Exercises Multivariate Analysis

Judith Pe?afiel

25 de junio de 2014

1. An?lisi discriminant

1.1. Find the mean vectors and the covariance matrices.

Comen?em llegint les dades

```
rm(list = ls())
library(xtable)
## setwd('C:/Documents and Settings/jpenafiel/Mis
## documentos/Dropbox/rutines/')
setwd("C:/programs/Dropbox/rutines")
dat <- read.table("Copepodes.csv", header = TRUE, sep = ";")
dat <- dat[-nrow(dat), ]
```

Calculem les mitjanes del vector:

- $\tilde{\mu}$ de tota la poblaci?.
- $\tilde{\mu_1}$ del estadi 1.
- $\tilde{\mu_2}$ del estadi 2.

```
xbar0 <- apply(dat[, 1:2], 2, mean, na.rm = T)  ## sample mean vector of the all es-
tadi
xbar1 <- apply(subset(dat, estadi == 1)[, 1:2], 2, mean)  ## sample mean vector of the
tadi
xbar2 <- apply(subset(dat, estadi == 2)[, 1:2], 2, mean)  ## sample mean vector of the
cond estadi
```

Obtenim com a resultat:

Tabla 1: Vector de mitjanes

	long	amp
Tota la poblaci?	231.56	143.41
Estadi 1	219.49	138.08
Estadi 2	241.64	147.86

Calculem les matrius de covari?ncia:

```
cov.mat0 <- cov(dat[, 1:2], use = "complete.obs")  ## covariance matrix of the all es-
tadi
cov.mat1 <- cov(subset(dat, estadi == 1)[, 1:2])  ## covariance matrix of the first es-
tadi
cov.mat2 <- cov(subset(dat, estadi == 2)[, 1:2])  ## covariance matrix of the se-
cond estadi
```

- $\tilde{\Sigma}$ de tota la poblaci?.

```
##        long    amp
## long 421.91  84.86
## amp   84.86 245.04
```

- $\tilde{\Sigma_1}$ del estadi 1.

```
##          long     amp
## long 409.930  -1.316
## amp   -1.316 306.194
```

- $\tilde{\Sigma_2}$ del estadi 2.

```
##          long     amp
## long 409.930  -1.316
## amp   -1.316 306.194
```

1.2. Perform a multivariate comparison of mean groups (In this case you case use an R specific function)

Utilitzem la proba Hotelling's T^2 que tasta la hip?tesi :
$H_0 : \mu = \mu_0$
$H_1 : \mu \neq \mu_0$
La f?rmula utilitzada en la seg?ent funci? ?s:
$T^2 = \sqrt{n}(\hat{X} - \mu_0) S^{-1}_{pooled} \sqrt{n}(\hat{X} - \mu_0)$
On,
$T^2 \sim \frac{(n-1)p}{(n-p)} F_{p,n-p}$

```
# Funci? test de hotelling
hotel2T2 = function(x1, x2, a = 0.05) {
    p = ncol(x1)  ## dimenisonality of the data
    n1 = nrow(x1)  ## size of the first sample
    n2 = nrow(x2)  ## size of the second sample
    n = n1 + n2  ## total sample size
    xbar1 = apply(x1, 2, mean)  ## sample mean vector of the first sample
    xbar2 = apply(x2, 2, mean)  ## sample mean vector of the second sample
    dbar = xbar1 - xbar2  ## difference of the two mean vectors
    v = ((n1 - 1) * var(x1) + (n2 - 1) * var(x2))/(n - 2)  ## pooled covariance ma-
trix
    t2 = (n1 * n2 * dbar %*% solve(v) %*% dbar)/n
    test = ((n - p - 1) * t2)/((n - 2) * p)  ## test statistic
    crit = qf(1 - a, p, n - p - 1)  ## critical value of the F distribution
    pvalue = 1 - pf(test, p, n - p - 1)  ## p-value of the test statistic
    list(test = test, critical = crit, p.value = pvalue, df1 = p, df2 = n -
        p - 1)
}


# Subgrup Estadi 1
x1 <- subset(dat, estadi == 1)[, 1:2]
# Subgrup Estadi 2
x2 <- subset(dat, estadi == 2)[, 1:2]
res <- hotel2T2(x1, x2, a = 0.05)
```

Obtenim:
$T^2 = 38{,}76 \leq \frac{(167-1)p}{(167-2)} F_{2,164}$
Aleshores , 38.76$\geq \frac{(167-1)p}{(167-2)} F_{2,164}$. Per tant, rebutgem la hip?tesis nul?la d'igualtat en les matrius de mitjanes.

