5.3 Lab: Cross-Validation and the Bootstrap

2023-05-25

5.3 Lab: Cross-Validation and the Bootstrap

5.3.1 The Validation Set Approach

First to generate a seed for numbers, and then use the `sample()` function to split the set of observations into two parts (196 out of 392)

```
library(ISLR)
set.seed (1)
train=sample(392,196)
```

Then we can do the regression

```
lmod=lm(mpg~horsepower,data=Auto,subset=train)
```

Then use the `predict()` function to estimate the response for all 392 observations, and the `mean()` function to see the MSE of the 196 observations.

```
attach(Auto)
mean((mpg-predict(lmod,Auto))[-train]^2)
```

```
## [1] 23.26601
```

Then we can use the `poly()` function to estimate the test error for the quadratic and cubic regressions.

```
lmod2=lm(mpg~poly(horsepower,2),data=Auto,subset=train)
mean((mpg-predict(lmod2,Auto))[-train]^2)
```

```
## [1] 18.71646
```

```
lmod3=lm(mpg~poly(horsepower,3),data=Auto,subset=train)
mean((mpg-predict(lmod3,Auto))[-train]^2) # Here should be 2 not 3
```

```
## [1] 18.79401
```

If we choose a different data set, we will obtain a different result.

5.3.2 Leave-One-Out CV

This can be written as LOOCV, which can be computed automatically using `glm()` or `lm()` functions

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

```
## (Intercept)  horsepower
##  39.9358610  -0.1578447
```

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

```
## (Intercept)  horsepower
##  39.9358610  -0.1578447
```

Then we can do CV as follows:

```
library(boot)
glm.fit=glm(mpg~horsepower ,data=Auto)
cv.err=cv.glm(Auto,glm.fit)
cv.err$delta
```

```
## [1] 24.23151 24.23114
```

We can create a loop function to iteratively fits polynomial regressions for polynomials from i=1 to 5.

```
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] 24.23151 19.24821 19.33498 19.42443 19.03321
```

5.3.3 k-Fold Cross-Validation

The `cv.glm()` function can also be used to implement k-fold CV as below:

```
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] 24.27207 19.26909 19.34805 19.29496 19.03198 18.89781 19.12061 19.14666
##  [9] 18.87013 20.95520
```

Here we can see that the computation time is much shorter than that of LOOCV.

5.3.4 The Bootstrap

Estimating the Accuracy of a Statistic of Interest

There are two steps to - First, we create a function that computes the statistic of interest. - Second, we use the `boot()` function, which is part of the boot library, to perform the bootstrap by repeatedly sampling observations from the data set with replacement.

```
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)))
 }
```

It returns an estimate for α based on applying to the observations indexed by the argument index.

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

```
## [1] 0.5758321
```

Then we will use the `sample()` function to randomly select 100 observations ranging from 1 to 100.

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

```
## [1] 0.7368375
```

Also we can poroduce a `R=1000` bootstrap estimates for α

```
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* 0.5758321 -0.001695873  0.09366347
```

Estimating the Accuracy of a Linear Regression Model

Here we use the bootstrap approach in order to assess the variability of the estimates for β_0 and β_1. We will need to using SE to compare.

We first create a simple function, `boot.fn()`. We then apply this function to the full set of 392 observations.

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

```
## (Intercept)  horsepower
##  39.9358610  -0.1578447
```

The `boot.fn()` function can also be used in order to create bootstrap esti- mates for the intercept and slope terms by randomly sampling from among the observations with replacement.

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

```
## (Intercept)  horsepower
##  40.3404517  -0.1634868
```

Then we will use the `boot()` function to compute the standard errors of 1,000 bootstrap estimates for the intercept and slope terms.

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

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Auto, statistic = boot.fn, R = 1000)
##
##
## Bootstrap Statistics :
##       original        bias    std. error
## t1* 39.9358610  0.0549915227 0.841925746
## t2* -0.1578447 -0.0006210818 0.007348956
```

We can get a summary of the bootstrap as below:

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

```
##                Estimate  Std. Error   t value      Pr(>|t|)
## (Intercept) 39.9358610 0.717498656  55.65984 1.220362e-187
## horsepower  -0.1578447 0.006445501 -24.48914  7.031989e-81
```

Below we compute the bootstrap standard error estimates and the standard linear regression estimates that result from fitting the quadratic model to the data, and then we ccan compare their standard errors.

```r
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* 56.900099702  3.511640e-02 2.0300222526
## t2* -0.466189630 -7.080834e-04 0.0324241984
## t3*  0.001230536  2.840324e-06 0.0001172164
```

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

```
##                     Estimate   Std. Error   t value      Pr(>|t|)
## (Intercept)     56.900099702 1.8004268063  31.60367 1.740911e-109
## horsepower      -0.466189630 0.0311246171 -14.97816  2.289429e-40
## I(horsepower^2)  0.001230536 0.0001220759  10.08009  2.196340e-21
```