QUIZ 5

Amanjit Gill

November 28, 2022

```
library(MASS)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(heplots)
```

```
## Loading required package: car

## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

## Loading required package: broom
```

```
library(e1071)

# QUESTION 1A - DATA

Sigma <- matrix(
    c(0.4, -0.2, 0.5, -0.2, 2.1, 1.1, 0.5, 1.1, 3.6),
    3, 3)

Mu <-  matrix(
    c(0.2, -0.7, 0.3, -0.3, -1.2, -0.5, -0.8, -0.2, -0.8),
    3, 3, byrow=TRUE,
    dimnames = list(c("Normal", "Anxiety", "Obsession"), c("A", "B", "C")))

p <- 1/3
```

```r
# conditions:
# - 3 populations (normal, anxiety, obsession)
# - multivariate normal
# - equal covariances
# - equal priors
# - equal misclassification costs

# QUESTION 1A - PART (I)

ans_a_i_N <- function(A, B, C) {

    x <- rbind(A, B, C)
    mu1 <- Mu["Normal",]
    S <- Sigma
    p <- 1/3

    t(mu1) %*% solve(S) %*% x - 1/2 * t(mu1) %*% solve(S) %*% mu1 + log(p)
}

ans_a_i_A <- function(A, B, C) {

    x <- rbind(A, B, C)
    mu2 <- Mu["Anxiety",]
    S <- Sigma
    p <- 1/3

    t(mu2) %*% solve(S) %*% x - 1/2 * t(mu2) %*% solve(S) %*% mu2 + log(p)
}

ans_a_i_O <- function(A, B, C) {

    x <- rbind(A, B, C)
    mu3 <- Mu["Obsession",]
    S <- Sigma
    p <- 1/3

    t(mu3) %*% solve(S) %*% x - 1/2 * t(mu3) %*% solve(S) %*% mu3 + log(p)
}

# QUESTION 1A - PART (II)

A <- -0.1
B <- -1.6
C <- 1.2

d1 <- ans_a_i_N(A, B, C)
d2 <- ans_a_i_A(A, B, C)
d3 <- ans_a_i_O(A, B, C)

sum_exp_d <- sum(exp(d1), exp(d2), exp(d3))

prob_N <- exp(d1) / sum_exp_d
prob_A <- exp(d2) / sum_exp_d
```

```r
prob_O <- exp(d3) / sum_exp_d

ans_a_ii <- c(prob_N, prob_A, prob_O)

dump(c("ans_a_ii"), file="")
```

```
## ans_a_ii <-
## c(0.28308258583296009, 0.51966725269384084, 0.19725016147319899
## )
```

```r
# QUESTION 1A - PART (III)

p1 <- 0.4
p2 <- 0.3
p3 <- 0.3

mu1 <- Mu["Normal",]
mu2 <- Mu["Anxiety",]
mu3 <- Mu["Obsession",]

x <- rbind(A, B, C)
S <- Sigma

d1 <- t(mu1) %*% solve(S) %*% x - 1/2 * t(mu1) %*% solve(S) %*% mu1 + log(p1)
d2 <- t(mu2) %*% solve(S) %*% x - 1/2 * t(mu2) %*% solve(S) %*% mu2 + log(p2)
d3 <- t(mu3) %*% solve(S) %*% x - 1/2 * t(mu3) %*% solve(S) %*% mu3 + log(p3)

sum_exp_d <- sum(exp(d1), exp(d2), exp(d3))

prob_N <- exp(d1) / sum_exp_d
prob_A <- exp(d2) / sum_exp_d
prob_O <- exp(d3) / sum_exp_d

ans_a_iii <- c(prob_N, prob_A, prob_O)

dump(c("ans_a_iii"), file="")
```

```
## ans_a_iii <-
## c(0.34489852561675777, 0.4748591353774867, 0.18024233900575556
## )
```

```r
# QUESTION 1A - PART (IV)

# conditions:
# - 2 populations (anxiety, obsession)
# - multivariate normal
# - equal covariances
# - equal priors
# - equal misclassification costs

# mahalanobis distance
Delta <- sqrt(t(mu2 - mu3) %*% solve(S) %*% (mu2 - mu3))

ans_a_iv_pAO <- pnorm(-Delta/2)[1]
ans_a_iv_pOA <- pnorm(-Delta/2)[1]
```

```r
dump(c("ans_a_iv_pAO", "ans_a_iv_pOA"), file="")
```

```
## ans_a_iv_pAO <-
## 0.31556108186808635
## ans_a_iv_pOA <-
## 0.31556108186808635
```

```r
# QUESTION 1B - DATA

G <- c('Normal', 'Anxiety', 'Anxiety', 'Obsession', 'Anxiety', 'Obsession',
    'Obsession', 'Obsession', 'Anxiety', 'Normal', 'Anxiety', 'Normal',
    'Anxiety', 'Obsession', 'Normal', 'Normal', 'Normal', 'Anxiety',
    'Anxiety', 'Normal', 'Anxiety', 'Obsession', 'Normal', 'Normal',
    'Normal', 'Normal', 'Anxiety', 'Obsession', 'Obsession', 'Normal',
    'Obsession', 'Anxiety', 'Normal', 'Obsession', 'Obsession', 'Anxiety',
    'Normal', 'Anxiety', 'Anxiety')

X <- matrix(
    c(-0.8, -2.5, -3, -0.1, -3.2, -2.2, -1, -2.3, -2.5, -1.6, 2.3, -0.6, -1.7,
    -0.9, -4.3, -0.7, 1.7, -1.1, -1, 0.3, -0.8, -0.6, 3, 2.1, -1.6, -0.6,
    -3, 0.7, 0.4, 1.7, 0.3, -2.4, -0.7, 0.7, -3, 0.5, -0.3, -3.8, -0.2, -1.7,
    -1, -4.2, 0.1, -2.6, -2, 0, 0.6, 0.4, 1, -2.7, -1.7, 0.3, -1.4, 1.8, 0.9,
