Linear Discriminant Analysis and Logistics Regression

Sanjaya Mananage

(a) linear discriminant function analysis

Create a new variable "Admission_Status" based on the criteria, if Chance_of_Admit >= 0.7 - Admit(1) and otherwise Do not admit (0).

```
Admission<-read.csv("Admission_Predict.csv")
Admission<-Admission[,-1]
Admission$Admission_Status<-ifelse(Admission$Chance_of_Admit>=0.7,1,0)
```

Linear discriminant function analysis to classify future applicant as admit or do not admit on other variables.

```
attach(Admission)

library(MASS)
dis<-lda(Admission_Status ~ GRE_Score+ TOEFL_Score +University_Rating + SOP + LOR + CGPA + Research,
         data=Admission,prior=c(1/2,1/2))
dis
```

```
## Call:
## lda(Admission_Status ~ GRE_Score + TOEFL_Score + University_Rating +
##     SOP + LOR + CGPA + Research, data = Admission, prior = c(1/2,
##     1/2))
##
## Prior probabilities of groups:
##   0   1
## 0.5 0.5
##
## Group means:
##   GRE_Score TOEFL_Score University_Rating      SOP      LOR     CGPA  Research
## 0  307.0131    102.5882          2.261438 2.741830 2.875817 8.087974 0.2483660
## 1  322.8745    110.3968          3.599190 3.807692 3.809717 8.915425 0.7327935
##
## Coefficients of linear discriminants:
##                           LD1
## GRE_Score          0.04859038
## TOEFL_Score        0.01617842
## University_Rating  0.14987272
## SOP               -0.08706974
## LOR                0.18329397
## CGPA               0.88150770
## Research           0.39318518
```

```r
names(dis)
```

```
##  [1] "prior"   "counts"  "means"   "scaling" "lev"     "svd"     "N"
##  [8] "call"    "terms"   "xlevels"
```

```r
dis$scaling #coefficients are saved here
```

```
##                          LD1
## GRE_Score          0.04859038
## TOEFL_Score        0.01617842
## University_Rating  0.14987272
## SOP               -0.08706974
## LOR                0.18329397
## CGPA               0.88150770
## Research           0.39318518
```

Classification rule:

$\hat{a_1} = [0.04859038, 0.01617842, 0.14987272, -0.08706974, 0.18329397, 0.88150770, 0.39318518]$

Group means:

$\bar{x}_k$	GRE_Score	TOEFL_Score	University_Rating	SOP	LOR	CGPA	Research
0	307.0131	102.5882	2.261438	2.741830	2.875817	8.087974	0.2483660
1	322.8745	110.3968	3.599190	3.807692	3.809717	8.915425	0.7327935

for k^{th} group compute $\sum_{j=1}^{2}(\hat{y_j} - \bar{y}_{kj})^2 = \sum_{j=1}^{2}(\hat{\mathbf{a}}_j'\mathbf{x} - \hat{\mathbf{a}}_j'\bar{\mathbf{x}}_k)^2$

where k=0,1

The group that has minimum value of the above sum of squared distance is assigned x.

Suppose a new application comes with GRE Score = 310, TOEFL Score = 110, University Rating = 3, Statement of Purpose = 3, Letter of Recommendation Strength = 3, Undergraduate GPA = 8.5 and Research Experience = 1.

The admission status for this new applicant.

```r
#Observations on  new bulls that need to be classified
newdata<-data.frame(320,110,3,3,3,8.5,1)
colnames(newdata)<-colnames(Admission[-c(8:9)])
newdata
```

```
##   GRE_Score TOEFL_Score University_Rating SOP LOR CGPA Research
## 1       320         110                 3   3   3  8.5        1
```

```r
# prediction of classes for the new observations
predict(dis,newdata=newdata)$class
```

```
## [1] 1
## Levels: 0 1
```

According to the classification rule the new observation $x = (320, 110, 3, 3, 3, 8.5, 1)$ is classified to the group 1 (Admit)

The plug-in (APER) and leave-one-out (AER) estimates of misclassification rates

```
cat("##APER\n")
```

```
## ##APER
```

```
pred.group1<-predict(dis,method="plug-in")$class
table(Admission_Status, pred.group1)
```

```
##                  pred.group1
## Admission_Status   0   1
##                0 139  14
##                1  49 198
```

```
APER<-(49+14)/400
APER
```

```
## [1] 0.1575
```

```
cat("\n##AER\n")
```

```
##
## ##AER
```

```
dis2<-lda(Admission_Status ~ GRE_Score+ TOEFL_Score +University_Rating + SOP + LOR + CGPA + Research,
          data=Admission,prior=c(1/2,1/2), CV=TRUE)
table(Admission_Status, dis2$class)
```

```
##
## Admission_Status   0   1
##                0 138  15
##                1  52 195
```

```
AER<-(52+15)/400
AER
```

```
## [1] 0.1675
```

Plug-in(APER) = 0.1575

Leave-one-out(AER) = 0.1675

AER is greater than APER.

