Lab 7

Attila Lazar

18.11.2020

1)

data

We load the dataset

```
#install.packages("rrcov")
data(olitos, package="rrcov")

set.seed(1234)
n <- nrow(olitos)
train <- sample(1:n, round(n*2/3))
test <- (1:n) [-train]
```

a)

We train *lda* with the train dataset and calculate the test set error

```
library(MASS)
model1 <- lda(grp~., data=olitos, subset=train)
pred1 <- predict(model1, olitos[test,])
T <- table(olitos[test,'grp'], pred1$class)
e1 <- 1-sum(diag(T))/sum(T)
e1
```

```
## [1] 0.375
```

and compare with the error rate obtained with CV

```
model1.cv <- lda(grp~., data=olitos, subset=train, CV=TRUE)
T <- table(olitos[train,'grp'], model1.cv$class)
e1.cv <- 1-sum(diag(T))/sum(T)
e1.cv
```

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

b)

We use *qda* for the same task

```
#model2 <- qda(grp~., data=olitos, subset=train)
```

c)

We use *RDA*

```r
library(klaR)

model3 <- rda(grp~., data=olitos, subset=train)
model3$regularization
```

```
##        gamma       lambda
## 6.180857e-06 1.000000e+00
```

The *gamma* and *lambda* parameters show that the used covariance structure is simular to a common covariance matrixt like in *lda*

We compute the error rate on the test-set

```r
pred3 <- predict(model3, olitos[test,])
T <- table(olitos[test,'grp'], pred3$class)
e3 <- 1-sum(diag(T))/sum(T)
e3
```

```
## [1] 0.125
```

and the train set with CV

```r
model3.cv <- lda(grp~., data=olitos, subset=train, CV=TRUE)
T <- table(olitos[train,'grp'], model3.cv$class)
e3.cv <- 1-sum(diag(T))/sum(T)
e3.cv
```

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

2)

We load the bank dataset

```r
bank <- read.csv2("data/bank.csv")
```

a)

We select 3000 observations as our training set for *lda*

```r
set.seed(1234)
train <- sample(1:nrow(bank), 3000)
test <- (1:nrow(bank)) [-train]

model2.1 <- lda(y~., data=bank, subset=train)
model2.1$prior
```

```
##        no       yes
## 0.8803333 0.1196667
```

With *lda* we calculated the priori probabilities of the two classes. We can see that the "no" class has a much bigger probablity

```r
pred2.1 <- predict(model2.1, bank[test,])
T <- table(bank[test,'y'], pred2.1$class)
T
```

```
##
##         no  yes
##   no  1308   51
##   yes   89   73
```

We see the effect of higher probability of the "no" class on the confusion table: much more samples are predicted with "no" than "yes". We compute the missclassification Rate:

```
e4 <- 1-sum(diag(T))/sum(T)
e4
```

```
## [1] 0.09204471
```

b)

We can move the decision boundary by specifying prior probabilities for *lda*. If we ie provide a much smaller probability for "no" = 0.2, there will be much more "yes" predictions.

```
f <- 0.2
model2.2 <- lda(y~., data=bank, subset=train, prior=c(f, 1-f))
pred2.2 <- predict(model2.2, bank[test,])
T <- table(bank[test,'y'], pred2.2$class)
T
```

```
##
##        no yes
##   no  983 376
##   yes  17 145
```

This way we get much worse error rate.

```
e5 <- 1-sum(diag(T))/sum(T)
e5
```

```
## [1] 0.2583826
```