + \ln p_i \end{aligned}$$

where

$$z_{iA}^Q = -\frac{1}{2}\left[\ln|\Sigma_{A_i}| + (\mathbf{x}_A-\mu_{iA})'\Sigma_{A_i}^{-1}(\mathbf{x}_A-\mu_{iA})\right] \text{ for } i = 1,2,\cdots g.$$

$$z_{iB}^Q = -\frac{1}{2}\left[\ln|\Sigma_{B_i}| + (\mathbf{x}_B-\mu_{iB})'\Sigma_{B_i}^{-1}(\mathbf{x}_B-\mu_{iB})\right] \text{ for } i = 1,2,\cdots g.$$

Thus the sample quadratic discrimination score for the i^{th} population is:

$$\hat{d}_i^Q = -\frac{1}{2}\ln|\mathbf{W}_i| - \frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}}_i)'\mathbf{S}_i^{-1}(\mathbf{x}-\bar{\mathbf{x}}_i) + \ln\hat{p}_i$$
$$= \hat{z}_{iA}^Q + \hat{z}_{iB}^Q + \ln\hat{p}_i, \text{ for } i = 1,2$$

where

$$\hat{z}_{iA}^Q = -\frac{1}{2}\left[\ln|\mathbf{S}_{A_i}| + (\mathbf{x}_A - \bar{\mathbf{x}}_{iA})'\mathbf{S}_{A_i}^{-1}(\mathbf{x}_A - \bar{\mathbf{x}}_{iA})\right]$$

$$\hat{z}_{iB}^Q = -\frac{1}{2}\left[\ln|\mathbf{S}_{B_i}| + (\mathbf{x}_B - \bar{\mathbf{x}}_{iB})'\mathbf{S}_{B_i}^{-1}(\mathbf{x}_B - \bar{\mathbf{x}}_{iB})\right]$$

$$\hat{p}_i = \frac{n_i}{\sum_{k=1}^{g} n_k} \qquad i = 1,2,\cdots g$$

In this situation, H computes $\hat{z}_{iA}^Q$ and sends it to R; then R can calculate $\hat{z}_{iB}^Q$ and $\ln\hat{p}_i$ and obtain $\hat{d}_i^Q$ (here $g = 2$). According to the classification rule:

Allocate x to π_k, if

$$d_k^Q = \max\{d_i^Q, i = 1,2,\cdots g\} \qquad [3]$$

So R allocates x to π_k, if

$$\hat{d}_k^Q = \max\{\hat{d}_i^Q, i = 1,2\}$$

In summary for Section 4.1.2,

H (Hospital administration computes):	R(Research group computes):	H and R compute jointly:	H sends to R:	R sends to H:
$\lvert\mathbf{S}_{A_i}\rvert, i = 1,2$	$\lvert\mathbf{S}_{B_i}\rvert, i = 1,2$	$\hat{p}_i$	$\hat{z}_{iA}^Q, i = 1,2$	$\hat{z}_{iB}^Q, i = 1,2$
$\mathbf{S}_{A_i}^{-1}, i = 1,2$	$\mathbf{S}_{B_i}^{-1}, i = 1,2$	$\hat{d}_i^Q, i = 1,2$		
$\mathbf{x}_A - \bar{\mathbf{x}}_{iA}, i = 1,2$	$\mathbf{x}_B - \bar{\mathbf{x}}_{iB}, i = 1,2$			
$\hat{z}_{iA}^Q, i = 1,2$	$\hat{z}_{iB}^Q, i = 1,2$			

4.2 Horizontally Partitioned Data

For this situation, there are a total of n variables and H and R have different exclusive cases for these n variables. .That is H has data matrix **A** which is a $N_A \times n$ matrix, and R has data matrix **B** is a $N_B \times n$ matrix. So here we have **A** partitioned into $\mathbf{A}_{1N_{A1}\times n}$ and $\mathbf{A}_{2N_{A2}\times n}$ and **B** partitioned into $\mathbf{B}_{1N_{B1}\times n}$, and $\mathbf{B}_{2N_{B2}\times n}$. Now we can write

$$\mathbf{X}_1 = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{B}_1 \end{pmatrix}$$

$$\mathbf{X}_2 = \begin{pmatrix} \mathbf{A}_2 \\ \mathbf{B}_2 \end{pmatrix}$$

and the the sample mean vector of population $i, (i = 1, 2)$can be written as :

$$\bar{\mathbf{x}}_i = \frac{N_{Ai}\bar{\mathbf{x}}_{Ai} + N_{Bi}\bar{\mathbf{x}}_{Bi}}{N_{Ai} + N_{Bi}} = \tilde{\mathbf{x}}_{Ai} + \tilde{\mathbf{x}}_{Bi}$$

where

$$\tilde{\mathbf{x}}_{Ai} = \frac{N_{Ai}\bar{\mathbf{x}}_{Ai}}{N_{Ai} + N_{Bi}} \text{ and } \tilde{\mathbf{x}}_{Bi} = \frac{N_{Bi}\bar{\mathbf{x}}_{Bi}}{N_{Ai} + N_{bi}} \quad , i = 1, 2$$

Now $\tilde{\mathbf{x}}_{Ai}(\tilde{\mathbf{x}}_{Bi})$ can be obtained by H (R) independently. The sample mean vector for the entire data is given by:

$$\begin{aligned} \bar{\mathbf{x}} &= \frac{1}{N}\sum_{i=1}^{2}(N_{Ai} + N_{Bi})\bar{\mathbf{x}}_i \\ &= \frac{1}{N}\sum_{i=1}^{2} N_{Ai}\bar{\mathbf{x}}_{Ai} + \frac{1}{N}\sum_{i=1}^{2} N_{Bi}\bar{\mathbf{x}}_{Bi} \\ &= \tilde{\mathbf{x}}_A + \tilde{\mathbf{x}}_B \end{aligned}$$

where

$$\tilde{\mathbf{x}}_A = \frac{1}{N}\sum_{i=1}^{2} N_{Ai}\bar{\mathbf{x}}_{Ai} = \sum_{i=1}^{2}\left(\frac{N_{Ai}+N_{Bi}}{N}\right)\tilde{\mathbf{x}}_{Ai}$$

and

$$\tilde{\mathbf{x}}_B = \frac{1}{N}\sum_{i=1}^{2} N_{Bi}\bar{\mathbf{x}}_{Bi} = \sum_{i=1}^{2}\left(\frac{N_{Ai}+N_{Bi}}{N}\right)\tilde{\mathbf{x}}_{Bi}$$

Now $\tilde{\mathbf{x}}_A$ ($\tilde{\mathbf{x}}_B$) can be obtained by H (R) independently. Thus we have

H(hospital administration computes and keeps):	R(Research group computes and keeps):
$\tilde{\mathbf{x}}_{Ai}(i = 1,2)$	$\tilde{\mathbf{x}}_{Bi}(i = 1,2)$
$\tilde{\mathbf{x}}_A$	$\tilde{\mathbf{x}}_B$

Now we consider several possible situations.

