SDSC 3006 L02
Class 5. Bootstrap and model selection

Name: Yiren Liu
Email: yirenliu2-c@my.cityu.edu.hk

School of Data Science
City University of Hong Kong

Outline

- **Bootstrap**
- **Model Selection**

Bootstrap

What is Bootstrap?

How Bootstrapping Resamples Your Data to Create Simulated Datasets

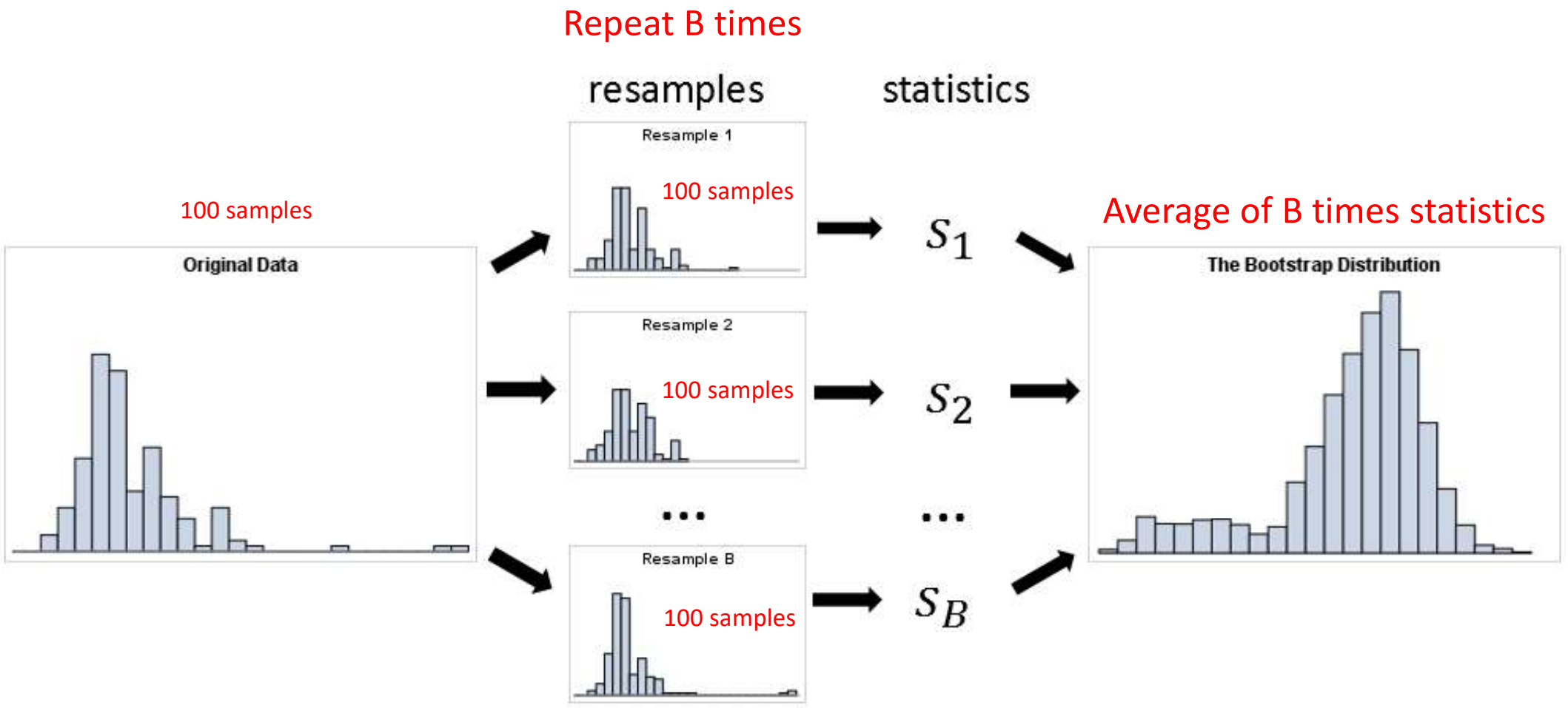
Bootstrapping resamples the original dataset with replacement many thousands of times to create simulated datasets. This process involves drawing random samples from the original dataset. Here's how it works:

1. The bootstrap method has an equal probability of randomly drawing each original data point for inclusion in the resampled datasets.
2. The procedure can select a data point more than once for a resampled dataset. This property is the "with replacement" aspect of the process.
3. The procedure creates resampled datasets that are the same size as the original dataset.

The process ends with your simulated datasets having many different combinations of the values that exist in the original dataset. Each simulated dataset has its own set of sample statistics, such as the mean, median, and standard deviation. Bootstrapping procedures use the distribution of the sample statistics across the simulated samples as the sampling distribution.

Reference: Jim Frost, Introduction to Bootstrapping in Statistics with an Example. URL: https://statisticsbyjim.com/hypothesis-testing/bootstrapping/

Bootstrap


Repeat B times
resamples
statistics
100 samples
Original Data
Resample 1
100 samples
s_1
Resample 2
100 samples
s_2
...
...
Resample B
100 samples
s_B
Average of B times statistics
The Bootstrap Distribution

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.
- Give a try for 10 samples from uniform(-5,5). Try to estimate the mean.

```
> set.seed(1)
> sample(100,100,replace = T)
  [1]  27  38  58  91  21  90  95  67  63   7  21  18  69  39  77  50  72 100  39  78  94
 [22]  22  66  13  27  39   2  39  87  35  49  60  50  19  83  67  80  11  73  42  83  65
 [43]  79  56  53  79   3  48  74  70  48  87  44  25   8  10  32  52  67  41  92  30  46
 [64]  34  66  26  48  77   9  88  34  84  35  34  48  90  87  39  78  97  44  72  40  33
 [85]  76  21  72  13  25  15  24   6  65  88  78  80  46  42  82  61
```

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.

Algorithms

Algorithm 6.1 *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Algorithm 6.2 *Forward stepwise selection*

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   (a) Consider all $p - k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   (b) Choose the *best* among these $p - k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Algorithms

Algorithm 6.3 *Backward stepwise selection*

1. Let $\mathcal{M}_p$ denote the *full* model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the *best* among these k models, and call it $\mathcal{M}_{k-1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

What is AIC, BIC?

- The Akaike information criterion (AIC) and the Bayesian information criterion (BIC) **provide measures of model performance that account for model complexity**. AIC and BIC combine a term reflecting how well the model fits the data with a term that penalizes the model in proportion to its number of parameters.

What is AIC, BIC?

$$\mathrm{AIC}_i = -2\log L_i + 2p_i$$

$$\mathrm{BIC}_i = -2\log L_i + p_i \log n$$

L = Max likelihood: the parameter with the highest probability of correctly representing the relationship between the input and output

p = the number of parameters

n = the number of values in the data set

What is Maximum likelihood (L)

- Maximum likelihood is the probability that the predicted parameter is an accurate representation of the relationship between the independent and dependent variable. For example, what is the likelihood that our predicted gradient is correct? Maximum likelihood is derived from the comparison of all the possible parameters followed by the selection of the parameter that has the highest chance of being correct. In both formulas, the natural log (ln) of the maximum likelihood is used as the measurement of the model's accuracy in explaining the sample data.

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"), install it if not installed
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.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)