Big Data and Better Decisions
Spring 2018

Module 11: Bootstrapping

1. Introduction to Bootstrapping

Bootstrapping is a statistical tool that we can use to quantify uncertainty associated with an estimator or statistical learning method. For a simple example, we can use it to estimate the standard errors for linear regression coefficients. Of course, R calculates standard errors automatically for us, so we would likely not need to use bootstrapping for a simple OLS regression. However, the same principal can be applied to other circumstances in which a measure of variability would be difficult to obtain and not automatically outputted by R.

The following will draw heavily from ISL Section 5.2, and it is recommended that you consult this chapter for a more in-depth discussion of bootstrapping. We'll use a conceptual example to illustrate the procedure for bootstrapping. Consider a situation where you would like to invest $0 \leq \alpha \leq 1$ fraction of your funds in a financial asset which yields the random quantity X, and the remaining amount $1 - \alpha$ in a different financial asset which yields the random quantity Y. Since there is variability associated with the returns on the two assets, we want to pick α to minimize the total risk or variance of our investment. So, we want to minimize $Var(\alpha X + (1 - \alpha)Y)$. It's possible to show that the value that minimizes the variance is equivalent to

$$\alpha = \frac{Var(Y) - Cov(X, Y)}{Var(X) + Var(Y) - 2Cov(X, Y)}$$

Let's start with a situation where we know the true underlying value of the variances for each of the random variables X and Y:

$$Var(Y) = \sigma_Y^2 = 1.25$$
$$Var(X) = \sigma_X^2 = 1$$
$$Cov(XY) = \sigma_{XY} = 0.5$$

Here, we can calculate the value of alpha, the optimal portion of our income to invest in X, exactly as 0.6. However, in the real world, we don't know the true underlying variances, so we must use past observations of X and Y to estimate the variances, and ultimately to estimate the optimal alpha. Suppose we are able to observe a random sample of 100 past observations for X and Y and calculate $Sample\ Var(Y) = \hat{\sigma}_Y^2, Sample\ Var(X) = \hat{\sigma}_Y^2, Sample\ Cov(X) = \hat{\sigma}_{XY}$ for this set of data. We have

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}$$

(1)

So, for one sample, you obtain a single value of $\hat{\alpha}$. In this case, let's say that you obtain $\hat{\alpha} = 0.576$ based on the sample variances from your sample of 100 observations. Without knowing the true underlying value of alpha, how can you know how accurate this estimate is? For your investment purposes, you might want to know the **standard error** of your estimate for alpha, or how much you expect $\hat{\alpha}$ to differ from the true α on average. One way to calculate this standard error would be to draw more samples of 100 observations for X and Y, estimating $\hat{\alpha}$ each time, and then take the standard deviations of these estimates. In a simulated example from ISL, 1000 samples of 100 observations for X and Y are drawn, with the $\hat{\alpha}$ calculated each time. The average of these $\hat{\alpha}$ is calculated:

$$\bar{\alpha} = \frac{1}{1000}\sum_{r=1}^{1000} \hat{\alpha}_r = 0.5996$$

Which is remarkably close to the true value of alpha. The standard deviation of the estimates of $\hat{\alpha}$ is:

$$SE(\hat{\alpha}) = \sqrt{\frac{1}{1000-1}\sum_{r=1}^{1000} (\hat{\alpha}_r - \bar{\alpha})^2} = 0.083$$

This means that if we just relied on one random sample of size 100 from the population to estimate alpha, we would expect our estimate to differ from the true alpha by 0.08 on average. But how can we calculate the standard error of our estimate if we have only one dataset of 100 observations to analyze? It seems a silly exercise to demonstrate calculating the standard error for a sample of 100 observations by sampling additional data. If we had this data available, why not use it to get a better estimate of $\hat{\alpha}$?

However, we can modify this procedure to allow us to calculate the standard error of $\hat{\alpha}$ using ONLY our original dataset of 100 observations. Rather than repeatedly obtaining independent data sets from the population, we instead obtain distinct data sets by repeatedly sampling observations *from the original data set* **with replacement**. In this case, imagine that we take our 100 observations of X and Y and write each pair of X and Y on a slip of paper, so we have 100 total slips in a bucket. Then we randomly draw a slip, write down the value of X and Y, and toss the slip back into bucket. We repeat this until we have 100 total observations in a new randomly drawn sample. If we compare this new randomly drawn sample to our original, we might have, for example, 3 copies of observation 27, and no copies of observation 50. We'll calculate alpha in this new sample and call that $\hat{\alpha}^{*1}$. Then we'll repeat this procedure, drawing a new sample with replacement from our original sample, and calculating alpha again. We repeat a total of 1000 times, so we have $\hat{\alpha}^{*1}, \hat{\alpha}^{*2} \ldots \hat{\alpha}^{*1000}$, all based on our original sample. Then we take these values, as well as their average $\bar{\alpha}^* = \frac{1}{1000}\sum_{r'=1}^{1000} \hat{\alpha}^{*r'}$, we can calculate the standard error for our original $\hat{\alpha}$ using the formula:

$$SE(\hat{\alpha}) = \sqrt{\frac{1}{1000-1}\sum_{r=1}^{1000} (\hat{\alpha}^{*r} - \bar{\alpha}^*)^2}$$

