

Bootstrap

Christophe Dutang, Nhan Nguyen and Phuong Nguyen

25 mars 2008

Abstract

The Bootstrap is the most widely used resampling statistical method. This method becomes very popular because of the big breakthroughs in computer science and computer industry which allow practitioners (both on academic and professional sides) to (re)sample very large and very complex datasets. Bootstrap provides a very powerful solution to statistical inference when traditional statistic cannot be applied. For instance, if we study a ratio of means or a ratio of standard deviations, there is no explicit formula: that is where bootstrap is useful.

From bootstrap, we can extract a lot of statistical quantities (such as standard error, bias, confidence interval, hypothesis test, . . .) on a particular problem because it computes the sampling distribution. This is carried out by resampling the dataset and computing bootstrap replicates of the statistic of interest. And then, we just have to determine the statistical characteristics such as the mean (used for the bias), the standard deviation (used for the standard error), etc. . .

The bootstrap method needs a good statistical software, where intensive computation can be done. R statistical software is a free software based on the S language, providing an extremely wide range of statistical functionalities (linear and non-linear modelling, classical statistical test, time series analysis, . . .). R allows users to add additional functionality with packages. The package **boot** of Pr. Brian Ripley implements all the necessary functions of bootstrap.

This report aims to give a quick overview of bootstrap. As bootstrap is the topic of many books, many theses and many courses, this report is not intended to be exhaustive. The following work only scratches the surface of bootstrap, but it gives basic and understandable examples of bootstrap applications.

Keywords : **Bootstrap; Resampling methods; R package boot;**

Sommaire

Introduction	3
1 Theory	4
1.1 non-parametric bootstrap	4
1.2 Parametric bootstrap	5
2 Applications of bootstrap	7
2.1 Bootstrap standard error and bias	7
2.2 Bootstrap confidence interval	9
2.3 Bootstrap hypothesis test	13
Conclusion	14
Bibliography	14
Glossary	16
A R commands for numerical applications	17
A.1 Bootstrap standard error and bias	17
A.2 Bootstrap confidence interval	18
A.3 Hypothesis test	19

Introduction

The purpose of statistical inference is to extract and summarize the key characteristics of a dataset. For example, the mean and the standard deviation are the most simple statistics of interest. To know the accuracy of data summaries, we can find some explicit results for certain statistics in the literature. However, for some complex statistics, there are no formulas available. Bootstrap has been invented to solve this issue.

The name bootstrap comes from the Rudolph Erich Raspe's tale, where Baron Munchausen had fallen to the bottom of a deep lake. And as he was to succumb to his fate he thought to pull himself up by his own bootstraps. In French, bootstrap is sometimes translated by "à la Cirano" (Act 3, Scene II) in reference to this hero who projected to reach the moon by iteratively throwing a magnet while being on an iron sheet. However, this is not a wide-accepted French translation in the scientific literature.

The purpose of this report is to give a quick introduction of bootstrap method with direct applications. As numerical applications are all done in R*, our work shows possibilities (at most a tenth) of the R package **boot**[†] implementing bootstrap methods. This report is divided into two chapters. Firstly, we introduce the fundamental concepts of bootstrap, both parametric and non-parametric bootstrap. Secondly, we apply the bootstrap method on four particular statistics: standard errors, bias, confidence intervals and hypothesis tests. Finally, we conclude.

*R Development Core Team (2007).

[†]original by Angelo Canty. R port by Brian Ripley. (2007).

Chapter 1

Theory

In this chapter, we present the core of the theory of bootstrap. In the following, we denote by (x_1, \dots, x_n) the observed data, realizations of random variables (X_1, \dots, X_n) , which are assumed to be identically and independently distributed (i.i.d.) characterized through the cumulative distribution function (cdf) F . We focus on a statistic of interest $s = s(x_1, \dots, x_n)$ (realization of the random variable $s = s(X_1, \dots, X_n)$) to estimate the quantity S .

1.1 non-parametric bootstrap

1.1.1 Introduction

non-parametric bootstrap tries to estimate the sampling distribution of a statistic of interest $s = s(X_1, \dots, X_n)$ given the observed sample (x_1, \dots, x_n) assuming the cdf F is unknown*. Bootstrap carries out this purpose by generating “new” data from (x_1, \dots, x_n) by sampling with replacement. So we get b sets of data $(\tilde{x}_{i,1}, \dots, \tilde{x}_{i,n})_{1 \leq i \leq b}$ where an estimation of $\tilde{s}_1, \dots, \tilde{s}_b$ can be made. Hence, we are able to study the sampling distribution of s given (x_1, \dots, x_n) .

1.1.2 Method and concepts

Bootstrap method is structured as follows :

1. random sample with replacement on the observed sample (x_1, \dots, x_n) : in output, we get b bootstrap samples $(\tilde{x}_{i,1}, \dots, \tilde{x}_{i,n})_{1 \leq i \leq b}$,
2. then compute the statistic s on bootstrap samples : bootstrap replicates are $\tilde{s}_1, \dots, \tilde{s}_b$,
3. finally use the bootstrap data $(\tilde{s}_1, \dots, \tilde{s}_b)$ to quantify the accuracy of s (e.g. standard error, confidence set, ...).

