재추출방법을 위한 보충 R 코드와 결과

이재용, 임요한

서울대학교 통계학과

August 23, 2017

Contents

1 전역옵션들

```
opts_chunk$set(eval=TRUE, cache=TRUE, fig.width=7, fig.height=4)
```

2 자동차자료 읽기

```
auto = read.csv("Auto.csv", header=T, sep=",")
auto$horsepower = as.numeric(auto$horsepower)
head(auto)
```

```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307         17   3504         12.0   70      1
## 2  15         8          350         35   3693         11.5   70      1
## 3  18         8          318         29   3436         11.0   70      1
## 4  16         8          304         29   3433         12.0   70      1
## 5  17         8          302         24   3449         10.5   70      1
## 6  15         8          429         42   4341         10.0   70      1
##                       name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

알지못하는 이유로 horsepower의 클래스가 factor로 입력이 된다. 이를 numeric으로 바꾸었다.

```
library(ISLR)
```

3 검증자료방법을 이용한 다항회귀 차수 결정

```
set.seed(1)
train=sample(397,198)
```

랜덤시드를 고정하고 1:397 중에서 198개의 숫자를 랜덤하게 추출해서 train으로 명명한다. 이를 훈련자료의 관측치 번호이다.

책의 코드에는 397 대신 392를 썼다. 그런데 내가 업로드한 자료의 개수는 397이므로 여기서는 397을 쓴다.

```
lm.fit=lm(mpg~horsepower,data=auto,subset=train)
attach(auto)
mean((mpg-predict(lm.fit,auto))[-train]^2)
```

```
## [1] 49.85
```

```
lm.fit2=lm(mpg~poly(horsepower,2),data=auto,subset=train)
mean((mpg-predict(lm.fit2,auto))[-train]^2)
```

```
## [1] 49.68

lm.fit3=lm(mpg~poly(horsepower,3),data=auto,subset=train)
mean((mpg-predict(lm.fit3,auto))[-train]^2)

## [1] 31.65
```

훈련자료에 세 개의 모형을 적합하고 시험자료로 시험오차를 계산한다.

```
set.seed(2)
train=sample(397,198)
lm.fit=lm(mpg~horsepower,subset=train)
mean((mpg-predict(lm.fit,auto))[-train]^2)

## [1] 45.25

lm.fit2=lm(mpg~poly(horsepower,2),data=auto,subset=train)
mean((mpg-predict(lm.fit2,auto))[-train]^2)

## [1] 44.95

lm.fit3=lm(mpg~poly(horsepower,3),data=auto,subset=train)
mean((mpg-predict(lm.fit3,auto))[-train]^2)

## [1] 30.43
```

시드만 바꿔서 동일한 방법으로 시험오차를 계산했다.

4 하나남기기 교차검증

```
glm.fit=glm(mpg~horsepower,data=auto)
coef(glm.fit)

## (Intercept)  horsepower
##     17.8076      0.1108

lm.fit=lm(mpg~horsepower,data=auto)
coef(lm.fit)

## (Intercept)  horsepower
##     17.8076      0.1108
```

glm 명령에서 family의 디폴트 값은 gaussian이다. 따라서 위의 모형은 동일한 선형모형을 적합한 것이다.

```
library(boot)
glm.fit=glm(mpg~horsepower,data=auto)
cv.err=cv.glm(auto, glm.fit)
names(cv.err)

## [1] "call"  "K"     "delta" "seed"

str(cv.err)

## List of 4
##  $ call : language cv.glm(data = auto, glmfit = glm.fit)
##  $ K    : num 397
##  $ delta: num [1:2] 50.6 50.6
##  $ seed : int [1:626] 403 198 241038711 -724551010 -982165160 -1828904773 1219649113 639708648 -2491
```

cv.glm은 K-겹 교차검증오차를 계산한다. cv.glm의 옵션으로 K가 있는데 이의 디폴트 값은 자료의 개수이다. 따라서 하나남기기 교차검증오차를 계산한다.

```
cv.err$delta

## [1] 50.59 50.59
```

항상 길이가 2인 벡터이다. 첫번째는 예측오차의 교차검증추정량이다. 두번째 것은 하나남기기 교차검증을 사용하지 않을 때 생기는 편이를 교정한 값이다. 여기서는 두 개의 값이 거의 동일하다.