4.2.1 The $n \times n$ population covariance matrices of different populations are equal

Here we assume that both population covariance matrices are equal

$$\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma},$$

The sample estimate of the covariance matrices of $\mathbf{X}_1$ and $\mathbf{X}_2$ is:

$$\begin{aligned}\mathbf{W} &= (N_{A1}-1)\mathbf{S}_{A1} + (N_{A2}-1)\mathbf{S}_{A2} + (N_{B1}-1)\mathbf{S}_{B1} + (N_{B2}-1)\mathbf{S}_{B2} \\ &= \mathbf{W}_A + \mathbf{W}_B\end{aligned}$$

where

$$\mathbf{W}_A = (N_{A1}-1)\mathbf{S}_{A1} + (N_{A2}-1)\mathbf{S}_{A2}$$

and

$$\mathbf{W}_B = (N_{B1}-1)\mathbf{S}_{B1} + (N_{B2}-1)\mathbf{S}_{B2}$$

Now we have

H(hospital administration computes and keeps): $\mathbf{W}_A$ R(Research group computes and keeps): $\mathbf{W}_B$

4.2.1.1 The $n \times n$ population covariance matrices are of full rank as well.

For this case, using the matrix inverse protocol mentioned in Section 3.2, H and R can get $\mathbf{W}^{-1}$ without disclosing their own original data.

$$
\begin{aligned}
\mathbf{B}_{SS} &= \sum_{i=1}^{2}(N_{Ai}+N_{Bi})(\bar{\mathbf{x}}_i-\bar{\mathbf{x}})(\bar{\mathbf{x}}_i-\bar{\mathbf{x}})' \\
&= \sum_{i=1}^{2}(N_{Ai}+N_{Bi})\left[(\bar{\mathbf{x}}_{Ai}+\bar{\mathbf{x}}_{Bi})-\left(\sum_{j=1}^{2}\left(\frac{N_{Aj}+N_{Bj}}{N}\right)\bar{\mathbf{x}}_{Aj}+\sum_{j=1}^{2}\left(\frac{N_{Aj}+N_{Bj}}{N}\right)\bar{\mathbf{x}}_{Bj}\right)\right] \\
&\quad \left[(\bar{\mathbf{x}}_{Ai}+\bar{\mathbf{x}}_{Bi})-\left(\sum_{j=1}^{2}\left(\frac{N_{Aj}+N_{Bj}}{N}\right)\bar{\mathbf{x}}_{Aj}+\sum_{j=1}^{2}\left(\frac{N_{Aj}+N_{Bj}}{N}\right)\bar{\mathbf{x}}_{Bj}\right)\right]' \\
&= \sum_{i=1}^{2}(N_{Ai}+N_{Bi})\left[\left(\bar{\mathbf{x}}_{Ai}-\sum_{j=1}^{2}\frac{N_{Aj}+N_{Bj}}{N}\bar{\mathbf{x}}_{Aj}\right)+\left(\bar{\mathbf{x}}_{Bi}-\sum_{j=1}^{2}\frac{N_{Aj}+N_{Bj}}{N}\bar{\mathbf{x}}_{Bj}\right)\right] \\
&\quad \left[\left(\bar{\mathbf{x}}_{Ai}-\sum_{j=1}^{2}\frac{N_{Aj}+N_{Bj}}{N}\bar{\mathbf{x}}_{Aj}\right)+\left(\bar{\mathbf{x}}_{Bi}-\sum_{j=1}^{2}\frac{N_{Aj}+N_{Bj}}{N}\bar{\mathbf{x}}_{Bj}\right)\right]' \\
&= \sum_{i=1}^{2}(N_{Ai}+N_{Bi})(\mathbf{x}^*_{Ai}+\mathbf{x}^*_{Bi})(\mathbf{x}^*_{Ai}+\mathbf{x}^*_{Bi})' \\
&= \sum_{i=1}^{2}(N_{Ai}+N_{Bi})(\mathbf{x}^*_{Ai}\mathbf{x}^{*\prime}_{Ai}+\mathbf{x}^*_{Bi}\mathbf{x}^{*\prime}_{Ai}+\mathbf{x}^*_{Ai}\mathbf{x}^{*\prime}_{Bi}+\mathbf{x}^*_{Bi}\mathbf{x}^{*\prime}_{Bi})
\end{aligned}
$$

where

$$\mathbf{x}^*_{Ai}=\bar{\mathbf{x}}_{Ai}-\sum_{j=1}^{2}\frac{N_{Aj}+N_{Bj}}{N}\bar{\mathbf{x}}_{Aj}=\bar{\mathbf{x}}_{Ai}-\bar{\mathbf{x}}_A$$

$$\mathbf{x}^*_{Bi}=\bar{\mathbf{x}}_{Bi}-\sum_{j=1}^{2}\frac{N_{Aj}+N_{Bj}}{N}\bar{\mathbf{x}}_{Bj}=\bar{\mathbf{x}}_{Bi}-\bar{\mathbf{x}}_B$$

Using the matrix product protocol in Section 3.1.1 or Section 3.1.2, H and R can get

$\mathbf{V}_{Ai} + \mathbf{V}_{Bi} = \mathbf{x}^*_{Bi}\mathbf{x}^{*\prime}_{Ai}$ and then get

$$\mathbf{x}^*_{Ai}\,\mathbf{x}^{*\prime}_{Bi} = (\mathbf{x}^*_{Bi}\mathbf{x}^*_{Ai})' = (\mathbf{V}_{Ai} + \mathbf{V}_{Bi})' = \mathbf{V}'_{Ai} + \mathbf{V}'_{Bi}.$$

Now H computes $\mathbf{T}_A = \sum_{i=1}^{2}(N_{Ai} + N_{Bi})(\mathbf{V}_{Ai} + \mathbf{V}'_{Ai} + \mathbf{x}^*_{Ai}\mathbf{x}^{*\prime}_{Ai})$ and R computes $\mathbf{T}_B = \sum_{i=1}^{2}(N_{Ai} + N_{Bi})(\mathbf{V}_{Bi} + \mathbf{V}'_{Bi} + \mathbf{x}^*_{Bi}\mathbf{x}^{*\prime}_{Bi})$. H sends $\mathbf{T}_A$ to R and R send $\mathbf{T}_B$ to H. Now H and R jointly get $T_A + T_B = B_{SS}$. They can calculate the eigenvalue λ of $\mathbf{W}^{-1}\mathbf{B}$, for $g = 2$, and the corresponding eigenvector $\mathbf{e}$. H computes

$$z_{Ai} = \mathbf{e}'\tilde{\mathbf{x}}_{Ai}$$

where

$$\tilde{\mathbf{x}}_{Ai} = \frac{N_{Ai}\bar{\mathbf{x}}_{Ai}}{N_{Ai} + N_{Bi}}, i = 1,2$$

and sends z_{Ai} to H

Then R can get

$$\begin{aligned} z_i &= \mathbf{e}'(\mathbf{x} - \bar{\mathbf{x}}_i) \\ &= \mathbf{e}'\mathbf{x} - \mathbf{e}'(\tilde{\mathbf{x}}_{Ai} + \tilde{\mathbf{x}}_{Bi}) \\ &= \mathbf{e}'\mathbf{x} - \mathbf{e}'\tilde{\mathbf{x}}_{Bi} - \mathbf{e}'\tilde{\mathbf{x}}_{Ai} \\ &= \mathbf{e}'\mathbf{x} - \mathbf{e}'\tilde{\mathbf{x}}_{Bi} - z_{Ai} \end{aligned}$$

Finally we use the classification rule mentioned in Section 4.1.1.1 to allocate x to one of the two populations.

