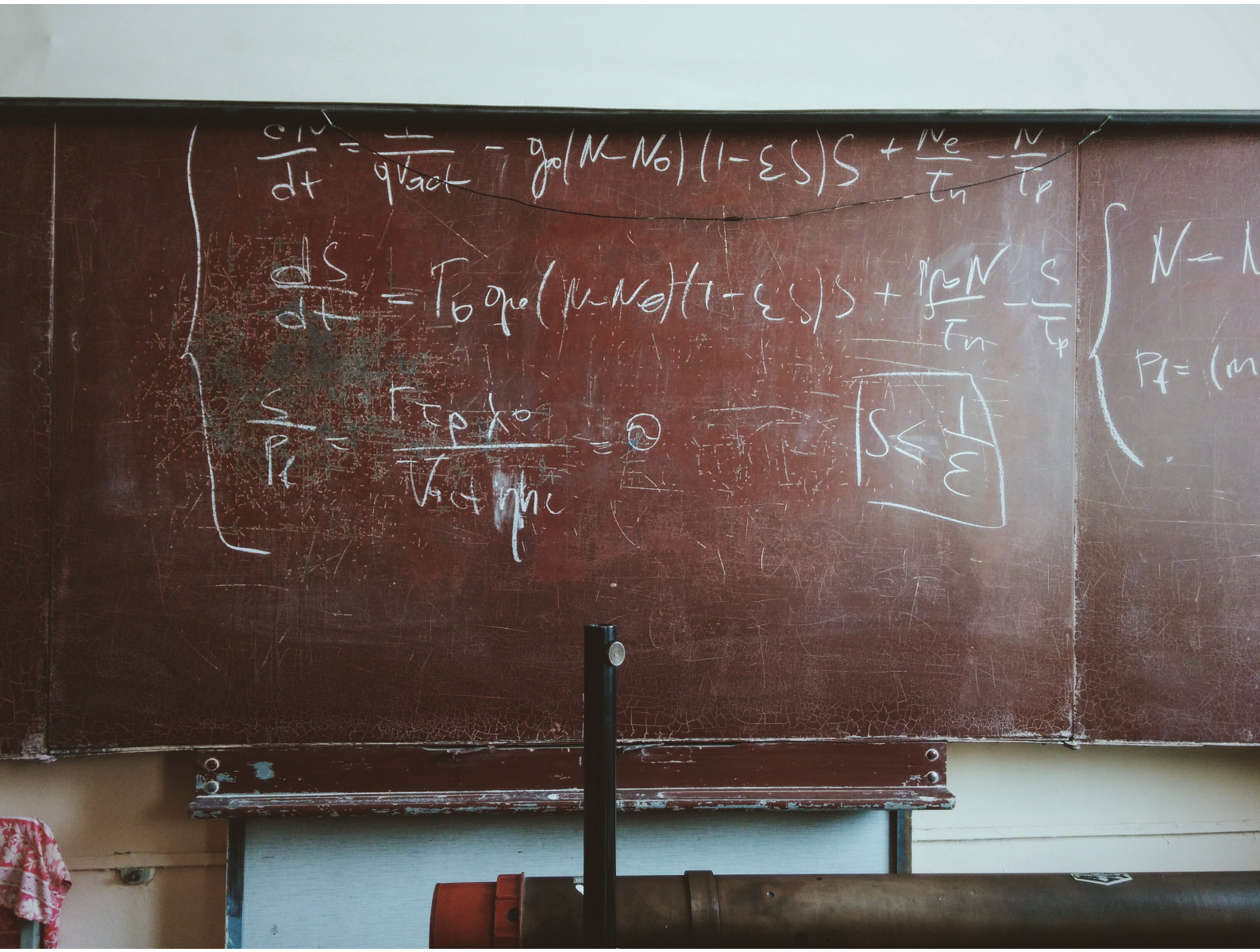
An Introduction to the Bootstrap Method

An exploration about bootstrap method, the motivation, and how it works

Lorna Yen Jan 26, 2019 · 17 min read



Bootstrap is a powerful, computer-based method for statistical inference without relying on too many assumption. The first time I applied the bootstrap method was in an A/B

inference. Not only that, in fact, it is widely applied in other statistical inference such as confidence interval, regression model, even the field of machine learning. That's lead me go through some studies about bootstrap to supplement the statistical inference knowledge with more practical other than my theory mathematical statistics classes.

