The Bootstrap Estimate for Standard Error

Zihan Zhang*

March 26, 2019

Abstract

Bootstrap is a simple method used to estimate the standard error of an unknown population with only a sample. In this note, I would like to write an brief introduction about the Bootstrap as well as its realization with some other softwares except for R.

1 Nonparametric Bootstrap

Bootstrap, firstly introduced in 1979, provides a relatively simple way to calculate the standard error of the estimator just based on the sample we known. Actually, Bootstrap that we have discussed in class is the non-parametric one, which means it does not be involved in any theoretical derivations. It enjoys the advantages of being being able to finish the whole process by computers themselves.

Suppose we have an underlying population F with a coefficient θ describing its feature and both of them are unknown. Also, we have a sample $\mathbf{x} = (x_1, x_2, ..., x_n)$ comprising some individuals (x_i) generated from F. In statistics, we are able to estimate θ with the estimator $\hat{\theta} = s(\mathbf{x})$ through some derivations. The Bootstrap was introduced to measure the accuracy of $\hat{\theta}$.

From the known sample, we are able to generate the Bootstrap sample, which is defined to be a random sample of size n with replacement from the known sample $\mathbf{x}$. It could be donated as $\mathbf{x}^* = (x_1^*, x_2^*, ..., x_n^*)$, where the star notation manifest the $\mathbf{x}^*$ is not the actual set of $\mathbf{x}$, but rather a randomized and replaced version of initial sample $\mathbf{x}$.

Next, we would like to evaluate the Bootstrap replication corresponding to each Bootstrap sample and calculate the $\hat{\theta(i)} = s(\mathbf{x}^{*i})$ for each Bootstrap sample. Having those done, we are able to estimate the stand error by the sample standard deviation of N replications. The equation is as follows.

$$\widehat{SE_N} = \sqrt{\frac{\sum_{i=1}^{N}[\widehat{\theta^*(i)} - \widehat{\theta^*(\bullet)})]^2}{N-1}} \tag{1}$$

where $\theta^*(\bullet) = \frac{1}{N}\sum_{i=1}^{N}\theta^*(i)$.

---
*2016 WISE IUEC, Student ID: 15220162202518

2 Translate the code into MATLAB language

The instructor provides a method of realization of the Bootstrap in R, here I would like to make it available in MATLAB.

The program could be written in function, which is as follows.

```matlab
1  function main_bootstrap()
2  %% import the data from R, "Portfolio" is the same dataset from R.
3  Portfolio=xlsread('Portfolio.xlsx')
4
5  %% generate one bootstrap sample and calculate the alpha.
6  index=randint(1,length(Portfolio),[1 length(Portfolio)]);
7  alpha = return_alpha(Portfolio,index)
8
9  %% do bootstrap manually.
10 A=[]
11 B=1000
12 for i=1:B
13 A(i)=return_alpha(Portfolio,randint(1,length(Portfolio),[1 ...
      length(Portfolio)]));
14 end
15
16 mu = mean(A)
17 sd = std(A)
18
19 %% This is the function used for computing alpha.
20     function alpha= return_alpha(data,index)
21         x=data(index,1);
22         y=data(index,2);
23         c=cov(x,y)
24         alpha=((var(y)-c(1,2))/(var(x)+var(y)-2*c(1,2)))
25     end
26 end
```

Also, we can realize it without matlab functions.

```matlab
1  clear
2  Portfolio=xlsread('Portfolio.xlsx')
3  index=randint(1,length(Portfolio),[1 length(Portfolio)]);
4
5  A=[]
6  B=1000
7  for i=1:B
8      index=randint(1,length(Portfolio),[1 length(Portfolio)]);
9      x=data(index,1);
10     y=data(index,2);
11     c=cov(x,y)
12     alpha=((var(y)-c(1,2))/(var(x)+var(y)-2*c(1,2)))
13     A(i)=alpha;
14 end
15
```

```
16  mu = mean(A)
17  sd = std(A)
```

```
mu =

    0.5804


sd =

    0.0882
|
```

```
> mean(A)
[1] 0.5801821
> sd(A)
[1] 0.09119144
```

Figure 1: Result from MATLAB   Figure 2: Result from R

It is evident that the results are similar.

References

[1] Bayes Theorem, https://en.wikipedia.org/wiki/Bayes%27_theorem.

[2] Naive Bayes Classifier, https://en.wikipedia.org/wiki/Naive_Bayes_classifier.

[3] Jiaming Mao, Foundations of Statistical Learning,
https://jiamingmao.github.io/data-analysis/assets/Lectures/Foundations_of_Statistical_Learning.pdf