    -2.8, 1.5, 1.1, -0.1, 2, -0.6, -1.2, 1.8, -0.4, -1.2, -1.3, 0.6, -3.2,
    -1.3, -0.2, -2.4, 0.2, 0.5, -1.9, 0.5, 0.8, 0.7, 1.8, 0.3, 0.3, 1.3,
    -0.7, 1.4, 2.9, -1.8, 0.3, -4.1, -1.1, -1.8, -2.9, 0, -1.5, -3, -0.5, 0,
    0, 0.3, 1.6, -0.8, -0.8, 0.5, -0.5, -0.6, -0.7, -1.5, 0.5, -2.5, -0.7,
    0.3, -0.3, 3.1, -1, 0.1, 0.7, 0.8, -0.9, 3.3),
    39, 3, byrow=TRUE,
    dimnames=list(NULL, c("A","B","C")))

neurotic <- cbind(data.frame(Diagnosis=G), X)

# QUESTION 1B - PART (I)

A <- -0.1
B <- -1.6
C <- 1.2

p <- 1/3

# convert diagnosis column to factor
# to maintain ordering of classes in model results
neurotic$Diagnosis <- factor(neurotic$Diagnosis,
                             levels=c("Normal", "Anxiety", "Obsession"))

# train model
neurotic.lda <- lda(Diagnosis ~ ., data=neurotic, prior=c(p, p, p))

# make prediction
x.new <- data.frame(A, B, C)
probs <- predict(neurotic.lda, newdata=x.new)$posterior

ans_b_i <- c(probs[1,"Normal"], probs[1, "Anxiety"], probs[1, "Obsession"])
```

```
dump(c("ans_b_i"), file="")
```

```
## ans_b_i <-
## c(0.20501688784243235, 0.73793569674089488, 0.05704741541667268
## )
```

```
# QUESTION 1B - PART (II)

cm <- table(Actual=neurotic$Diagnosis, Predicted=predict(neurotic.lda)$class)

ans_b_ii <- matrix(cm, 3, 3)

dump(c("ans_b_ii"), file="")
```

```
## ans_b_ii <-
## structure(c(10L, 4L, 1L, 4L, 7L, 1L, 0L, 3L, 9L), dim = c(3L,
## 3L))
```

```
# QUESTION 1B - PART (III)

hyp_test <- boxM(select(neurotic, -Diagnosis), neurotic$Diagnosis)

ans_b_iii_teststat <- as.numeric(hyp_test$statistic)

ans_b_iii_df <- as.numeric(hyp_test$parameter)

ans_b_iii_pval <- hyp_test$p.value

dump(c("ans_b_iii_teststat", "ans_b_iii_df", "ans_b_iii_pval"), file="")
```

```
## ans_b_iii_teststat <-
## 7.7987159412364777
## ans_b_iii_df <-
## 12
## ans_b_iii_pval <-
## 0.80065559996659341
```

```
# QUESTION 1C - SVM TUNING

Cs <- c(0.5,1,5,10,50)
gammas <- c(0.01, 0.1, 1, 10)

trials.lin <- data.frame(C=Cs, SVs=NA, TotAcc=NA)
trials.rad <- data.frame(C=rep(Cs, each=length(gammas)), gamma=gammas,
                         SVs=NA, TotAcc=NA)

for (i in 1:nrow(trials.lin)) {
    svm.lin <- svm(
        Diagnosis ~ ., data=neurotic, kernel="linear",
        cost=trials.lin[i, "C"], cross=10)
    trials.lin[i,"SVs"] <- svm.lin$tot.nSV
    trials.lin[i, "TotAcc"] <- svm.lin$tot.accuracy
}

# total number of radial trials = number of Cs * number of gammas
for (i in 1:nrow(trials.rad)) {
```

```r
    svm.rad <- svm(
        Diagnosis ~ ., data=neurotic, kernel="radial",
        cost=trials.rad[i, "C"], gamma=trials.rad[i, "gamma"], cross=10)
    trials.rad[i,"SVs"] <- svm.rad$tot.nSV
    trials.rad[i, "TotAcc"] <- svm.rad$tot.accuracy
}

trials.lin <- trials.lin[order(-trials.lin$TotAcc, trials.lin$SVs),]
trials.rad <- trials.rad[order(-trials.rad$TotAcc, trials.rad$SVs),]
head(trials.rad)
```

```
##     C gamma SVs   TotAcc
## 13 10  0.01  38 53.84615
## 9   5  0.01  38 51.28205
## 17 50  0.01  32 46.15385
## 6   1  0.10  38 43.58974
## 19 50  1.00  36 41.02564
## 14 10  0.10  34 38.46154
```

```r
head(trials.lin)
```

```
##      C SVs   TotAcc
## 5 50.0  30 48.71795
## 1  0.5  34 48.71795
## 4 10.0  30 46.15385
## 2  1.0  33 46.15385
## 3  5.0  32 38.46154
```

```r
# choose linear SVM with C = 0.5
# achieves same total accuracy as top-performing radial SVM

# QUESTION 1C - SVM PREDICTION

# final model
svm.tuned <- svm(Diagnosis ~ ., data=neurotic, kernel="linear",
                 cost=0.5, cross=10)

x.new <- data.frame(A, B, C)
predict(svm.tuned, newdata=x.new, decision.values=TRUE)
```

```
##       1
## Anxiety
## attr(,"decision.values")
##   Normal/Anxiety Normal/Obsession Anxiety/Obsession
## 1     -0.5028904        0.8847127          1.027873
## Levels: Normal Anxiety Obsession
```