Discriminant Analysis - 3rd TUTORIAL

The purpose of linear discriminant analysis (LDA) is to find the linear combinations of the original variables that gives the best possible separation between the groups in our data set. Linear discriminant analysis is also known as "canonical discriminant analysis", or simply "discriminant analysis".

Jobs data

A large international air carrier has collected data on employees in three different job classifications: 1) customer service personnel, 2) mechanics and 3) dispatchers. The director of Human Resources wants to know if these three job classifications appeal to different personality types. Each employee is administered a battery of psychological tests which includes measures of interest in outdoor activity, sociability and conservativeness.

The number of groups (G) is 3, and the number of variables is 3 (p = 3). The maximum number of useful discriminant functions that can separate the three jobs is less or equal to the minimum between G-1 and p, and so in this case it is less or equal to the minimum between 2 and 3, which is 2. Thus, we can find at most 2 useful discriminant functions to separate the employees by their jobs, using the 3 observed variables.

Load jobs data:

```
jobs <- read.csv("c:\temp\jobs.csv",sep=';',header=TRUE)
jobs[1:5,] # have a look at the data
> jobs[1:5,]
   OUTDOOR SOCIAL CONSERVATIVE JOB
1  10  22 5 1
2  14  17 6 1
3  19  33 7 1
4  14  29 12 1
5  14  25 7 1
```

Perform LDA from first principles:

```
n <- dim(jobs)[1]
p <- dim(jobs)[2]-1 # Subtract 1 because one of the columns specifies the job

# Separate the three groups of jobs
jobs.gp1 <- jobs[jobs$JOB==1,1:3]
jobs.gp2 <- jobs[jobs$JOB==2,1:3]
jobs.gp3 <- jobs[jobs$JOB==3,1:3]

# Need sample statistics
> n1 <- dim(jobs.gp1)[1]
> n2 <- dim(jobs.gp2)[1]
> n3 <- dim(jobs.gp3)[1]
> n1
[1] 85
> n2
[1] 93
> n3
[1] 66
```
# Group mean
> mean.gp1 <- apply(jobs.gp1,2,mean)
> mean.gp2 <- apply(jobs.gp2,2,mean)
> mean.gp3 <- apply(jobs.gp3,2,mean)
> mean.tot <- (mean.gp1*n1+mean.gp2*n2+mean.gp3*n3)/(n1+n2+n3)

> mean.gp1
OUTDOOR SOCIAL CONSERVATIVE
12.517647 24.223529 9.023529

> mean.gp2
OUTDOOR SOCIAL CONSERVATIVE
18.53763 21.13978 10.13978

> mean.gp3
OUTDOOR SOCIAL CONSERVATIVE
15.57576 15.45455 13.24242

> mean.tot
OUTDOOR SOCIAL CONSERVATIVE
15.63934 20.67623 10.59016

# Within group covariance matrices
> S.gp1 <- var(jobs.gp1)
> S.gp1
OUTDOOR SOCIAL CONSERVATIVE
OUTDOOR 21.609804 4.418627 -2.548039
SOCIAL 4.418627 18.794678 2.506583
CONSERVATIVE -2.548039 2.506583 9.880392

> S.gp2 <- var(jobs.gp2)
> S.gp2
OUTDOOR SOCIAL CONSERVATIVE
OUTDOOR 12.707807 -1.575970 2.543595
SOCIAL -1.575970 20.708519 0.175900
CONSERVATIVE 2.543595 0.175900 10.512856

> S.gp3 <- var(jobs.gp3)
> S.gp3
OUTDOOR SOCIAL CONSERVATIVE
OUTDOOR 16.894172 1.672727 0.689044
SOCIAL 1.672727 14.190209 0.134266
CONSERVATIVE 0.689044 0.134266 13.632634

> W <- ((n1-1)*S.gp1 + (n2-1)*S.gp2 + (n3-1)*S.gp3)/(n1+n2+n3-3)

