Lab: Cross-Validation and the Bootstrap

Jonathan Bryan

May 18, 2018

5.3.1 The Validation Set Approach

```
library(ISLR)

#Sample random integers to extract data from the Auto set
set.seed(1)
train = sample(nrow(Auto), nrow(Auto)/2)

#Linear regression
lm.fit = lm(mpg~horsepower, data=Auto, subset = train)

#MSE
mean((Auto$mpg-predict(lm.fit, Auto))[-train]^2)
```

```
## [1] 26.14142
```

```
#Polynomial (2) linear regression
lm.fit2 = lm(mpg~poly(horsepower,2), data = Auto, subset = train)

#MSE
mean((Auto$mpg-predict(lm.fit2, Auto))[-train]^2)
```

```
## [1] 19.82259
```

```
#Polynomial (3) linear regression
lm.fit3 = lm(mpg~poly(horsepower,3), data = Auto, subset = train)

#MSE
mean((Auto$mpg-predict(lm.fit3, Auto))[-train]^2)
```

```
## [1] 19.78252
```

```
#Repeating OLS fits with different seed
set.seed(2)
train = sample(nrow(Auto), nrow(Auto)/2)

#Linear regression
lm.fit = lm(mpg~horsepower, data=Auto, subset = train)
lm.fit2 = lm(mpg~poly(horsepower,2), data = Auto, subset = train)
lm.fit3 = lm(mpg~poly(horsepower,3), data = Auto, subset = train)

#MSE
mean((Auto$mpg-predict(lm.fit, Auto))[-train]^2)
```

```
## [1] 23.29559
```

```
mean((Auto$mpg-predict(lm.fit2, Auto))[-train]^2)
```

```
## [1] 18.90124
```

```r
mean((Auto$mpg-predict(lm.fit3, Auto))[-train]^2)
```

```
## [1] 19.2574
```

5.3.2 Leave-One-Out Cross-Validation

```r
#Demonstration of equivalence between glm and lm
glm.fit = glm(mpg~horsepower, data=Auto)
coef(glm.fit)
```

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

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

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

```r
library(boot)

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

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

```r
#LOOCV test error average with polynomial functions up to 5
cv.error = rep(NA,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

```r
#LOOCV test error average with polynomial functions up to 10
set.seed(17)
cv.error.10 = rep(NA,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] 24.20520 19.18924 19.30662 19.33799 18.87911 19.02103 18.89609
##  [8] 19.71201 18.95140 19.50196
```

5.3.4 The Bootstrap

```
#Functon to calculate alpha for investment portfolio
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)))
}

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

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

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

```
## [1] 0.5963833
```

```
#Bootstrap function
boot(data = Portfolio, statistic = 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 -7.315422e-05  0.08861826
```

```
#Bootstrap estimates for linear regression coefficients (using full data)
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
```

```
#Randomly sampling all observations with replacement
set.seed(1)
boot.fn(Auto,sample(392,392, replace = T))
```

```
## (Intercept)  horsepower
##  38.7387134  -0.1481952
```

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

```
## (Intercept)  horsepower
##  40.0383086  -0.1596104
```

```
boot(data = Auto, statistic = boot.fn, R = 1000)
```

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

```
##
##
## Call:
## boot(data = Auto, statistic = boot.fn, R = 1000)
##
##
## Bootstrap Statistics :
##       original      bias    std. error
## t1* 39.9358610  0.02972191 0.860007896
## t2* -0.1578447 -0.00030823 0.007404467
```

```
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
```

```
boot.fn = function(data,index){
  return(coef(lm(mpg~horsepower +I( horsepower ^2), data = data, subset=index)))
}
set.seed(1)
boot(data = Auto, statistic = boot.fn, R = 1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Auto, statistic = boot.fn, R = 1000)
##
##
## Bootstrap Statistics :
##         original        bias     std. error
## t1* 56.900099702  6.098115e-03 2.0944855842
## t2* -0.466189630 -1.777108e-04 0.0334123802
## t3*  0.001230536  1.324315e-06 0.0001208339
```

```
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
```