Stat 361 - Recitation 11

Bootstrap - Jackknife

Orçun Oltulu

07 - 08 / 05 / 2020

Basic Bootstrap

The bootstrap is a method of Monte Carlo simulation where no parametric assumptions are made about the underlying population that generated the random sample.

We use the sample as an estimate of the population. This estimate is called the emprical distribution $\hat{F}$ where each x_i has probability mass $\frac{1}{n}$. Thus, each x_i has the same likelihood of being selected in the new sample from $\hat{F}$

When we use $\hat{F}$ as our pseudo-population, then we **resample with replacement** from the original sample.

To get an idea about the distribution of desired statistic, we obtain B bootstrap samples by sampling with replacement from the original sample. Then calculate this statistic for each bootstrap sample.

Then, these bootstrap replicates provide us with an estimate of the distribution of desired statistic.

Procedure

1. Given a random sample, $x = (x_1, x_2, ..., x_n)$, calculate $\hat{\theta}$.
2. Sample with replacement from the original sample to get $x^{*^b} = (x_1^{*^b}, x_2^{*^b}, ..., x_n^{*^b})$
3. Calculate the same statistic using the bootstrap sample in step 2 to get $\hat{\theta}_b^*$.
4. Repeat steps 2 through 3, B times.
5. Use this estimate of the distribution of $\hat{\theta}$ (i.e.the bootstrap replicates, $\hat{\theta}^{*^b}$) to obtain the desired characteristic (eg. standard error, bias or confidence interval).

Question 1:

Use Iris dataset (which is available in R) to estimate the skewness of the Sepal.Length of the population. Then, we estimate the standard error and bias of this statistic using the bootstrap method.

You may need **moments** package while you are calculating skewness.

Steps to follow

1. Obtain the skewness of the given sample, $\hat{\theta}$.
2. Resample from the original sample. (sample(..., replace = T))
3. Calculate the parameter of interest from bootstrap sample, say $\hat{\theta}^*$.
4. Repeat steps 2 and 3, B times .
5. Estimate the standard error of $\hat{\theta}$ by using

$$\hat{SE}_B(\hat{\theta}) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(\hat{\theta}_b^* - \bar{\hat{\theta}}^*)^2}$$

where $\bar{\hat{\theta}}^* = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}_b^*$

Estimate the bias of $\hat{\theta}$ by

$$\text{bias}(T) = E(T) - \theta$$

$$\hat{\text{bias}}_B = \bar{\hat{\theta}}^* - \hat{\theta}$$

```
#install.packages("moments")
library(moments)
#load iris dataset
data("iris")
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
B <- 1000 # bootstrap replicates
Skew <- numeric(B)
```

```r
sample_skew <- skewness(iris$Sepal.Length)

for(i in 1:B){
  index <- sample(1:nrow(iris), size = nrow(iris),
                  replace = TRUE)
  x <- iris[index,"Sepal.Length"]
  Skew[i] <- skewness(x)

}

est_skew <- mean(Skew)
se_skew <- sd(Skew)
bias_skew <- est_skew - sample_skew

out <- c(sample_skew, est_skew, se_skew, bias_skew)
names(out) <- c("sample", "estimated", "std.error", "bias")
out
```

```
##        sample     estimated     std.error          bias
##  0.3117530585  0.3115863043  0.1331254449 -0.0001667542
```

Basic Jackknife

Procedure

1. Given a random sample, $x = (x_1, x_2, ..., x_n)$, calculate $\hat{\theta}$.
2. Let $x_{(i)}$ be the sample but with i^{th} observation removed:

$$x_{(i)} = (x_1, ..., x_{i-1}, x_{i+1}, ..., x_n)$$

3. Calculate the same statistic using the jackknife sample in step 2 to get $\hat{\theta}_{(i)}$.
4. Repeat steps 2 through 3, n times.
5. Use this estimate of the distribution of $\hat{\theta}$ to obtain the desired characteristic (eg. standard error, bias or confidence interval).

Further reading

Question 2:

Again use Iris dataset to estimate the skewness of the Sepal.Length of the population. Then, we estimate the standard error and bias of this statistic using the jackknife method.

