Computational Statistics Assignment

Bootstrapping

124384 - Luycer Bosire

4/14/2021

1. Explain what bootsrapping is.

The bootstrap is a widely applicable and extremely powerful statistical tool used to quantify the uncertainty associated with a given estimate or statistical learning. The bootstrap resampling method can be used to measure the accuracy of a predictive model. It can be used to estimate the standard errors of the coefficient from a linear regression fit.

The bootstrap algorithm begins by generating a large number of independent bootstrap samples $\mathbf{Y}^{*1}, \mathbf{Y}^{*2}, \ldots, \mathbf{Y}^{*B}$ each of size n. Typical values for B, the number of bootstrap samples, range from 50 to 200 for standard error estimation.

Corresponding to each bootstrap sample is a *bootstrap replication* of s, namely $s(\mathbf{y}^{*b})$, the value of the statistic s evaluated for $\mathbf{y}^{*b}$.

If $s(y)$ is the sample median, for instance, then $s(y\,*)$ is the median of the bootstrap sample. The bootstrap estimate of standard error is the standard deviation of the bootstrap replications,

$$\widehat{se}_{boot} = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}[s(\mathbf{Y}^{*b}) - s(.)]^2},$$

where $s(.) = \frac{1}{B}\sum_{b=1}^{B} s\,(\mathbf{Y}^{*b})$

2. Use examples to show the difference between parametric and non-parametric bootsrap.

Non-parametric bootstrapping is where the data under analysis comes from the same distribution and sample size as the data at hand and thus is sampled with replacement to preserve the probability density function from the dataset itself.

A brief overview of the non-parametric bootstrap method below serves as our example;

Given

$$X_i = 1, \ldots., n$$

) iid from

$$\mathbf{f}(.)$$

.

$$\hat{\theta}_n = h(X_1, \ldots., X_n$$

is an estimator of

$$\theta$$

and and

$$T_n$$

is a centered or standardized random variable constructed from for example;

$$T_n = \frac{\sqrt{n}(\hat{\theta} - \theta_0)}{v_n}$$

where

$$v_n$$

is a known standard deviation or estimated standard deviation, say

$$v_n = h_1(X_1, \ldots., X_n)$$

.

Consider

$$X_i^*$$

iid from

$$\mathbf{F}_n$$

, the empirical distribution function of

$$X_1, X_2, \ldots., X_n$$

. Based on these iid random variables construct using the same formula as for

$$\hat{\theta}_n$$

and

$$v_n$$

that is the same as h, h_i;

$$\hat{\theta}_n^* = h(X_1^*, \ldots., X_n^*), (v_n^*)^2 = h(X_1^*, \ldots., X_n^*)$$

and

$$T_n^* = \frac{\sqrt{n}(\hat{\theta}_n^* - \hat{\theta}obs)}{v_n^*}$$

Parametric bootsrapping assumes that the data comes from a known distribution with unknown parameters. The parameters are estimated from the data at hand and used to estimate distributions to simulate the samples.

A brief overview of the parametric bootstrap method below serves as our example;

Given

$$X_i = 1, \ldots., n$$

) iid from

$$\mathbf{f}(.;\theta), \theta \epsilon \phi)$$

.

$$\theta_0$$

is used to denote the true value of the parameter.

$$\hat{\theta}_n = h(X_1, \ldots., X_n)$$

is an estimator of

$$\theta$$

and

$$T_n$$

is a centered or standardized random variable constructed from for example;

$$T_n = \frac{\sqrt{n}(\hat{\theta} - \theta_0)}{v_n}$$

where

$$v_n$$

is a known standard deviation or estimated standard deviation, say

$$v_n = h_1(X_1, \ldots., X_n)$$

.

Consider

$$X_i^*$$

iid from

$$\mathbf{f}(.;\hat{\theta}_{obs})$$

. Based on these iid random variables construct using the same formula as for

$$\hat{\theta}_n$$

and

$$v_n$$

that is the same as h, h_i;

$$\hat{\theta}_n^* = h(X_1^*, \ldots., X_n^*)$$

and

$$T_n^* = \frac{\sqrt{n}(\hat{\theta}_n^* - \hat{\theta} obs}{v_n^*}$$

The difference between parametric and non-parametric lies in the distribution of the sample

$$X_i^*$$

We use Monte Carlo simulation with **M** simulation steps of size *n* each to approximate the sampling distribution of

$$T_n^*$$

for this given

$$\theta_{obs}$$

.

The parametric bootstrap theory is that

$$T_n^*$$

and

$$T_n$$

have approximately the same distribution, thus approximately the same quantiles.

3. Explain the bootsrap Confidence Interval.

Using an example of a population from a normal distribution of sample size

$$n$$

and parametric boostraping resampling;

a. Compute the sample mean

$$\overline{x}$$

b. and variance

$$s^2$$

c. . The bootsrap samples can be taken by generating random samples of size

$$n$$

d. from

$$N(x, s^2)$$

e. .

f. Suppose we take a sample of 1000, the set of 1000 bootstrap sample means should be a good estimate of the sampling distribution of x.

g. A 95% CI for the population mean is then formed by sorting the bootstrap means from lowest to highest, and dropping the 2.5% smallest and largest remaining values at the ends of the CI.

