

Bootstrap Methods

Suppose a sample X_1, X_2, \dots, X_n is to be used to estimate a population parameter, θ .

We form a statistic T that estimates θ . Our interest is then in the sampling distribution of T .

It's often intractable.

The basic idea of the bootstrap is that the true population can be approximated by an infinite population in which each of the n sample points are equally likely.

The parameter is a functional of a population distribution function:

$$\theta = \int g(x) dP(x)$$

The estimator is often the same functional of the empirical distribution function:

$$T = \int g(x) dP_n(x)$$

Various properties of the distribution of T can be estimated by use of “bootstrap samples” .

- bias
- variance and standard deviation
- other moments (lower-order)

The bootstrap is a “resampling” method, as is the jackknife.

Plug-In Estimators

Problem: given a random sample (x_1, x_2, \dots, x_n) from an unknown distribution with CDF P , we want to estimate a parameter, $\theta = \theta(P)$.

A possible estimate is the “*plug-in estimate*”, $T = \theta(P_n)$, where P_n is the empirical distribution function (“*plug-in principle*”).

If no additional information on P is available, the plug-in estimator is the best estimator in terms of most asymptotic measures ($n \rightarrow \infty$). (What kinds of additional information might be available? P is a member of some parametric family, or along with each x_i we have another variable u , so we might use a regression estimate.)

Bootstrap Observations

A *bootstrap sample*: a sample, $(x_1^*, x_2^*, \dots, x_n^*)$, drawn from the empirical distribution resulting from (x_1, x_2, \dots, x_n) . (Although this is operationally equivalent to sampling with replacement, conceptually it is a random sample from a discrete distribution whose support has cardinality n .)

Suppose we have an estimate $T = t(x)$, where x is the sample, (x_1, x_2, \dots, x_n) .

$T^* = t(x^*)$, where x^* is the bootstrap sample, is a *bootstrap observation* of T .

The bootstrap estimate of some function of the estimator T is a plug-in estimate that uses the empirical distribution P_n in place of P . This is the *bootstrap principle*, and this bootstrap estimate is called the *ideal bootstrap*.

Basic Ideas of the Bootstrap

The bootstrap is a method of statistical inference; that is, the objective in using a bootstrap method is to estimate some parameter of a distribution (population), or to perform a test of an hypothesis about the parameter.

A parameter of a population is a functional of the distribution function, P ; for example, the mean:

$$\mu = \int x \, dP(x),$$

or the variance:

$$\sigma^2 = \int (x - \mu)^2 \, dP(x).$$

In general, we may write a parameter both as a (real-valued) functional and as the value of the functional:

$$\theta = \theta(P)$$

The usual assumption is that the sample is a collection of independent entities, all from the same distribution.

The central idea behind the bootstrap is to let the sample play the role of the population. The same functional of the distribution function of the sample should then be a good estimate of the parameter.

Other methods of statistical inference arise from this same idea. Think of the method of moments: for estimating a first moment (the mean) of a population, use the sample first moment.

If we let the sample play the role of the population, the distribution function for that population is the *empirical distribution function* from the sample.

Because it corresponds to the population distribution function, P , we will denote it by P_n to emphasize that it relies on a sample of size n . It can be formed from the (discrete) probability density function that puts mass $1/n$ at each point in the sample of size n .

A bootstrap statistic corresponding to $\theta = \theta(P)$ is

$$T = \theta(P_n).$$

The mean, for example, would be

$$\hat{\mu} = \int x \, dP_n(x).$$

The mean is pretty easy to compute. This is not the case for all bootstrap statistics.

Notation:

Sample – the collection $\{X_1, X_2, \dots, X_n\}$.

Sometimes, we will think of these as random variables, other times, as realizations of random variables.

It is not convenient to think of the sample as a set, because we do not wish to emphasize the uniqueness of the elements; it is better to think of the sample as a vector,

$$X = (X_1, X_2, \dots, X_n),$$

except we are not concerned about the ordering.

Resample – the collection $\{X_1^*, X_2^*, \dots, X_n^*\}$, where each of the X_i^* 's is chosen independently and with equal probability from the sample. (That implies sampling “with replacement”.)

