Module 11
The Bootstrap (Part 2)

In this module we discuss some more details about the bootstrap (these details pertain to regression models, the two-sample problem, and the block bootstrap), and we also introduce a variation of the bootstrap called the Jackknife. All of these subjects are enumerated below.

1 More on the Bootstrap

1.1 The Bootstrap for Regression Models

Recall the standard regression model: $Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon$ where $\epsilon \sim N(0, \sigma^2)$. An easy way to write this is

$$Y = \mathbf{x}^T \boldsymbol{\beta} + \epsilon,$$

where $\mathbf{x} = (1, x_1, x_2, \ldots, x_p)^T$ and $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^T$. An obvious question is: "How is $\boldsymbol{\beta}$ estimated?" This involves, of course, collecting multiple, say n, observations (responses and covariates) and constructing the vector and matrix $\mathbf{y}$ and $\mathbf{X}$, where

$$\mathbf{y} = (y_1, y_2, \ldots, y_n)^T, \quad \textit{and} \quad \mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{pmatrix},$$

and with these construct the model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_n)^T \sim N\left(\mathbf{0}, \sigma^2 \mathbf{I}\right)$.

The estimated value of $\boldsymbol{\beta}$ is that value which minimizes the sum-of-squares, and this turns out to be

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T \mathbf{X}\right)^{-1} \mathbf{X}^T \mathbf{y}.$$

The variability of this estimator can be easily calculated with

$$\text{Var}\left(\hat{\boldsymbol{\beta}}\right) = \sigma^2 \left(\mathbf{X}^T \mathbf{X}\right)^{-1},$$

and this variability is estimated as

$$\widehat{\text{Var}}\left(\hat{\boldsymbol{\beta}}\right) = \hat{\sigma}^2 \left(\mathbf{X}^T \mathbf{X}\right)^{-1}.$$

But a question may come up regarding some function of the parameters. Assume, for example, that we are interested in $\theta = g(\beta_0, \beta_1, \beta_2)$ where g is some strange non-linear function. We can easily estimate θ with $\hat{\theta} = g\left(\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2\right)$, but how is it possible to estimate $\text{Var}\left(\hat{\theta}\right)$? This can be done using the Bootstrap, and there are two ways to do the Bootstrap.

You can bootstrap the pairs, in which case you let $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n)$ where $\mathbf{z}_j = (\mathbf{x}_j, y_j)$. In this case, the bootstrapping algorithm is:

`For` $j = 1 : B$

1. `Generate` j^{th} `Bootstrap sample` $\mathbf{Z}_j = \left(\mathbf{z}^*_{j,1}, \mathbf{z}^*_{j,2}, \ldots, \mathbf{z}^*_{j,n}\right)$ `where` $\mathbf{z}_{j,i}$ `is a value of` $\mathbf{Z}$ `randomly sampled (with replacement).`
2. `With this value of` $\mathbf{Z}_j$`, construct a corresponding vector of responses,` $\mathbf{y}^*_j$ `and a corresponding design matrix,` $\mathbf{X}^*_j$`, and calculate an estimate of` $\boldsymbol{\beta}$`. Call it` $\hat{\boldsymbol{\beta}}^*_j = \left(\mathbf{X}^{*,T}_j \mathbf{X}^*_j\right)^{-1} \mathbf{X}^{*,T}_j \mathbf{y}^*_j$.
3. `Calculate an estimate of` θ `from this value of` $\hat{\boldsymbol{\beta}}^*$`. Call ilt` $\hat{\theta}^*_j$.

`Can now estimate` $\text{Var}\left(\hat{\theta}\right)$ `with` $\widehat{\text{Var}}\left(\hat{\theta}\right) = \frac{1}{B-1} \sum_{j=1}^{B} \left(\hat{\theta}^*_j - \hat{\theta}(\cdot)\right)^2$.

You can also bootstrap the residuals. The algorithm for this is given below

Calculate the residuals $\hat{\boldsymbol{\epsilon}} = (\hat{\epsilon}_1, \hat{\epsilon}_2, \ldots, \hat{\epsilon}_n)$ where $\hat{\epsilon}_j = \hat{y}_j - \mathbf{x}_j^T \hat{\boldsymbol{\beta}}$.

