A note on Bootstrap

gustavoc@msu.edu

10/13/2021

Bootstrap

For a reference on this topic I suggest Chapters 10 & 11 of Computer Age Statistical Inference (Efron & Hastie, 2017).

Context

Up until now we considered inference methods that are either based on assumptions about the distribution of the data (e.g., in a linear model if we assume that data is normally distributed, then esitmates also follow multivariate normal distributions) or hold in large samples (e.g., in linear models, even if data is not normally distributed, for large samples OLS estimates will follow multivariate normal distributions, this is an instance of the central limit theorem. Likewise, maximum likelihood estimates are asymptotically unbiased and asymptotically normally distributed).

Bootstrap is a re-sampling technique for evaluating features of the sampling distribution of an estimator (e.g., the standard error, CIs) that does not require making assumptions. There is an implicit large-sample requirement for Botstrap to work, but the rate of convergence may be faster than the one required for other inference procedures.

Conceptual repeated sampling

Frequentist inferences are based on the sampling distribution of the estimator over conceptual repeated sampling.

Let $\hat{\theta}(S_n)$ be an estimator, i.e., a function that maps from data (S_n a sample of size n, e.g., $S_n = \{(y_1, x_1), ..., (y_n, x_n)\}$) into estimates, and let F denote the true distribution of the data. Ideally, to approximate the sampling distribution of an estimator we should:

- Draw a large number of samples $S_{(n)j}$ from F , each of size n, $S_{(n)1}, S_{(n)2}, ..., S_{(n)N}$,
- Evaluate the estimator for each samples to produce a sequence $\hat{\theta_1}, \hat{\theta_2}, ..., \hat{\theta_N}$,
- Construct an empirical distribution for the estimator from the sequence $\hat{\theta_1}, ..., \hat{\theta_N}$
- Make inferences based on the empirical distribution.

Non-parametric Bootstrap

In practice we have only one samle $S_{(n)} = \{(y_1, x_1), ..., (y_n, x_n)\}$, and we do not get the benefit of drawing a large number of samples from the population.

In Bootstrap we *pretend* that our sample $S_n = \{(y_1, x_1), ..., (y_n, x_n)\}$ is the populutation and generate *bootstrap samples* by drawing data with replacement from S_n.

This is illustrated in the following snippet which uses serum urate data from the gout data set. Our goal is to estimate the average serum urate level. We use the sample mean as our estimator and estimate the SE of the mean using the standard formula ($\hat{SE}(\bar{x}) = \sqrt{\hat{Var}(x)/n}$) and using Bootstrap.

Example 1: estimating the SE of the mean using Bootstrap

```r
fname='https://raw.githubusercontent.com/gdlc/STAT_COMP/master/DATA/goutData.txt'
DATA=read.table(fname,header=TRUE)
SU=DATA$su
n=nrow(DATA)
Estimate=mean(SU)
SE0=sqrt(var(SU)/n)

## Bootstrap
B=5000 # numbrer of bootstrap samples
means=rep(NA,B)
for(i in 1:B){
  tmp=sample(1:n,size=n,replace=TRUE)
  bootstrap_sample=SU[tmp]
  means[i]=mean(bootstrap_sample )
}

SE.Bootstrap=sd(means)
c(SE0,SE.Bootstrap)
```

```
## [1] 0.08131497 0.08235140
```

We see that the two estimates of the SE are very close (this is expected since both are correct!).

Example 2: Inference on the correlation coefficient (this was an exam problem in previous years)

For the previous example, Boostrap is not needed because we have a closed-form exact formula for the SE that require very minimal assumptions (random sampling). Bootstrap becomes more useful when we do not have a closed-form estimator for the SE. Although there are approximate formulas for the SE of the correlation coefficient, these formulas are only approximate and hold either in large samples or when the the two RVs follow a bi-variate normal distribution.