Una altra manera de comparar les mitjanes, ?s el an?lisi MANOVA. On la hipotesis nula ?s igualtat en la matriu de mitjanes dels diferents grups.
Utilitzarem la t?cnica de Wilk's. On l'estad?stic ?s:
$\Lambda * = \frac{|W|}{|B+W|}$

```
groupF <- factor(dat[, 3])
Yvar <- as.matrix(dat[, 1:2])
x <- summary(manova(Yvar ~ groupF, data = dat), test = "Wilks")
x

##             Df Wilks approx F num Df den Df  Pr(>F)
## groupF       1 0.679     38.8      2    164 1.6e-14 ***
## Residuals 165
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

En aquest cas tampoc podem assumir igualtat en les matrius de mitjanes.

1.3. Perform a multivariate comparison of covariance matrices.

Per compara els dos vectors de mitjanes, primerament utilitzarem el Box's test:

```
# Box's test
cov.Mtest = function(x, ina, a = 0.05) {
    p = ncol(x)  ## dimension of the data set
    n = nrow(x)  ## sample size
    k = max(ina)  ## number of groups
    nu = rep(0, k)  ## the sample size of each group will be stored here later
    pame = rep(0, k)  ## the determinant of each covariance will be stored he-
re
    ## t calculate the covariance matrix of each group
    for (i in 1:k) {
        nu[i] = sum(ina == i)
    }
    z = cbind(x, ina)
    mat = array(dim = c(p + 1, p + 1, k))
    mat1 = array(dim = c(p, p, k))

    for (i in 1:k) {
        mat[, , i] = cov(z[ina == i, ])
    }
    mat = mat[1:p, 1:p, 1:k]
    for (i in 1:k) {
        mat1[, , i] = mat[, , i] * nu[i]
```

```
    }

    ## calculate the pooled covariance matrix
    Sp = apply(mat1, 1:2, sum)
    Sp = Sp/(n - k)
    for (i in 1:k) {
        ## this 'for' function calculates the determinant of each covariance ma-
trix
        pame[i] = det((nu[i]/(nu[i] - 1)) * mat[, , i])
    }
    pamela = det(Sp)  ## determinant of the pooled covariance matrix

    ## construct the test statistic
    test1 = log(pamela/pame)
    test2 = sum((nu - 1) * test1)
    gama1 = (2 * (p^2) + 3 * p - 1)/(6 * (p + 1) * (k - 1))
    gama2 = gama1 * (sum(1/(nu - 1)) - 1/(n - k))
    gama = 1 - gama1 * gama2
    test = gama * test2  ## this is the M
    df = 0.5 * p * (p + 1) * (k - 1)  ## degrees of freedom of the chi-square dis-
tribution
    pvalue = 1 - pchisq(test, df)  ## p-value of the test statistic
    crit = qchisq(1 - a, df)  #critical value of the chi-square distribution
    list(M.test = test, degrees = df, critical = crit, p.value = pvalue)
}


res1 <- cov.Mtest(dat[, 1:2], as.numeric(dat[, 3]), a = 0.05)
```

Tamb? utilitzem el test de la m?xima likelihood per comparar les matrius de covari?ncies.

```
# Test de la m?xima likelihod
cov.likel = function(x, ina, a = 0.05) {
    p = ncol(x)  ## dimension of the data set
    n = nrow(x)  ## sample size
    k = max(ina)  ## number of groups
    nu = rep(0, k)  ## the sample size of each group will be stored later
    pame = rep(0, k)  ## the determinant of each covariance will be stored

    for (i in 1:k) {
        nu[i] = sum(ina == i)
    }
    z = cbind(x, ina)
    mat = array(dim = c(p + 1, p + 1, k))
    mat1 = array(dim = c(p, p, k))

    for (i in 1:k) {
        mat[, , i] = cov(z[ina == i, ])
    }
    mat = mat[1:p, 1:p, 1:k]
```