For $j = 1 : B$

1. Generate j^{th} bootstrap sample of residuals, $\hat{\boldsymbol{\epsilon}}_j^* = \left(\hat{\epsilon}_{j,1}^*, \hat{\epsilon}_{j,2}^*, \ldots, \hat{\epsilon}_{j,n}^*\right)$ where $\hat{\epsilon}_{j,i}^*$ is a value from $\hat{\boldsymbol{\epsilon}}$ randomly sampled (with replacement).
2. Now consider the dataset $\{(\hat{y}_1^*, \mathbf{x}_1), (\hat{y}_2^*, \mathbf{x}_2), \ldots, (\hat{y}_n^*, \mathbf{x}_n)\}$ where $\hat{y}_i^* = \mathbf{x}_i^T \hat{\boldsymbol{\beta}} + \hat{\epsilon}_{j,i}^*$.
3. Calculate an estimate of $\boldsymbol{\beta}$ using this bootstrapped sample. Call it $\hat{\boldsymbol{\beta}}_j^* = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\hat{\mathbf{y}}^*$.
4. Can now estimate $\text{Var}\left(\hat{\theta}\right)$ with $\widehat{\text{Var}}\left(\hat{\theta}\right) = \frac{1}{B-1}\sum_{j=1}^{B}\left(\hat{\theta}_j^* - \hat{\theta}(\cdot)\right)^2$.

Which method is better? Is it better to bootstrap the pairs or the residuals. It turns out that when you believe the true regression model (that $\epsilon \sim N(0, \sigma^2)$) and that the distribution of error does not depend on the value of $\mathbf{x}$, then the errors can be transposale and the second method is better. If you're suspicious of this belief, though, then it is better to bootstrap the pairs.

1.2 The Two-Sample Problem

Let's assume you have two samples: $X_1, X_2, \ldots, X_n \sim f_X(x)$ and $Y_1, Y_2, \ldots, Y_m \sim f_Y(y)$, where $E(X_i) = \mu_X$ for all i and $E(Y_j) = \mu_Y$ for all j. And let's assume you want to construct a confidence interval for $\mu_X - \mu_Y$ (to see if the difference between the two means is significantly different from 0). In traditional statistics (assuming normality, etc.), a way to construct a confidence interval is simple. It is calculated as

$$\overline{X} - \overline{Y} \pm t_{n+m-2}\sqrt{\frac{s_X^2}{n} + \frac{s_Y^2}{m}}.$$

If the normality assumption is relaxed, however, we can't use this formula. We can, however, use the bootstrap. The algorithm for doing this is given below

For $j = 1 : B$

1. Generate bootstrap sample $\mathbf{X}_j^* = \left(X_{j,1}^*, X_{j,2}^*, \ldots, X_{j,n}^*\right)$ where $X_{j,i}^*$ is some value of the original X's randomly sampled (with replacement).
2. Generate bootstrap sample $\mathbf{Y}_j^* \left(U_{j,1}^*, Y_{j,2}^*, \ldots, Y_{j,m}^*\right)$ where $Y_{j,i}^*$ is some value of the original Y's randomly sampled (with replacement).
3. Calculate $\hat{\theta}_j^* = \overline{X}_j^* - \overline{Y}_j^*$

With these bootrap calculations, order the B values of $\hat{\theta}$ from smallest to largest. The 95% confidence interval for $\mu_X - \mu_Y$ would thus be the 2.5^{th} percentile and the 97.5^{th} percentile of the $\hat{\theta}$s.

1.3 The Moving Block Bootstrap

To motivate the principles of the "moving block bootstrap," I will first present to you the following data set of the percent return of a stock for 1000 days. The data is graphed in Figure 1 below.

A model commonly fit to this dataset is the auto-regressive model of order one (denoted AR(1)). For a sequence of time series data, $Y_1, Y_2, Y_3, \ldots, Y_t, Y_{t+1}, \ldots$, this model takes the form

$$Y_t = \phi Y_{t-1} + \epsilon,$$

where $\epsilon \sim N(0, \sigma^2)$ and ϕ_1 is some parameter that needs to be estimated. This parameter can easily be estimated in R using the `ar.ols` function.