You can see that this is very similar to the formula we used above to calculate the SE from distinct samples of the full population. In a simulation of bootstrapping from a single sample, we calculate the SE as 0.087. See the figure below from ISL illustrating these two processes:

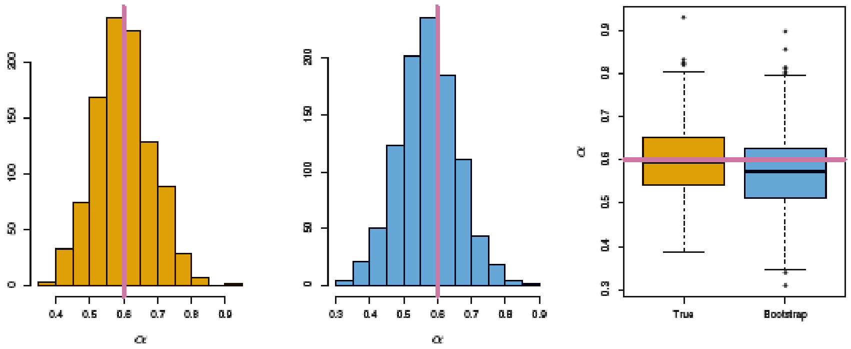

FIGURE 5.10. Left: *A histogram of the estimates of α obtained by generating 1,000 simulated data sets from the true population.* Center: *A histogram of the estimates of α obtained from 1,000 bootstrap samples from a single data set.* Right: *The estimates of α displayed in the left and center panels are shown as boxplots. In each panel, the pink line indicates the true value of α.*

2. IMPLEMENTATION IN R

We will practice two examples of bootstrapping reproduced from section 5.3.4 of ISL, which use the Portfolio and Auto datasets (loadable from JupyterHub). The first example replicates bootstrapping of the standard error for the statistic of interest above: variance in portfolio returns.

Estimating the Accuracy of a Statistic of Interest

One of the great advantages of the bootstrap approach is that it can be applied in almost all situations. No complicated mathematical calculations are required. Performing a bootstrap analysis in R entails only two steps. First, we must create a function that computes the statistic of interest. Second, we use the `boot()` function, which is part of the boot library, to perform the bootstrap by repeatedly sampling observations from the data set with replacement.

We will load the Portfolio data described in the example above. To illustrate the use of the bootstrap on this data, we must first create a function, `alpha.fn()`, which takes as input the (X, Y) data as well as a vector indicating which observations should be used to estimate α. The function then outputs the estimate for α based on the selected observations.

```
> 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)))
> }
```

This function returns, or outputs, an estimate for α based on applying formula (1) above to the observations indexed by the argument `index`. For instance, the following command tells R to estimate α using all 100 observations.

```
> alpha.fn(Portfolio ,1:100)
[1] 0.576
```

The next command uses the `sample()` function to randomly select 100 observations from the range 1 to 100, with replacement. This is equivalent to constructing a new bootstrap data set and recomputing $\hat{\alpha}$ based on the new data set.

```
> set.seed(1)
> alpha.fn(Portfolio ,sample (100,100, replace=T))
[1] 0.596
```

We can implement a bootstrap analysis by performing this command many times, recording all of the corresponding estimates for α, and computing the resulting standard deviation. However, the `boot()` function automates this approach. Below we produce R = 1000 bootstrap estimates for α.

```
> boot(Portfolio ,alpha.fn,R=1000

ORDINARY NONPARAMETRIC BOOTSTRAP
Call: boot(data = Portfolio , statistic = alpha.fn, R = 1000)
Bootstrap Statistics :
     original   bias        std. error
t1* 0.5758     -7.315e-05 0.0886
```

The final output shows that using the original data, $\hat{\alpha}$= 0.5758, and that the bootstrap estimate for SE($\hat{\alpha}$) is 0.0886.