This article is mainly focus on introducing the core concepts of Bootstrap than its application. But some embed codes will be used as a concept illustrating. We will do a introduction of Bootstrap resampling method, then illustrate the motivation of Bootstrap when it was introduced by Bradley Efron(1979), and illustrate the general idea about bootstrap.

Related Fundamental knowledge

The ideas behind bootstrap, in fact, are containing so many statistic topics that needs to be concerned. However, it is a good chance to recap some statistic inference concepts! The related statistic concept covers:

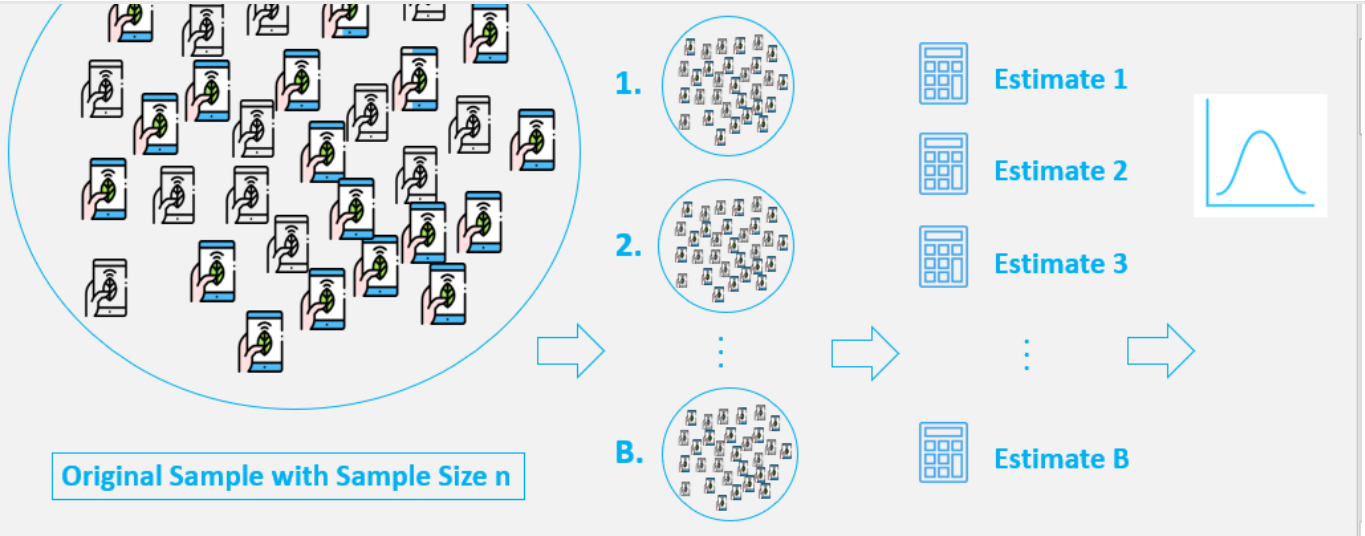
- Basic Calculus and concept of function
- Mean, Variance, and Standard Deviation
- Distribution Function (CDF) and Probability Density Function (PDF)
- Sampling Distribution
- Central Limit Theory, Law of Large Number and Convergence in Probability
- Statistical Functional, Empirical Distribution Function and Plug-in Principle

Having some basic knowledge above would help for gaining basic ideas behind bootstrap. Some ideas may cover with advance statistic, but I will use a simple way and not very formal mathematics expressions to illustrate basic idea as simple as I can. Links at the end of the article will be provided if you want to learn more about these concepts.