*We do not make assumptions on the cdf F .

Actually, the step 1 is equivalent to draw samples from the empirical cumulative distribution function F_n^* of the data (x_1, \dots, x_n) . In step 1, we choose a random index j in the set of integers $\{1, \dots, n\}$ and define a bootstrap replication as $\tilde{x} = x_j^\dagger$. Hence we have that the probability $P(\tilde{x} = x_j) = \frac{1}{n}$ for all j in $\{1, \dots, n\}$. When using the empirical distribution function F_n , we draw a uniform random variable u and find $x_k^* = \tilde{x}$ such that $F_n(x_k^*) \leq u < F_n(x_{k+1}^*)$ where (x_1^*, \dots, x_n^*) denotes the sorted data. Since F_n is a step function of vertical step $1/n$, we have $\frac{k}{n} \leq u < \frac{k+1}{n}$ i.e. $k = \lfloor u \cdot n \rfloor^\ddagger$. Hence we have the equivalence of drawing random samples on indexes of data and sampling with the empirical distribution function F_n .

All concepts introduced so far are gathered in the schematic diagram 1.1.

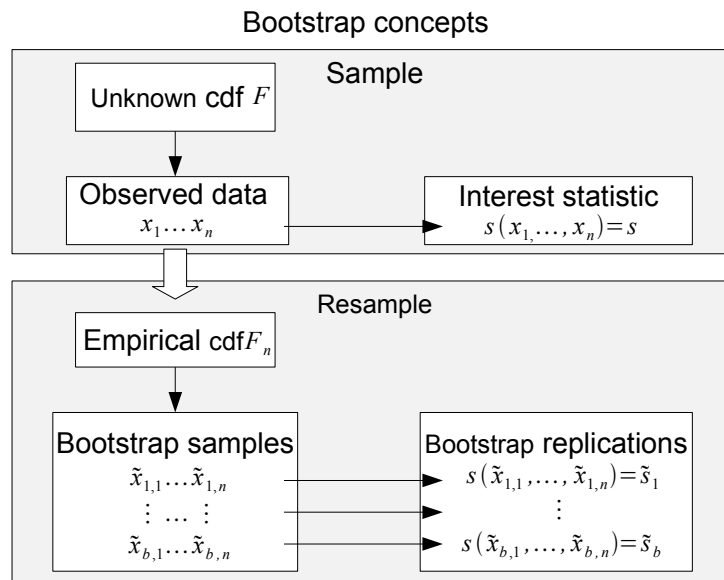


Figure 1.1: non-parametric Bootstrap concept

1.2 Parametric bootstrap

1.2.1 Introduction

Although the non-parametric bootstrap is useful for estimating the bias and standard error of sample statistics, it is not very useful for constructing hypothesis tests. This is because the non-parametric bootstrap distribution is constructed based on the properties of the actual underlying population, as they appear in the original sample. For a hypothesis test, the sampling distribution of the statistic s under the null hypothesis - not the actual population - is needed.

*defined as $F_n(x) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{(x_i \leq x)}$

†this action is carried out n times to have a full sample $(\tilde{x}_1, \dots, \tilde{x}_n)$

‡the integer part of $u \times n$.

1.2.2 Method and concepts

In the parametric bootstrap, we logically supposed the observed data (x_1, \dots, x_n) come from a parametric model. That is to say the cdf F is a distribution of a particular family (e.g. exponential distribution, normal distribution, ...) dependent on a parameter θ . All the uncertainty relies on the fact we do not know the value of parameter θ .

The difference with the non-parametric bootstrap is that resampling are carried out with the parametric model once θ is estimated. Then from the bootstrap samples $(\tilde{x}_{i,1}, \dots, \tilde{x}_{i,n})_{1 \leq i \leq b}$, we get new estimation of statistic s : $\tilde{s}_1, \dots, \tilde{s}_b$.

The parametric bootstrap consist in

1. estimating the parameter θ of the parametric model given the data (x_1, \dots, x_n) say $\hat{\theta}$,
2. random sampling from the parametric model with $\theta = \hat{\theta}$ to get b bootstrap samples $(\tilde{x}_{i,1}, \dots, \tilde{x}_{i,n})_{1 \leq i \leq b}$,
3. then computing the statistic s on bootstrap samples : bootstrap replicates are denoted $\tilde{s}_1, \dots, \tilde{s}_b$,
4. finally using the bootstrap data $(\tilde{s}_1, \dots, \tilde{s}_b)$ to quantify the accuracy of s (e.g. standard error, confidence set, ...).

Let us notice that the statistic s can be different of the model parameter θ . The schematic diagram 1.2 summarize concepts.