```
    ## create the pooled covariance matrix
    for (i in 1:k) {
        mat1[, , i] = mat[, , i] * nu[i]
    }
    Sp = apply(mat1, 1:2, sum)
    Sp = Sp/n

    ## calculate the determinant of each covariance matrix
    for (i in 1:k) {
        pame[i] = det(mat[, , i])
    }
    pamela = det(Sp)  ## determinant of the pooled covariance matrix

    test1 = log(pamela/pame)  ## divides the determinant of the pooled covarian-
ce
    ## matrix with every covariance matrix
    test = sum(nu * test1)  ## test statistic
    df = 0.5 * p * (p + 1) * (k - 1)  ## degrees of freedom of the asymptotic chi-
square
    pvalue = 1 - pchisq(test, df)  ## p-value of the test statistic
    crit = qchisq(1 - a, df)  #critical value of the chi-square distribution
    list(test = test, degrees = df, critical = crit, p.value = pvalue)
}


res2 <- cov.likel(dat[, 1:2], as.numeric(dat[, 3]), a = 0.05)
```

Tabla 2: Igualtat en la matriu de cov?riancies

	Estad?stic	G.ll.	Valor Cr?tic	P-Valor
Box's test	26.338	3.000	7.815	0.000
Test de la m?xima likelihood	26.781	3.000	7.815	0.000

1.4. Construct the Fisher's linear discriminant function and the quadratic discriminant function using your own functions

1.4.1. Funci? lineal de Fisher

Per tal de poder realitzar la funci? lineal de Fisher, calculem S_{pooled} amb la f?rmula:

$$S_{pooled}=\frac{(n_1-1)S_1+(n_2-1)S_2}{n-2}$$

```
# Estadi 1
x1 <- subset(dat, estadi == 1)[, 1:2]
# Estadi 2
x2 <- subset(dat, estadi == 2)[, 1:2]
# Dimensi? de les dades
p = ncol(x1)
```

```
# Mostra en el estadi 1
n1 = nrow(x1)
# Mostra en el estadi 2
n2 = nrow(x2)
# Total mostra
n = n1 + n2
# Vector mitjana mostral estadi 1
xbar1 = as.matrix(rbind(apply(x1, 2, mean)))  #
# Vector mitjana mostral estadi 2
xbar2 = as.matrix(rbind(apply(x2, 2, mean)))
# Diferencia vector de mitjanes
dbar = xbar1 - xbar2
## pooled covariance matrix
v = ((n1 - 1) * var(x1) + (n2 - 1) * var(x2))/(n - 2)
```

Obtenim:

$$\hat{S}_{pooled}=\begin{pmatrix} 301{,}4 & 31{,}02 \\ 31{,}02 & 222{,}52 \end{pmatrix}$$

Per obtenir la funci? discriminant de fisher calculem:

$$\text{Y}=(\mu_1-\mu_1)\text{'S}^{-1}_{pooled}\text{x}$$

```
# Inversa de Spooled
v_inverse <- solve(v)
# Funci? discriminant de fisher
y = dbar %*% v_inverse
```

El resultat d'aquest ?s: $\begin{pmatrix} -22{,}1439 & -9{,}7782 \end{pmatrix} \begin{pmatrix} 0{,}0034 & -0{,}0005 \\ -0{,}00054 & 0{,}0046 \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} =$
=-0.06995286X_1 -0.03418953X_2

Per tal d 'assignar les noves observacions, apliquem la regla:

- X_0 en el estadi 1 si : $(\mu_2-\mu_1)$' $\Sigma^{-1}X_0-m\leq 0$
- X_0 en el estadi 2 si : $(\mu_2-\mu_1)$' $\Sigma^{-1}X_0-m\geq 0$

on,
$\tilde{m}=(\mu_1-\mu_1)\text{'S}^{-1}_{pooled}(\mu_1+\mu_1)$

```
lda_fisher <- function(xbar1, xbar2, v_inverse, newdat) {
    # Calcula la diferencia dels vectors de mitjanes
    dbar <- xbar2 - xbar1
    # Calcula fisher
    y = dbar %*% v_inverse
    m <- (dbar %*% v_inverse %*% t(xbar2 + xbar1))/2


    d <- y %*% t(newdat)
    xo <- ifelse(d - m >= 0, 2, ifelse(d - m < 0, 1, 0))
```