The Bootstrap Sampling Method

The basic idea of bootstrap is make inference about a **estimate**(such as sample mean) for a population parameter θ (such as population mean) on sample data. It is a resampling method by independently sampling with replacement from an existing sample data with same sample size n, and performing inference among these resampled data.

Generally, bootstrap involves the following steps:



1. A sample from population with sample size n.
2. Draw a sample from the original sample data **with replacement** with size n, and replicate B times, each re-sampled sample is called a Bootstrap Sample, and there will totally B Bootstrap Samples.
3. Evaluate the **statistic** of θ for each Bootstrap Sample, and there will be totally B estimates of θ.
4. Construct a **sampling distribution** with these B Bootstrap statistics and use it to make further statistical inference, such as:

- Estimating the standard error of statistic for θ.
- Obtaining a Confidence Interval for θ.

We can see we generate new data points by re-sampling from an **existing sample**, and make inference just based on these new data points.

How and why does bootstrap work?

In this article, I will divide this big question into three parts:

1. What's the **initial motivation** that Efron introduced the bootstrap?
2. Why use the **simulation technique**? In other word, how can I find a estimated variance of statistic by resampling?
3. What's the main idea that we need to draw a sample from the original sample **with replacement** ?

with the help of modern computer power. When Efron introduced the method, it was particularly motivated by evaluating of the **accuracy of an estimator** in the field of statistic inference. Usually, estimated standard error are an first step toward thinking critically about the accuracy statistical estimates.

Now, to illustrate how bootstrap works and how an estimator's standard error plays an important role, let's start with a simple case.

Scenario Case

> *Imagine that you want to summarize how many times a day do students pick up their smartphone in your lab with totally **100** students. It's hard to summarize the number of pickups in whole lab like a census way. Instead, you make a online survey which also provided the pickup-counting APP. In the next few days, you receive **30** students responses with their number of pickups in a given day. You calculated the **mean** of these 30 pickups and got an **estimate for** pickups is 228.06 times.*

```
sample
```

```
Out[5]: array([166, 201, 458, 190, 445,  87, 385, 427, 387, 166, 474,
        49, 430,
               205,  54, 343, 413, 389,  20,  58, 191,  87, 463,  88,
        389,  52,
               102,   1, 102,  20])
```

```
In [6]: # our first sample mean
        sample_mean = sample.mean()
        sample_mean
```

```
Out[6]: 228.06666666666666
```

```
In [7]: # standard deiveation for this sample
        sample_std = np.std(sample, ddof=1)
        sample_std
```

```
Out[7]: 166.96890756052164
```

```
In [8]: # estimated standard error for the sapmle mann
        sample_std/(30 ** 0.5)
```

```
Out[8]: 30.48421235763086
```

```
In [ ]:
```

Medium_06_1.ipynb hosted with ❤️ by GitHub view raw

Codes for this case, just feel free to check out.

can do is just evaluate the **population parameter** through an **estimator** based on an observed sample, and then get an **estimate** as the evaluation of average smartphone usage in the lab.

- **Estimator/Statistic**: A rule for calculating an estimate**.** In this case is Sample mean, always denoted as $\bar{X}$.
- **Population Parameter:** Numeric summary about a population. In this case is the average time of phone pickups per day in our lab, always denoted as μ.

One key question is — **How accurate is this estimate result?**

Because of the ***sampling variability***, it is virtually never that $\bar{X} = \mu$ occurs**.** Hence**,** besides reporting the value of a point estimate, some indication about the precision should be given. The common measure of accuracy is the standard error of the estimate.

The Standard Error

The *standard error* of an estimator is it's standard deviation. It tells us how far your sample estimate deviates from the actual parameter. If the standard error itself involves unknown parameters, we used the *estimated standard error* by replacing the unknown parameters with an estimate of the parameters.