(b) LDA with three class variable

Now create the second new variable "Admission_Status2" with three classes based on the criteria, if (Chance_of_Admit >= 0.8) - Admit (1) else if (0.8 > Chance_of_Admit >= 0.5) - Borderline (2) and otherwise Do not admit (3).

```
#Admission<-read.csv("Admission_Predict.csv")
#Admission<-Admission[,-1]
Admission$Admission_Status2<-ifelse(Admission$Chance_of_Admit>=0.8,1,
                                    ifelse(Admission$Chance_of_Admit>=0.5,2,3))
table(Admission$Admission_Status2)
```

```
##
##   1   2   3
## 128 239  33
```

```
attach(Admission)

##b
library(MASS)
dis<-lda(Admission_Status2 ~ GRE_Score+ TOEFL_Score +University_Rating + SOP + LOR + CGPA + Research,
         data=Admission,prior=c(1/3,1/3,1/3))
dis
```

```
## Call:
## lda(Admission_Status2 ~ GRE_Score + TOEFL_Score + University_Rating +
##     SOP + LOR + CGPA + Research, data = Admission, prior = c(1/3,
##     1/3, 1/3))
##
## Prior probabilities of groups:
##         1         2         3
## 0.3333333 0.3333333 0.3333333
##
## Group means:
##   GRE_Score TOEFL_Score University_Rating      SOP      LOR     CGPA  Research
## 1  328.3281   113.58594          4.148438 4.242188 4.132812 9.241953 0.9296875
## 2  312.6736   105.19665          2.661088 3.073222 3.228033 8.378033 0.3933054
## 3  302.0606    99.48485          2.060606 2.500000 2.439394 7.704545 0.1818182
##
## Coefficients of linear discriminants:
##                            LD1           LD2
## GRE_Score          -0.009260823  0.0044667328
## TOEFL_Score        -0.072394662 -0.0008867583
## University_Rating  -0.014757895  0.9563982049
## SOP                 0.203656079  0.6048251753
## LOR                -0.230082565 -0.8935530145
## CGPA               -1.907665862 -1.5892951724
## Research           -0.458033441  0.9968816146
##
## Proportion of trace:
##    LD1    LD2
## 0.9798 0.0202
```

```
names(dis)
```

```
##  [1] "prior"   "counts"  "means"   "scaling" "lev"     "svd"     "N"
##  [8] "call"    "terms"   "xlevels"
```

```
dis$scaling #coefficients are saved here
```

```
##                            LD1           LD2
## GRE_Score          -0.009260823  0.0044667328
## TOEFL_Score        -0.072394662 -0.0008867583
## University_Rating  -0.014757895  0.9563982049
## SOP                 0.203656079  0.6048251753
## LOR                -0.230082565 -0.8935530145
## CGPA               -1.907665862 -1.5892951724
## Research           -0.458033441  0.9968816146
```

```r
#Observations on  new bulls that need to be classified
newdata<-data.frame(320,110,3,3,3,8.5,1)
colnames(newdata)<-colnames(Admission[-c(8:10)])
newdata
```

```
##   GRE_Score TOEFL_Score University_Rating SOP LOR CGPA Research
## 1       320         110                 3   3   3  8.5        1
```

```r
# prediction of classes for the new observations
predict(dis,newdata=newdata)$class
```

```
## [1] 2
## Levels: 1 2 3
```

Two Classification rules:

$\hat{a_1} = [-0.009260823, -0.072394662, -0.014757895, 0.203656079, -0.230082565, -1.907665862, -0.458033441]$

$\hat{a_2} = [0.0044667328, -0.0008867583, 0.9563982049, 0.6048251753, -0.8935530145, -1.5892951724, 0.9968816146]$

Group means:

$\bar{x}_k$	GRE_Score	TOEFL_Score	University_Rating	SOP	LOR	CGPA	Research
1	328.3281	113.58594	4.148438	4.242188	4.132812	9.241953	0.9296875
2	312.6736	105.19665	2.661088	3.073222	3.228033	8.378033	0.3933054
3	302.0606	99.48485	2.060606	2.500000	2.439394	7.704545	0.1818182

for k^{th} group compute $\sum_{j=1}^{2}(\hat{y_j} - \bar{y}_{kj})^2 = \sum_{j=1}^{2}(\hat{\mathbf{a}}_j'\mathbf{x} - \hat{\mathbf{a}}_j'\bar{\mathbf{x}}_k)^2$

where k=1,2,3

The group that has minimum value of the above sum of squared distance is assigned x.

According to the classification rule the above new observation $x = (320, 110, 3, 3, 3, 8.5, 1)$ is classified to the group 2 (Borderline)

The plug-in (APER) and leave-one-out (AER) estimates of misclassification rates for two LDAs

```r
cat("##APER\n")
```

```
## ##APER
```

```r
pred.group1<-predict(dis,method="plug-in")$class
table(Admission_Status2, pred.group1)
```

```
##                   pred.group1
## Admission_Status2   1   2   3
##                 1 119   9   0
##                 2  31 169  39
##                 3   0   6  27
```

```r
APER<-(9+31+39+6)/400
APER
```

```
## [1] 0.2125
```

```
cat("\n##AER\n")
```

```
##
## ##AER
```

```
dis2<-lda(Admission_Status2 ~ GRE_Score+ TOEFL_Score +University_Rating + SOP + LOR + CGPA + Research,
          data=Admission,prior=c(1/3,1/3,1/3), CV=TRUE)
table(Admission_Status2, dis2$class)
```

```
##
## Admission_Status2   1   2   3
##                 1 119   9   0
##                 2  32 167  40
##                 3   0   6  27
```

```
AER<-(9+32+40+6)/400
AER
```

```
## [1] 0.2175
```

Plug-in(APER) = 0.2125

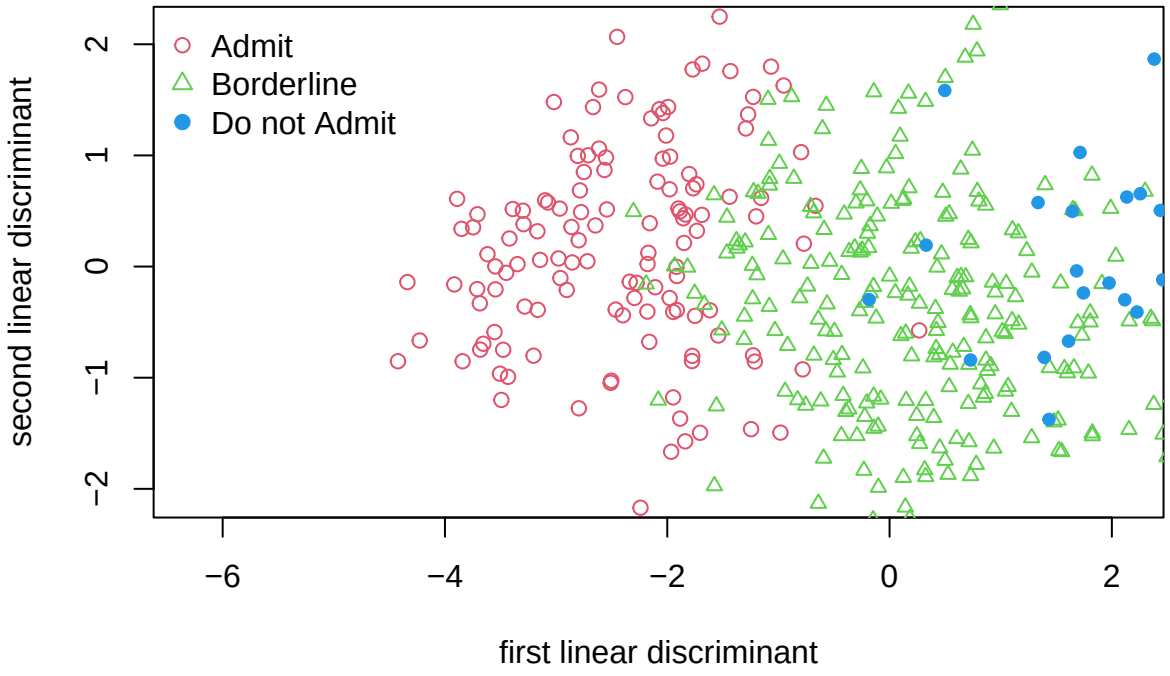
Leave-one-out(AER) = 0.2175

AER is greater than APER.