Hint:

The jackknife estimate of the standard error is

$$\widehat{SE}_J(\hat{\theta}) = \sqrt{\frac{n-1}{n}\sum_{i=1}^{n}(\hat{\theta}_i - \bar{\theta})^2}$$

The jackknife estimate of the bias is

$$\text{bias}(\theta) = (n-1) * (\bar{\theta} - \hat{\theta})$$

```
Skew_jack <- numeric(nrow(iris))
sample_skew <- skewness(iris$Sepal.Length)

for(i in 1:nrow(iris)){

   Skew_jack[i] <- skewness(iris$Sepal.Length[-i])

}

est_skew2 <- mean(Skew_jack)
se_skew2 <- sqrt((nrow(iris) - 1) * mean((Skew_jack - est_skew2)^2))
bias_skew2 <- (nrow(iris)-1) * (est_skew2 - sample_skew)

out2 <- rbind(out,c(sample_skew, est_skew2, se_skew2, bias_skew2))
rownames(out2) <- c("bootstrap", "jackknife")
out2
```

```
##              sample estimated std.error          bias
## bootstrap 0.3117531 0.3115863 0.1331254 -0.0001667542
## jackknife 0.3117531 0.3117041 0.1315430 -0.0072903956
```

```
library(bootstrap)
alternative <- jackknife(iris$Sepal.Width, skewness)
alternative$jack.se
```

```
## [1] 0.1922494
```

```
alternative$jack.bias
```

```
## [1] -0.0112776
```

Question 3:

Use Iris data set to estimate the correlation between Sepal.Length and Petal.Length, and compute the bootstrap estimate of the standard errror and bias of the sample correlation.

```r
sample_cor <- cor(iris$Sepal.Length, iris$Petal.Length)

B <- 1000
Corr <- numeric(B)

for(i in 1:B){

  index <- sample(1:nrow(iris), size = nrow(iris),
                  replace = TRUE)
  x <- iris[index, c("Sepal.Length","Petal.Length")]
  Corr[i] <- cor(x)[1,2]
}

est_cor <- mean(Corr)
se_cor <- sd(Corr)
bias_cor <- est_cor - sample_cor

out <- c(sample_cor, est_cor, se_cor, bias_cor)
names(out) <- c("sample", "estimated", "std.error","bias")
out
```

```
##        sample     estimated     std.error          bias
##  0.8717537759  0.8710998222  0.0172349135 -0.0006539537
```

Additionally, do it with boot function.