In 4.2.1.1 section,

H(Hospital administration computes):	R(Research group computes):	H and R compute jointly:	H sends to R:	R sends to H:
$\tilde{\mathbf{x}}_{Ai}(i = 1,2)$	$\tilde{\mathbf{x}}_{Bi}(i = 1,2)$	$\mathbf{x}^*_{Bi}\mathbf{x}^{*\prime}_{Ai}$	$\mathbf{T}_A$	$\tilde{\tilde{\mathbf{e}}}_B$
$\tilde{\mathbf{x}}_A$	$\tilde{\mathbf{x}}_B$	$\mathbf{W}^{-1}$	$\mathbf{V}_{Ai}(i = 1,2)$	$\mathbf{V}_{Bi}(i = 1,2)$
$\mathbf{W}_A$	$\mathbf{W}_B$	$\mathbf{B}_{SS}$	$\mathbf{T}_A$	$\mathbf{T}_B$
$\mathbf{V}_{Ai}(i = 1,2)$	$\mathbf{V}_{Bi}(i = 1,2)$	λ	$z_{Ai}(i = 1,2)$	
$\mathbf{T}_A$	$\mathbf{T}_B$	$\mathbf{e}$		
$z_{Ai}(i = 1,2)$	$z_i(i = 1,2)$	$z_i(i = 1,2)$		

4.2.1.2 The $n \times n$ population covariance matrices are equal but not of full rank

In this case, since both population covariance matrices are equal

$$\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$$

Let $\mathbf{P}_i(i = 1,2)$ denote the eigenvectors of $\boldsymbol{\Sigma}_i(i = 1,2)$ and $\mathbf{P}$ be the eigenvectors of Σ., therefore

$$\mathbf{P}_1 = \mathbf{P}_2 = \mathbf{P}$$

which tells us that

$$\tilde{\boldsymbol{\Sigma}}_1 = \mathbf{P}_1'\boldsymbol{\Sigma}_1\mathbf{P}_1 = \tilde{\boldsymbol{\Sigma}}_2 = \mathbf{P}_2'\boldsymbol{\Sigma}_2\mathbf{P}_2 = \boldsymbol{\Sigma} = \mathbf{P}'\boldsymbol{\Sigma}\mathbf{P}$$

are of full rank.

Let $\mathbf{P}_A = \begin{pmatrix} \mathbf{e}_{A1} & \mathbf{e}_{A2} & \cdots & \mathbf{e}_{Aq} \end{pmatrix}$ ($\mathbf{P}_B = \begin{pmatrix} \mathbf{e}_{B1} & \mathbf{e}_{B2} & \cdots & \mathbf{e}_{Bp} \end{pmatrix}$) be the eigenvectors of W_A (W_B), the estimate of Σ, corresponding to nonzero eigenvalues $\begin{pmatrix} \lambda_{A1} & \lambda_{A2} & \cdots & \lambda_{Aq} \end{pmatrix}$ ($\begin{pmatrix} \lambda_{B1} & \lambda_{B2} & \cdots & \lambda_{Bp} \end{pmatrix}$).Then we replace **A** (**B**) by

$$\tilde{\mathbf{A}} = \mathbf{A}\mathbf{P}_A$$

and

$$\widetilde{\mathbf{B}} = \mathbf{B}\mathbf{P}_B$$

Now since $\mathbf{X}_A\mathbf{P}_A$ ($\mathbf{X}_B\mathbf{P}_B$) has a full rank covariance matrix

$$\widetilde{\mathbf{\Sigma}}_A = \mathbf{P}_A'\mathbf{\Sigma}_A\mathbf{P}_A$$
$$\widetilde{\mathbf{\Sigma}}_B = \mathbf{P}_B'\mathbf{\Sigma}_B\mathbf{P}_B$$

The covariance matrix of $\widetilde{\mathbf{X}}$ is $\widetilde{\mathbf{\Sigma}} = \widetilde{\mathbf{\Sigma}}_A = \mathbf{\Sigma}_B$ and $\widetilde{\mathbf{\Sigma}}$ is of full rank. So for this Section 4.2.1.2 we have

H(hospital administration computes and keeps):	R(Research group computes and keeps):
$\mathbf{P}_A$	$\mathbf{P}_B$
$\widetilde{\mathbf{A}}$	$\widetilde{\mathbf{B}}$

Then H and R can just calculate the Fisher's discriminants of $\widetilde{\mathbf{X}}$ and get the classification results as described in Section 4.2.1.1 .

4.2.2 The two $n \times n$ population covariance matrices are unequal

We now consider the case where the two population covariance matrices are not the same. i.e. $\mathbf{\Sigma}_1 \neq \mathbf{\Sigma}_2$.

Again, we need more information to solve such a problem. In this thesis we assume $\mathbf{X}$ follows a multivariate normal density and set the estimate of the prior probability (the probability of an observation $\mathbf{x}$ coming from population π_i) to p_i:

$$\widehat{p_i} = \frac{N_i}{\sum_{k=1}^{g} N_k}$$

We may wrote the sample estimate of the population covariance matrix $\mathbf{\Sigma}_i$ as

$$\mathbf{S}_i = (N_{Ai} - 1)\mathbf{S}_{Ai} + (N_{Bi} - 1)\mathbf{S}_{Bi} \qquad i = 1, 2, \cdots g$$

Now $\mathbf{S}_i$ is invertible. Now H and R use the matrix inverse protocol to get $\mathbf{S}_i^{-1}$; they then can calculate $|\mathbf{S}_i| = \frac{1}{|\mathbf{S}_i^{-1}|}$. Now the sample mean vector of population i $(i = 1,2)$is given by :

$$\bar{\mathbf{x}}_i = \frac{N_{Ai}\bar{\mathbf{x}}_{A_i} + N_{Bi}\bar{\mathbf{x}}_{Bi}}{N_{Ai} + N_{Bi}} = \tilde{\mathbf{x}}_{Ai} + \tilde{\mathbf{x}}_{Bi}$$

where

$$\tilde{\mathbf{x}}_{Ai} = \frac{N_{Ai}\bar{\mathbf{x}}_{A_i}}{N_{Ai} + N_{Bi}},$$
$$\tilde{\mathbf{x}}_{Bi} = \frac{N_{Bi}\bar{\mathbf{x}}_{Bi}}{N_{Ai} + N_{bi}} \qquad \text{for } i = 1,2$$

Therefore, according to Fisher's quadratic discriminants analysis(QDA)[3], the sample quadratic discrimination score for the i^{th} population is:

$$\begin{aligned}\widehat{d_i^Q} &= -\frac{1}{2}\ln|\mathbf{S}_i| - \frac{1}{2}(\mathbf{x} - (\tilde{\mathbf{x}}_{Ai} + \tilde{\mathbf{x}}_{Bi}))'\mathbf{S}_i^{-1\prime}(\mathbf{x} - (\tilde{\mathbf{x}}_{Ai} + \tilde{\mathbf{x}}_{Bi})) + \ln p_i \\ &= -\frac{1}{2}\ln|\mathbf{S}_i| \\ &\quad - \frac{1}{2}\left[(\mathbf{x} - \tilde{\mathbf{x}}_{Bi})'\mathbf{S}_i^{-1}(\mathbf{x} - \tilde{\mathbf{x}}_{Bi}) - (\mathbf{x} - \tilde{\mathbf{x}}_{Bi})'\mathbf{S}_i^{-1}\tilde{\mathbf{x}}_{Ai} - \tilde{\mathbf{x}}'_{Ai}\mathbf{S}_i^{-1}(\mathbf{x} - \tilde{\mathbf{x}}_{Bi}) + \tilde{\mathbf{x}}'_{A_i}\mathbf{S}_i^{-1}\tilde{\mathbf{x}}_{A_i}\right] + \ln p_i\end{aligned}$$

Since $\mathbf{S}_{Ai}$ is symmetric, thus $\mathbf{S}_i$ and $\mathbf{S}_i^{-1}$ are symmetric. Now

$$(\mathbf{x} - \tilde{\mathbf{x}}_{Bi})'\mathbf{S}^{-1}\tilde{\mathbf{x}}_{A_i} = \left(\tilde{\mathbf{x}}'_{A_i}\mathbf{S}^{-1}(\mathbf{x} - \tilde{\mathbf{x}}_{Bi})\right)'$$

