Resampling Methods: The Bootstrap

Rebecca C. Steorts, Duke University

STA 325, Chapter 5 ISL

Agenda

- Bootstrap
- General set up
- Application to investing problem
- Application to linear regression

The Bootstrap

The bootstrap is a widely applicable and extremely powerful statistical tool that can be used to **quantify the uncertainty** associated with a given estimator or statistical learning method.

The Bootstrap

As a simple example, the bootstrap can be used to estimate the standard errors of the coefficients from a linear regression fit.

Of course, we can get these from packages, so this isn't particularly useful, but this is just one simple example of the bootsrap.

The power of the bootstrap lies in the fact that it can be easily applied to a wide range of statistical learning methods, including some for which a measure of variability is otherwise **difficult to obtain** and is **not automatically output** by statistical software.

Toy example: Investing

Suppose we wish to determine the best investment allocation under a simple model.

Later, we explore the use of the bootstrap to assess the variability associated with the regression coefficients in a linear model fit.

Toy example: Investing

Suppose that we wish to invest a fixed sum of money in two financial assets that yield returns of X and Y, where X and Y are random quantities.

We will invest a fraction α of our money in X, and will invest the remaining $1 - \alpha$ in Y.

Since there is variability associated with the returns on these two assets, we wish to choose α to minimize the total risk, or variance, of our investment.

Toy example: Investing

That is we want to minimize

$$Var(\alpha X + (1 - \alpha)Y).$$

One can show (exercise) that the value that minimizes the risk is given by

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_Y^2 + \sigma_X^2 - 2\sigma_{XY}}, \tag{1}$$

where $\sigma_X^2 = \mathrm{Var}(X), \sigma_Y^2 = \mathrm{Var}(Y)$, and $\sigma_{XY} = \mathrm{Cov}(X, Y)$.

Toy example: Investing

- In reality, $\sigma_X^2, \sigma_Y^2, \sigma_{XY}$ are unknown.
- We can compute estimates of these quantities $\hat{\sigma}_X^2, \hat{\sigma}_Y^2, \hat{\sigma}_{XY}$ using a data set that contains past measurements for X and Y.
- We can then estimate the value of α that minimizes the variance of our investment using

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

Toy example: Investing

- It is natural to wish to quantify the accuracy of our estimate of α.
- We can understand how this might work for simulated data but in general, we cannot apply this to real data since we cannot generate new samples from the original population (since its unknown).

The Bootstrap

The bootstrap approach allows us to use a computer to emulate the process of obtaining new sample sets, so that we can estimate the variability of $\hat{\alpha}$ without generating additional samples.

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.

The Boostrap

Suppose we have a simple dataset Z with n observations.

1. Randomly select n observations from the data set in order to produce a bootstrap data set, Z^{*1}.

   - The sampling is performed with replacement, which means that the same observation can occur more than once in the bootstrap data set.

2. We can use Z^{*1} to produce a new bootstrap estimate for α, which we call $\hat{\alpha}^{*1}$.

The Boostrap (continued)

- This procedure is repeated B times for some large value of B, in order to produce B different bootstrap data sets,

  $$Z^{*1}, Z^{*2}, \ldots, Z^{*B}.$$

  and B corresponding α estimates,

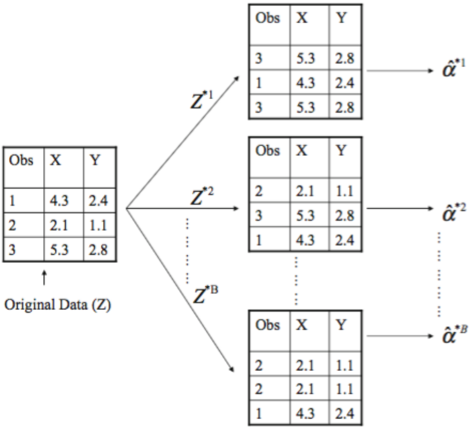
  $$\hat{\alpha}^{*1}, \hat{\alpha}^{*2}, \ldots, \hat{\alpha}^{*B}.$$

3. We can compute the standard error of these bootstrap estimates using the formula

$$\mathrm{SE}_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1}\sum_{r=1}^{B}(\hat{\alpha}^{*r} - \frac{1}{B}\sum_{r'=1}^{B}\hat{\alpha}^{*r'})}$$

4. This serves as an estimate of the standard error of $\hat{\alpha}$ estimated from the original data set.

The Bootstrap



Obs	X	Y
1	4.3	2.4
2	2.1	1.1
3	5.3	2.8

Obs	X	Y
3	5.3	2.8
1	4.3	2.4
3	5.3	2.8

Obs	X	Y
2	2.1	1.1
3	5.3	2.8
1	4.3	2.4

Obs	X	Y
2	2.1	1.1
2	2.1	1.1
1	4.3	2.4

Figure 1: A graphical illustration of the bootstrap approach on a small sample containing $n = 3$ observations. Each bootstrap data set contains n observations, sampled with replacement from the original data set. Each bootstrap data set is used to obtain an estimate of α.

The Boostrap in Practice

Performing a bootstrap analysis in R entails only two steps.

1. We must create a function that computes the statistic of interest.
2. 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.

The Boostrap on the Portfolio data set

The Portfolio data set in the ISLR package is the investment data set that motivated the bootstrap earlier.

