STAT 641: BOOTSTRAPPING METHODS

Jiyoun Myung

Department of Statistics and Biostatistics
California State University, East Bay

Spring 2021, Day 5

Review

The Bootstrap Principle

Real world

Unknown probability distribution

$$F \rightarrow x = (x_1, x_2, \ldots, x_n)$$

Observed random sample

$$\widehat{\theta} = s(x)$$

Statistic of interest

Bootstrap world

Empirical distribution

$$\widehat{F} \rightarrow x^* = (x_1^*, x_2^*, \ldots, x_n^*)$$

Bootstrap sample

$$\widehat{\theta}^* = s(x^*)$$

Bootstrap replication

Review

Non-parametric vs Parametric Bootstrap

Non-parametric Bootstrap

$$X_1^{*1}, X_2^{*1}, \ldots, X_n^{*1} \sim \hat{F}$$
$$X_1^{*2}, X_2^{*2}, \ldots, X_n^{*2} \sim \hat{F}$$
$$\vdots$$
$$X_1^{*B}, X_2^{*B}, \ldots, X_n^{*B} \sim \hat{F}$$

Parametric Bootstrap

$$X_1^{*1}, X_2^{*1}, \ldots, X_n^{*1} \sim \hat{F}_{\text{par}}$$
$$X_1^{*2}, X_2^{*2}, \ldots, X_n^{*2} \sim \hat{F}_{\text{par}}$$
$$\vdots$$
$$X_1^{*B}, X_2^{*B}, \ldots, X_n^{*B} \sim \hat{F}_{\text{par}}$$

Algorithm 6.1

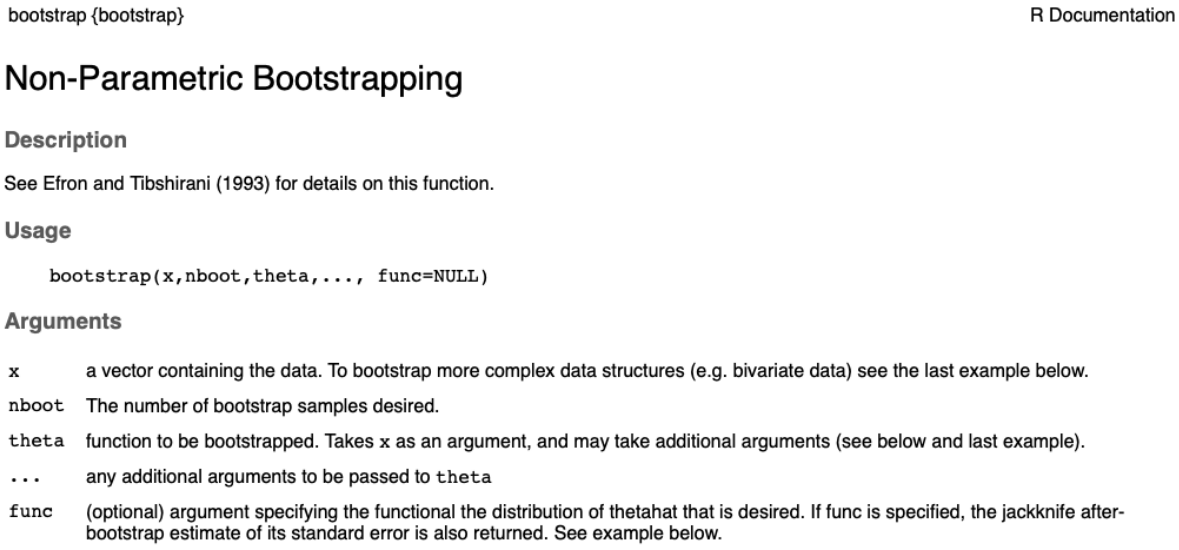
The bootstrap algorithm for estimating standard errors