Let's take an example. In our case, our estimator is sample mean, and for sample mean(and nearly only one!), we have an simple formula to easily obtain it's standard error.

Standard Error of Sample Mean

$$\sigma_{\bar{x}} = {}^{\sigma}/_{\sqrt{n}}$$

σ is the standard deviation of population.
n is the size of the sample

However, the standard deviation of population σ is always **unknown** in real world, so the most common measurement is the ***estimated standard error,*** which use the sample standard deviation S as a estimated standard deviation of the population:

$s = \sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n-1}$, *the sample standard deviation for n independent data points, and n is the size of the sample.*

In our case, we have sample with 30, and sample mean is 228.06, and the sample standard deviation is 166.97, so our *estimated standard error for our sample mean is* $166.97/\sqrt{30} = 30.48$.

Standard Error in Statistic Inference

Now we have got our estimated standard error. How can the standard error be used in the statistic inference? Let's use a simple example to illustrate.

Roughly speaking, if a estimator has a **normal distribution or a approximately a normal distributed**, then we expect that our estimate to be less than one standard error away from its expectation about 68% of the time, and less than two standard errors away about 95% of the time.

In our case, recall that the sample we collected is 30 response sample, which is sufficiently large in thumb rule, the **Central Limit Theorem** tells us the sampling distribution of $\bar{X}$ is closely approximated to a normal distribution. Combining the estimated standard error that, we can get:

> *We can be reasonably confident that the true of* μ, the the average times a day do students pick up their smartphone in our lab, *lies within approximately 2 standard error of* $\bar{X}$, *which is* $(228.06 - 2\times30.48,\ 228.06+2\times30.48) = (167.1,\ 289.02)$.

The Ideal and Reality in Statistic World

We have made our statistic inference. However, how this inference was going well is under some rigorous assumptions.

Let's recall what assumption or classical theorem we may have used so far:

estimated standard error.

- We assume we know or can estimate about the estimator's population. In our case is the **approximated normal distribution.**

However, in our real world, sometimes it's hard to meet assumptions or theorem like above:

- It's hard to know the information about population, or it's distribution.
- The standard error of a estimate is hard to evaluate in general. **Most of time, there is no a precise formula like standard error of sample mean**. If now, we want to make a inference for the *median* of the smart phone pickups, what's the standard error of sample *median*?

This is why the bootstrap comes in to address these kind of problems. When these assumptions are violated, or when no formula exists for estimating standard errors , bootstrap is the powerful choice.

II. Explanation about Bootstrap

To illustrate the main concepts, following explanation will evolve some mathematics definition and denotation, which are kind of informal in order to provide more intuition and understanding.

1. Initial Scenario

Assume we want to estimate the standard error of our statistic to make an inference about population parameter, such as for constructing the corresponding confidence interval (just like what we have done before!). And:

- We don't know anything about population.
- There is no precise formula for estimating the standard error of statistic**.**

Let X1, X2, … , Xn be a random sample from a population P with distribution function F**.** And let ***M*= g(X1, X2, …, Xn)**, be our **statistic for parameter of interest,** meaning that the statistics a function of sample data X1, X2, …, Xn. What we want to know is the **variance** of M*,* denoted as Var(M).

- First, since we don't know anything about population, we can't determine the value of Var(M) that requires known parameter of population*,* so we need to estimate

- Second, in real world we always don't have a simple formula for evaluating the EST_Var(M) other than the sample mean's.

It leads us need to **approximate** the EST_Var(M). How? Before answer this , let's introduce an common practical way is **simulation, assume we know P**.

