Multidimensional Statistical Analysis

Code

Lamyaa BOUZBIBA Alexandre POUPEAU Eloise JULIEN

Hide

```
NAm2 = read.table("NAm2.txt", header = TRUE)
#NAm2
cont <- function(x)
{ if (x %in% c("Canada"))
  cont <-"NorthAmerica"
  else if (x %in% c("Guatemala","Mexico","Panama","CostaRica"))
  cont <-"CentralAmerica"
  else
  cont<-"SouthAmerica"
  return (factor(cont))
}
contID <- sapply(as.character(NAm2[,4]),FUN=cont)
library("class")
library("MASS")
```

a) 3 out of 3 points
Be careful, we say “non-invertable” matrix and not “non-reversable”

a)

Hide

```
labels <- rep (1:2, each = 494/2)
set = sample(labels, 494)
# continent + genetic markers
NAcont <- cbind(contID=contID,NAm2[,-(1:8)])
NAcont[,1] <- factor(NAcont[,1])
# we make our training phase using the elements which have its set value equal 1.
lda(contID ~ . , data = NAcont[, 1:1001], subset = (set == 1))
```

```
Error in lda.default(x, grouping, ...) :
  variables   1   4  15  21  23  40  49  58  59  64  66  67  73  79  96 107 132 14
5 152 165 201 202 214 219 227 254 277 279 300 301 307 318 324 327 328 357 364 368
370 394 395 410 435 471 478 485 505 519 521 542 557 568 573 586 605 628 635 659 66
2 669 671 678 683 685 696 703 716 723 734 743 766 787 800 802 810 817 827 844 845
868 920 921 937 938 939 942 960 961 962 975 980 983 appear to be constant within g
roups
```

Here, lda does not work because there are genetic markers with a variance of 0 so the intra_group covariance matrix is non-reversing and R cannot solve it. Actually, collinear variables should be removed.

b)

Hide

```
withinvar <- apply(NAcont[(set==1),-1],FUN=function(x){tapply(x,NAcont[(set==1),1]
,FUN=var)},MARGIN=2)

bool <- as.logical(apply(withinvar,FUN=function(x){prod(x!=0)},MARGIN=2))

NAcont2 <- cbind(contID=contID,(NAm2[,-(1:8)])[,bool])

NAcont2[,1]<-factor(NAcont[,1])

classModel <- lda(contID ~ . , data = subset(NAcont2, set == 1))

#classModel
```

Instead of \(5717\), there are \(3136\) genetic markers remaining after removing those which variance is equal to zero.

c)

Now we predict the original populations of individuals who have not been trained.

Hide

```
# Here we make our test phase using the elements which have its set value equal 2.
prediction <- predict(classModel, newdata = subset(NAcont2, set == 2))
#prediction$class
```

b) 1.5 out of 1.5 points
c) 1.5 out of 1.5 points
d) 5 out of 6 points
The results are correct but you did not do any interpretation of it. For instance, you could have noticed that the quality of the classification on each population is related to the number of trials we have for each one of them. Also, the South and Central american populations are less well classified, an observation that you could hypothesise that comes from the mixing of these people due to geographical proximity

d)

