ANALISIS DISCRIMINANTE

Mathias Bourel

12/10/2019

1- Ejemplo1: Para entender el análisis discriminante

Datos de dos especies de mosquitos al que se le mide longitud trompa y longitud del ala.

Preparación de la base:

```
X=matrix(c(138,164,140,170,124,172,136,174,138,182,148,182,154,182,138,190,156,
208,114,178,120,186,118,196,130,196,126,200,128,200),15,2,byrow=T)

Y=c(rep(0,9),rep(1,6))


X=cbind(X,Y)
X
```

```
##               Y
##  [1,] 138 164 0
##  [2,] 140 170 0
##  [3,] 124 172 0
##  [4,] 136 174 0
##  [5,] 138 182 0
##  [6,] 148 182 0
##  [7,] 154 182 0
##  [8,] 138 190 0
##  [9,] 156 208 0
## [10,] 114 178 1
## [11,] 120 186 1
## [12,] 118 196 1
## [13,] 130 196 1
## [14,] 126 200 1
## [15,] 128 200 1
```

```
colnames(X)=c("Trompa","Ala","Cat")
X_1=X[1:9,-3]
```

Matrices de variances y covarianzas y eje discrimiante

```
X_1
```

```
##       Trompa Ala
##  [1,]    138 164
##  [2,]    140 170
##  [3,]    124 172
##  [4,]    136 174
##  [5,]    138 182
##  [6,]    148 182
##  [7,]    154 182
##  [8,]    138 190
##  [9,]    156 208
```

```r
S_1=var(X_1)

X_2=X[10:15,-3]
S_2=var(X_2)

S=1/13*(8*S_1+5*S_2)
solve(S)
```

```
##             Trompa         Ala
## Trompa  0.02354207 -0.01169284
## Ala    -0.01169284  0.01328076
```

```r
d=colMeans(X_1)-colMeans(X_2)

w=solve(S)%*%as.matrix(d)
```

Se obtiene que el eje es

$$z = w'x = 0.582x_1 - 0.382x_2$$

La distancia de Mahalanobis es:

```r
D2=t(as.matrix(d))%*%solve(S)%*%as.matrix(d)
D=sqrt(D2)
```

Vamos a comparar ahora con la función LDA del paquete MASS:

```r
library(MASS)
l=lda(Cat~.,data=as.data.frame(X))
l$scaling
```

```
##                LD1
## Trompa -0.14781385
## Ala     0.09659929
```

```r
w
```

```
##              [,1]
## Trompa  0.5823644
## Ala    -0.3805867
```

```r
#Encontramos lo mismo:
l$scaling/w
```

```
##               LD1
## Trompa -0.2538168
## Ala    -0.2538168
```

```r
q=qda(Cat~.,data=as.data.frame(X))
q
```

```
## Call:
## qda(Cat ~ ., data = as.data.frame(X))
##
## Prior probabilities of groups:
##   0   1
## 0.6 0.4
##
## Group means:
##     Trompa      Ala
```

```
## 0 141.3333 180.4444
## 1 122.6667 192.6667
```

```
ls(q)
```

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

Clasificamos:

```
dis=t(w)%*%((colMeans(X_1)+colMeans(X_2))/2)
#Asignaremos la clase 0 si z>dis y la clase 1 si z<dis

    #Por ejemplo
    x=as.matrix(c(118,196))
    z=t(w)%*%x
    z
```

```
##           [,1]
## [1,] -5.875997
```

```
    x=as.matrix(c(138,164))
    z=t(w)%*%x
    z
```

```
##          [,1]
## [1,] 17.95007
```

```
    x=as.matrix(c(130,190))
    z=t(w)%*%x
    z
```

```
##          [,1]
## [1,] 3.395896
```

2- Ejemplo 2: analisis discriminante sobre dos tipos raciales (De J.M Marin, universidad Carlos III, Madrid.)