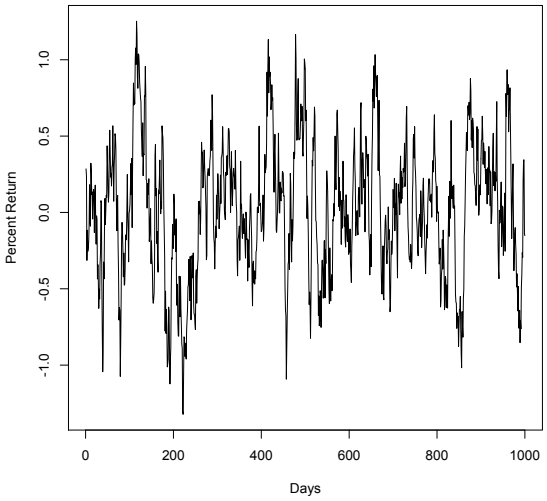

Figure 1: Percent Return of Stock over 1000 Days

An obvious question arises, though: what is the variance estimate of $\hat{\phi}$? In this case, a standard bootstrap procedure can not be applied (or would be very ill-advised) because the observations $Y_1, Y_2, \ldots, Y_{t-1}, Y_t, Y_{t+1}, \ldots$ have a time-dependent structure. In other words, Y_t depends (obviously) on Y_{t-1}, Y_{t-2}, and so on.

The block bootstrap is meant to account for problems like this. In the block bootstrap, blocks of data (as opposed to individual points) are bootstrapped. The details of the algorithm are given below:

- Let the data set be $Y_1, Y_2, Y_3, \ldots, Y_{n-2}, Y_{n-1}, Y_n$.
- Create blocks of data (of size k).

$$\begin{array}{lll} \text{Block} & 1: & Y_1, Y_2, \ldots, Y_{k-1}, Y_k \\ \text{Block} & 2: & Y_2, Y_3, \ldots, Y_k, Y_{k+1} \\ \text{Block} & 3: & Y_3, Y_4, \ldots, Y_{k+1}, Y_{k+2} \\ \vdots & & \vdots \end{array}$$

  .

- For $j = 1 : B$
  1. Randomly sample m of these blocks and string them together end-to-end to create a datset roughly the size of n ($n \approx m \cdot k$).
  2. Estimate the parameters of this bootstrapped dataset. In the context of the example we've discussed, this is $\hat{\phi}_j^*$.
- With all of the bootstrapped estimates of ϕ, the variance of its estimate can be estimated with $\widehat{\text{Var}}\left(\hat{\phi}\right) = \frac{1}{B-1}\sum_{j=1}^{B}\left(\hat{\phi}_j^* - \hat{\phi}(\cdot)\right)^2$,

The intuition behind the block bootstrap should be clear. It attempts to preserve the inherent dependence in the data by bootstrapping blocks (of size k) of data rather than individual observations. The accuracy of the method assumes, however, that observations beyond k time steps of one another do not (or weakly) depend on one another. It is thus important that k is selected carefully.

2 The Jackknife

Recall that in the bootsrap, the bias and standard error (or estimated variability) of an estimated parameter can be calculated by setting $B = 200$ or far greater than that. This can be computationally expensive to do. In the cases where the sample size, n, is small, you can get around this by using the Jackknife.

In the Jackknife, instead of randomly generating B samples, you deterministically consider n samples. Recall that n is the original sample size. And with each of these samples, calculate an estimate of the parameter of interest. Let's call this parameter of interest θ. These datasets used in the Jackknife are given below and are denoted as $\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \ldots, \mathbf{x}_{(n)}$.

$$\begin{aligned}
\mathbf{x} = (x_1, x_2, \ldots, x_n) &\longrightarrow \hat{\theta} = g(\mathbf{x}) \\
\mathbf{x}_{(1)} = (x_2, x_3, \ldots, x_n) &\longrightarrow g\left(\mathbf{x}_{(1)}\right) \\
\mathbf{x}_{(2)} = (x_1, x_3, \ldots, x_n) &\longrightarrow \theta_{(2)} = g\left(\mathbf{x}_{(2)}\right) \\
&\vdots \qquad\quad \vdots \\
\mathbf{x}_{(n)} = (x_1, x_2, \ldots, x_{n-1}) &\longrightarrow \hat{\theta}_{(n)} = g\left(\mathbf{x}_{(n)}\right)
\end{aligned}$$

