Bootstrapping

EC 607 Metrics, Tutorial 10

Philip Economides
Spring 2021

Today

- Recap
- `boot` package
- Multiple Statistics
- Parametric Bootstrap
- Looking Forward

Recap

Bootstrapping is a resampling method, where we draw samples from our dataset Z and refit our model of interest in order to study the behaviour of key statistics across a large number of iterations.

Like conventional methods, however, bootstrap methods rely on asymptotic theory and are only exact in infinitely large samples.

When and Why do we Bootstrap?

Generally suited to cases where

1. Distribution of a statistic is unknown, perhaps due to low sample size.
2. When the sample size is too small to draw a valid inference.
3. We are approaching a research idea and wish to approximate the distribution.

boot package

`boot` provides extensive facilities for bootstrapping and related resampling methods. You can bootstrap a single statistic (e.g. a median), or a vector (e.g., regression weights).

- `boot()` generates bootstrap replicates of a statistic applied to data.

boot package

```
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)
```

- **statistic:** Usually a prepared function. The function should include an indices parameter that the `boot()` function can use to select cases for each replication.
- **R:** Number of replications.
- **sim:** Indicating the type of simulation required. Choice of "ordinary" (the default), "parametric", "balanced", "permutation", or "antithetic".
- Returns observed value of statistic applied to data.

boot package

Preparing the function for **statistic**.

```
p_load(boot)

# R-squared for the model, through resampling
rsq ← function(formula, data, indices) {
  d ← data[indices,] # allows boot to select sample
  fit ← lm(formula, data=d)
  return(summary(fit)$r.square)
}

# bootstrapping with 1000 replications
results ← boot(data=mtcars, statistic=rsq,
   R=1000, formula=mpg~wt+disp)
```

boot package

```
plot(results)
```

boot package

`boot.ci()` will generate a confidence interval for you on your estimate. Five different types of equi-tailed two-sided nonparametric confidence intervals are available.

```
boot.ci(boot.out, conf = 0.95, type = "all",
index = 1:min(2,length(boot.out$t0)), var.t0 = NULL,
var.t = NULL, t0 = NULL, t = NULL, L = NULL,
h = function(t) t, hdot = function(t) rep(1,length(t)),
hinv = function(t) t, ... )
```

type:

"norm": first order normal approximation

"basic": basic bootstrap interval

"stud": studentized bootstrap interval

"perc": bootstrap percentile interval

"bca": adjusted bootstrap percentile

boot package

Using our example:

```
boot.ci(results, type="bca")
```

```
#> BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
#> Based on 1000 bootstrap replicates
#>
#> CALL :
#> boot.ci(boot.out = results, type = "bca")
#>
#> Intervals :
#> Level       BCa
#> 95%   ( 0.6476,  0.8590 )
#> Calculations and Intervals on Original Scale
#> Some BCa intervals may be unstable
```

Multiplie Statistics

The statistics function you provide can also return a vector.

```
# Statistic(s)
bs ← function(formula, data, indices) {
  d ← data[indices,] # allows boot to select sample
  fit ← lm(formula, data=d)
  return(coef(fit))
}

# Boot
results ← boot(data=mtcars, statistic=bs,
   R=1000, formula=mpg~wt+disp)
```

Multiplie Statistics

```
plot(results, index=1) # intercept
```

Multiplie Statistics

```
plot(results, index=2) # wt
```

Multiplie Statistics

```
plot(results, index=3) # disp
```

Parametric Bootstrap

```
# Generate a dataframe, exponential distribution
y ← data.frame(list(sample = rexp(50)))

# Describes how random values are to be generated.
# It should be a function of two arguments.
# First argument should be the observed data
# Second argument consists of any other information needed (e.g. parameter estimat

expboot ← function(x,B){

  y ← x
  y$sample ← rexp(nrow(y), B)
  y
}

bootperc ← function(y, p){
  quantile(y$sample,p )
}

b2 ← boot(y, bootperc, R=1000,
          sim="parametric", ran.gen=expboot,
          mle = mean(y$sample), p=.95)
```

Parametric Bootstrap

```
plot(b2)
```

Recommendations

1. Get started on **Github**, see Jenny Bryan's guide. Will be very useful for tracking your own work flow, joint research and keeping up to date with recent developments across various programming languages.

2. Develop your data wrangling and tidying skills. Check out `data.table`, Grant McDermott introduces. Very fast, no dependencies.

3. Beyond R, explore your research interests! Start reading papers, get acquainted with data sources in your prospective field(s).

Recommendations

Highlighted Data Sources

- **IPUMS:** Publicly available individual level data from US censuses (every 10 years) and American Community Survey (Every year since 2000). Contains demographic and geographic information. Data are standarized and easy to work with. Website

- **Stanford Open Policing:** Collects and standardizes data on vehicle and pedestrian stops from law enforcement departments across the country. Over 200 million records from dozens of state and local police departments. Website

- **BigQuery:** As part of the Google Cloud Platform, BigQuery is a fully-managed, serverless data warehouse that enables scalable analysis over petabytes of data. Wrangling big data will require some advanced data-sci skills. See Grant's material.

Recommendations

Efficient Lit Search

Log into **Web of Science** through the university's library portal. Very useful search features relative to the library itself, with ranking in order of citations available. Huge time saver.

Researcher: A mobile app that works as a direct news feed for new publications. Simply prepare a list of preferred journals and stay informed on how your prospective field is developing.

Mendeley: Set up an account and immediately start creating libraries of readings through PDF's synced onto a private cloud-platform. Will help you keep track of papers, share material more easily with co-authors and jointly keep track of notes.

For collecting my BibTeX references, I usually refer to econpapers.

Recommendations

Dissemination/Branding

- Its never too early to make a website, infact the sooner the better. You'll pick up more knowledge as time proceeds and have a banger come final year.
- I'd recommend checking out a wowchemy guide on this. For a less hands-on approach, google sites if another popular option.
- Be sure to hop on Twitter and keep informed on recent developments in your practice/field. I come across a lot of information on graduate student opportunities and conferences/seminars.
- Give me a shout if you have questions on any of these recommendations.