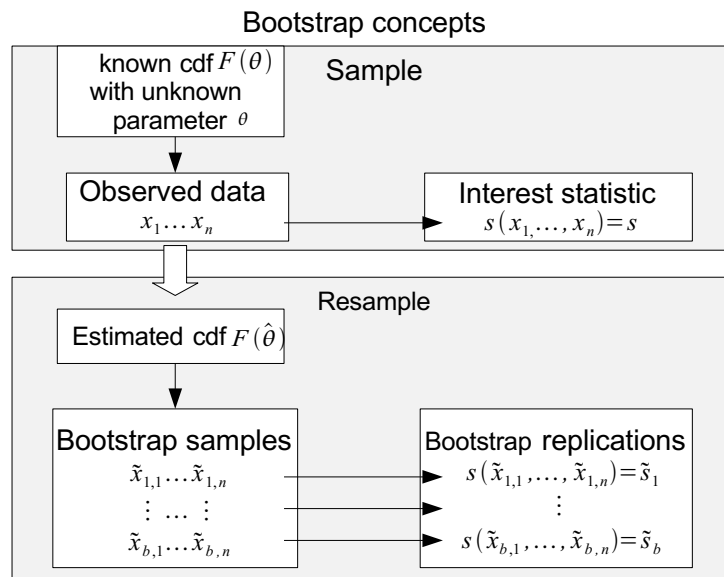


Figure 1.2: parametric Bootstrap concept

Chapter 2

Applications of bootstrap

In all this chapter, we carried out numerical applications thanks the statistical software R and the bootstrap dedicated package `boot` of Pr. Brian Ripley. We will work on different datasets: US arrests in 1973, sleeping time in 1905 *.

2.1 Bootstrap standard error and bias

2.1.1 Bootstrap estimate of standard error

Arguably, one of the most important practical applications of the bootstrap is in providing conceptually simple estimates of the standard error and bias for a statistic of interest. As usual s stands for the statistic of interest based on the observed sample. In output of the bootstrap method, we get bootstrap replicates $(\tilde{s}_1, \dots, \tilde{s}_b)$. Let us denote its standard error by σ_s defined as $\sqrt{Var(s(X_1, \dots, X_n))}$. Since the form of the statistic s may be very complicated, the exact formulas for the corresponding bootstrap estimate of standard error (BESE) is approximated by the empirical standard deviation of $(\tilde{s}_1, \dots, \tilde{s}_b)^\dagger$ i.e.

$$\tilde{\sigma}_b = \sqrt{\frac{1}{b-1} \sum_{i=1}^b (\tilde{s}_i - \bar{\tilde{s}}_b)^2} \xrightarrow{b \rightarrow \infty} \sigma_s,$$

where $\bar{\tilde{s}}_b$ is the mean of bootstrap replicates (i.e. $\frac{1}{b} \sum_{i=1}^b \tilde{s}_i$).

Let us note that the total number of resamples, b , may be taken as large as we wish, since we are in complete control of the resampling process. For instance, it has been shown that for estimating BESE, b equal to about 500 typically gives already a satisfactory approximation.

*more details with `help(USArrest)` and `help(sleep)` in R.

[†]In the following formula, the term $\frac{1}{b-1}$ comes from the fact the naive estimator of the standard deviation with $\frac{1}{b}$ is biased.

2.1.2 Bootstrap estimate of bias

Similarly, we have an analogous estimate of the bias of statistics s . We recall that the bias is the difference between the expectation of our statistic s and the mathematical quantity S we want to estimate*. For instance, the bias of the mean is defined as $B_{\bar{X}_n} = E[\bar{X}_n] - E[X]$. Whereas in a parametric context, the bias of $\hat{\lambda}_n$ is $B_{\hat{\lambda}_n} = E[\hat{\lambda}_n] - \lambda$, where λ can denote the parameter of an exponential distribution for example.

We estimate the bias of the statistic s by

$$\tilde{B}_b = \frac{1}{b} \sum_{i=1}^b \tilde{s}_i - s(x_1, \dots, x_n) \xrightarrow{b \rightarrow \infty} B_s.$$

We denote B_b as the bootstrap estimate of bias (BEB). When $b = 1000$, we generally have a good estimation of the true bias B_s .

2.1.3 Numerical applications

Non-parametric bootstrap

We apply this procedure on the data `USArrests`[†] with a non-parametric bootstrap. This data set contains statistics, in arrests per 100,000 residents for assault, murder, and rape in each of the 50 US states in 1973. We focus only on murders.

With the R commands shown in appendix, we find that the mean and the median of murder arrest number are 7.788 and 7.25 respectively. The bootstrap standard error on these statistics are 0.583 and 0.891, while bias are 0.034 and 0.012. The respective bias estimates are 0.035 and 0.013.

Finally, we plot in figure 2.1 the bootstrap estimate of standard error (of the mean and the median) in function of the resample number b . The convergence of the BESE is not so quick as we might expect. With a resample number b below 500, the BESE will be inaccurate both for the mean and median statistic. This is partially due to the small number of data (50 i.e. the number of US states).