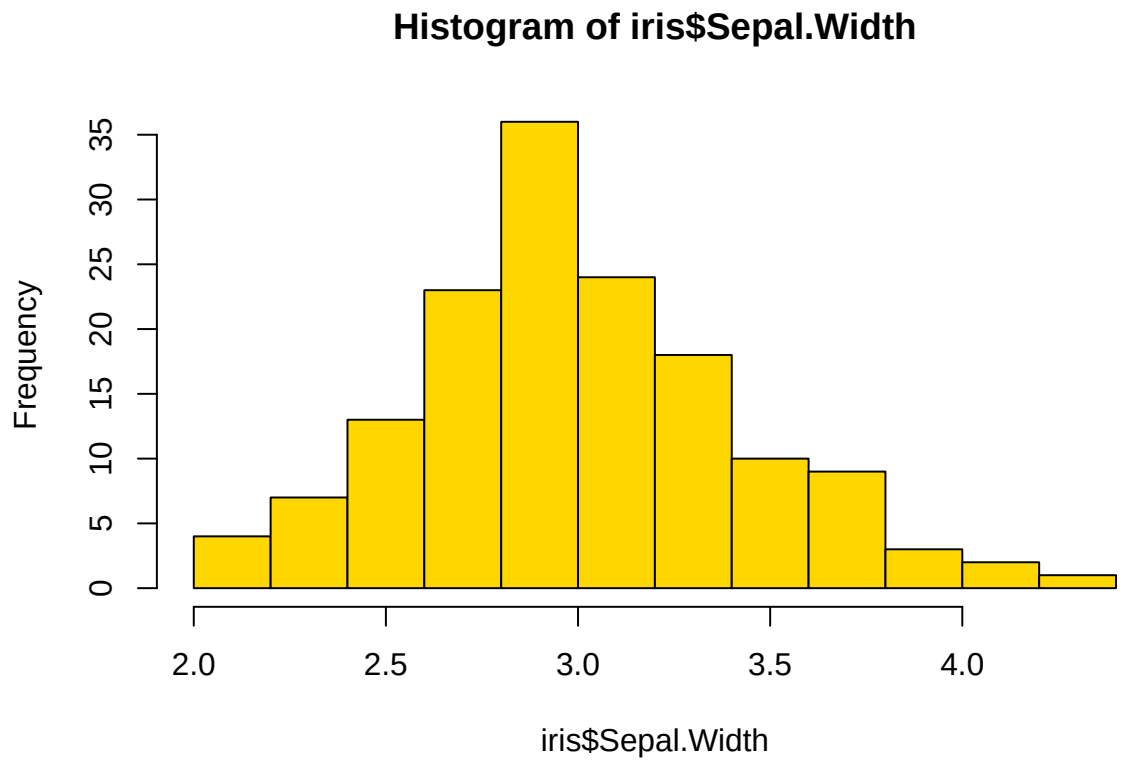
```r
library(boot)

corr_function<- function(x,i){
  cor(x[i,"Sepal.Length"],x[i,"Petal.Length"])
}

boot(data = iris, statistic = corr_function, R = 1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = iris, statistic = corr_function, R = 1000)
##
##
## Bootstrap Statistics :
##      original       bias    std. error
## t1* 0.8717538 0.0003156776  0.01660071
```

Question 4:

Construct a 95% Confidence Interval for the mean of the Sepal.Width.

```
hist(iris$Sepal.Width, col = "Gold")
```




```
shapiro.test(iris$Sepal.Width)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  iris$Sepal.Width
## W = 0.98492, p-value = 0.1012
```

```
sample_avg <- mean(iris$Sepal.Width)
B <- 1000
avg <- numeric(B)

alpha <- .05

for(i in 1:B){
  index <- sample(1:nrow(iris), size = nrow(iris),
                  replace = TRUE)
```

```
  x <- iris[index, "Sepal.Width"]
  avg[i] <- mean(x)
}

est_avg <- mean(avg)
se_avg <- sd(avg)

lower <- sample_avg - qnorm(1-alpha/2, 0,1) * se_avg
upper <- sample_avg + qnorm(1-alpha/2, 0,1) * se_avg

out <- c(lower, upper)
names(out) <- c("Lower", "Upper")
out
```

```
##    Lower    Upper
## 2.985891 3.128776
```

Construct also Percentile Interval for the mean of Sepal.Width

```
sample_avg <- mean(iris$Sepal.Width)
B <- 1000
avg <- numeric(B)

alpha <- .05

for(i in 1:B){
  index <- sample(1:nrow(iris), size = nrow(iris),
                  replace = TRUE)
  x <- iris[index, "Sepal.Width"]
  avg[i] <- mean(x)
}

lower <- quantile(avg, alpha/2)
upper <- quantile(avg, 1-alpha/2)

out <- c(lower, upper)
names(out) <- c("Lower", "Upper")
out
```

```
##    Lower    Upper
## 2.989983 3.124683
```

Cross Validation & Model Selection

Cross validation is a data partitioning method that can be used to assess the stability of parameter estimates, the accuracy of a classification algorithm, the adequacy of a fitted model, and in many other applications. The jackknife could be considered a special case of cross validation, because it is primarily used to estimate bias and standard error of an estimator.

In building a classifier, a researcher can partition the data into training and test sets. The model is estimated using the data in the training set only, and the misclassification rate is estimated by running the classifier on the test set.

Another version of cross validation is the "n-fold" cross validation, which partitions the data into n test sets (now test points). This "leave-one-out" procedure is like the jackknife. The data could be divided into any number K partitions, so that there are K test sets. Then the model fitting leaves out one test set in turn, so that the models are fitted K times.