```
    x <- c(xo, d)

    return(x)
}
lda <- NULL
for (i in 1:nrow(dat)) {
    lda <- rbind(lda, lda_fisher(xbar1, xbar2, v_inverse, as.matrix(dat[, 1:2][i,
        ])))
}
```

```
p1 <- nrow(x1)/nrow(dat)
p2 <- nrow(x2)/nrow(dat)
```

En aquest cas no em tingut en compte les probabilitats a priori, que en aquest cas s?n diferents pels dos grups.Segons el llibre **Applied Multivariate Statistical Analysis** de Johnson utilitzarem la seg?ent regla per assignar noves observacions en els estadis:

- $\text{p}_1\text{=n}_1\frac{}{n} = 0.4551$
- $\text{p}_2\text{=n}_1\frac{}{n} = 0.5449$

```
p1 <- nrow(x1)/nrow(dat)
p2 <- nrow(x2)/nrow(dat)
lda_fisher_priori <- function(xbar1, xbar2, v_inverse, newdat, p1, p2) {
    # Calcula la diferencia dels vectors de mitjanes
    dbar <- xbar2 - xbar1
    # Calcula fisher
    y = dbar %*% v_inverse
    m <- log(p1/p2)


    d <- y %*% t(newdat)
    xo <- ifelse(d - m >= 0, 1, ifelse(d - m < 0, 2, 0))

    x <- c(xo, d)

    return(x)
}
lda_priori <- NULL
for (i in 1:nrow(dat)) {
    lda_priori <- rbind(lda_priori, lda_fisher_priori(xbar1, xbar2, v_inverse,
        as.matrix(dat[, 1:2][i, ]), p1, p2))
}
```

1.4.2. Funci? quad?tica discriminant

Per tal de portar a terme la funci? discriminant quadr?tica utilitzarem la regla:
$\frac{1}{2}x_0'(S_1^{-1} - S_2^{-1})x_0 + (\hat{x}_1'S_1^{-1} - \hat{x}_2'S_2^{-1})x_0 - k \geq log(\frac{p_1}{p2})$
on,
$\text{k=}\frac{1}{2}\left(\frac{|S_1|}{|S_2|}\right) + \frac{1}{2}(\hat{x}_1'S_1^{-1}\hat{x}_1 - \hat{x}_2'S_2^{-1}\hat{x}_2)$

```
# S1
s1 <- cov(x1)
# s2
s2 <- cov(x2)
p1 <- nrow(x1)/nrow(dat)
p2 <- nrow(x2)/nrow(dat)
k <- 1/2 * log(det(s1)/det(s2)) + 1/2 * ((xbar1) %*% solve(s1) %*% t(xbar1) -
    xbar2 %*% solve(s2) %*% t(xbar2))
qda <- NULL
X <- as.matrix(dat[, 1:2])
for (i in 1:nrow(dat)) {
    yq <- -1/2 * t(X[i, ]) %*% (solve(s1) - solve(s2)) %*% X[i, ] + (xbar1 %*%
        solve(s1) - xbar2 %*% solve(s2)) %*% X[i, ]
    xo <- ifelse(yq - k >= log(p1/p2), 1, ifelse(yq - k < log(p1/p2), 2, NA))
    qda <- rbind(qda, cbind(yq, xo))
}
res <- cbind(dat, qda[, 2])
table(res[, 3], res[, 4])
```

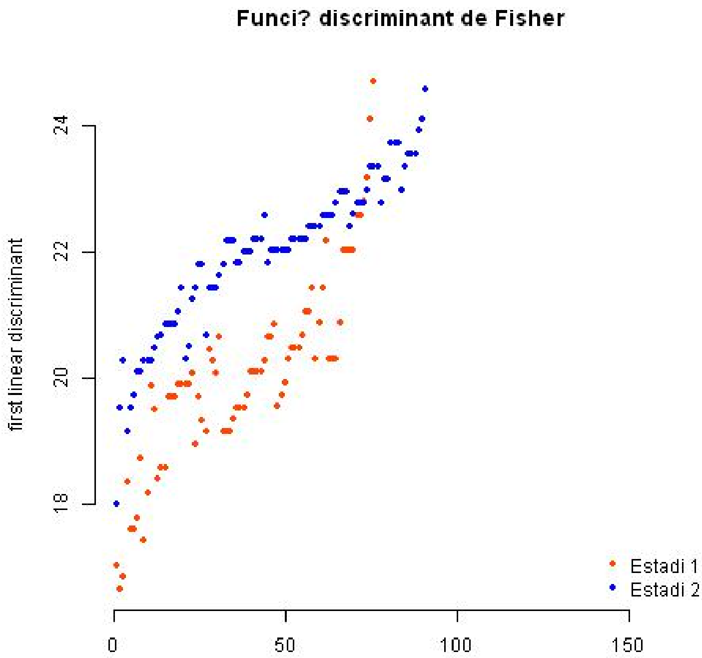
1.5. Show the derived discriminant functions on a scatter plot of the original data.