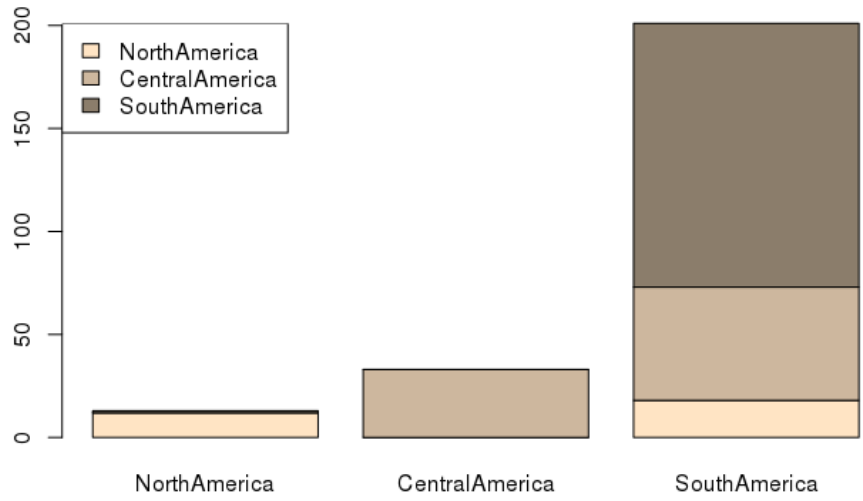
Hide

```
confusionMatrix <- table(subset(contID, set == 2), prediction$class)
confusionMatrix
```

```
                NorthAmerica CentralAmerica SouthAmerica
  NorthAmerica            12              0           18
  CentralAmerica           0             33           55
  SouthAmerica             1              0          128
```

Hide

```
barplot(confusionMatrix, col=c("bisque", "bisque3", "bisque4"))
legend("topleft", legend=c("NorthAmerica", "CentralAmerica", "SouthAmerica"), fill
=c("bisque", "bisque3", "bisque4"))
```



With this barplot, we can see for instance that we predicted that 1 person is North American instead of South American. Moreover, we predicted that 55 people are South American instead of Central American, this the highest error.

e) 2 out of 2 points
f) 5 out of 6 points
Be careful here. You guys are messing up an important concept here. The K-Fold cross validation scheme is used to better validate the classification model. It does so by estimating more accurately the accuracy of the classifier. This does not mean that it improves the quality of the LDA ! It just helps us be more safe to say whether it is working well or no.

e)

Hide

```
rate <- 1-((sum(diag(confusionMatrix)))/sum(confusionMatrix))
sprintf("The rate of misclassified individuals is equal to %s .", rate)
```

```
[1] "The rate of misclassified individuals is equal to 0.299595141700405 ."
```

The classification error is equal to about 30% .

f)

A possible improvement to the method implemented in the previous question could be the 13-fold cross-validation. So, we will divide randomly the data into 13 subsets where 12 subsets are used as the training sets and the remaining subset is used as a validation subset to compute a prediction error. We repeat this procedure ten times by considering each subset as the validation subset and the other subsets as the training sets.

Hide

```
labels <- rep(1:13, each=13)
set = sample(labels, 494, replace=TRUE)
NAcont <- cbind(contID=contID,NAm2[,-(1:8)])
NAcont[,1] <- factor(NAcont[,1])
rate <- c(1:13)
for (i in 1:13){

  withinvar <- apply(NAcont[(set==i),-1],FUN=function(x){tapply(x,NAcont[(set==i),
1],FUN=var)},MARGIN=2)
  bool <- as.logical(apply(withinvar,FUN=function(x){prod(x!=0)},MARGIN=2))
  NAcont2 <- cbind(contID=contID,(NAm2[,-(1:8)])[,bool])
  NAcont2[,1]<-factor(NAcont[,1])
  classModel <- lda(contID ~ . , data = subset(NAcont2, set != i))

  prediction <- predict(classModel, newdata = subset(NAcont2, set == i))

  confusionMatrix <- table(subset(contID, set == i), prediction$class)
  rate[i] <- 1-((sum(diag(confusionMatrix)))/sum(confusionMatrix))

}
```

```
variables are collinearvariables are collinearvariables are collinearvariables are
collinearvariables are collinearvariables are collinearvariables are collinearvari
ables are collinearvariables are collinearvariables are collinearvariables are col
linearvariables are collinearvariables are collinear
```

Hide

```
sprintf("The rate of misclassified individuals is equal to %s .", mean(rate))
```

```
[1] "The rate of misclassified individuals is equal to 0.183506811286169 ."
```

With the 13-fold cross validation, the classification error is equal to about 18% .

To conclude, this cross validation is a good improvement to the method implemented in the previous question. Indeed, the classification error is equal to about 18% against 30% previously.