To illustrate the use of the bootstrap on this data, we must

1. 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.
2. Then the function will output the estimate for α based on the selected observations.

The Boostrap on the Portfolio data set

```r
library(ISLR)
# Input: data set and index
# the index could be the indices
# of the entire data set (or a subset)
alpha.fn=function(data,index){
  X=data$X[index]
  Y=data$Y[index]
  # estimate alpha_hat
  return((var(Y)-cov(X,Y))/(var(X)+var(Y)-2*cov(X,Y)))
}
alpha.fn(Portfolio, 1:100)
```

```
## [1] 0.5758321
```

This function returns, or outputs, an estimate for α based on applying equation (5.7) to the observations indexed by the argument index.

The Boostrap on the Portfolio data set

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=TRUE))
```

```
## [1] 0.5963833
```

Implementing the Bootstrap

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.

The boot() function **automates** this approach.

Implementing the Bootstrap

Load the boot package in R.

```r
library(boot)
```

Implementing the Bootstrap

```
# produce R=1000 bootstrap estimates
# for alpha using boot()
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* 0.5758321 -7.315422e-05  0.08861826
```

Bootstrap Summary Results

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.

Bootstrap: Linear Regression

The bootstrap approach can be used to assess the variability of the coefficient estimates and predictions from a statistical learning method.

We will using bootstrapping to assess the variability of the estimates of β_0 and β_1 for a regression model using horsepower to predict mpg in the Auto data set.

We will then compare the estimates to those obtained using the formulas for $\mathrm{SE}(\hat{\beta}_0)$ and $\mathrm{SE}(\hat{\beta}_1)$ (refer to Section 3.1.2).

Bootstrap: Linear Regression

```r
# Bootstrap function
# Input: data, indices for data points
# Output: intercept and slope estimates
boot.fn <- function(data,index) {
return(coef(lm(mpg~horsepower, data=data,
               subset=index)))
  }
# Call the bootstrap on the entire data set
boot.fn(Auto, 1:392)
```

```
## (Intercept)  horsepower
##  39.9358610  -0.1578447
```

Bootstrap: Linear Regression

We can also get the indices by randomly sampling with replacement.

```r
set.seed(1)
boot.fn(Auto, sample(392,392, replace=TRUE))
```

```
## (Intercept)  horsepower
##  38.7387134  -0.1481952
```

```r
boot.fn(Auto, sample(392,392, replace=TRUE))
```

```
## (Intercept)  horsepower
##  40.0383086  -0.1596104
```

Bootstrap: Linear Regression

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

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.

```
boot(Auto, boot.fn, 1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Auto, statistic = boot.fn, R = 1000)
##
##
## Bootstrap Statistics :
##       original      bias    std. error
## t1* 39.9358610  0.02972191 0.860007896
## t2* -0.1578447 -0.00030823 0.007404467
```

Bootstrap: Linear Regression

As discussed in Section 3.1.2, standard formulas can be used to compute the standard errors for the regression coefficients in a linear model.

We find that the estimates for the slope and intercept respectively are 0.717 and 0.0064.

```r
summary(lm(mpg~horsepower, data=Auto))$coef
```

```
##               Estimate  Std. Error   t value      Pr(>|t|)
## (Intercept) 39.9358610 0.717498656  55.65984 1.220362e-187
## horsepower  -0.1578447 0.006445501 -24.48914  7.031989e-81
```

Bootstrap versus Estimates via Regression

Bootstrap: 0.86 (slope) and 0.0074 (intercept)

Formula: 0.717 (slope) and 0.0064 (intercept)

Why are they so different?

Bootstrap versus Estimates via Regression

The formula used for the estimates relies on assumtions.

One assumption is that we estimate σ^2 using the RSS.

This means we are assuming that the linear model is correct.

In fact, we can check that the the data here has a non-linear relationship, and hence this assumption is most likely violated.

This means the residuals and $\hat{\sigma}^2$ will be inflated.

The bootstrap does not rely on such assumptions, and thus, the errors are likely to be more accurate in this example.

Bootstrap versus Quadratic Model

What's one way we could fix the issue regarding the issue with the residuals above?

Bootstrap versus Quadratic Model

We can try fitting a quadratic model to the data.

```r
boot.fn <- function(data,index)
coefficients(lm(mpg~horsepower + I(horsepower^2),
                data=data,
subset=index))
set.seed(1)
```

Bootstrap to Quadractic Model

```r
boot(Auto, boot.fn, 1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Auto, statistic = boot.fn, R = 1000)
##
##
## Bootstrap Statistics :
##         original        bias     std. error
## t1* 56.900099702  6.098115e-03 2.0944855842
## t2* -0.466189630 -1.777108e-04 0.0334123802
## t3*  0.001230536  1.324315e-06 0.0001208339
```

Quadractic Linear Regression Model

```
summary(lm(mpg~horsepower+I(horsepower^2),
           data=Auto))$coef
```

```
##                     Estimate   Std. Error   t value      Pr(>|t|)
## (Intercept)     56.900099702 1.8004268063  31.60367 1.740911e-109
## horsepower      -0.466189630 0.0311246171 -14.97816  2.289429e-40
## I(horsepower^2)  0.001230536 0.0001220759  10.08009  2.196340e-21
```

Now, the coefficient standard estimates are very close to each other.