Esos datos correponden a dos tipos raciales diferentes en los que se practicaron diferentes medidas de longitudes y anchos de craneo y cara.

```
datos=source("datostibet.dat")
datos=datos$value
```

Hagamos la construcción paso a paso:

```
attach(datos)
    Tibet1 <- datos[Type==1,c(1:5)]
    Tibet2 <- datos[Type==2,c(1:5)]

    n1=nrow(Tibet1)
    n2=nrow(Tibet2)
    n=n1+n2
    p=ncol(Tibet1)

# Vector de medias de las dos poblaciones:
    mean1=apply(Tibet1, 2, mean)
```

```
    mean2=apply(Tibet2, 2, mean)

# Estimaci?n de la matriz de varianzas covarianzas de toda la poblaci?n:
    S=((n1-1)*var(Tibet1) + (n2-1)*var(Tibet2))/(n-2)
    inv.S=solve(S)

# Eje discrimante:
    w=inv.S %*% (mean1-mean2)
```

Comparamos con la función MASS del paquete MASS:

```
dis=lda(Type ~ Length + Breadth + Height + Fheight + Fbreadth,
           data=datos, prior=c(0.5,0.5))

# Vector de la función discriminante:
    dis$scaling
```

```
##                   LD1
## Length    0.047726591
## Breadth  -0.083247929
## Height   -0.002795841
## Fheight   0.094695000
## Fbreadth  0.094809401
```

```
#Volvemos a ver que encontramos un vector colineal:
    dis$scaling / w
```

```
##                 LD1
## Length   -0.5344124
## Breadth  -0.5344124
## Height   -0.5344124
## Fheight  -0.5344124
## Fbreadth -0.5344124
```

Clasificación de una nueva observación:

```
newdata <- rbind(c(171, 140.5, 127, 69.5, 137),
                 c(179, 132, 140, 72, 138.5))

# hallo el punto de corte en la direcci?n discriminante para ambos grupos:
    lda.1 <- mean1 %*% w
    lda.1
```

```
##           [,1]
## [1,] -28.71277
```

```
    lda.2 <- mean2 %*% w
    lda.2
```

```
##           [,1]
## [1,] -32.21421
```

```
    puntodecorte=(lda.1 + lda.2)/2
    puntodecorte
```

```
##           [,1]
## [1,] -30.46349
```

La regla de clasificación es la siguiente: el individuo será de tipo 1 si el score encontrado es mayor que

-30.46349, y de tipo 2 en otro caso. Calculo el score para newdata

```
    newdata %*% w
```

```
##           [,1]
## [1,] -29.34069
## [2,] -32.02032
```

la primera observación es de tipo 1 y la segunda observación es de tipo 2.

Clasificación de una nueva observación usando MASS

```
# necesitamos que newdata sea un data.frame
    dimnames(newdata)=list(NULL, c("Length","Breadth","Height",
                                   "Fheight","Fbreadth"))
    newdata.frame=data.frame(newdata)

# usamos la funci?n predict:
    pred=predict(dis, newdata=newdata.frame)
    pred
```

```
## $class
## [1] 1 2
## Levels: 1 2
##
## $posterior
##           1         2
## 1 0.7545066 0.2454934
## 2 0.1741016 0.8258984
##
## $x
##          LD1
## 1 -0.6000350
## 2  0.8319908
```

```
# clase que se predice:
    pred$class
```

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

Performance de la predicción

```
# predicci?n sobre la muestra usando la lda encontrada
    group=predict(dis, method="plug-in")$class
    group
```