```
cv.error=rep(0,5)
for (i in 1:5){
 glm.fit=glm(mpg~poly(horsepower,i),data=auto)
 cv.error[i]=cv.glm(auto,glm.fit)$delta[1]
 }
cv.error

## [1] 50.59 50.65 32.52 32.72 27.32
```

다항회귀의 차수별로 하나남기기 교차검증으로 추정한 예측오차를 구한다.

5 k겹 교차검증

```
set.seed(17)
cv.error.10=rep(0,10)
for (i in 1:10){
 glm.fit=glm(mpg~poly(horsepower,i),data=auto)
 cv.error.10[i]=cv.glm(auto,glm.fit, K=10)$delta[1]
 }
cv.error.10
```

```
##  [1] 50.37 50.65 32.47 32.50 27.20 27.44 24.17 24.35 23.54 23.80
```

10겹 교차검증방법으로 예측오차를 추정한다. 다항회귀의 차수는 10차까지 이다.

6 붓스트랩

```
alpha.fn=function(data,index){
 X=data$X[index]
 Y=data$Y[index]
 return((var(Y)-cov(X,Y))/(var(X)+var(Y)-2*cov(X,Y)))
 }
```

X와 Y를 콤포넌트로 갖고 있는 data와 data의 관측치 중 일부분의 인덱스를 나타내는 index를 받아들여서, 그 인덱스에 해당하는 자료만을 이용하여

$$\frac{\mathbb{V}ar(Y)-\mathbb{C}ov(X,Y)}{\mathbb{V}ar(X)+\mathbb{V}ar(Y)-2\mathbb{C}ov(X,Y)}$$

를 반환한다.

```
str(Portfolio)
```

```
## 'data.frame': 100 obs. of  2 variables:
##  $ X: num  -0.895 -1.562 -0.417 1.044 -0.316 ...
##  $ Y: num  -0.235 -0.885 0.272 -0.734 0.842 ...
```

```
alpha.fn(Portfolio,1:100)
```

```
## [1] 0.5758
```

Portfolio라는 자료에 alpha.fn을 적용해보았다. Portfolio는 X, Y를 콤포넌트로 갖고 있는 데이터프레임이다.

```
set.seed(1)
alpha.fn(Portfolio, sample(100,100,replace=T))

## [1] 0.5964
```

alpha.fn에 들어가는 인덱스를 `sample(100,100,replace=T)`를 이용해서 랜덤하게 구했다.

```
boot(Portfolio, alpha.fn, R=1000)

##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Portfolio, statistic = alpha.fn, R = 1000)
##
##
## Bootstrap Statistics :
##     original     bias    std. error
## t1*   0.5758 -7.315e-05     0.08862
```

boot은 자료 Portfolio에 통계량 alpha.fn의 붓스트랩을 적용하여 bias와 표준오차를 구한다.

7 붓스트랩을 이용해서 선형모형의 정확도를 구하는 코드

```
# Estimating the Accuracy of a Linear Regression Model

boot.fn=function(data,index)
 return(coef(lm(mpg~horsepower,data=data,subset=index)))
boot.fn(Auto,1:392)

## (Intercept)  horsepower
##     39.9359     -0.1578

set.seed(1)
boot.fn(Auto,sample(392,392,replace=T))
```

```
## (Intercept)  horsepower
##     38.7387     -0.1482
```

```
boot.fn(Auto,sample(392,392,replace=T))
```

```
## (Intercept)  horsepower
##     40.0383     -0.1596
```

```
boot(Auto,boot.fn,1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Auto, statistic = boot.fn, R = 1000)
##
##
## Bootstrap Statistics :
##     original     bias    std. error
## t1*  39.9359  0.0297219    0.860008
## t2*  -0.1578 -0.0003082    0.007404
```

```
summary(lm(mpg~horsepower,data=Auto))$coef
```

```
##              Estimate Std. Error t value   Pr(>|t|)
## (Intercept)  39.9359   0.717499   55.66 1.220e-187
## horsepower   -0.1578   0.006446  -24.49  7.032e-81
```

```
boot.fn=function(data,index)
 coefficients(lm(mpg~horsepower+I(horsepower^2),data=data,subset=index))
set.seed(1)
boot(Auto,boot.fn,1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
```

```
## Call:
## boot(data = Auto, statistic = boot.fn, R = 1000)
##
##
## Bootstrap Statistics :
##      original     bias    std. error
## t1* 56.900100  6.098e-03   2.0944856
## t2* -0.466190 -1.777e-04   0.0334124
## t3*  0.001231  1.324e-06   0.0001208
```

```
summary(lm(mpg~horsepower+I(horsepower^2),data=Auto))$coef
```

```
##                   Estimate Std. Error t value   Pr(>|t|)
## (Intercept)     56.900100  1.8004268   31.60 1.741e-109
## horsepower      -0.466190  0.0311246  -14.98  2.289e-40
## I(horsepower^2)  0.001231  0.0001221   10.08  2.196e-21
```