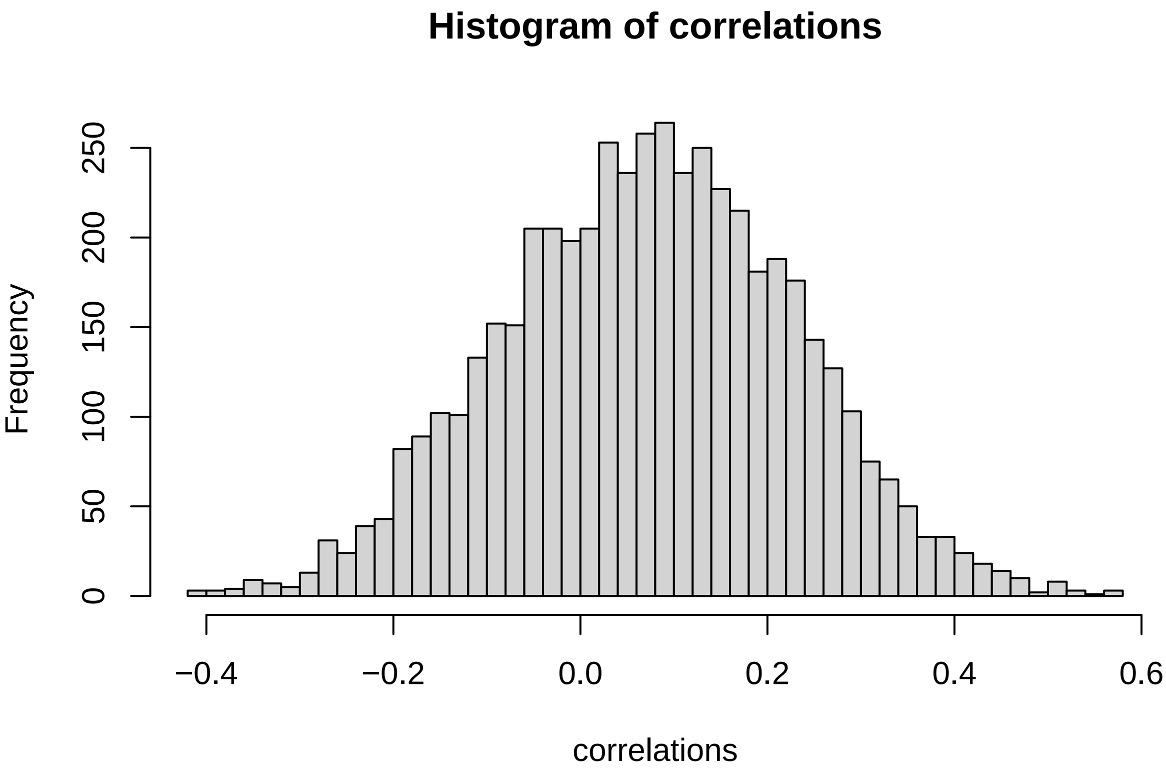
The following example illustrates how Bootsrap approximates the sampling distribution of the correlation coefficient. To illustrate we begin by using a small sample size (n=50).

```r
n=50
set.seed(195021)
tmp=sample(1:n,size=n)
SU=DATA$su[tmp]
AGE=DATA$age[tmp]

# point estimate and approximate SE
COR=cor(SU,AGE)
SE0=sqrt((1-COR^2)/(n-2))

## Bootstrap
B=5000 # numbrer of bootstrap samples
correlations=rep(NA,B)
for(i in 1:B){
  tmp=sample(1:n,size=n,replace=TRUE)
  correlations[i]=cor(SU[tmp],AGE[tmp])
}

SE.Bootstrap=sd(correlations)
```

```r
c(SE0,SE.Bootstrap)
```

```
## [1] 0.1439726 0.1549960
```

```r
hist(correlations,50)
```