```
jpeg("./fig1.jpg")
x <- cbind(lda, dat[, 3])

plot(x[, 2], bty = "n", type = "n", main = "Funci? discriminant de Fisher",
    ylab = "first linear discriminant", xlab = "")
points(subset(x[, 2], x[, 3] == 1), col = "orangered", pch = 20)
points(subset(x[, 2], x[, 3] == 2), col = "blue", pch = 20)
legend("bottomright", c("Estadi 1", "Estadi 2"), col = c("orangered", "blue"),
    pch = 20, bty = "n")

dev.off()
```

No ?s pot realitzar el gr?fic de la funci? discriminant quadr?tica, ja que es basa en un score/regla d'assignaci?.



1.6. Estimate the misclassification rate.

1.6.1. Funci? lineal de Fisher

No tinguen en compte les probabilitats a priori:

```
ct <- table(dat$estadi, lda[, 1])
colnames(ct) <- c("Allocated to Estadi 1", "Allocated to Estadi 2")
rownames(ct) <- c("Is  Estadi 1", "Is Estadi 2")
x <- ct[1, 2] + ct[2, 1]  #Misclassified
AR <- xtable(ct, caption = "Misclassification")
print(AR, sanitize.text.function = function(x) {
    x
}, caption.placement = "top", include.rownames = TRUE)
```

Tabla 3: Misclassification

	Allocated to Estadi 1	Allocated to Estadi 2
Is Estadi 1	61	15
Is Estadi 2	21	70

Tenim un total de 36 individus mal classificats
Que correspon a una proporci? del mal classificats de: 0.22

Tinguen en compte les probabilitats a priori:

```
ct <- table(dat$estadi, lda_priori[, 1])
colnames(ct) <- c("Allocated to Estadi 1")
rownames(ct) <- c("Is  Estadi 1", "Is Estadi 2")
x <- 0  #Misclassified
AR <- xtable(ct, caption = "Misclassification")
print(AR, sanitize.text.function = function(x) {
    x
}, caption.placement = "top", include.rownames = TRUE)
```

Tabla 4: Misclassification

	Allocated to Estadi 1
Is Estadi 1	76
Is Estadi 2	91

Tenim un total de 0 individus mal classificats
Que correspon a una proporci? del mal classificats de: 0
Utilitzant les probabilitat a priori, trobem que no classifiquem malament cap individu.

1.6.2. Funci? quad?tica discriminant

```
ct <- table(dat$estadi, qda[, 2])
colnames(ct) <- c("Allocated to Estadi 1", "Allocated to Estadi 2")
rownames(ct) <- c("Is  Estadi 1", "Is Estadi 2")
x <- ct[1, 2] + ct[2, 1]  #Misclassified
AR <- xtable(ct, caption = "Misclassification")
print(AR, sanitize.text.function = function(x) {
    x
}, caption.placement = "top", include.rownames = TRUE)
```

Tabla 5: Misclassification

	Allocated to Estadi 1	Allocated to Estadi 2
Is Estadi 1	64	12
Is Estadi 2	18	73

Tenim un total de 30 individus mal classificats
Que correspon a una proporci? del mal classificats de: 0.18