Note that the dataset $\mathbf{x}_{(-i)}$ is the original data set with the i^{th} value eliminated. And with these estimates of $\hat{\theta}$, one can calculate another estimate of bias and estimated variability. These are

$$\widehat{\text{bias}}_{\text{Jcknf}}\left(\hat{\theta}\right) = (n-1)\left(\hat{\theta}(\cdot) - \hat{\theta}\right),$$

where $\hat{\theta}(\cdot) = \frac{1}{n}\sum_{j=1}^{n}\hat{\theta}_{(j)}$ and

$$\widehat{\text{Var}}_{\text{Jcknf}}\left(\hat{\theta}\right) = \left\{\frac{n-1}{n}\sum_{j=1}^{B}\left(\hat{\theta}_{(j)} - \hat{\theta}(\cdot)\right)^2\right\}^{\frac{1}{2}}.$$

Notice the inflation factors here: $(n-1)$ for bias and the estimated variability. This is done to account for the fact that Jackknife estimators are much more close to the original estimators than the bootstrap (because remember, the Jackknife eliminates just one observation).

There are a few setbacks of the Jackknife, however. The Jackknife estimate of variance and bias are not good for statistics that are not "smooth" functions of the data. A statistics that is a "mooth" function of the data is one that does not change much with small changes in the data itself. Or one that does not change at all with large changes in the data. An example of a statistic that is not a smooth function of the data is the median. Consider the dataset.

So good 4 smooth

$$1.1, 2, 3.6, 4.1, 4.4, 5.1, 5.7, 7.9.$$

. The median of the sample is 4.25. Now assume that 7.9, the largest value in the dataset becomes 17.9. If this were the case, the median would not change at all. The median is thus not a "smooth" function of the data. To see how the Jackknife fails in cases like this, let's calculate the various statistics that would give us an estimate of the variance via Jackknife.

$$\begin{aligned}
\mathbf{x}_{(-1)} = 2, 3.6, 4.1, 4.4, 5.1, 5.7, 7.9 &\implies \text{median} = 4.4 \\
.\mathbf{x}_{(-2)} = 1.1, 3.6, 4.1, 5.1, 5.7, 7.9 &\implies \text{median} = 4.4 \\
&\vdots \\
\mathbf{x}_{(-8)} = 1.1, 2, 3.6, 4.1, 4.4, 5.1, 5.7 &\implies \text{median} = 4.1.
\end{aligned}$$

So the eight medians that would be calculated in this case would be

$$4.4, 4.4, 4.4, 4.4, 4.1, 4.1, 4.1, 4.1.$$

These are just four repeats of two numbers, and this leads to a very inaccurate estimate of the median's variance.

Jackknife can't do medians. Medians don't change much. Medians are not smooth.

A way to possibly correct for the problems like this is with the "delete-d" Jackknife. In the "delete-d" Jackknife, you leave out d observations, where $n = r \cdot d$ and r is some integer. If this were the case, $\begin{pmatrix} n \\ d \end{pmatrix}$ samples would have to be considered and the estimate for the variance would be

$$\widehat{\text{Var}}_{\text{Jackknife}}\left(\hat{\theta}\right) = \frac{r}{\begin{pmatrix} n \\ d \end{pmatrix}} \sum \left(\hat{\theta}_{(s)} - \hat{\theta}_{(\cdot)}\right)^2.$$

where $\hat{\theta}_{(\cdot)} = \sum \hat{\theta}_{(s)} \Big/ \begin{pmatrix} n \\ d \end{pmatrix}$ and the sum is over all subsets of size $n - d$. For the data set above, if we set $d = 2$, we would have to consider $\begin{pmatrix} 8 \\ 2 \end{pmatrix} = 28$ samples. Below is an illustration of some of the samples that we would get if we employed the "delete-d" Jackknife with $d = 2$..

$$\begin{aligned} \mathbf{x}_{(-1,-2)} &= 3.6, 4.1, 4.4, 5.1, 5.7, 7.9 \\ &\vdots \\ \mathbf{x}_{(-2,-4)} &= 1.1, 3.6, 4.4, 5.1, 5.7, 7.9 \\ &\vdots \\ \mathbf{x}_{(-7,-8)} &= 1.1, 2, 3.6, 4.1, 4.4, 5.1 \end{aligned}$$