The following code compares the two SEs for sample sizes equal to 20,30,50,100, and 500.

```
## Warning in cor(data$su[tmp], data$age[tmp]): the standard deviation is zero

## Warning in cor(data$su[tmp], data$age[tmp]): the standard deviation is zero
```

```
##        n        SE0 SE.Bootstrap
## [1,]  10 0.33022433           NA
## [2,]  20 0.22888142   0.21612618
## [3,]  30 0.18527215   0.17718627
## [4,]  50 0.14240290   0.13778140
## [5,] 100 0.10015205   0.09873318
## [6,] 400 0.04992928   0.04978745
```

```
## [1]       NA 1.059018 1.045635 1.033542 1.014371 1.002849
```

Bias

An estimator is biased if the expected value of the estimator is different than the true population parameter. In boostrap we pretend that our sample is the population; thus we can have an approximation to the bias of the estimator by comparing the estimate that we get with the sample ($\hat{\theta}$) with the average boostrap estimate ($\bar{\hat{\theta}}_b$), that is: $\hat{\theta} - \bar{\hat{\theta}}_b$. The following code evaluate potential biases for Examples 1 (sample mean) and 2 (sample correlation). The results suggest that, as expected, that the sample mean is an unbiased estimator of the population mean, but the sample correlation may be slightly upwardly biased (for correlations close to 1, the estimator may be downardly biased).

```r
 DATA=read.table(fname,header=TRUE)
 n=50
 set.seed(195021)
 tmp=sample(1:n,size=n)
 DATA=DATA[tmp,]
 SU=DATA$su[tmp]
 AGE=DATA$age[tmp]

# Point estimates
 meanSU=mean(SU)
 COR=cor(SU,AGE)


 ## Bootstrap
 B=10000 # numbrer of bootstrap samples
 correlations=rep(NA,B)
 means=correlations
 for(i in 1:B){
   tmp=sample(1:n,size=n,replace=TRUE)
   correlations[i]=cor(SU[tmp],AGE[tmp])
   means[i]=mean(SU[tmp])
 }

 TMP=cbind('Sample Estimates'=c('mean-su'=meanSU,'cor su-age'=COR),'Average Bootstrap'=c('mean-su'=mean
 TMP=cbind(TMP,'Ratio'=TMP[,2]/TMP[,1])
 round(TMP,4)
```

```
##              Sample Estimates Average Bootstrap  Ratio
## mean-su                5.8720            5.8702 0.9997
## cor su-age             0.0711            0.0741 1.0431
```

Bootstrap Confidence Interval

Recall that a 95%CI is a decision rule that renders intervals (DATA=> CI=[Low,Up]) which, over conceptual repeated sampling, includes the true population parmeter 95% of the times. There are multiple ways to use Bootstrap to estimate CIs:

- If we are willing to assume that our estimate follows a normal distribution we can use $\hat{\theta} + / - 1.96 \times SE_{bootstrap}$,
- **Percentile method**: we can simply use the 0.025 and 0.975 empirical percentiles of the Bootstrap estimates to approximate the CI,
- Bias-corrected intervals.

The first approach assumes nromality, thus, if normaility does not hold the percentile method should be preferred. The percentile method may not be accurate if the estimate is biased, in those cases bias-corrected intervals are preferred; the function `boot.ci()` of the `boot` package can be used to compute biased-corrected CIs.

```r
 DATA=read.table(fname,header=TRUE)
 n=50
 set.seed(195021)
 tmp=sample(1:n,size=n)
 DATA=DATA[tmp,]
 SU=DATA$su[tmp]
```

```
AGE=DATA$age[tmp]


## Bootstrap
B=10000 # numbrer of bootstrap samples
correlations=rep(NA,B)
for(i in 1:B){
  tmp=sample(1:n,size=n,replace=TRUE)
  correlations[i]=cor(SU[tmp],AGE[tmp])
}

CI_1=COR+c(-1,1)*1.96*sd(correlations)
CI_2=quantile(correlations,p=c(.025,.975))

round(CI_1,3)
```

```
## [1] -0.233  0.376
```

```
round(CI_2,3)
```

```
##   2.5%  97.5%
## -0.232  0.374
```