Parametric bootstrap

We apply bootstrap standard error and bias on an exponentially $\mathcal{E}(\theta)$ random generated sample[‡]. To estimate the parameter θ , we use the unbiased maximum likelihood estimator $\hat{\theta}_n = \frac{n-1}{\sum_{i=1}^n X_i}$. We are interested in the standard deviation of $\hat{\theta}_n$, where an explicit formula exist $\sqrt{Var(\hat{\theta}_n)} = \frac{\theta^2}{n-2}$. Hence we can compare the BESE on $\hat{\theta}_n$ with the true value. As we can see on figure 2.2, the convergence of the BESE is very low, this can be explained we have a relative small sample ($n = 1000$).

*e.g. to estimate the expectation, we have $S = E[X]$ and $s = \bar{X}_n$.

[†]packaged in R.

[‡]cf. appendix for the R code.

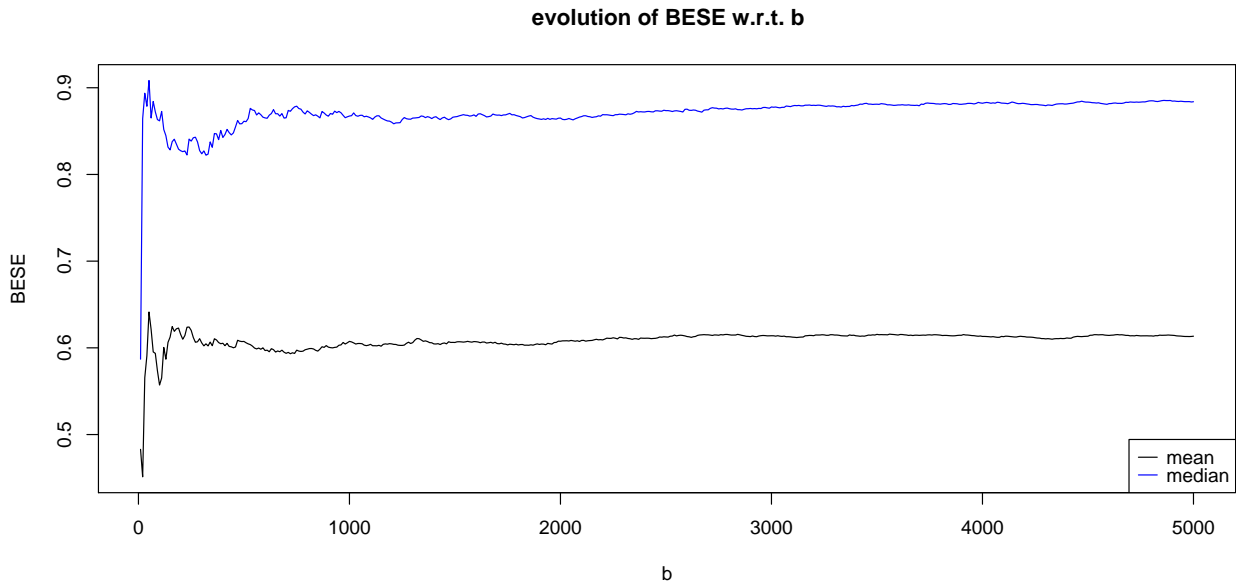


Figure 2.1: Convergence of the bootstrap estimate of standard error

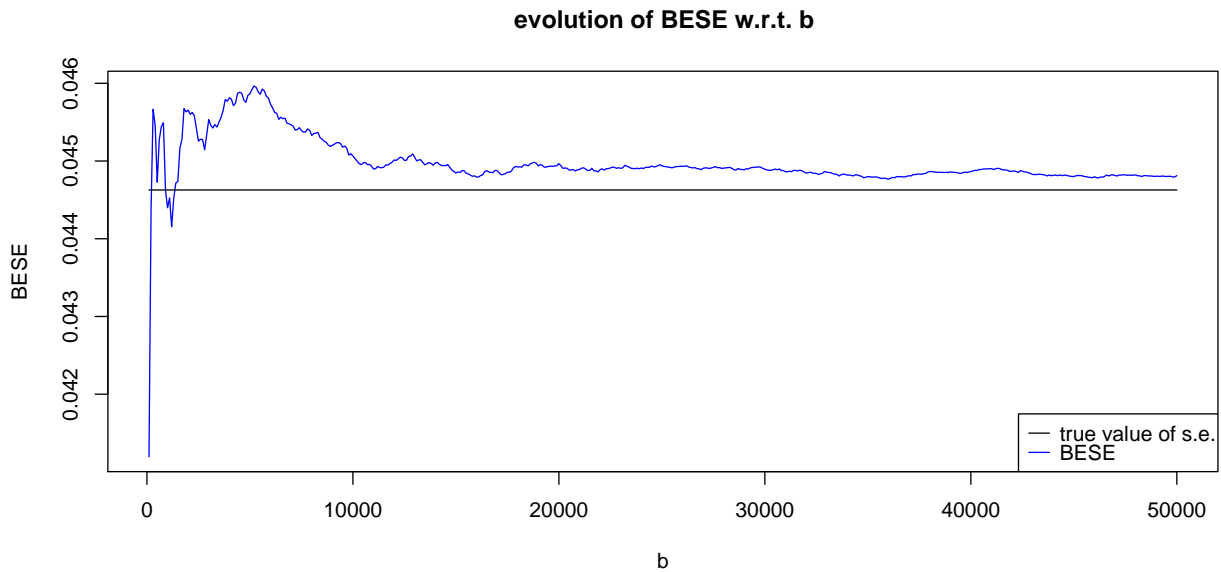


Figure 2.2: Convergence of the bootstrap estimate of standard error

2.2 Bootstrap confidence interval

Confidence sets are widely used in statistics because they provide bounds for s our statistic of interest. A subset C_n of \mathbb{R} is said to be a confidence set of level α if $P(S \in C_n) = 1 - \alpha$.