Using the matrix product protocol, H (R) can get $\mathbf{V}_{Ai}$ $(\mathbf{V}_{Bi})$, where

$$\mathbf{V}_{Ai} + \mathbf{V}_{Bi} = (\mathbf{x} - \tilde{\mathbf{x}}_{Bi})'\mathbf{S}^{-1}\tilde{\mathbf{x}}_{A_i}$$

and

$$\mathbf{V}'_{Ai} + \mathbf{V}'_{Bi} = \tilde{\mathbf{x}}'_{A_i}\mathbf{S}^{-1}(\mathbf{x} - \tilde{\mathbf{x}}_{Bi})$$

and H computes

$$t_{Ai} = \widetilde{x_{Ai}}'\mathbf{S}_i^{-1}\widetilde{x_{Ai}} + \mathbf{V}_{Ai} + \mathbf{V}'_{Ai}$$

and R computes

$$t_{Bi} = (\mathbf{x} - \bar{\mathbf{x}}_{Bi})' \mathbf{S}_i^{-1} (\mathbf{x} - \bar{\mathbf{x}}_{Bi}) + \mathbf{V}_{Bi} + \mathbf{V}'_{Bi}$$

and they then exchange these values . The can then obtain the sample quadratic discrimination score for the i^{th} population by computing

$$\widehat{d_i^Q} = -\frac{1}{2} \ln|\mathbf{S}_i| - \frac{1}{2}(t_{iA} + t_{iB}) + \ln p_i \text{ where } i = 1,2$$

$$\widehat{p_i} = \frac{n_i}{\sum_{k=1}^{2} n_k} \text{ for } i = 1,2$$

According to the classification rule:

Allocate $\mathbf{x}$ to π_k, if

$$d_k^Q = \max\{d_i^Q, i = 1,2,\cdots g\} \qquad [3]$$

So R allocates $\mathbf{x}$ to π_k, if

$$\widehat{d_k^Q} = \max\{\widehat{d_i^Q}, i = 1,2\}.$$

Thus for this Section 4.2.2 we have

H(Hospital administration computes):	R(Research group computes):	H and R compute jointly:	H sends to R:	R sends to H:
$\bar{\mathbf{x}}_{A_i}(i = 1,2)$	$\bar{\mathbf{x}}_{Bi}(i = 1,2)$	$\mathbf{S}_i^{-1}$	$\mathbf{V}_{Ai}(i = 1,2)$	$\mathbf{V}_{Bi}(i = 1,2)$
	$\mathbf{x} - \bar{\mathbf{x}}_{Bi}$	$\|\mathbf{S}_i\|$	t_{iA}	t_{iB}
$\mathbf{V}_{Ai}(i = 1,2)$	$\mathbf{V}_{Bi}(i = 1,2)$	$\widehat{p_i}$		
t_{iA}	t_{iB}	$\widehat{d_i^Q}(i = 1,2)$		

5. Implementation and Application

We illustrate the above methodology using a multiple-sclerosis dataset (the CD-ROM offered by [1]). In the dataset, the observations are divided into non-multiple-sclerosis and multiple-sclerosis cases and five (5) associated variables are given. We have developed R code to carry out the computations. To check the accuracy of our methods, we randomly selected two-thirds of the observations as the training data to build the classifer and used the left out one-third to do the test. We then computed the misclassification rate. This procedure was repeated 1000 times and the average of all the 1000 misclassification rates was computed.

5.1 Vertically Partitioned Data

In [1], the multiple-sclerosis data are given in one table and the last variable in the public data shows to which population.the observation belongs (i.e. multiple sclerosis or non multiple sclerosis). To test the procedure for vertically partitioned data, we divided the data matrix vertically into two exclusive parts **A** and **B**. Assuming that hospital administration has **A** and the research group has **B**, and then partitioning the data according to the population from which the data come, we separate **A** into $\mathbf{A}_1$ and $\mathbf{A}_2$, and **B** into $\mathbf{B}_1$ and $\mathbf{B}_2$. The hospital administration and the research group now can follow our procedures outlined in Chapter 4 to do the classification jointly without sharing their original data and thus protecting patients' privacy.

5.1.1 The $(m+n) \times (m+n)$ population covariance matrices of different populations are equal

We compare the average misclassification rates of the solution procedure in 4.1.1, with that of Fisher's linear discriminant method, based on the same sample and validation data The results are

(The code is given in the Appendix):

Average Misclassification Rate based on 1000 tests	
Privacy Preserving Methods	Fisher's linear discriminant
0.2328485	0.1390303

The average misclassification rate of for our privacy preserving method is greater than Fisher's mainly because, in the R program, we use the Taylor series expansion of a matrix instead of the original matrix to compute the eigenvalue.

5.1.2 The $(m+n) \times (m+n)$ population covariance matrices of different populations are unequal

Here the test results of our method versus the QDA method are as follows(The code is given in the Appendix):

Average Misclassification Rate of 1000 tests	
Privacy Preserving Methods	QDA
0.1138485	0.1180909

5.2 Horizontally Partitioned Data

Using the multiple sclerosis dataset again, we now consider the case of horizontally partitioned data. Here according to the population to which the observation belongs, we divided the matrix into two parts $\mathbf{X}_1$ and $\mathbf{X}_2$. We then select some observations of $\mathbf{X}_1$ ($\mathbf{X}_2$) as $\mathbf{A}_1$ ($\mathbf{A}_2$) and set the remainder as $\mathbf{B}_1$ ($\mathbf{B}_2$). Assume that one hospital has $\mathbf{A}_i (i = 1,2)$ and the other hospital has $\mathbf{B}_i (i = 1,2)$. To protect privacy, one party can not disclose the original data to the other party. To carry out our evaluation of the methodology, in the program we generated the two groups randomly

to obtain $\mathbf{W}^{-1}$, $\mathbf{B}_{SS}$ and $|\mathbf{S}_i|$ according to the protocol in Chapter 3.

5.2.1 The $n \times n$ population covariance matrices of different populations are equal

The test results of our method and the one of Fisher's discriminant method are: (The code is given in the Appendix):

Average Misclassification Rate of 1000 tests

Privacy Preserving Methods	Fisher's Discriminants Method
0.1310303	0.1425758

5.2.2 The $n \times n$ population covariance matrices of different populations are unequal

Given this condition, then the test results of our method and the one of QDA method are: (The code is given in the Appendix):

Average Misclassification Rate of 1000 tests

Privacy Preserving Methods	QDA
0.1646970	0.1206364

6. Conclusion and Future work

In this thesis, we introduce a new privacy-preserving method to do classification of observations into one of two populations when parties have vertically or horizontally partitioned data. The procedures are based upon Fisher's linear and quadratic discriminant functions modified to use privacy preserving protocols presented by Wu et al [2]. The procedures allow two parties to allocate a new observation to one of two populations without disclosing their own private data. We also implemented the methodologies using R software and applied them to a dataset on multiple sclerosis patients; the code appear oin the Appendix.

Our test results show a higher average misclassification rate for the privacy preserving method than the rate for Fisher's linear discriminant on vertically partitioned data under the assumption of equal population covariance matrices. This is due to the using a Taylor expansion as approximation of a matrix in the R code; this indicates that a better matrix approximation method could be used to improve the privacy preserving classification technique.. In all other cases, the average misclassification of the privacy preserving methods are all very close to the corresponding Fisher's discriminant methods.

