STAT 428 Statistical Computing

Homework 4 Solutions

Department of Statistics, University of Illinois at Urbana-Champaign

13 Mar 2017

Problem 1 (Rizzo problem 7.7)

```
set.seed(428)
library("bootstrap")
# Load the data (five-dimensional score data)
data(scor)

# Covariance matrix
Sigma = cov(scor)
# Calculate the eigenvalues
svd.decomp = svd(Sigma)
eig.val = svd.decomp$d
# Sample estimate of theta
theta.hat = eig.val[1]/sum(eig.val)

# Use bootstrap to estimate the bias and standard error
# Number of replications
B = 1000
# Sample size
n = 100
theta.boot = rep(0,B)
for(b in 1:B){
    ind = sample(1:88,n,replace = TRUE)
    resamp = scor[ind,]
    eig.val.boot = svd(cov(resamp))$d
    theta.boot[b] = eig.val.boot[1]/sum(eig.val.boot)
}
# Bias
bias.boot = mean(theta.boot)-theta.hat
# Standard error
se.boot = sd(theta.boot)

# The result
cat(paste("The sample estimate of theta:",round(theta.hat,4),sep=" "),
    paste("The bootstrap estimate of bias:",round(bias.boot,4),sep=" "),
    paste("The bootstrap estimate of Standard error:",round(se.boot,4),sep=" "),
    sep="\n")
```

```
## The sample estimate of theta: 0.6191
## The bootstrap estimate of bias: -0.0024
## The bootstrap estimate of Standard error: 0.0457
```

Problem 2 (Rizzo problem 7.8)

```
# Use jackknife to estimate the bias and standard error
n = dim(scor)[1]

theta.jack = rep(0,n)
for(i in 1:n){
    resamp = scor[-i,]
    eig.val.jack = svd(cov(resamp))$d
    theta.jack[i] = eig.val.jack[1]/sum(eig.val.jack)
}

# Jackknife estimate of bias
bias.jack = (n-1)*(mean(theta.jack) - theta.hat)
# Jackknife estimate of standard error
se.jack = sqrt((n-1)*mean((theta.jack-mean(theta.jack))^2))

# The result
cat(paste("The jackknife estimate of bias:",round(bias.jack,4),sep=" "),
    paste("The jackknife estimate of Standard error:",round(se.jack,4),sep=" "),
    sep="\n")
```

```
## The jackknife estimate of bias: 0.0011
## The jackknife estimate of Standard error: 0.0496
```

Problem 3 (Rizzo problem 7.10)

(1) Select model by cross validation procedure

```
library("DAAG")
data(ironslag)
n = dim(ironslag)[1]
# Cubic polynomial model
L = lm(magnetic~chemical+I(chemical^2)+I(chemical^3), data = ironslag)

# Estimate prediction error by leave-one-out cross validation
err = 0
for(k in 1:n){
    # Fit a cubic model with (n-1) observations
    L = lm(magnetic~chemical+I(chemical^2)+I(chemical^3), data = ironslag[-k,])
    yhat = predict(L, newdata = data.frame(chemical = ironslag[k,1]))
    err = err + (yhat - ironslag[k,2])^2
}
# Prediction error for cubic polynomial model
pred.err = err/n
cat(paste("Prediction error for cubic polynomial model:",round(pred.err,5),sep=" "),
    sep="\n")
```

```
## Prediction error for cubic polynomial model: 18.17756
```

Compared with the result in Example 7.18, Model 2, the quadratic model (cv error: 17.85248) is selected by leave-one-out cross validation procedure.

(2) Select model according to adjusted R^2

```
data(ironslag)
n = dim(ironslag)[1]
# Linear model
L1 = lm(magnetic~chemical, data = ironslag)
# Quadratic model
L2 = lm(magnetic~chemical+I(chemical^2), data = ironslag)
# Exponential model
L3 = lm(log(magnetic)~chemical, data = ironslag)
# Cubic polynomial model
L4 = lm(magnetic~chemical+I(chemical^2)+I(chemical^3), data = ironslag)

cat(paste("Adjusted R^2 for Linear model:",round(summary(L1)$adj.r.squared,5),sep=" "),
    paste("Adjusted R^2 for Quadratic model:",round(summary(L2)$adj.r.squared,5),sep=" "),
    paste("Adjusted R^2 for Exponential model:",round(summary(L3)$adj.r.squared,5),sep=" "),
    paste("Adjusted R^2 for Cubic polynomial model:",round(summary(L4)$adj.r.squared,5),sep=" "),
    sep="\n")
```

```
## Adjusted R^2 for Linear model: 0.52815
## Adjusted R^2 for Quadratic model: 0.57682
## Adjusted R^2 for Exponential model: 0.52806
## Adjusted R^2 for Cubic polynomial model: 0.57404
```

Model 2, the quadratic model is selected according to maximum adjusted R^2.