In most cases, confidence sets are constructed by considering a “pivotal” function $g(x_1, \dots, x_n, F)$ whose distribution G_n is independent of unknown distribution F . But the problem is that the distribution G_n is not necessarily known. In classical theory, we replace the distribution G_n by its limit, and so the derived confidence set is only valid for large n .

The bootstrap approach of this problem consists in using the bootstrap estimation G_{boot} of G_n , and then construct the confidence set. All the methods for confidence interval come from Shao & Tu (1996) and Efron & Tibshirani (1993)

2.2.1 The Gaussian bootstrap

The normal-theory interval assumes that the statistic s is normally distributed, which can be the case for statistics in sufficiently large samples. And then we use the bootstrap estimate of standard error to construct a $100(1 - \alpha)$ percent confidence interval for S of the form

$$\left[s - u_{1-\frac{\alpha}{2}} \sigma_b; s + u_{1-\frac{\alpha}{2}} \sigma_b \right],$$

where σ_b is the bootstrap estimate of the standard error of s , and $u_{1-\frac{\alpha}{2}}$ is the $1 - \frac{\alpha}{2}$ quantile of the standard-normal distribution (e.g. roughly 1.96 for a 95-percent confidence interval, where $\alpha = 0.05$).

2.2.2 The bootstrap-t

The problem of the previous approach is that we use the normal approximation on the statistic s . That is to say we assume $\frac{s-S}{\sigma_s}$ follows a normal distribution, where s is our statistic estimating the quantity S and σ_s denotes the true standard error of s . The principle of the bootstrap-t is to estimate the distribution of $\frac{s-S}{\sigma_s}$ by its bootstrap distribution.

If we denote by G_{boot} the distribution computed with the bootstrap replicates, then the bootstrap-t confidence interval for S is

$$\left[s - \sigma_s G_{boot}^{-1} \left(1 - \frac{\alpha}{2} \right), s + \sigma_s G_{boot}^{-1} \left(1 - \frac{\alpha}{2} \right) \right].$$

The main drawback is we need an estimate of the standard error σ_s of s . One may think we could use the bootstrap analog $\tilde{\sigma}_b$. But we have to use different bootstrap replicates for G_{boot} and $\tilde{\sigma}_s$, so the computation explodes.

2.2.3 The bootstrap percentile

An alternative approach, called the bootstrap percentile interval, is to use the empirical quantiles of bootstrap replicates $(\tilde{s}_1, \dots, \tilde{s}_n)$ to form a confidence interval for s :

$$\left[\tilde{s}_j^*; \tilde{s}_k^* \right],$$

where $j = \lfloor \alpha \frac{b+1}{2} \rfloor$, $k = \lfloor (1 - \frac{\alpha}{2}) \frac{b+1}{2} \rfloor$, and \tilde{s}^* denotes the ordered bootstrap replicates (e.g. $j = 25$ and $k = 975$ for $b = 999$ and $\alpha = 0.05$).

We continue the example concerning US murder. We know the empirical mean and median are 7.788 and 7.25 respectively. Their bootstrap percentile confidence intervals are calculated with R. We get [6.51385, 9.05210] and [5.8, 9.0] as bootstrap 95-percent confidence intervals for the mean and the median.

2.2.4 The bootstrap accelerated bias-corrected percentile (BCa)

Although they do not assume normality, percentile confidence intervals often do not perform well. So-called bias-corrected, accelerated (or BCa) percentile intervals are preferable. To find the BCa interval for s , we first compute $z = \Phi^{-1}(G_{boot}(s))$, where Φ^{-1} stands for the standard-normal quantile function and G_{boot}^* is the empirical distribution function of s calculated with bootstrap replicates. If the bootstrap sampling distribution is symmetric, and if s is unbiased, then this proportion will be close to 0.5, and the ‘‘correction factor’’ z will be close to 0.