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

```
# Quiero comparar con la verdadera etiqueta:
    table(group, Type)
```

```
##      Type
## group  1  2
##     1 14  3
##     2  3 12
```

```
# tasa de error de clasificaci?n:
    (3+3)/n
```

```
## [1] 0.1875
```

```
# un mejor metodo es usando validaci?n cruzada
    predicciones= array(NA, n)
    for (i in 1:n){
    dat <- Tibet[-i,]
    dis <- lda(Type ~ Length+Breadth+Height+Fheight+Fbreadth,
              data=dat, prior=c(0.5,0.5))
    predicciones[i] <- predict(dis, newdata=Tibet[i,c(1:5)])$class
    }
predicciones
```

```
##  [1] 2 1 1 1 2 1 1 1 1 1 1 2 2 2 1 1 1 2 2 1 2 2 1 2 1 1 2 2 1 2 2 1
```

```
table(predicciones, Type)
```

```
##             Type
## predicciones  1  2
##            1 12  6
##            2  5  9
```

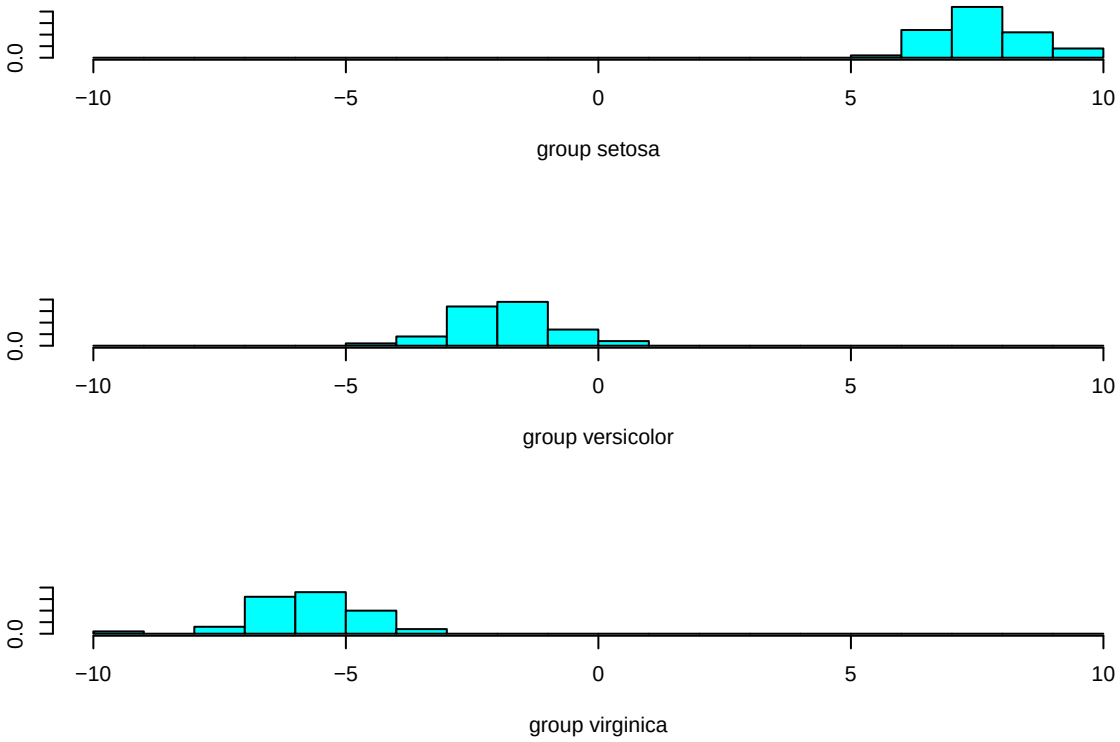
```
# error de clasificaci?n
(6+5)/n
```

```
## [1] 0.34375
```