1. Select B independent bootstrap samples $\mathbf{x}^{*1}, \mathbf{x}^{*2}, \cdots, \mathbf{x}^{*B}$, each consisting of n data values drawn with replacement from $\mathbf{x}$, as in (6.1) or (6.4). [For estimating a standard error, the number B will ordinarily be in the range $25 - 200$, see Table 6.1.]
2. Evaluate the bootstrap replication corresponding to each bootstrap sample,
$$\hat{\theta}^*(b) = s(\mathbf{x}^{*b}) \qquad b = 1, 2, \cdots, B. \tag{6.5}$$
3. Estimate the standard error $\text{se}_F(\hat{\theta})$ by the sample standard deviation of the B replications
$$\widehat{\text{se}}_B = \left\{ \sum_{b=1}^{B} [\hat{\theta}^*(b) - \hat{\theta}^*(\cdot)]^2 / (B-1) \right\}^{1/2}, \tag{6.6}$$
where $\hat{\theta}^*(\cdot) = \sum_{b=1}^{B} \hat{\theta}^*(b)/B$.

Packages in R

Two well known R packages concerned with the bootstrap.

- The "bootstrap" package, which is documents by our textbook.



bootstrap {bootstrap} R Documentation

Non-Parametric Bootstrapping

Description

See Efron and Tibshirani (1993) for details on this function.

Usage

```
bootstrap(x,nboot,theta,..., func=NULL)
```

Arguments

`x`	a vector containing the data. To bootstrap more complex data structures (e.g. bivariate data) see the last example below.
`nboot`	The number of bootstrap samples desired.
`theta`	function to be bootstrapped. Takes `x` as an argument, and may take additional arguments (see below and last example).
`...`	any additional arguments to be passed to `theta`
`func`	(optional) argument specifying the functional the distribution of thetahat that is desired. If func is specified, the jackknife after-bootstrap estimate of its standard error is also returned. See example below.

- The "boot" package with the textbook Davison and Hinkley (1997): Bootstrap Methods and Their Applications.

The boot package

boot {boot} R Documentation

Bootstrap Resampling

Description

Generate R bootstrap replicates of a statistic applied to data. Both parametric and nonparametric resampling are possible. For the nonparametric bootstrap, possible resampling methods are the ordinary bootstrap, the balanced bootstrap, antithetic resampling, and permutation. For nonparametric multi-sample problems stratified resampling is used: this is specified by including a vector of strata in the call to boot. Importance resampling weights may be specified.

Usage

```
boot(data, statistic, R, sim = "ordinary", stype = c("i", "f", "w"),
     strata = rep(1,n), L = NULL, m = 0, weights = NULL,
     ran.gen = function(d, p) d, mle = NULL, simple = FALSE, ...,
     parallel = c("no", "multicore", "snow"),
     ncpus = getOption("boot.ncpus", 1L), cl = NULL)
```

The Boot package

Arguments:

- data: The data as a vector, matrix or data frame.
- statistic: A function which when applied to data returns a vector containing the statistic(s) of interest. When *sim* = *"parametric"*, the first argument to statistic must be the data. For each replicate a simulated dataset returned by *ran.gen* will be passed.
- R: The number of bootstrap replicates.
- sim: A character string indicating the type of simulation required. Possible values are "ordinary" (the default), "parametric", "balanced", etc.
- ran.gen: This function is used only when *sim* = *"parametric"* when it describes how random values are to be generated. It should be a function of two arguments. The first argument should be the observed data and the second argument consists of any other information needed (e.g. parameter estimates). The second argument may be a list, allowing any number of items to be passed to ran.gen
- mle: The second argument to be passed to ran.gen. Typically these will be maximum likelihood estimates of the parameters.