Let $s_{(-i)}$ represent the value of s produced when the i th observation is deleted from the original sample (x_1, \dots, x_n) . There are n of these quantities ($s_{(-1)} = s(x_2, \dots, x_n)$, $s_{(-2)} = s(x_1, x_3, \dots, x_n), \dots$). Let \bar{s}_- represent the average of the 1-deleted statistic sample $(s_{(-1)}, \dots, s_{(-n)})$ i.e. $\bar{s}_- = \frac{1}{n} \sum_{i=1}^n s_{(-i)}$. Then we calculate

$$a = \frac{\sum_{i=1}^n (s_{(-i)} - \bar{s}_-)^3}{6 \left(\sum_{i=1}^n (s_{(-i)} - \bar{s}_-)^2 \right)^{\frac{3}{2}}}.$$

With the correction factors z and a in hand, we compute

$$a_1 = \Phi \left(z + \frac{z - u_{1-\frac{\alpha}{2}}}{1 - a(z - u_{1-\frac{\alpha}{2}})} \right) \text{ and } a_2 = \Phi \left(z + \frac{z + u_{1-\frac{\alpha}{2}}}{1 - a(z - u_{1-\frac{\alpha}{2}})} \right).$$

Finally, the values a_1 and a_2 are used to locate the endpoints of the corrected percentile confidence interval:

$$[\tilde{s}_j^*; \tilde{s}_k^*],$$

where $j = \lfloor ba_1 \rfloor$ and $k = \lfloor ba_2 \rfloor$. When the correction factors a and z are both 0, we get $a_1 = \Phi(-u_{1-\frac{\alpha}{2}}) = \Phi(u_{\frac{\alpha}{2}}) = \frac{\alpha}{2}$ and $a_2 = \Phi(u_{1-\frac{\alpha}{2}}) = \Phi(u_{\frac{\alpha}{2}}) = 1 - \frac{\alpha}{2}$, which corresponds to the (uncorrected) percentile interval. Shao & Tu (1996) explains that the correction factor z is used to correct the bias of the confidence interval, while a is dedicated to speed up the convergence of the confidence interval.

The whole idea behind this correction two correction factors is to find a relation between the statistic s and our quantity of interest S such as $P(f(s) - f(S) \leq x) = \varphi(x)$ where the transformation function f is such that φ is a known distribution (typically the normal distribution).

* $G_{boot}(x) = \frac{1}{b} \sum_{i=1}^b \mathbb{1}_{(\tilde{s}_i \leq x)}$.

But the previous equality does not necessarily hold. So Efron & Tibshirani (1993) suggests an equation of the form: $P\left(\frac{f(s)-f(S)}{1+af(S)} + z \leq x\right) = \varphi(x)$.

To obtain sufficiently accurate 95-percent BCa confidence intervals, the number of bootstrap samples, b , should be at least 1000.

2.2.5 Comparison of bootstrap confidence interval

We still study the US arrests (in the case of murder) but with a different statistic: the standard deviation. The purpose of its application is to show that increasing the bootstrap number reduces the variation of bootstrap confidence intervals and mainly to compare the difference between different bootstrap confidence intervals.

The R code that produces the following figure is given in appendix. On this figure, we can clearly see that the variability of bounds of the confidence interval decreases quickly with the number of bootstrap replicates. Furthermore, we see the big difference between the crude confidence interval and the most sophisticated BCa confidence interval. Unexpectedly, the Gaussian version of the bootstrap confidence interval works well. This is probably due to the fact the distribution of standard deviations is Gaussian.

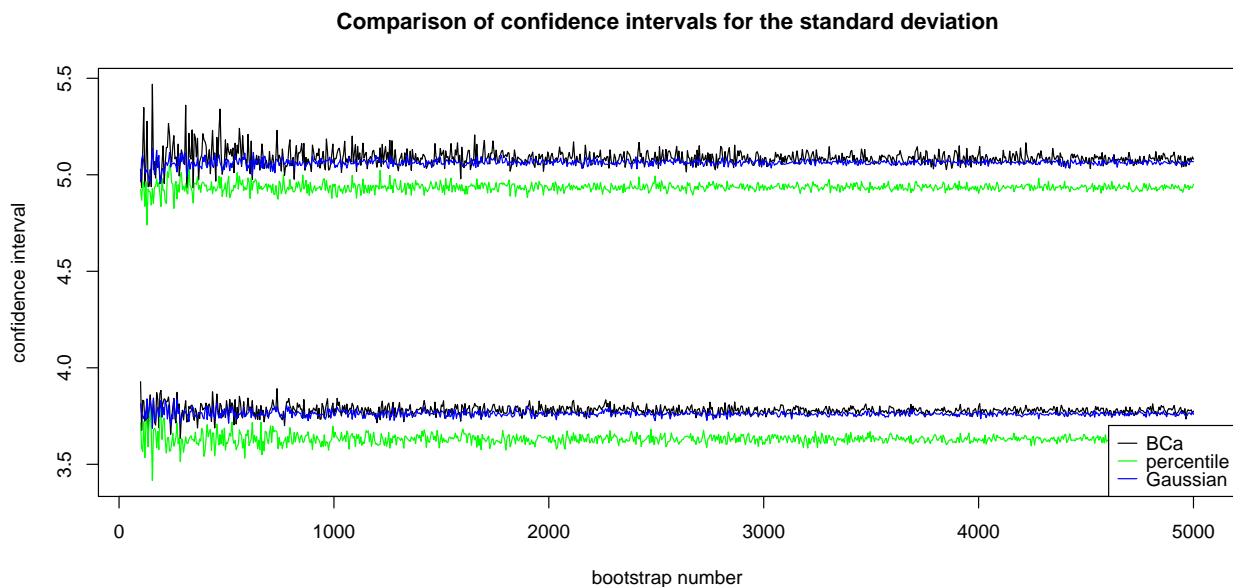


Figure 2.3: Comparison of confidence intervals (estimated standard deviation is 4.35551)