2. Simulation

Let's talk about the idea of simulation. It's useful for obtaining information about a statistic's sampling distribution with the aid of computers. But it has an important assumption — Assume we know the population ***P.***

Now let X1, X2, … , Xn be a random sample from a population and assume M= g(X1, X2, …, Xn) is the statistic of interest, we could approximate mean and variance of statistic ***M*** by simulation as follows:

1. Draw random sample with size n from P**.**
2. Compute statistic for the sample.
3. Replicate B times for process 1. and 2 and get B statistics.
4. Get the **mean** and **variance** for these B statistics.

Why does this simulation works? Since by a classical theorem, the ***Law of Large Numbers*****:**

- The mean of these B statistic converges to the true mean of statistic *M* as B → ∞.

And by Law of Large Numbers and several theorem related to **Convergence in Probability**:

With the aid of computer, we can make B as large as we like to approximate to the sampling distribution of statistic *M*.

Following is the example Python codes for simulation in the previous phone-picks case. I use B=100000, and the simulated mean and standard error for sample mean is very close to the theoretical results in the last two cells. Feel free to check out.

Example codes for simulation applied with the previous phone-picks case start from cell [10].

3. The Empirical Distribution Function and Plug-in Principle

We have learned the idea of simulation. Now, can we **approximate** the *EST_Var(M)* by simulation? Unfortunately, to do the simulation above, we need to know the information about population P. The truth is that we don't know anything about the P. For addressing this issue, one of most important component in bootstrap Method is adopted:

Using **Empirical distribution function** to approximate the **distribution function** of population**,** and applying **Plug-in Principle to get an estimate for** Var(M) — the **Plug-in estimator**.

(1) Empirical Distribution Function

The idea of **Empirical distribution function (EDF)** is building an distribution function (CDF) from an existing data set. The EDF usually approximates the CDF quite well, especially for large sample size. In fact, it is a common, useful method for estimating a CDF of a random variable in pratical.

The EDF is a discrete distribution that gives equal weight to each data point (i.e., it assigns probability 1/ n to each of the original n observations), and form a cumulative

(2) Statistical Functional

Bootstrap use the EDF as an estimator for CDF of population. However, we know the EDF is a type of cumulative distribution function(CDF). **To apply the EDF as an estimator for our statistic M, we need to make the form of M as a function of CDF type, even the parameter of interest as well to have the some base line.** To do this, a common way is the concept called Statistical Functional. Roughly speaking, a statistical functional is any function of a distribution function. Let's take an example:

Suppose we are interested in parameters of population. In statistic field , there is always a situation where **parameters of interest is a function of the distribution function**, these are called statistical functionals. Following list that population mean E(X) is a statistical functional:

From above we can see the mean of population E(X) can also be expressed as a **form of CDF of population *F* —** this is a statistical functional. Of course, this expression can be applied to any function other than mean, such as variance.

Statistical functional can be viewed as quantity describing the features of the population. The mean, variance, median, quantiles of F are features of population. Thus, using statistical functional, we have a more rigorous way to define the concepts of population parameters. Therefore, we can say, our statistic M can be : $M=g(F)$, with the population CDF F.

(3) Plug-in Principle = EDF + Statistical Functional

We have made our statistic is $M= g(X1, X2, \ldots, Xn)=g(F)$ be a statistical functional form. However, we don't know F. So we have to "plug-in" a estimator for F, "into" our $M=g(F)$, in order to make this M can be evaluate.

It is called **plug-in principle.** Generally speaking, the plug-in principle is a method of **estimation of statistical functionals** from a population distribution by evaluating the same functionals, but with the empirical distribution which is based on the sample. This estimation is called a **plug-in estimate** for the population parameter of interest. For example, a median of a population distribution can be approximated by the median of the empirical distribution of a sample. The empirical distribution here, is form just by the sample because we don't know population. Put it simply:

- If our parameter of interest , say θ, has the statistical function form $\theta=g(F)$, which F is population CDF.
- The plug-in estimator for $\theta=g(F)$, is defined to be $\theta\_hat=g(F\_hat)$:

- From above formula we can see we "plug in" the θ_hat and F_hat for the unknown θ and F. F_hat here, is purely estimated by sample data.
- **Note that both of the θ and θ_hat are determined by the same function g(.).**

Let's take an mean example as follows, we can see g(.) for mean is — averaging all data points, and it is also applied for sample mean. F_hat here, is form by sample as an

So, what is the F_hat? Remember bootstrap use Empirical distribution function(EDF) as an estimator of CDF of population? In fact, EDF is also a common estimator that be widely used in plug-in principle for F_hat.