For the case of vertically partitioned data, we assumed the variables held by the two parties are independent and we restricted ourselves to discriminating between two populations. Future work will examine the case of more than two populations and vertically partitioned data where the independence of the variables is relaxed.

References

[1] Johnson, R. A. and Wichern, D. W.(2002). Discrimination and Classification. *Applied Multivariate Statistical Analysis*(*5th ed*), pages 581-660, USR, New Jersey: Prentice-Hall, Inc..

[2] Du, W., Han, Y. S. and Chen, S. (2004). Multivariate Statistical Analysis Privacy-Preserving Multivariate Statistical Analysis: Linear Regression and Classification. In Proceedings of the Fourth SIAM International Conference on Data Mining.

Appendix

A R code of each test in 5.1

Read data and split it into training and testing data, and sperate them according to description in 5.1:

```
drive <- "D:"
data.dir <- paste(drive, "DATA", "Datamining", sep="/")
(d.file <- paste(data.dir, "T1-6.dat", sep="/"))
n.col=6
d.alldata<- matrix(scan(d.file), ncol=n.col, byrow=T)
train=c(sample(1:69,46),sample(70:98,19))
a=d.alldata[train,c(1,2,4)]
b=d.alldata[train,c(3,5)]
a1=a[1:46,]
a2=a[47:65,]
b1=b[1:46,]
b2=b[47:65,]
```

A.1 R code in 5.1.1

Use privacy preserving Fisher's dicriminants method on vertically patitioned data and Fisher's dicriminants method on the original data to do the classification as described in 5.1.1, given the assumption that the covariance matrices of different populations are equal:

```
S.a1=cov(a1)
S.a2=cov(a2)
W.a=(46-1)*S.a1+(19-1)*S.a2
W.a.inv=solve(W.a)
x.a1.mean=colSums(a1)/46
```

```
x.a2.mean=colSums(a2)/19
U.a=46*19/(46+19)*(x.a1.mean-x.a2.mean)%*%t(x.a1.mean-x.a2.mean)
S.b1=cov(b1)
S.b2=cov(b2)
W.b=(46-1)*S.b1+(19-1)*S.b2
W.b.inv=solve(W.b)
x.b1.mean=colSums(b1)/46
x.b2.mean=colSums(b2)/19
U.b=46*19/(46+19)*(x.b1.mean-x.b2.mean)%*%t(x.b1.mean-x.b2.mean)
z=rep(0,33)
j=1
for(t in 1:98){
for(i in 1:65){
if (t!=train[i])
z[j]=t
else {
z[j]=0
break}}
if (z[j]!=0)
j=j+1}
z=z[1:33]
x=d.alldata[train,1:5]
validation=d.alldata[z,1:5]
x1=x[1:46,]
x2=x[47:65,]
S1=cov(x1)
S2=cov(x2)
```

```
x1.mean=colSums(x1)/46
x2.mean=colSums(x2)/19
W=(46-1)*S1+(19-1)*S2
B=46*19/(46+19)*(x1.mean-x2.mean)%*%t(x1.mean-x2.mean)
z.1=eigen(solve(W)%*%B)$vector[,1]%*%t(validation-rep(1,33)%*%t(x1.mean))
z.2=eigen(solve(W)%*%B)$vector[,1]%*%t(validation-rep(1,33)%*%t(x2.mean))
for(i in 1:33){if(abs(z.1[i])<abs(z.2[i])){z.1[i]=1} else z.1[i]=2}
```

Here to calculate Z_A mentioned in 4.1.1.1, because R just can solve linear equation, we use Taylor series to expand it at $\lambda = 1$:

$$\begin{aligned} Z_A &= (\overline{x_{A_1}} - \overline{x_{A_2}})' Q_A^{-1} (\overline{x_{A_1}} - \overline{x_{A_2}}) \\ &= (\overline{x_{A_1}} - \overline{x_{A_2}})' \left(\frac{n_1 n_2}{n_1 + n_2} (\overline{x_{A_1}} - \overline{x_{A_2}})(\overline{x_{A_1}} - \overline{x_{A_2}})' - \lambda W_A \right)^{-1} (\overline{x_{A_1}} - \overline{x_{A_2}}) \\ &= (\overline{x_{A_1}} - \overline{x_{A_2}})' ((U_A W_A^{-1} - \lambda I) W_A)^{-1} (\overline{x_{A_1}} - \overline{x_{A_2}}) \\ &= (\overline{x_{A_1}} - \overline{x_{A_2}})' W_A^{-1} (U_A W_A^{-1} - \lambda I)^{-1} (\overline{x_{A_1}} - \overline{x_{A_2}}) \end{aligned}$$

where

$$U_A = \frac{n_1 n_2}{n_1 + n_2} (\overline{x_{A_1}} - \overline{x_{A_2}})(\overline{x_{A_1}} - \overline{x_{A_2}})'$$

Since from the Taylor Series expansion,

$$(U_A W_A^{-1} - \lambda I)^{-1} \approx (U_A W_A^{-1} - I)^{-1} + (\lambda - 1)(U_A W_A^{-1} - I)^{-2} + o(I)(\lambda - 1)^2$$

thus

$$\begin{aligned} Z_A &\approx (\overline{x_{A_1}} - \overline{x_{A_2}})' W_A^{-1} \left((U_A W_A^{-1} - I)^{-1} - (U_A W_A^{-1} - I)^{-2} + \lambda (U_A W_A^{-1} - I)^{-2} \right) (\overline{x_{A_1}} - \overline{x_{A_2}}) \\ &= (\overline{x_{A_1}} - \overline{x_{A_2}})' W_A^{-1} \left((U_A W_A^{-1} - I)^{-1} - (U_A W_A^{-1} - I)^{-2} \right) (\overline{x_{A_1}} - \overline{x_{A_2}}) \\ &\quad + \lambda (\overline{x_{A_1}} - \overline{x_{A_2}})' W_A^{-1} (U_A W_A^{-1} - I)^{-2} (\overline{x_{A_1}} - \overline{x_{A_2}}) \\ &= Z_{A1} + \lambda Z_{A2} \end{aligned}$$

where

$$Z_{A1} = (\overline{x_{A_1}} - \overline{x_{A_2}})' W_A^{-1} \left((U_A W_A^{-1} - I)^{-1} - (U_A W_A^{-1} - I)^{-2} \right) (\overline{x_{A_1}} - \overline{x_{A_2}})$$