2.3 Bootstrap hypothesis test

An hypothesis test is a statistical rule based on a random sample which aims to decide if an hypothesis (a statement on a characteristic of data) can be rejected or accepted. Formally, we choose two hypothesis H_0 and H_1 (respectively the null and the alternative hypothesis). An hypothesis test give us a criterion to choose between H_0 and H_1^* . We generally denote by α the error probability to reject H_0 when H_0 is true[†]

As for the other applications of bootstrap method, two cases have to be distinguished: non-parametric and parametric bootstrap.

- For parametric bootstrap testing, we assumed data come from a known distribution F but with an unknown parameter θ . Under the null hypothesis, θ is contained in a particular subset of parameter say Θ_0 (which could be just a singleton). So we just have to sample bootstrap replicates $\tilde{s}_{0,1}, \dots, \tilde{s}_{0,b}$ under the null hypothesis. And then, approximate the probability that we reject H_0 by $\frac{\text{card}(\{\tilde{s}_{0,i} > s\})}{b}$ [‡].
- For non-parametric bootstrap testing, we generally test some characteristics on multi sample (e.g. the independence or the difference in means of two samples). This has consequences on the resampling method, because we need to resample under the null hypothesis.

Here are some very basic examples of bootstrap hypothesis tests. First, we use a non-parametric bootstrap method to compare two soporific drugs on patents by the extra number of sleeps compare to control. We want to test if the two samples (x and y) have the mean i.e. $H_0 : \bar{x}_n = \bar{y}_n$ against $H_0 : \bar{x}_n \neq \bar{y}_n$. The dataset `sleep` comes from an article of Cushny and Peebles in 1905 in the journal of physiology.

With R, we find a p-value[§] of 99%, hence we utterly reject the null hypothesis. More complicated non-parametric bootstrap test can be done with a studentized “pivot”.

Secondly, we illustrate the parametric bootstrap hypothesis test. We consider an exponential distributed sample, i.e. we assume $(X_i)_{1 \leq i \leq n} \stackrel{iid}{\sim} \mathcal{E}(\lambda)$. The maximum likelihood estimator is $\frac{1}{\bar{X}_n}$. With R, we simulate a sample (x_1, \dots, x_{10}) (i.e. 1.9588 0.1278 0.1699 0.1376 0.1768 0.0548 1.8644 0.1304 0.3774 0.4542).

The maximum likelihood estimator is $\hat{\lambda} = \frac{1}{\bar{x}_{10}} = 1.8339$. We want to test the hypothesis that $H_0 : \lambda = 1.8$. We sample 100 sample exponentially distributed with a λ parameter equalled to 1.8 in order to compute bootstrap replicate. Then the p-value is 0.43 approximated by the ratio of bootstrap replicate greater than $\hat{\lambda}$ and bootstrap number ($b = 100$). Hence to reject the null hypothesis, we can have a better confidence than 44%.

* Actually, it binarily chooses or rejects the null hypothesis H_0 .

† called type I error

‡ where *card* stands for the cardinal, i.e. the number of items of a set.

§ the treshold α_t of type I error probability such that we reject the null hypothesis with $\alpha < \alpha_t$.

Conclusion

Bootstrap is an inference statistical method based on resampling method. Due to the increase in computational power, bootstrap is definitely a very effective and efficient statistical tool, which can be used to solve difficult statistical problem. In this report, we only present four applications of the bootstrap, namely the bias, the standard error, confidence intervals and hypothesis tests. However, there is a lot of other applications such as prediction intervals in regression, meaningfulness test in time series. . . We have just scratched the surface of bootstrap.

Despite the fact that bootstrap is a powerful tool, bootstrap is not intended to replace classical statistics nor be used when classical statistics apply. Bootstrap needs to be used with care. And the fact, its computer implementation is quite quick, must not mask that the reliability and the quality of inference statistic is mainly due to the availability and the quality of data.

Bibliography

Efron, B. & Tibshirani, R. J. (1993), *An Introduction to Bootstrap*, Chapman and Hall.

original by Angelo Canty. R port by Brian Ripley., S. (2007), *boot: Bootstrap R (S-Plus) Functions (Canty)*. R package version 1.2-30.

R Development Core Team (2007), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

URL: <http://www.R-project.org>

Shao, J. & Tu, D. (1996), *The Jackknife and Bootstrap*, Springer.