Based on the example in *question 2* we can obtain quantiles for the central 0.95 region by solving for

$$\theta_0$$

from

$$c_L^* \leq \frac{\sqrt{n}(\hat{\theta}_n - \theta_0)}{v_n} \leq c_U^*$$

4. Use the in-built function boot to illustrate various aspects of boostrapping.

Performing a bootstrap analysis in R entails two steps. First, we create a function that computes the statistic of interest. Then we use the **boot()** function, which is be of the **boot** library to perform the bootstrap by repeatedly sampling observations from the dataset with replacement.

We make use of the **Portfolio** dataset in the package **ISLR**.

First we create a function **bootanalysis** that takes input X and Y data and a vendor indicating which observations should be used to estimate a parameter

$$\alpha$$

. The function then outputs the estimate for

$$\alpha$$

based on the selected observations.

```r
library(ISLR)

## Warning: package 'ISLR' was built under R version 4.0.5

attach(Portfolio)

bootanalysis<- 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 an estimate for

$$\alpha$$

. For example, the following command tells

$$R$$

to estimate

$$\alpha$$

using all 100 observations.

```r
set.seed(1)
bootanalysis(Portfolio, sample(100,100, replace = T))

## [1] 0.7368375
```

The above procedure will require us to perform the command many times, recording all of the corresponding estimates for

$$\alpha$$

and computing the resulting standard deviation.

We produce R=1000 bootstrap estimates for

$$\alpha$$

```r
library(boot)
boot(Portfolio, bootanalysis, R=1000)

##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Portfolio, statistic = bootanalysis, R = 1000)
##
##
## Bootstrap Statistics :
```

```
##      original        bias    std. error
## t1* 0.5758321 -0.001695873  0.09366347
```

The output shows that using the original data,

$$\hat{\alpha} = 0.5758$$

and that the bootstrap estimate for

$$SE(\hat{\alpha}) = 0.094$$

Next we analyze the accuracy of a linear regression model. We assess the variability of the estimates for

$$\beta_0$$

and

$$\beta_1$$

, the intercept and the slope terms for the linear model

```
bootanalysis<- function(data, index){
  return(coef(lm(mpg~horsepower, data = data, subset = index)))
}

bootanalysis(Auto, 1:392)

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

Next we create bootstrap estimates for the intercept and slope terms by randomly sampling from among the observations with replacement.

```
set.seed(1)
bootanalysis(Auto, sample(392,392, replace = T))

## (Intercept)  horsepower
##  40.3404517  -0.1634868

bootanalysis(Auto, sample(392,392, replace = T))

## (Intercept)  horsepower
##  40.1186906  -0.1577063
```

We then use the *boot()* function to compute the standard errors of 1000 bootstrap estimates for the intercept and slope terms

```
boot(Auto, bootanalysis, 1000)

##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
```

```
## Call:
## boot(data = Auto, statistic = bootanalysis, R = 1000)
##
##
## Bootstrap Statistics :
##       original        bias    std. error
## t1* 39.9358610  0.0544513229 0.841289790
## t2* -0.1578447 -0.0006170901 0.007343073
```

This indicates that the bootstrap estimate for

$$SE(\hat{\beta}_0) = 0.84, SE(\hat{\beta}_1) = 0.007$$

#5. Using wages dataset in R and use the bootstrap estimates to estimate the variance of the sample median as well as the 95% CI for the population median.

```
library(boot)
library(ISLR)

data(Wage)
median(Wage$wage)

## [1] 104.9215

var(Wage$wage)

## [1] 1741.276

data_medians<- function(x, indices){
  return(median(x[indices]))
}
wage_bootstrapanalysis<- boot(Wage$wage, data_medians, 10000)

#We compute the original sample median
median(Wage$wage, 1:length(Wage$wage))

## Warning in if (na.rm) x <- x[!is.na(x)] else if (any(is.na(x)))
return(x[FALSE]
## [NA]): the condition has length > 1 and only the first element will be
used

## [1] 104.9215

wage_bootstrapanalysis$t0

## [1] 104.9215

#Note that the 2 medians are similar

wage_bootstrapanalysis$R

## [1] 10000
```

```r
wage_bootstrapanalysis$call
```

```
## boot(data = Wage$wage, statistic = data_medians, R = 10000)
```

```r
boot(data = Wage$wage, statistic = data_medians, R = 10000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Wage$wage, statistic = data_medians, R = 10000)
##
##
## Bootstrap Statistics :
##     original   bias    std. error
## t1* 104.9215 0.284396   0.6626873
```

```r
boot.ci(wage_bootstrapanalysis)
```

```
## Warning in boot.ci(wage_bootstrapanalysis): bootstrap variances needed for
## studentized intervals
```

```
## Warning in norm.inter(t, adj.alpha): extreme order statistics used as
endpoints
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = wage_bootstrapanalysis)
##
## Intervals :
## Level      Normal              Basic
## 95%   (103.3, 106.0 )   (102.9, 105.8 )
##
## Level     Percentile            BCa
## 95%   (104.0, 106.9 )   (101.8, 104.9 )
## Calculations and Intervals on Original Scale
## Warning : BCa Intervals used Extreme Quantiles
## Some BCa intervals may be unstable
```