Scatterplot of the first two discriminant scores by labeling different Admission status with different symbols and colors.

```
dis.ld<-predict(dis)$x
dis.ld<-data.frame(cbind(dis.ld,Admission_Status2=Admission$Admission_Status2))
eqscplot(dis.ld[dis.ld$Admission_Status2==1, 1], dis.ld[dis.ld$Admission_Status2==1, 2],
         xlab = "first linear discriminant", ylab = "second linear discriminant",col=2)
points(dis.ld[dis.ld$Admission_Status2==2, 1], dis.ld[dis.ld$Admission_Status2==2, 2],
       pch = 2, cex = 0.8, col = 3)
points(dis.ld[dis.ld$Admission_Status2==3, 1], dis.ld[dis.ld$Admission_Status2==3, 2],
       pch = 19, cex = 0.8, col = 4)

legend('topleft',legend=c("Admit", "Borderline", "Do not Admit"),
       pch=c(1,2,19), col=c(2,3,4), bty="n")
```



(c) Classification rule using logistic regression.

```
attach(Admission)
fit1 <- glm(Admission_Status ~ GRE_Score+ TOEFL_Score +University_Rating + SOP + LOR + CGPA + Research,
            family=binomial, data=Admission)
summary(fit1)
```

```
##
## Call:
## glm(formula = Admission_Status ~ GRE_Score + TOEFL_Score + University_Rating +
##     SOP + LOR + CGPA + Research, family = binomial, data = Admission)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.63991  -0.37047   0.06996   0.39138   2.29484
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -61.58287    8.31492  -7.406  1.3e-13 ***
## GRE_Score           0.10460    0.02871   3.643 0.000269 ***
## TOEFL_Score         0.07713    0.05431   1.420 0.155565
## University_Rating   0.44936    0.23963   1.875 0.060757 .
## SOP                -0.39012    0.26543  -1.470 0.141625
## LOR                 0.56873    0.27708   2.053 0.040115 *
## CGPA                2.22765    0.61663   3.613 0.000303 ***
## Research            0.56866    0.34454   1.650 0.098845 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 532.22  on 399  degrees of freedom
## Residual deviance: 244.53  on 392  degrees of freedom
## AIC: 260.53
##
## Number of Fisher Scoring iterations: 6
```

```
newdata<-data.frame(320,110,3,3,3,8.5,1)
colnames(newdata)<-colnames(Admission[-c(8:10)])
newob<- predict(fit1, newdat= newdata,type="response")
newob
```

```
##         1
## 0.8534825
```

```
cat("##APER")
```

```
## ##APER
```

```
#Plug-in estimate
table(Admission_Status,(predict(fit1, type="response")>0.5))
```

```
##
## Admission_Status FALSE TRUE
##                0   129   24
##                1    25  222
```

```
APER<-(25+24)/400
APER
```

```
## [1] 0.1225
```

```
cat("##AER")
```

```
## ##AER
```

```
#Cross-Validation (Leave-one-out method)
newpred <- numeric(length(Admission_Status))

for (i in 1:length(Admission_Status)){
newdat <- Admission[-i,]
newfit <- glm(Admission_Status ~ GRE_Score+ TOEFL_Score +University_Rating + SOP + LOR + CGPA + Research,
              family=binomial, data=newdat)
newpred[i] <- predict(newfit, newdat=Admission[i,-c(8:10)],  type="response")

}

table(Admission_Status,(newpred>0.5))
```

```
##
## Admission_Status FALSE TRUE
##                0   126   27
##                1    30  217
```

```r
AER<-(30+27)/400
AER
```

```
## [1] 0.1425
```

If $\hat{p}(x_0) > 0.5$ then classify x_0 to 1(Admit) otherwise Do not admit.

$\hat{p}(x_0) = 0.8534825 > .5$. So we assign new observation to 1(Admit).

Plug-in(APER) = 0.1225

Leave-one-out(AER) = 0.1425

AER is greater than APER.