SDSC3006 Lab
5-Bootstrap and model selection

Langming LIU langmiliu2-c@my.cityu.edu.hk

School of Data Science
City University of Hong Kong

Contents

- Bootstrap
- Model Selection

Bootstrap

Introduction

- Dataset: Portfolio (X,Y) in library ISLR2.
- Target: Minimize the total risk(var). Minimizer is given by:

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}},$$

- Method: Utilize Bootstrap to get the estimate of Var and Cov.
- Steps: Simulate 100 pairs of returns for X and Y for 1000 times. Take average. Calculate Sd of estimate
- Notice: Use the boot() function in the boot library to perform the bootstrap.

Code

```
library(ISLR2)
attach(Portfolio)
##create a function to calculate alpha
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))
}
##estimate alpha using all 100 observations
alpha.fn(Portfolio,1:100)
#randomly select 100 observations from the dataset with replacement
set.seed(1)
alpha.fn(Portfolio,sample(100,100,replace=T))
#estimates based on 1000 bootstrap samples
library(boot)
boot(Portfolio,alpha.fn,R=1000)
```

Q and A

- Question: How can we get 100 simulated values from 100 true observations?
- Answer: Each time we can choose randomly from the range 1 to 100 with replacement, which means we can choose the same value for several times.

```
> set.seed(1)
> sample (100, 100, replace = T)
  [1]  68  39   1  34  87  43  14  82  59  51
 [19]  85  37  89  37  34  89  44  79  33  84
 [37]  44  87  70  40  44  25  70  39  51  42
 [55]  78  65  70  87  70  75  81 100  13  40
 [73]  93  28  48  33  45  21  31  17  73  87
 [91]  93  34  10   1  43  59  26  15  58  29
```

Model Selection

Introduction

- Dataset: Hitters in library ISLR.
- Target: Predict a baseball player's Salary based on several predictors.
- Methods: Best Subset Selection, Stepwise Selection (Forward and Backward) and Cross Validation
- Steps: refer Algorithm 6.1, 6.2, 6.3.
- Keys: The criterion for different model of fixed model size: RSS or R^2. The criteria for the best among different model size: Cp (AIC), BIC, or adjusted R^2. Install packages leaps.

Pre-processing

```
library(ISLR)
names(Hitters)
dim(Hitters)
sum(is.na(Hitters$Salary))   #total number of missing salary ("NA")
Hitters=na.omit(Hitters) #remove rows with missing values in any variable
dim(Hitters)
sum(is.na(Hitters))
```

Best Subset Selection

```
install.packages("leaps")
library(leaps)
regfit.full = regsubsets(Salary~.,Hitters)
##print the best set of predictors for each model size; by
default, only return results up to the best 8-predictor model
summary(regfit.full)
##to return as many predictors as specified(Max=19)
regfit.full = regsubsets(Salary~.,data=Hitters,nvmax=19)
reg.summary = summary(regfit.full)
names(reg.summary)
```

Best Subset Selection

```
##create figure contains four subfigure(2*2)
par(mfrow=c(2,2))
#Figure 1
plot(reg.summary$rss,xlab="Number of predictors",ylab="RSS",type="l")
#Figure 2
plot(reg.summary$adjr2,xlab="Number of predictors",ylab="Adjusted RSq",type="l")
a=which.max(reg.summary$adjr2)      #highlight maximizer
points(a,reg.summary$adjr2[a], col="red",cex=2,pch=20)
```

Best Subset Selection

```
#Figure 3
plot(reg.summary$cp,xlab="Number of
predictors",ylab="Cp",type='l')
b=which.min(reg.summary$cp)
points(b,reg.summary$cp[b],col="red",cex=2,pch=20)
#Figure 4
plot(reg.summary$bic,xlab="Number of
predictors",ylab="BIC",type='l')
c=which.min(reg.summary$bic)
points(c,reg.summary$bic[c],col="red",cex=2,pch=20)

##print the coefficient estimates of the best model by BIC
coef(regfit.full,c)
```

Stepwise Selection

```
regfit.fwd=regsubsets(Salary~.,data=Hitters,nvmax=19,
method="forward")
summary(regfit.fwd)
#summary(regfit.fwd)$bic(or cp, adjr2)
regfit.bwd=regsubsets(Salary~.,data=Hitters,nvmax=19,
method="backward")
summary(regfit.bwd)
##print the coefficient estimates of the 7-predictor model
coef(regfit.full,7)
coef(regfit.fwd,7)
coef(regfit.bwd,7)
```

CV for model selection

```
##Randomly split data into a training set and a test set
set.seed(1)
train=sample(c(TRUE,FALSE), nrow(Hitters), rep=TRUE)
test=(!train)
##Perform best subset selection
regfit.best=regsubsets(Salary~.,data=Hitters[train,],nvmax=19)
##building an "X" matrix from test data
test.mat=model.matrix(Salary~.,data=Hitters[test,])
```

CV for model selection

```
##Compute test MSE of the 19 models(size from 1 to 19)
val.errors=rep(NA,19)
for(i in 1:19){
coefi=coef(regfit.best,id=i)
pred=test.mat[,names(coefi)]%*%coefi          #matrix product
val.errors[i]=mean((Hitters$Salary[test]-pred)^2)
}
val.errors
```

CV for model selection

```
##Find the best model
best_size=which.min(val.errors)
coef(regfit.best,best_size)

##after finding the best model, we need to fit this model using
the full data set to obtain more accurate coefficient estimates
regfit.best=regsubsets(Salary~.,data=Hitters,nvmax=19)
coef(regfit.best,best_size)
```