Let's take a look what does our estimator $M = g(X1, X2, \ldots, Xn) = g(F)$ will look like if we plug-in with EDF into it.

- Let Statistic of interest be $M = g(X1, X2, \ldots, Xn) = g(F)$ from a population CDF F.
- We don't know F, so we build a Plug-in estimator for M, M becomes M_hat = g(F_hat). Let's rewrite M_hat as follows:

According this, for our mean example, we can find the plug-in estimator for mean **μ** is just the sample mean:

Hence, we through Plug-in Principle, to make an estimate for M=g(F), say M_hat=g(F_hat). And remember that, what we want to find out is Var(M), and we approximate Var(M) by Var(M_hat). But in general case, there is no precise formula for Var(M_hat) other than sample mean! It leads us to apply a simulation.

(4) Bootstrap Variance Estimation

It's nearly the last step! Let's refresh the whole process with the Plug-in Principle concept.

Our goal is to estimate the ***variance of our estimator M, which is Var(M).*** The Bootstrap principle is as follows:

1. We don't know the population P with CDF denoted as F, so bootstrap use **Empirical distribution function(EDF) as estimate of F.**

M=g(F) becomes M_hat= g(F_hat), it's the plugged-in estimator with EDF — F_hat.

4. Take **simulation** to **approximate to the *Var(M_hat)*.**

Recall that to do the original version of simulation, we need to draw a sample data from population, obtain a statistic M=g(F) from it, and replicate the procedure B times, then get variance of these B statistic to approximate the true variance of statistic.

Therefore, to do simulation in step 4, we need to:

1. Draw a sample data from **EDF.**
2. Obtain a **plug-in** statistic M_hat= g(F_hat).
3. Replicate the two procedure B times.
4. Get the variance of these B statistic, **to approximate the true variance of plug-in statistic**.(It's an easily confused part.)

What's the **simulation? In fact, it is the bootstrap sampling process that we mentioned in the beginning of this article!**

Two questions here(I promise these are last two!):

1. **How does draw from EDF look like in step 1?**
2. How does this simulation work?

How does draw from EDF look like?

We know EDF builds an CDF from existing sample data X1, …, Xn, and by definition it puts mass 1/n at each sample data point. Therefore, drawing an random sample from an EDF, can be seen as drawing n observations, with replacement, from our existing sample data X1, …, Xn. **So that's why the bootstrap sample is sampled with replacement as shown before.**

How does simulation work?

The **variance** of plug-in estimator M_hat=g(F_hat) is what the bootstrap simulation want to simulate. At the beginning of simulation, we draw observations with replacement from our existing sample data X1, …, Xn. Let's denote these re-sampled data X1* , …, Xn*. Now, let's compare bootstrap simulation with our original simulation version again .

Original Simulation Version- Approximate EST_Var(M|F) with known F

Let X1, X2, … , Xn be a random sample from a population *P* and assume *M*= g(X1, X2, …, Xn) is the statistic of interest, we could approximate variance of statistic *M* by simulation as follows:

1. Draw random sample with size n from P.
2. Compute statistic for the sample.
3. Replicate B times for process 1. and 2 and get B statistics.
4. Get the variance for these B statistics.

Same with previous Simulation part for simulating Var(M).