Problem 4 (Rizzo problem 7.11)

```
library(knitr)
data(ironslag)
n = dim(ironslag)[1]
# Linear model
L1 = lm(magnetic~chemical, data = ironslag)
# Quadratic model
L2 = lm(magnetic~chemical+I(chemical^2), data = ironslag)
# Exponential model
L3 = lm(log(magnetic)~chemical, data = ironslag)
# Log-log model
L4 = lm(log(magnetic)~log(chemical), data = ironslag)

index = combn(1:n, m=2)
err.ltocv = matrix(nrow = 1, ncol = 4, data = 0)
colnames(err.ltocv) = c("linear","quadratic","exponential","log-log")
for(i in 1:choose(n,2)){
    # Index of the test data
    ind = index[,i]
    # Linear model
    L1 = lm(magnetic~chemical, data = ironslag[-ind,])
    yhat1 = predict(L1, newdata = data.frame(chemical = ironslag[ind,1]))
    err.ltocv[1] = err.ltocv[1] + sum((yhat1 - ironslag[ind,2])^2)
    # Quadratic model
    L2 = lm(magnetic~chemical+I(chemical^2), data = ironslag[-ind,])
```

```
    yhat2 = predict(L2, newdata = data.frame(chemical = ironslag[ind,1]))
    err.ltocv[2] = err.ltocv[2] + sum((yhat2 - ironslag[ind,2])^2)
    # Exponential model
    L3 = lm(log(magnetic)~chemical, data = ironslag[-ind,])
    yhat3 = predict(L3, newdata = data.frame(chemical = ironslag[ind,1]))
    err.ltocv[3] = err.ltocv[3] + sum((exp(yhat3) - ironslag[ind,2])^2)
    # Log-log model
    L4 = lm(log(magnetic)~log(chemical), data = ironslag[-ind,])
    yhat4 = predict(L4, newdata = data.frame(chemical = ironslag[ind,1]))
    err.ltocv[4] = err.ltocv[4] + sum((exp(yhat4) - ironslag[ind,2])^2)
}
err.ltocv = err.ltocv/(2*choose(n,2))
knitr::kable(err.ltocv, caption = "Leave-two-out Cross Validation Error")
```

Table 1: Leave-two-out Cross Validation Error

linear	quadratic	exponential	log-log
19.57227	17.87018	18.45491	20.46718

Model 2, the quadratic model is selected by leave-two-out cross validation procedure.

Problem 5 (Rizzo problem 8.1)

Use `chickwts` data from faraway package comparing chicks fed **casein** with chicks fed **horsebean**. The hypotheses of interest are

$$H_0: \ F = G \quad vs \quad H_1: \ F \neq G,$$

where F is the distribution of weight of chicks fed casein supplements and G is the distribution of weight of chicks fed horsebean supplements.

```
library("faraway")
data(chickwts)
attach(chickwts)
x = sort(as.vector(weight[feed == "casein"]))
y = sort(as.vector(weight[feed == "horsebean"]))
detach(chickwts)

# Function to estimate the integrated squared distance between the distributions
cvm.dist <- function(x,y){
    n = length(x)
    m = length(y)
    # Two-sample Cramer-von Mises test for equal distributions as a permutation test
    # Empirical distributions
    F = ecdf(x)
    G = ecdf(y)
    W2 = m*n/((m+n)^2)*(sum((F(x)-G(x))^2) + sum((F(y)-G(y))^2))
    return(W2)
}

# Permutation test
R = 999 # Number of replicates
z = c(x,y) # pooled sample
```

```r
K = 1:(length(x)+length(y))
D = numeric(R)
D0 = cvm.dist(x,y)
for(i in 1:R){
    k = sample(K, size = length(x), replace = FALSE)
    x1 = z[k]
    y1 = z[-k]
    D[i] = cvm.dist(x1,y1)
}
p = mean(c(D0,D) >= D0)
cat(paste("The estimate of p value:",p,sep=" "),
    sep="\n")
```

```
## The estimate of p value: 0.001
```

$\hat{p} = 0.001 < 0.05 \implies$ Reject H_0 at significance level $\alpha = 0.05$. Thus, we can't conclude that the distributions of chick weights for the casein and horsebean groups are the same.