Note: There are several steps to model selection, but we will focus on the prediction error. The prediction error can be estimated by cross validation, without making strong distributional assumptions about the error variable.

Steps to follow:

1. Remove k^{th} $\quad k = 1, ..., n$ observation in order to be the test point, then fit a model with $n-1$ observation, (x_i, y_i), $\quad i \neq k$.
2. Compute the predicted response $\hat{y}_k = \hat{\beta}_0 + \hat{\beta}_1 x_{1k} + ... + \hat{\beta}_p x_{pk}$
3. Compute the predicton error $e_k = y_k - \hat{y}_k$
4. Estimate the mean of the squared errors $\hat{\sigma}^2_\varepsilon = \frac{1}{n} \sum_{k=1}^{n} e_k^2$

Question 5:

For this question, use ISLR package and load 'Auto' dataset into R, get only mpg, horsepower and weight.

Use 'mpg' as response variable, and in order to estimate 'mgp' use horsepower and weight.

Apply Leave-one-out Cross Validation technique to find the best model among those 5 models;

1. Linear: $Y_{mpg} = \beta_0 + \beta_1 X_{hp} + \beta_2 X_{year} + \beta_3 X_w + \varepsilon$
2. Add interaction effect: $Y_{mpg} = \beta_0 + \beta_1 X_{hp} + \beta_2 X_{year} + \beta_3 X_w + \beta_4 X_{year} X_w + \varepsilon$
3. 2^{nd} order Polynomial: $Y_{mpg} = \beta_0 + \beta_1 X_{hp} + \beta_2 X_{year} + \beta_3 X_w + \beta_4 X_w^2 + \varepsilon$
4. 3^{rd} order Polynomial: $Y_{mpg} = \beta_0 + \beta_1 X_{hp} + \beta_2 X_{year} + \beta_5 X_w + \beta_4 X_w^2 + \beta_5 X_w^3 + \varepsilon$

```r
library(ISLR)
data(Auto)
Auto <- Auto[,c("mpg","horsepower","weight", "year")]
Auto <- as.data.frame(scale(Auto))
head(Auto,10)
```

```
##           mpg horsepower    weight      year
## 1  -0.6977467  0.6632851 0.6197483 -1.623241
## 2  -1.0821153  1.5725848 0.8422577 -1.623241
## 3  -0.6977467  1.1828849 0.5396921 -1.623241
## 4  -0.9539925  1.1828849 0.5361602 -1.623241
## 5  -0.8258696  0.9230850 0.5549969 -1.623241
## 6  -1.0821153  2.4299245 1.6051468 -1.623241
## 7  -1.2102382  3.0014843 1.6204517 -1.623241
## 8  -1.2102382  2.8715843 1.5710052 -1.623241
## 9  -1.2102382  3.1313843 1.7040399 -1.623241
## 10 -1.0821153  2.2220846 1.0270935 -1.623241
```

```r
attach(Auto)
models <- list(as.formula(mpg ~ horsepower + year + weight),
               as.formula(mpg ~ horsepower + year * weight),
               as.formula(mpg ~ horsepower + year + poly(weight,2)),
               as.formula(mpg ~ horsepower + year + poly(weight,3)))

errors <- matrix(0, ncol = length(models),
                         nrow = nrow(Auto))
colnames(errors) <- c(paste("model",1:4,sep=""))

for(i in 1:nrow(Auto)){
  for(j in 1:length(models)){
    model <- lm(models[[j]], data = Auto[-i,])
    errors[i,j] <- predict(model, Auto[i,]) - Auto$mpg[i]
  }
}
detach(Auto)

comparison <- apply(errors^2,2,mean)
comparison
```

```
##    model1    model2    model3    model4
## 0.1953273 0.1663312 0.1512112 0.1516711
```

```r
models[[which.min(comparison)]]
```

```
## mpg ~ horsepower + year + poly(weight, 2)
```

```r
plot(comparison, type = "b", lwd = 2,
     pch = 1:length(models))
legend("topright", legend = models,
       pch = 1:length(models))
```