Estimating the Accuracy of a Linear Regression Model

The bootstrap approach can be used to assess the variability of the coefficient estimates and predictions from a statistical learning method. Here we use the bootstrap approach in order to assess the variability of the estimates for β_0 and β_1, the intercept and slope terms for the linear regression model that uses `horsepower` to predict `mpg` in the Auto data set. We will compare the estimates obtained using the bootstrap to those obtained using the formulas for $SE(\hat{\beta}_0)$ and $SE(\hat{\beta}_1)$:

$$SE(\hat{\beta}_0)^2 = \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right], \quad SE(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

These formulas require the assumption that the errors for ϵ_i for each observation are uncorrelated with common variance σ^2. We first create a simple function, `boot.fn()`, which takes in the Auto data set as well as a set of indices for the observations, and returns the intercept and slope estimates for the linear regression model. We then apply this function to the full set of 392 observations in order to compute the estimates of β₀ and β₁ on the entire data set using the usual linear regression coefficient estimate formulas. Note that we do not need the “{“ and “}” at the beginning and end of the function because it is only one line long.

```
> boot.fn=function (data ,index)
> return(coef(lm(mpg~horsepower ,data=data , subset=index)))
> boot.fn(Auto ,1:392)
(Intercept )     horsepower
39.936           -0.158
```

The `boot.fn()` function can also be used in order to create bootstrap estimates for the intercept and slope terms by randomly sampling from among the observations with replacement. Here we give two examples.

```
> set.seed(1)
> boot.fn(Auto ,sample (392,392, replace=T))
(Intercept )     horsepower
 38.739          -0.148
> boot.fn(Auto ,sample (392,392, replace=T))
(Intercept )     horsepower
40.038           -0.160
```

Next, we use the boot() function to compute the standard errors of 1,000 bootstrap estimates for the intercept and slope terms.

```
> boot(Auto ,boot.fn ,1000)
ORDINARY NONPARAMETRIC BOOTSTRAP
Call:
boot(data = Auto , statistic = boot.fn, R = 1000)
Bootstrap Statistics :
     original      bias std.      error
t1*   39.936       0.0297        0.8600
t2*    -0.158     -0.0003        0.0074
```

This indicates that the bootstrap estimate for $SE(\hat{\beta}_0)$ is 0.86, and that the bootstrap estimate for $SE(\hat{\beta}_1)$ is 0.0074. The standard formulas above can be used to compute the standard errors for the regression coefficients in a linear model. These can be obtained using the `summary()` function.

```
        > summary (lm(mpg~horsepower ,data=Auto))$coef
                  Estimate   Std. Error      t value    Pr(>|t|)
(Intercept)       39.936     0.71750         55.7       1.22e-187
horsepower        -0.158     0.00645         -24.5      7.03e-81
```

Interestingly, these are somewhat different from the estimates obtained using the bootstrap. Does this indicate a problem with the bootstrap? In fact, it suggests the opposite. As stated above, the standard error formulas rely on certain assumptions. For example, they depend on the unknown parameter σ^2 , the noise variance. We then estimate σ^2 using the RSS. Now although the formula for the standard errors do not rely on the linear model being correct, the estimate for σ^2 does. It turns out that there is a non-linear relationship in the data, and so the residuals from a linear fit will be inflated, and so will $\hat{\sigma}^2$. Secondly, the standard formulas assume (somewhat unrealistically) that the x_i are fixed, and all the variability comes from the variation in the errors i. The bootstrap approach does not rely on any of these assumptions, and so it is likely giving a more accurate estimate of the standard errors of β_0 and β_1 than is the `summary()` function.

Below we compute the bootstrap standard error estimates and the standard linear regression estimates that result from fitting the quadratic model to the data. Since this model provides a better fit to the data in comparison to the linear model, there is now a better correspondence between the bootstrap estimates and the standard estimates of for $SE(\hat{\beta}_0)$, $SE(\hat{\beta}_1)$ and $SE(\hat{\beta}_2)$.

```
> boot.fn=function (data ,index)
>  coefficients(lm(mpg~horsepower +I(horsepower ^2),data=data  ,
subset=index))
> set.seed(1)
> boot(Auto ,boot.fn ,1000)
ORDINARY NONPARAMETRIC BOOTSTRAP
Call:
boot(data = Auto , statistic = boot.fn, R = 1000)
Bootstrap Statistics :
     original         bias std.       error
t1*   56.900          6.098e-03       2.0945
t2*   -0.466          -1.777e-04      0.0334
t3*    0.001          1.324e-06       0.0001
> summary (lm(mpg~horsepower +I(horsepower ^2),data=Auto))$coef
           Estimate   Std. Error  t value         Pr(>|t|)
(Intercept) 56.9001    1.80043    32         1.7e-109
horsepower  -0.4662    0.03112    -15        2.3e-40
I(horsepower ^2) 0.0012 0.00012   10         2.2e-21
```

Problem Set Questions 11.1

a) Use the lm function to run an OLS regression of weight (explanatory variable) on acceleration (outcome) using the Auto dataset. Report and interpret the standard errors and coefficients.
b) Bootstrap the standard errors. How do your estimates compare to those generated by lm?
c) Do you think the assumptions required to use the traditional standard errors are fulfilled? Use the results of (a) and (b) as well as a scatterplot of weight and acceleration to support your answer.

3. BIBLIOGRAPHY/FURTHER READING

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani (2013). An Introduction to Statistical Learning, Volume 112. Springer