$$Z_{A2} = (\overline{x_{A_1}} - \overline{x_{A_2}})' W_A^{-1} (U_A W_A^{-1} - I)^{-2} (\overline{x_{A_1}} - \overline{x_{A_2}})$$

```
Z.a1=t(x.a1.mean-x.a2.mean)%*%W.a.inv %*%
    (solve(U.a%*%W.a.inv-diag(1,3,3)) -
    solve(U.a%*%W.a.inv-diag(1,3,3))%*%solve(U.a%*%W.a.inv-diag(1,3,3))) %*%
    x.a1.mean-x.a2.mean)
Z.a2=t(x.a1.mean-x.a2.mean)%*%W.a.inv%*%(solve(U.a%*%W.a.inv-diag(1,3,3))%*%
    solve(U.a%*%W.a.inv-diag(1,3,3)))%*%(x.a1.mean-x.a2.mean)
U.b.star=(46*19/(46+19))^2*(x.b1.mean-x.b2.mean)%*%Z.a2%*%
    t(x.b1.mean-x.b2.mean)+W.b
U.b.bar=(46*19/(46+19))^2*(x.b1.mean-x.b2.mean)%*%Z.a1%*%
    t(x.b1.mean-x.b2.mean)+U.b
D.b=U.b.bar-eigen(U.b.bar%*%solve(U.b.star))$values[1]*U.b.star
e.b.hat=solve(D.b)%*%W.b%*%D.b%*%eigen(U.b.bar%*%solve(U.b.star))$vectors[,1]
e.b.hat2=t(x.b1.mean-x.b2.mean)%*%solve(D.b)%*%W.b%*%D.b%*%
    eigen(U.b.bar%*%solve(U.b.star))$vectors[,1]
e.a.hat=-46*19/(19+46)*solve(U.a-eigen(U.b.bar%*%solve(U.b.star))$values[1]*W.a)%*%
    (x.a1.mean-x.a2.mean)%*%e.b.hat2
z.a1=t(e.a.hat)%*%t(validation[,c(1,2,4)]-rep(1,33)%*%t(x.a1.mean))
z.a2=t(e.a.hat)%*%t(validation[,c(1,2,4)]-rep(1,33)%*%t(x.a2.mean))
z.b1=t(e.b.hat)%*%t(validation[,c(3,5)]-rep(1,33)%*%t(x.b1.mean))
z.b2=t(e.b.hat)%*%t(validation[,c(3,5)]-rep(1,33)%*%t(x.b2.mean))
z1=z.a1+z.b1
```

```
z2=z.a2+z.b2
for(i in 1:33){if(abs(z1[i])<abs(z2[i])){z1[i]=1} else z1[i]=2}
mis.rate.f.m=0
for(i in 1:23){
if(z1[i]!=1)
mis.rate.f.m=mis.rate.f.m+1
}
for(i in 24:33){
if(z1[i]==1)
mis.rate.f.m=mis.rate.f.m+1}
mis.classification.f.m=mis.rate.f.m/33
mis.rate.f=0
for(i in 1:23){
if(z.1[i]!=1)
mis.rate.f=mis.rate.f+1
}
for(i in 24:33){
if(z.1[i]==1)
mis.rate.f=mis.rate.f+1}
mis.classification.f=mis.rate.f/33
```

Attention: in the above privacy preserving method's code, because it is written according to the procedures proved in 4.1.1.1 where $|Q_A| \neq 0$, the code may stop and occur error when the randomly selected training data has $|Q_A| = 0$. If so, you can just ran the code again.

A.2 R code in 5.1.2

Use privacy preserving QDA on vertically patitioned data and QDA on the original data to do

the classification as described in 5.1.2, given the assumption that the covariance matrices of different populations are unequal:

```
p1=46/(46+19)
p2=1-p1
for(i in 1:33){
x.a=validation[i,c(1,2,4)]
z.a1=-log(det(S.a1))/2-t(x.a-x.a1.mean)%*%solve(S.a1)%*%(x.a-x.a1.mean)/2
z.a2=-log(det(S.a2))/2-t(x.a-x.a2.mean)%*%solve(S.a2)%*%(x.a-x.a2.mean)/2
x.b=validation[i,c(3,5)]
z.b1=-log(det(S.b1))/2-t(x.b-x.b1.mean)%*%solve(S.b1)%*%(x.b-x.b1.mean)/2
z.b2=-log(det(S.b2))/2-t(x.b-x.b2.mean)%*%solve(S.b2)%*%(x.b-x.b2.mean)/2
dq1=z.a1+z.b1+log(p1)
dq2=z.a2+z.b2+log(p2)
if(dq1>dq2)
z2[i]=1
else
z2[i]=2
}
for(i in 1:33){
y=validation[i,]
dq1=-log(det(S1))/2-t(y-x1.mean)%*%solve(S1)%*%(y-x1.mean)/2
dq2=-log(det(S2))/2-t(y-x2.mean)%*%solve(S2)%*%(y-x2.mean)/2
dq1=dq1+log(p1)
dq2=dq2+log(p2)
if(dq1>dq2)
z.2[i]=1
else
```

```
z.2[i]=2
}
mis.rate.qd.m=0
for(i in 1:23){
if(z2[i]!=1)
mis.rate.qd.m=mis.rate.qd.m+1
}
for(i in 24:33){
if(z2[i]==1)
mis.rate.qd.m=mis.rate.qd.m+1
}
mis.classification.qd.m=mis.rate.qd.m/33
mis.rate.qd=0
for(i in 1:23){
if(z.2[i]!=1)
mis.rate.qd=mis.rate.qd+1
}
for(i in 24:33){
if(z.2[i]==1)
mis.rate.qd=mis.rate.qd+1
}
mis.classification.qd=mis.rate.qd/33
```

B R code in 5.2

Read data and seperate it as requirement in 5.2. In the code we **assume those third party's matrices be generated as the protocol required**, while they are actually given by ourselves:

```
drive <- "D:"
```

```
data.dir <- paste(drive, "DATA", "Datamining", sep="/")
(d.file <- paste(data.dir, "T1-6.dat", sep="/"))
n.col=6
d.alldata<- matrix(scan(d.file), ncol=n.col, byrow=T)
train=c(sample(1:69,46),sample(70:98,19))
x1=d.alldata[train[1:46],1:5]
x2=d.alldata[train[47:65],1:5]
a1=x1[1:24,]
a2=x2[1:8,]
b1=x1[25:46,]
b2=x2[9:19,]
n=46+19
na1=24
na2=8
nb1=46-24
nb2=19-8
```