> W
OUTDOOR SOCIAL CONSERVATIVE
OUTDOOR 16.939680 1.389637 0.268727
SOCIAL 1.389637 12.833999 0.977024
CONSERVATIVE 0.268727 0.977024 11.133846

> W.inv <- solve(W)

> B <- diag((n1*(mean.gp1-mean.tot)%*% t(mean.gp1-mean.tot))+ (n2*(mean.gp2-mean.tot)%*% t(mean.gp2-mean.tot))+(n3*(mean.gp3-mean.tot)%*% t(mean.gp3-mean.tot)))

> B
OUTDOOR SOCIAL CONSERVATIVE
[1,] 804.8996 -397.1973 141.5855
[3,] 141.5855 -702.9200 345.8797

> A <- W.inv %*% B # Calculating the canonical matrix
> eigen(A)$values
[1]  1.302035e+02  3.862069e+01 -3.384425e-15

> eigen(A)$vectors
[,1]       [,2]       [,3]
[1,]  0.3470958 -0.9130164  0.06612384
[2,] -0.7331079 -0.2019912  0.45123092
[3,]  0.5848738  0.3544018  0.88995409

Since we need to separate 3 groups on which 3 variables have been observed, the rank of the matrix $W^{-1}B$ is at most equal to 2; in fact there are two not null eigenvalues and two corresponding eigenvectors which are the discriminant coordinates or canonical variables.

The first discriminant direction is a linear contrast between interest in social activity and interest in outdoor or conservative activities. This means that the 3 employee categories mainly differ as far as the social component is concerned.

On the second discriminant direction outdoor activities and conservative activities have the largest effect. This direction separates employee categories according to the employees’ attitudes towards static activities rather than more dynamic ones.

The discriminant directions $A$ are unique up to a change of sign. The exact multiple of $A$ returned by an algorithm depends on the normalizing constraint it imposes. The eigenvectors obtained through the eigen R function have unit norm. The $A$ vectors obtained through the lda R function that we will describe in the following are obtained by imposing that the within group variance along $A$ is equal to 1.

> a.vect<-eigen(A)$vectors[,1:2]

The linear combination is then $Y = XA$ where $X$ is now the $n \times p$ data matrix and $A$ is the $p \times 2$ matrix whose columns are the eigenvectors of $W^{-1}B$ corresponding to the non null eigenvalues.

> Y<-as.matrix(jobs[,1:3])%*%a.vect

Thus, each unit is represented by a $d$-dimensional vector (where $d=\text{rank } W^{-1}B$).

A new observation $x_0 = (x_{01},...,x_{0p})$, whose group membership is unknown, is then projected on the $d$ dimensional discriminant subspace defined by the eigenvectors of $W^{-1}B$ corresponding to non zero eigenvalues, thus obtaining a new $d$-dimensional vector $y_0$. The averages of the $g$ groups are also projected on the same subspace: we obtain $\bar{y}_1,\bar{y}_2,...,\bar{y}_G$ and $y_0$ is allocated to the group whose average it is closest (in terms of Euclidean distance) in the discriminant subspace.

> y.mean.gp1<-mean.gp1%*%a.vect
> y.mean.gp2<-mean.gp2%*%a.vect
> y.mean.gp3<-mean.gp3%*%a.vect
>
> y.mean.gp1
[,1]       [,2]
[1,]  -8.136013  -13.1238

> y.mean.gp2
[,1]       [,2]
[1,]  -3.132914  -17.60166
The following for-cycle returns a matrix containing the Euclidean distances between each 'new' point and the mean vector of the canonical variables in the three groups and the closest group according to the evaluated distances:

```r
dist.group<-matrix(0,nrow=nrow(Y),4)
colnames(dist.group)<-c("dist.gp1","dist.gp2","dist.gp3","Group")
for (i in 1:nrow(Y)){
dist.gp1<-sqrt(sum((Y[i,]-y.mean.gp1)^2)) #Euclidean distance
dist.gp2<-sqrt(sum((Y[i,]-y.mean.gp2)^2)) #Euclidean distance
dist.gp3<-sqrt(sum((Y[i,]-y.mean.gp3)^2)) #Euclidean distance
group<- which.min(c(dist.gp1,dist.gp2,dist.gp3))
dist.group[i,]<-c(dist.gp1,dist.gp2,dist.gp3,group)
}
```