English-French Glossary

1. Bootstrap : bootstrap ou rééchantillonnage
2. Random variable : variable aléatoire
3. Data set : série des données
4. Output data : données des sorties
5. Observed sample : échantillon d'observation
6. Observed data : données des observations
7. To draw (the data) : tirer (les données)
8. Integer (number) : (nombre) entier
9. Statistics : statistiques, données, observations
10. (Empirical) cumulative distribution function (cdf) : fonction de répartition (empirique)
11. Interest statistic : statistiques d'intérêt
12. Bootstrap replicate/replication : répétition de bootstrap
13. Accuracy (of an estimator) : exactitude (d'un estimateur)
14. Standard error and bias : standard erreur et biais
15. Bound : borne
16. Standard deviation : écart type
17. Confidence interval : interval de confiance
18. Hypothesis test : test d'hypothèse
19. Statistical rule : règle de statistique

Appendix A

R commands for numerical applications

A.1 Bootstrap standard error and bias

A.1.1 Non-parametric

```
data(murder)
murder <- USArrests[,1]
#compute the BESE for the mean function
boot(murder, function(x,y) mean(x[y]), R=1000, stype = "i")
#compute the BESE for the median function
boot(murder, function(x,y) median(x[y]), R=1000, stype = "i")

bootReplMean <- boot(murder, function(x,y) mean(x[y]), R=5000)$t
BESEmean <- sapply(1:500*10, function(n) sd(bootReplMean[1:n]))
bootReplMedian <- boot(murder, function(x,y) median(x[y]), R=5000)$t
BESEmedian <- sapply(1:500*10, function(n) sd(bootReplMedian[1:n]))
bound <- c(min(BESEmean, BESEmedian), max(BESEmean, BESEmedian))
plot(1:500*10, BESEmean, type = "l", main=
      "evolution of BESE w.r.t. b", xlab="b", ylab="BESE", ylim=bound)
lines(1:500*10, BESEmedian, col="blue")
legend("bottomright", c("mean", "median"), col=c("black", "blue"), lty=c(1,1))
```

A.1.2 Parametric

```
x <- rexp(1000, y <- rlnorm(1))
n <- length(x)
lambdaMLE <- (n-1)/mean(x)/n
para.rg <- function(data, mle) rexp(length(data), lambdaMLE)
```

```

para.boot <- function(data) (length(data)-1)/mean(data)/length(data)
para.boot.MLE <- boot(x, para.boot, R=50000, sim="parametric",
  ran.gen= para.rg, mle=lambdaMLE)$t
BESEmeanPara <- sapply(1:500*100, function(n) sd(para.boot.MLE[1:n]))
truevalue <- sqrt(y^2/(n-2))
plot(1:500*100, BESEmeanPara, type="l", main="evolution of BESE w.r.t. b",
  xlab="b", ylab="BESE", col="blue",
  ylim=c(min(truevalue,BESEmeanPara) , max(BESEmeanPara)))
lines(1:500*100, truevalue*rep(1,500))
legend("bottomright",c("true value of s.e.", "BESE"),col=c("black", "blue"),lty=1)

```

A.2 Bootstrap confidence interval

```

quantile(boot(murder, function(x,y) mean(x[y]), R=1000)$t, c(0.025, 0.975))
quantile(boot(murder, function(x,y) median(x[y]), R=1000)$t, c(0.025, 0.975))
n<- 1000
h<- 5
normal <- matrix(0, 2, n)
percent <- matrix(0, 2, n)
bca <- matrix(0, 2, n)
for(i in 20:n)
{
  BootSd <- boot(murder, function(x,y) sd(x[y]), R=i*h)
  resIC <- boot.ci(BootSd, type = c("norm", "perc", "bca"))
  normal[,i] <- resIC$normal[c(2,3)]
  percent[,i] <- resIC$percent[c(4,5)]
  bca[,i] <- resIC$bca[c(4,5)]
}

bootnumber <- 20:n*h
bound <- c(min(bca[1,20:n], normal[1,20:n], percent[1,20:n]),
  max(bca[2,20:n], normal[2,20:n], percent[2,20:n]) )
plot(bootnumber, bca[1,20:n], type="l", ylim=bound, main="Comparison
  of confidence intervals for the standard deviation",
  xlab="bootstrap number", ylab="confidence interval")
lines(bootnumber, bca[2,20:n])
lines(bootnumber, percent[1,20:n], col="green")
lines(bootnumber, percent[2,20:n], col="green")
lines(bootnumber, normal[1,20:n], col="blue")
lines(bootnumber, normal[2,20:n], col="blue")

legend("bottomright",c("BCa", "percentile", "Gaussian"),
  col=c("black", "green", "blue"),lty=1)

```

A.3 Hypothesis test

A.3.1 Non-parametric

```
data(sleep)
drg1 <- sleep[,1][1:10]
drg2 <- sleep[,1][11:20]
bootRepl <- boot(sleep[,1], function(x,y) mean(x[y]), R=1000)$t
           - boot(sleep[,1], function(x,y) mean(x[y]), R=1000)$t
sum((bootRepl > mean(drg1)-mean(drg2)))/1000
```

A.3.2 Parametric

```
rexp(10, y <- rlnorm(1))
xtilde <- matrix(rexp(10*100, 1.8), 10, 100)
1/colSums(xtilde)*10
```