B.1 R code in 5.2.1

Use privacy preserving Fisher's dicriminants method on horizontally patitioned data and Fisher's dicriminants method on the original data to do the classification as described in 5.2.1, given the assumption that the covariance matrices of different populations are equal:

```
x.a1.mean=colSums(a1)/na1
x.a2.mean=colSums(a2)/na2
x.b1.mean=colSums(b1)/nb1
x.b2.mean=colSums(b2)/nb2
x.a1.bar=na1/(na1+nb1)*x.a1.mean
x.a2.bar=na2/(na2+nb2)*x.a2.mean
```

```
x.b1.bar=nb1/(na1+nb1)*x.b1.mean
x.b2.bar=nb2/(na2+nb2)*x.b2.mean
x.a.bar=(na1+nb1)/n*x.a1.bar+(na2+nb2)/n*x.a2.bar
x.b.bar=(na1+nb1)/n*x.b1.bar+(na2+nb2)/n*x.b2.bar
x.a1.star=x.a1.bar-x.a.bar
x.a2.star=x.a2.bar-x.a.bar
x.b1.star=x.b1.bar-x.b.bar
x.b2.star=x.b2.bar-x.b.bar
R.a1=c(rep(1,5))
R.b1=c(rep(1,5))
r.a1=diag(1,5)
r.b1=R.a1%*%t(R.b1)-r.a1
x.a1.star2=R.a1+x.a1.star
x.b1.star2=R.b1+x.b1.star
Vb1=diag(c(1,2,3,4,5),5)
T1=x.a1.star2%*%t(x.b1.star)+r.b1-Vb1
Va1=T1+r.a1-R.a1%*%t(x.b1.star2)
R.a2=c(rep(2,5))
R.b2=c(rep(2,5))
r.a2=diag(3,5)
r.b2=R.a2%*%t(R.b2)-r.a2
x.a2.star2=R.a2+x.a2.star
x.b2.star2=R.b2+x.b2.star
Vb2=diag(c(6,7,8,9,10),5)
T2=x.a2.star2%*%t(x.b2.star)+r.b2-Vb2
Va2=T2+r.a2-R.a2%*%t(x.b2.star2)
T.a=(na1+nb1)*(x.a1.star%*%t(x.a1.star)+Va1+t(Va1))+
```

```
      (na2+nb2)*(x.a2.star%*%t(x.a2.star)+Va2+t(Va2))
T.b=(na1+nb1)*(x.b1.star%*%t(x.b1.star)+Vb1+t(Vb1))+
      (na2+nb2)*(x.b2.star%*%t(x.b2.star)+Vb2+t(Vb2))
B=T.a+T.b
p=c(1,2,3,4,5)%*%t(c(3,4,2,5,7))-diag(1,5)
q=c(4,21,7,0,5)%*%t(c(9,1,3,5,7))-diag(1,5)
W.a=(na1-1)*cov(a1)+(na2-1)*cov(a2)
W.b=(nb1-1)*cov(b1)+(nb2-1)*cov(b2)
Rap=c(8,4,3,7,1)%*%t(c(1,0,2,1,9))
Rbp=c(5,3,3,6,1)%*%t(c(1,3,2,1,4))
rap=c(rep(1,5))%*%t(c(2,3,4,1,1))
rbp=Rbp%*%Rap-rap
p2=Rbp+p
W.a2=Rap+W.a
Vap=diag(c(4,2,3,1,5),5)
Tap=p2%*%W.a+rap-Vap
Vbp=Tap+rbp-Rbp%*%W.a2
PW.a=Vap+Vbp
Raq=c(8,4,3,7,1)%*%t(c(1,0,2,1,9))
Rbq=c(5,3,3,6,1)%*%t(c(1,3,2,1,4))
raq=c(rep(1,5))%*%t(c(2,3,4,1,1))
rbq=Raq%*%Rbq-raq
q2=Rbq+q
PW.a2=Raq+PW.a
Vbq=diag(c(1,2,3,1,1),5)
Tbq=PW.a2%*%q+rbq-Vbq
Vaq=Tbq+raq-Raq%*%q2
```

```
PWQ.a=Vaq+Vbq
PWQ.b=p%*%W.b%*%q
PWQ=PWQ.a+PWQ.b
PWQ.inv=solve(PWQ)
PWQ.b=p%*%W.b%*%q
PWQ=PWQ.a+PWQ.b
PWQ.inv=solve(PWQ)
Raq1=c(1,4,3,7,1)%*%t(c(1,0,2,1,1))
Rbq1=c(1,3,3,6,1)%*%t(c(1,3,2,1,4))
raq1=c(rep(1,5))%*%t(c(2,3,4,1,1))
rbq1=Rbq1%*%Raq1-raq1
q.star=Rbq1+q
PWQ.inv1=Raq1+PWQ.inv
Vaq1=diag(c(1,2,3,1,1),5)
Taq1=q.star%*%PWQ.inv+raq1-Vaq1
Vbq1=Taq1+rbq1-Rbq1%*%PWQ.inv1
PW.inv=Vaq1+Vbq1
Rap1=c(1,2,3,1,1)%*%t(c(1,0,2,1,3))
Rbp1=c(4,3,2,1,2)%*%t(c(1,3,2,1,4))
rap1=c(rep(1,5))%*%t(c(2,3,0,1,1))
rbp1=Rap1%*%Rbp1-rap1
p.star=Rbp1+p
PW.inv1=Rap1+PW.inv
Vbp1=diag(c(1,2,0,1,1),5)
Tbp1=PW.inv1%*%p+rbp1-Vbp1
Vap1=Tbp1+rap1-Rap1%*%p.star
W.inv=Vap1+Vbp1
```

```
e=abs(eigen(W.inv%*%B)$vector[,1])
z.a1=e%*%x.a1.bar
z.a2=e%*%x.a2.bar
z.b1=e%*%x.b1.bar
z.b2=e%*%x.b2.bar
z=rep(0,33)
j=1
for(t in 1:98){
for(i in 1:65){
if (t!=train[i])
z[j]=t
else {
z[j]=0
break}}
if (z[j]!=0)
j=j+1}
z=z[1:33]
z.a1=e%*%x.a1.bar
z.a2=e%*%x.a2.bar
z.b1=e%*%x.b1.bar
z.b2=e%*%x.b2.bar
validation=d.alldata[z,1:5]
z1=e%*%t(validation)-z.a1%*%rep(1,33)-z.b1%*%rep(1,33)
z2=e%*%t(validation)-z.a2%*%rep(1,33)-z.b2%*%rep(1,33)
for(i in 1:33){if(abs(z1[i])<abs(z2[i])){z1[i]=1} else z1[i]=2}
mis.rate.f.m=0
for(i in 1:23){
```

```
if(z1[i]!=1)
mis.rate.f.m=mis.rate.f.m+1
}
for(i in 24:33){
if(z1[i]==1)
mis.rate.f.m=mis.rate.f.m+1}
mis.classification.f.m2=mis.rate.f.m/33
```