The same result can be obtained by working on the original observation and by using the Mahalanobis distance. The following for-cycle returns a matrix containing the Mahalanobis distances between each point and the mean vector of each group and the closest group according to the evaluated distances:

```r
> dist.group2<-matrix(0,nrow=nrow(jobs),4)
> colnames(dist.group)<-c("dist.gp1","dist.gp2","dist.gp3","Group")
> jobs.m<-as.matrix(jobs[,1:3])
> for (i in 1:nrow(jobs)){
+ dist.gp1<-t(jobs.m[i,]-mean.gp1)%*%W.inv%*%(jobs.m[i,]-mean.gp1) #mahalanobis
+ dist.gp2<-t(jobs.m[i,]-mean.gp2)%*%W.inv%*%(jobs.m[i,]-mean.gp2) #mahalanobis
+ dist.gp3<-t(jobs.m[i,]-mean.gp3)%*%W.inv%*%(jobs.m[i,]-mean.gp3) #mahalanobis
+ group<- which.min(c(dist.gp1,dist.gp2,dist.gp3))
+ dist.group2[i,]<-c(dist.gp1,dist.gp2,dist.gp3,group)
+ }
```

The ranking of the two approaches is actually the same.

The following commands produce the plot of the units on the plane defined by the canonical variables:

```r
plot(Y,type="n",xlab="First Canonical Variable",ylab="Second Canonical variable")
points(Y[which(jobs[,4]==1),],pch=21,col="black",bg="black")
points(Y[which(jobs[,4]==2),],pch=24,col="red",bg="red")
points(Y[which(jobs[,4]==3),],pch=22,col="blue",bg="blue")
legend(1,1,c("CustomerService","Mechanic","Dispatcher"),col=c("black","red","blue"),pch=c(21,24,22))
```
In order to check the goodness of our classification we can compute a cross-table:

```r
> misclassification <- table(jobs[,4], dist.group[,4])
> misclassification
   1  2  3
1 70 11  4
2 16 62 15
3  3 12 51
> (16+3+12+11+4+15)/sum(misclassification)
[1] 0.25

25% of the respondents have been misclassified.

The same result can be obtained by using the R function `lda`, contained in the library `MASS`:

```r
> library(MASS)
> jobs.lda <- lda(jobs[,1:3], jobs[,4])
> jobs.lda
Call:
  lda(jobs[, 1:3], jobs[, 4])

Prior probabilities of groups:
1         2         3
0.3483607 0.3811475 0.2704918

Group means:

<table>
<thead>
<tr>
<th>Class</th>
<th>OUTDOOR</th>
<th>SOCIAL</th>
<th>CONSERVATIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12.51765</td>
<td>24.22353</td>
<td>9.023529</td>
</tr>
<tr>
<td>2</td>
<td>18.53763</td>
<td>21.13978</td>
<td>10.139785</td>
</tr>
<tr>
<td>3</td>
<td>15.57576</td>
<td>15.45455</td>
<td>13.242424</td>
</tr>
</tbody>
</table>

Coefficients of linear discriminants:

<table>
<thead>
<tr>
<th></th>
<th>LD1</th>
<th>LD2</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTDOOR</td>
<td>0.09198065</td>
<td>-0.22501431</td>
</tr>
<tr>
<td>SOCIAL</td>
<td>-0.19427415</td>
<td>-0.04978105</td>
</tr>
<tr>
<td>CONSERVATIVE</td>
<td>0.15499199</td>
<td>0.08734288</td>
</tr>
</tbody>
</table>

Proportion of trace:

<table>
<thead>
<tr>
<th></th>
<th>LD1</th>
<th>LD2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD1</td>
<td>0.7712</td>
<td></td>
</tr>
<tr>
<td>LD2</td>
<td>0.2288</td>
<td></td>
</tr>
</tbody>
</table>
The A matrix (here indicated as 'Coefficients of linear discriminants') yielded by the \texttt{lda} R-function is a bit different from what we obtained with the single value decomposition of $W^{-1}B$ matrix: here it is normalized so that the within groups covariance matrix is spherical.

$$A^{TW}A = \Psi_{\text{diag}}$$

$$\Psi^{-1/2}A^{TW}A\Psi^{-1/2} = I$$

By multiplying A vectors for $\Psi^{-1/2}$ we find again the coefficients of the linear discriminants yield by \texttt{lda}:

```r
> psi <- t(a.vect) %*% W %*% a.vect
> a.vect %*% solve(psi)^(1/2)

[,1]        [,2]
[1,]  0.09198064 -0.22501431
[2,] -0.19427415 -0.04978106
[3,]  0.15499199  0.08734288
```

And the opposite is true, i.e. if we divide the coefficients of the linear discriminants yield by \texttt{lda} by $\Psi^{-1/2}$ we obtain the first principles:

```r
> coef(jobs.lda) %*% solve(solve(psi)^(1/2))

[,1]       [,2]
OUTDOOR    0.3470958 -0.9130164
SOCIAL     -0.7331079 -0.2019912
CONSERVATIVE  0.5848737  0.3544018
```

Please note that this equivalence holds as far as the conventional definition of matrix $B$ is considered, i.e. groups are weighted by their size in the dataset. The \texttt{lda} function also allows to consider a covariance matrix weighted by the prior probabilities of the classes if these are specified; otherwise observed frequencies are used.

**Banknotes dataset**

Suppose you are the manager of a bank and you have the problem of discriminating between genuine and counterfeit banknotes. Here you see a banknote worth 1000 Austrian schillings (about 72.67 Euro). You are measuring several distances on the banknote and the width and height of it. Measuring these values of about 100 genuine and 100 counterfeit banknotes, you will finally end up with a table of 200 observations with several variables.
Let's consider here dataset `notes` which contains data on Swiss francs. The aim is to set up a model which is capable of discriminating between genuine and counterfeit money. Banknotes BN1 to BN100 are genuine, all others are counterfeit.

```r
> notes<-read.csv("banknotes.csv",sep=";",header=TRUE)
> notes[1:3,]
   Length  Left Right Bottom  Top Diagonal
1  214.8 131.0 131.1    9.0  9.7    141.0
2  214.6 129.7 129.7    8.1  9.5    141.7
3  214.8 129.7 129.7    8.7  9.6    142.2
```

Create the group membership vector:

```r
group <- c(rep("Genuine",100),rep("Counterfeit",100))
```

Perform LDA from first principles:

```r
> lda.notes <- lda(notes,group)
> lda.notes
Call: lda(notes, group)
Prior probabilities of groups:
  Counterfeit     Genuine
     0.5         0.5
Group means:
                  Length  Left Right Bottom  Top Diagonal
Counterfeit  214.823 130.300 130.193 10.530 11.133  139.450
Genuine     214.969 129.943 129.720  8.305 10.168  141.517
```

Coefficients of linear discriminants:

```r
            LD1
Length 0.00501113
Left  0.832432523
Right -0.848993093
Bottom 1.117335597
Top   -1.178884468
Diagonal 1.556520967
```

Since we want to separate two groups in which 5 variables have been observed, the rank of matrix $W^{-1}B$ is equal to 1; there is one not null eigenvalue and one corresponding eigenvector which is the discriminant coordinate or canonical variable.

The discriminant direction is a linear contrast between the length of the diagonal and the length of the left rim versus height of the right, bottom and top rims. This means that genuine and counterfeit banknotes mainly differ for the length of the diagonal (which summarizes the total dimensions of the notes) and for the length of the left rim with respect to all the other ones.