Bootstrap Simulation for Var(M_hat=g(F_hat))

Bootstrap Simulation Version- Approximate Var(M_hat|F_hat) with EDF

Now let X1, X2, … , Xn be a random sample from a population *P with CDF F,* and assume *M*= g(X1, X2, …, Xn ;F) is the statistic of interest. But we don't know F, so we:

1.Form a EDF from the existing sample data by draw observations with replacement from our existing sample data X1, …, Xn. These are denote as X1∗, X2∗, …, Xn∗. We call this is a bootstrap sample.

2.Compute statistic *M_hat*= g(X1∗, X2∗, …, Xn∗ ;F_hat) for the bootstrap sample.

3. Replicate B times for steps 2 and 3, and get B statistics M_hat.

4. Get the variance for these B statistics to approximate the **Var(M_hat).**

Simulating for Var(M_hat).

Would you feel familiar with processes above? In fact, it's the same process with bootstrap sampling method we have mentioned before!

III. What Does the Bootstrap Work?

Finally, let's check out how does our simulation will work. **What we will get the approximation from this bootstrap simulation is for Var(M_hat)**, but what we really concern is whether **Var(M_hat) can approximate to Var(M).** So two question here:

1. **Will bootstrap variance simulation result, which is S^2, can approximate well for Var(M_hat)?**
2. **Can Var(M_hat) can approximate to Var(M)?**

To answer this ,let's use a diagram to illustrate the both types simulation error:

1. From bootstrap variance estimation, we will get an estimate for Var(M_hat) — the plug-in estimate for Var(M). And the Law of Large Number tell us, if our simulation times B is large enough, the bootstrap variance estimation S^2, is a good approximate for Var(M_hat). Fortunately, we can get a larger B as we like with aid of a computer. So this simulation error can be small.

estimator approximate well to the estimator of interest ? That's the key point what we really concern. In fact, the topic of asymptotic properties for plug-in estimators is classified in high level mathematical statistic. But let's explain the main issues and ideas.

- First, We know the empirical distribution will converges to true distribution function well if sample size is large, say F_hat → F.
- Second, if F_hat → F, and if it's corresponding statistical function g(.) is a **smoothness conditions, then** g(F_hat) → g(F). In our case, the statistical function g(.) is *Variance*, which satisfy the required continuity conditions. Therefore, that explains why the bootstrap variance is a good estimate of the true variance of the estimator M.

Generally, the **smoothness conditions** on some functionals is difficult to verify. Fortunately, most common statistical functions like mean, variance or moments satisfy the required continuity conditions. It provides that bootstrapping works. And of course, make the original sample size not too small as we can.

Below is my Bootstrap sample code for pickup case, fell free to check out.

Bootstrap Recap

Let's recap the main ideas of bootstrap with following diagram!

Bootstrap has been applied to a much wider level of practical cases, it is more constructive to learn start from the basic part. Thanks for reading so far and hope this article helps! Leave your comments if I've made any mistakes :) !

Reference

Most helpful book by Efron, with more general concept of Bootstrap and how it connects to statistical inference.

- An Introduction to the Bootstrap

Also a helpful book, form EDF to Bootstrap method

- All of Statistics: A Concise Course in Statistical Inference

Other book:

- Bootstrap Methods And Their Application
- An Introduction to Bootstrap Methods with Applications to R

Empirical Distribution Function and Plug-in Principle

- http://faculty.washington.edu/yenchic/17Sp_403/Lec9_theory.pdf
- https://www.statlect.com/asymptotic-theory/empirical-distribution
- http://bjlkeng.github.io/posts/the-empirical-distribution-function/
- http://pub.math.leidenuniv.nl/~szabobt/STAN/STAN7.pdf

Other Materials

- http://www.stat.cmu.edu/~larry/=stat705/Lecture13.pdf
- http://faculty.washington.edu/yenchic/17Sp_403/Lec5-bootstrap.pdf
- https://web.as.uky.edu/statistics/users/pbreheny/764-F11/notes/12-6.pdf

Data Science Statistics Bootstrapping Statistical Inference Towards Data Science