B.2 R code in 5.2.2

Use privacy preserving QDA on horizontally patitioned data and QDA on the original data to do the classification as described in 5.1.2, given the assumption that the covariance matrices of different populations are unequal:

```
S1=cov(x1)
S2=cov(x2)
x1.mean=colSums(x1)/46
x2.mean=colSums(x2)/19
W=(46-1)*S1+(19-1)*S2
B=46*19/(46+19)*(x1.mean-x2.mean)%*%t(x1.mean-x2.mean)
z.1=eigen(solve(W)%*%B)$vector[,1]%*%t(validation-rep(1,33)%*%t(x1.mean))
z.2=eigen(solve(W)%*%B)$vector[,1]%*%t(validation-rep(1,33)%*%t(x2.mean))
for(i in 1:33){if(abs(z.1[i])<abs(z.2[i])){z.1[i]=1} else z.1[i]=2}
mis.rate.f=0
for(i in 1:23){
if(z.1[i]!=1)
mis.rate.f=mis.rate.f+1
}
for(i in 24:33){
```

```
if(z.1[i]==1)
mis.rate.f=mis.rate.f+1}
mis.classification.f2=mis.rate.f/33
S.a1=(na1-1)*cov(a1)/(na1+nb1-1)
S.b1=(nb1-1)*cov(b1)/(na1+nb1-1)
S.a2=(na2-1)*cov(a2)/(na2+nb2-1)
S.b2=(nb2-1)*cov(b2)/(na2+nb2-1)
p1=c(1,2,1,3,1)%*%t(c(3,4,2,5,1))-diag(1,5)
q1=c(5,2,1,0,5)%*%t(c(1,1,3,5,0))-diag(1,5)
Rap=c(1,4,3,7,1)%*%t(c(1,0,2,1,5))
Rbp=c(0,3,3,6,1)%*%t(c(1,3,2,1,4))
rap=c(rep(1,5))%*%t(c(2,3,4,1,1))
rbp=Rbp%*%Rap-rap
p2=Rbp+p1
S.a1.star=Rap+S.a1
Vap=diag(c(4,2,3,1,5),5)
Tap=p2%*%S.a1+rap-Vap
Vbp=Tap+rbp-Rbp%*%S.a1.star
PS.a1=Vap+Vbp
Raq=c(3,4,3,7,1)%*%t(c(1,0,2,1,3))
Rbq=c(5,3,3,2,1)%*%t(c(1,3,2,1,0))
raq=c(rep(1,5))%*%t(c(2,1,4,1,3))
rbq=Raq%*%Rbq-raq
q2=Rbq+q1
PS.a1.star=Raq+PS.a1
Vbq=diag(c(1,2,0,1,1),5)
Tbq=PS.a1.star%*%q1+rbq-Vbq
```

```
Vaq=Tbq+raq-Raq%*%q2
PSQ.a1=Vaq+Vbq
PSQ.1=PSQ.a1+p1%*%S.b2%*%q1
PSQ.inv1=solve(PSQ.1)
Raq1=c(1,4,3,1,1)%*%t(c(1,0,2,1,1))
Rbq1=c(1,3,3,0,1)%*%t(c(1,3,2,1,4))
raq1=c(rep(1,5))%*%t(c(2,3,0,1,1))
rbq1=Rbq1%*%Raq1-raq1
q.star1=Rbq1+q1
PSQ.inv1.star=Raq1+PSQ.inv1
Vaq1=diag(c(1,2,5,1,1),5)
Taq1=q.star1%*%PSQ.inv1+raq1-Vaq1
Vbq1=Taq1+rbq1-Rbq1%*%PSQ.inv1.star
PS.inv1=Vaq1+Vbq1
Rap1=c(1,2,9,1,1)%*%t(c(1,0,2,1,3))
Rbp1=c(4,3,7,1,2)%*%t(c(1,3,2,1,4))
rap1=c(rep(1,5))%*%t(c(2,3,0,1,1))
rbp1=Rap1%*%Rbp1-rap1
p.star1=Rbp1+p1
PS.inv1.star=Rap1+PS.inv1
Vbp1=diag(c(1,2,0,1,1),5)
Tbp1=PS.inv1.star%*%p1+rbp1-Vbp1
Vap1=Tbp1+rap1-Rap1%*%p.star1
S.inv1=Vap1+Vbp1
d.s1=1/det(S.inv1)
p1=c(1,2,1,3,1)%*%t(c(3,4,2,5,1))-diag(1,5)
q1=c(5,2,1,0,5)%*%t(c(1,1,3,5,0))-diag(1,5)
```

```
Rap=c(1,4,3,7,1)%*%t(c(1,0,2,1,5))
Rbp=c(0,3,3,6,1)%*%t(c(1,3,2,1,4))
rap=c(rep(1,5))%*%t(c(2,3,4,1,1))
rbp=Rbp%*%Rap-rap
p2=Rbp+p1
S.a2.star=Rap+S.a2
Vap=diag(c(4,2,3,1,5),5)
Tap=p2%*%S.a2+rap-Vap
Vbp=Tap+rbp-Rbp%*%S.a2.star
PS.a2=Vap+Vbp
Raq=c(3,4,3,7,1)%*%t(c(1,0,2,1,3))
Rbq=c(5,3,3,2,1)%*%t(c(1,3,2,1,0))
raq=c(rep(1,5))%*%t(c(2,1,4,1,3))
rbq=Raq%*%Rbq-raq
q2=Rbq+q1
PS.a2.star=Raq+PS.a2
Vbq=diag(c(1,2,0,1,1),5)
Tbq=PS.a2.star%*%q1+rbq-Vbq
Vaq=Tbq+raq-Raq%*%q2
PSQ.a2=Vaq+Vbq
PSQ.2=PSQ.a2+p1%*%S.b2%*%q1
PSQ.inv2=solve(PSQ.2)
Raq1=c(1,4,3,1,1)%*%t(c(1,0,2,1,1))
Rbq1=c(1,3,3,0,1)%*%t(c(1,3,2,1,4))
raq1=c(rep(1,5))%*%t(c(2,3,0,1,1))
rbq1=Rbq1%*%Raq1-raq1
q.star1=Rbq1+q1
```

```
PSQ.inv2.star=Raq1+PSQ.inv2
Vaq1=diag(c(1,2,5,1,1),5)
Taq1=q.star1%*%PSQ.inv2+raq1-Vaq1
Vbq1=Taq1+rbq1-Rbq1%*%PSQ.inv2.star
PS.inv2=Vaq1+Vbq1
Rap1=c(1,2,9,1,1)%*%t(c(1,0,2,1,3))
Rbp1=c(4,3,7,1,2)%*%t(c(1,3,2,1,4))
rap1=c(rep(1,5))%*%t(c(2,3,0,1,1))
rbp1=Rap1%*%Rbp1-rap1
p.star1=Rbp1+p1
PS.inv2.star=Rap1+PS.inv2
Vbp1=diag(c(1,2,0,1,1),5)
Tbp1=PS.inv2.star%*%p1+rbp1-Vbp1
Vap1=Tbp1+rap1-Rap1%*%p.star1
S.inv2=Vap1+Vbp1
d.s2=1/det(S.inv2)
x.a1=S.inv1%*%x.a1.bar
x.a2=S.inv2%*%x.a2.bar
p1=46/(46+19)
p2=19/(46+19)
for(t in 1:33){
x=validation[t,]
x.b1=x-x.b1.bar
x.b2=x-x.b2.bar
R.a1=c(rep(1,5))
R.b1=c(rep(1,5))
r.a1=0.7
```

```
r.b1=t(R.a1)%*%R.b1-r.a1
x.a1.star=R.a1+x.a1
x.b1.star=R.b1+x.b1
Vb1=0.291
T1=t(x.a1.star)%*%x.b1+r.b1-Vb1
Va1=T1+r.a1-t(R.a1)%*%x.b1.star
R.a1=c(1,3,6,2,1)
R.b1=c(rep(1,5))
r.a1=0.7
r.b1=t(R.a1)%*%R.b1-r.a1
x.a1.star=R.a1+x.a1
x.b1.star=R.b1+x.b1
Vb1=0.291
T1=t(x.a1.star)%*%x.b1+r.b1-Vb1
Va1=T1+r.a1-t(R.a1)%*%x.b1.star
R.a2=c(1,2,5,7,1)
R.b2=c(rep(2,5))
r.a2=1.4
r.b2=t(R.a2)%*%R.b2-r.a2
x.a2.star=R.a2+x.a2
x.b2.star=R.b2+x.b2
Vb2=0.52
T2=t(x.a2.star)%*%x.b2+r.b2-Vb2
Va2=T2+r.a2-t(R.a2)%*%x.b2.star
T.a1=t(x.a1.bar)%*%x.a1-2*Va1
T.b1=t(x.b1)%*%S.inv1%*%x.b1-2*Vb1
T.a2=t(x.a2.bar)%*%x.a2-2*Va2
```

```
T.b2=t(x.b2)%*%S.inv2%*%x.b2-2*Vb2
(T1=T.a1+T.b1)
(T2=T.a2+T.b2)
(d.1=-log(d.s1)/2-T1/2+log(p1))
(d.2=-log(d.s2)/2-T2/2+log(p2))
if(d.1>d.2) z2[t]=1
else z2[t]=2
}
mis.rate.qd.m=0
for(i in 1:23){
if(z2[i]!=1)
mis.rate.qd.m=mis.rate.qd.m+1
}
for(i in 24:33){
if(z2[i]==1)
mis.rate.qd.m=mis.rate.qd.m+1
}
mis.classification.qd.m2=mis.rate.qd.m/33
for(i in 1:33){
y=validation[i,]
dq1=-log(det(S1))/2-t(y-x1.mean)%*%solve(S1)%*%(y-x1.mean)/2
dq2=-log(det(S2))/2-t(y-x2.mean)%*%solve(S2)%*%(y-x2.mean)/2
dq1=dq1+log(p1)
dq2=dq2+log(p2)
if(dq1>dq2)
z.2[i]=1
else
```

```
z.2[i]=2
}
mis.rate.qd=0
for(i in 1:23){
if(z.2[i]!=1)
mis.rate.qd=mis.rate.qd+1
}
for(i in 24:33){
if(z.2[i]==1)
mis.rate.qd=mis.rate.qd+1
}
mis.classification.qd2=mis.rate.qd/33
```