The linear combination is then $Y = XA$ where $X$ is now the $n \times p$ data matrix and $A$ is the $p \times 2$ matrix whose columns are the eigenvectors of $W^{-1}B$ corresponding to the non null eigenvalues.

Thus, each unit is represented by a one-dimensional vector.

```r
> prev<-predict(lda.notes,dimen=1)
> Y<-prev$x
```
Now we can tabulate the obtained classifications with the true ones:

```r
crosstab <- table(group, prev$class)
crosstab
```

<table>
<thead>
<tr>
<th></th>
<th>Counterfeit</th>
<th>Genuine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Counterfeit</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>Genuine</td>
<td>1</td>
<td>99</td>
</tr>
</tbody>
</table>

The classification is almost perfect, since there is only one units that is misclassified.

```r
plot(Y, type="n", ylab="First Canonical variable", xlim=c(0,100))
points(Y[1:100,], pch=21, col="black", bg="black")
points(Y[101:200,], pch=24, col="red", bg="red")
legend(1,1,c("Genuine","Counterfeit"), col=c("black","red"), pch=c(21,24))
```

Crops data

In this example, the remote-sensing data are used. In crops data set, the observations are grouped into five crops: clover, corn, cotton, soybeans, and sugar beets. Four measures called V2 through V5 make up the descriptive variables.

```r
crops <- read.table("crops.txt", sep='')
crops[1:5,] # have a look at the data
```

<table>
<thead>
<tr>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
<th>V5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Corn</td>
<td>16</td>
<td>27</td>
<td>31</td>
<td>33</td>
</tr>
<tr>
<td>2 Corn</td>
<td>15</td>
<td>23</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>3 Corn</td>
<td>16</td>
<td>27</td>
<td>27</td>
<td>26</td>
</tr>
<tr>
<td>4 Corn</td>
<td>18</td>
<td>20</td>
<td>25</td>
<td>23</td>
</tr>
<tr>
<td>5 Corn</td>
<td>15</td>
<td>15</td>
<td>31</td>
<td>32</td>
</tr>
</tbody>
</table>

```r
n <- dim(crops)[1]
n
[1] 36
p <- dim(crops)[2]-1 # Subtract 1 because one of the columns specifies the
p
[1] 4
```
# Training set and Test set
> set<-1:nrow(crops)
> ftest<-sample(set,8)
> ftrain<-set[-ftest]
> ftest
[33] 34 35
> ftrain
[1]  3  4  5  6  7  8  9 10 12 14 15 16 18 19 21 22 23 24 25 26 27 28 30 31 32
> crops.train.lda <- lda(crops[ftrain,-1], crops[ftrain,1])
> crops.pred <- predict(crops.train.lda,crops[ftrain,-1])
> table(crops[ftrain,1],crops.pred$class)

<table>
<thead>
<tr>
<th></th>
<th>Clover</th>
<th>Corn</th>
<th>Cotton</th>
<th>Soybeans</th>
<th>Sugarbeets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clover</td>
<td>4</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Corn</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Cotton</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Soybeans</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Sugarbeets</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

> misc.err.rate<-(2+1+2+1+2+1)/28
> misc.err.rate
[1] 0.4285714

# About the 43% of observations in training set were misclassified.
# Let's see how good is classification in the test set
> crops.test.pred <- predict(crops.train.lda,crops[ftest,-1])
> table(crops[ftest,1],crops.test.pred$class)

<table>
<thead>
<tr>
<th></th>
<th>Clover</th>
<th>Corn</th>
<th>Cotton</th>
<th>Soybeans</th>
<th>Sugarbeets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clover</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Corn</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Cotton</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Soybeans</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Sugarbeets</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

> # Here over 8 crops 5 were misclassified about the 62.5%. Maybe this is due to
| the too small size of the sample that makes the classification very weak (the
| misclassification error rate is pretty high even in the training set).