Non-Parametric bootstrap with boot package

Example:

statistic: A function that produces the k statistics to be bootstrapped (k=1 if bootstrapping a single statistic). The function should include an indices parameter that the boot() function can use to select cases for each replication

```
library(boot)
x <- rexp(50)
boot_percentile <- function(x, i, p) quantile(x[i], p)
boot(x, boot_percentile, R = 999, p = 0.95)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = x, statistic = boot_percentile, R = 999, p = 0.95)
##
##
## Bootstrap Statistics :
##     original      bias    std. error
## t1* 2.658892 -0.01973989   0.5483944
```

Parametric bootstrap with boot package

Example:

Let $X_1, X_2, \ldots, X_n \sim \text{Exp}(\lambda)$, where λ is an unknown quantity. We estimate λ by an estimator such as the MLE $\hat{\lambda}_n = 1/\overline{X}_n$. Let's find the sampling distribution of 95^{th} percentile. So we first generate the bootstrap samples:

$$
\begin{aligned}
X_1^{*1}, X_2^{*1}, \ldots, X_n^{*1} &\sim \text{Exp}(\hat{\lambda}_n) \\
X_1^{*2}, X_2^{*2}, \ldots, X_n^{*2} &\sim \text{Exp}(\hat{\lambda}_n) \\
&\vdots \\
X_1^{*b}, X_2^{*B}, \ldots, X_n^{*B} &\sim \text{Exp}(\hat{\lambda}_n)
\end{aligned}
$$

Example: R code

```
library(boot)

x <- rexp(50)
bootpercentile <- function(x, p) quantile(x, p)
exp_boot <- function(x, mle)  rexp(length(x), mle)
b <- boot(x, bootpercentile, R = 999, sim = "parametric",
          ran.gen = exp_boot, mle = 1/mean(x), p = 0.95)
b
```

```
##
## PARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = x, statistic = bootpercentile, R = 999, sim = "parametric",
##     ran.gen = exp_boot, mle = 1/mean(x), p = 0.95)
##
##
## Bootstrap Statistics :
##     original     bias    std. error
## t1* 2.658892 -0.1986283   0.4604715
```

```
mean(b$t) - b$t0
```

```
##        95%
## -0.1986283
```

```
sd(b$t)
```

```
## [1] 0.4604715
```

Examples

Do this example in R

Example1: Let us consider a sample containing two hundred values generated randomly from a standard normal population $N(0, 1)$. This is the original sample. Then the sampling distribution of the sample mean is approximately normal with a mean 0 and a standard deviation $1/\sqrt{200}$. Apply the nonparametric bootstrap method to infer the result.

```
x <- rnorm(200) # original sample

boot_mean <- function(x, i){
  mean(x[i]) # allows boot to select sample
}
b <- boot(x, boot_mean, R = 999)
sd(b$t)
```

```
## [1] 0.06773244
```

```
1/sqrt(200)
```

```
## [1] 0.07071068
```

Examples

Do this example in R

Example2.: Let $X_1, X_2, \ldots, X_n \sim N(0, \sigma^2)$, where σ^2 is unknown number. How do we estimate the variance of the sample variance?

- Note: Use the sample variance instead of MLE.
- Start with

```
x <- rnorm(50)
```

Presentation Topics

Group 1: Jackknife Resampling

- Describe the approach
- Brief introduction of history
- Compare with bootstrap estimate
- Show how you can do it in R

Group 2: Double Bootstrap

- Give an algorithm on how to do double bootstrap
- Brief introduction of history (Background)
- Why this method is useful
- Demonstrate with an example on how this can be performed in R

Group 3: Using infer (tidymodel) Package

- Introduce the infer package: 4 main verbs
  https://infer.netlify.app/index.html
- Show how you can use the package for bootstrapping method - CI

Group 4: Using rsample (tidymodel) Package

- Introduce the rsample package:

  https://rsample.tidymodels.org/

  https://rsample.tidymodels.org/articles/Applications/Intervals.html

- Show how you can use the package for bootstrapping method - CI

Group 5: Bootstrap with CI forecasting

- Let's discuss the topics to present