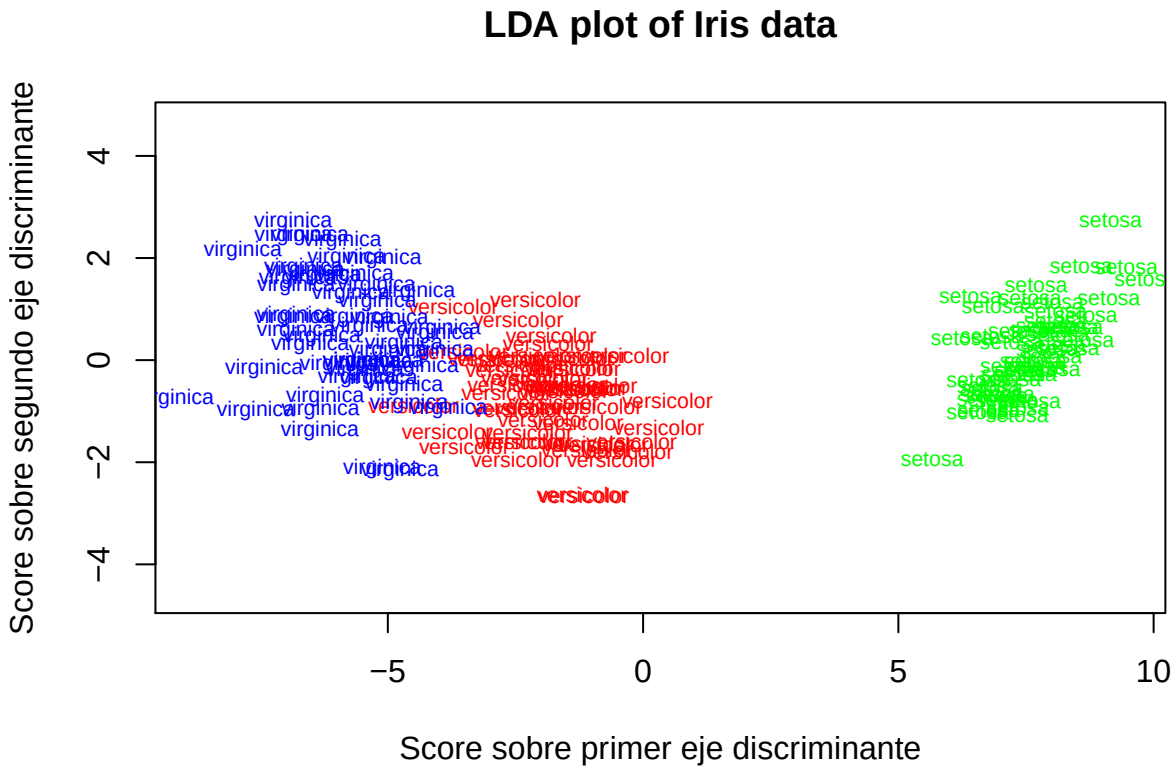
3- Ejemplo3: Iris de Fisher

```
library(MASS)
attach(iris)

data=data.frame(iris)
#Funci?n discriminante de MASS
    ss=lda(Species~.,data=iris)
#Histogramas de cada grupo
    plot(ss, dimen =1)
```




```
#Dibujo en 2D
    plot(ss, dimen =2, col=c("green","red","blue")[unclass(data[,5])],
    main="LDA plot of Iris data",xlab="Score sobre primer eje discriminante",
    ylab="Score sobre segundo eje discriminante")
```



```r
#Matriz de confusi?n

    newdata.frame=data.frame(data)
    pred=predict(ss, newdata=newdata.frame)
    #pred
    table(newdata.frame[,5],pred$class)
```

```
##
##              setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48         2
##   virginica       0          1        49
```

```r
    #Con validaci?n cruzada
    ss2=lda(Species~.,data=iris)
    pred=predict(ss2, newdata=newdata.frame)
    #pred
    table(newdata.frame[,5],pred$class)
```

```
##
##              setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48         2
##   virginica       0          1        49
```

```r
    #Quadratic discriminant analysis
    ss3=qda(Species~.,data=iris)
    pred=predict(ss3, newdata=newdata.frame)
    #pred
    table(newdata.frame[,5],pred$class)
```

```
##
##              setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48         2
##   virginica       0          1        49
```