We will use similar notation as above:

$$X^* = (X_1^*, X_2^*, \dots, X_n^*)$$

Bootstrap Bias Estimation

The problem in its broadest setting is to find a functional f_t (from some class of functionals) that allows us to relate the distribution function of the sample P_n to the population distribution function P , that is, such that

$$E(f_t(P, P_n)|P) = 0.$$

Suppose we are trying to estimate $h(\int g(x)dP(x))$. The h presents special problems.

For example, suppose we wish to estimate

$$\theta = \left(\int x dP(x) \right)^r$$

Start with

$$T = \left(\int x dP_n(x) \right)^r = \bar{x}^r$$

This is biased, however.

The Bootstrap Principle

Correcting for the bias is equivalent to finding t that solves the equation

$$f_t(P, P_n) = T(P_n) - \theta(P) + t$$

so that f_t has zero expectation with respect to P .

What about repeating this whole process?

Take a sample from the population with distribution function P_n . Let $P_n^{(1)}$ be the ECDF of that sample. Look for f_t so that

$$E(f_t(P_n, P_n^{(1)}) | P_n) = 0.$$

The difference is we know more about this equation because we know more about P_n .

This is the *bootstrap principle*.

Monte Carlo Estimation

How can we compute the (definite) integral

$$\int_S h(x) dx$$

One way to do it is to represent $h(x)$ as $g(x)p(x)$, where $p = dP$ is a “nice” probability density function (pdf) over S . (For example, suppose $S = [0, 2]$, then let $p(x) = 1/2$ over $[0, 2]$, and $p(x) = 0$ elsewhere. This is a nice density; it is uniform.)

Now, because

$$\int_S h(x) dx = E_P(g(X)),$$

where E_P means the expected value wrt the distribution with cumulative distribution function P , we can unbiasedly estimate $\int_S h(x) dx$ by

$$\frac{1}{n} \sum_{i=1}^n g(X_i),$$

where the X_i 's are a random sample from a population with CDF P .

This is called a *Monte Carlo estimate* of $\int_S h(x) dx$. It is a commonly used numerical method for evaluating a definite integral.

For example, suppose we want to evaluate the integral

$$\int_0^2 \log(x + 1)x^2(2 - x)^3 dx.$$

We can write this as

$$\int_0^2 2 \log(x + 1)x^2(2 - x)^3 f(x) dx,$$

where $f(x)$ is the uniform density over $[0, 2]$.

We would then generate n random deviates x_i from the uniform $[0, 2]$ and form

$$\sum_{i=1}^n 2 \log(x_i + 1)x_i^2(2 - x_i)^3 / n.$$

Alternatively, we can write the original integral as

$$\int_0^2 \frac{64}{35} \log(x + 1) f_B(x) dx,$$

where $f_B(x)$ is the density of a beta (3,4) random variable scaled to be over $[0, 2]$.

So another way of estimating the integral is to generate n random deviates from the beta (3,4), scale them to be x_i over $[0, 2]$, and form

$$\sum_{i=1}^n \frac{64}{35} \log(x_i + 1) / n.$$

Monte Carlo in the Bootstrap

In the bootstrap, we want to evaluate $T = \theta(P_n)$, which is often of the form $\int g(x) dP_n(x)$.

What is particularly attractive here is the fact that $dP_n(x)$ is just $1/n$ at each point in the sample:

$$dP_n(x) = \begin{cases} \frac{1}{n} & \text{for } x = x_i \quad (\text{a sample point}) \\ 0 & \text{elsewhere.} \end{cases}$$

So, to sample from the discrete distribution with CDF $P_n(x)$ is just to *resample*; that is, sample from the original sample.

In most applications of the bootstrap, this is exactly what we do.

Notice, that the Monte Carlo aspect is not the essential idea of the bootstrap; it is just a simple way of evaluating the integral.

Monte Carlo in the Bootstrap

So we have a statistic, T_n , from a sample of size n from a population with distribution function P , and we want to know something about the sampling distribution of T_n . The distribution of T_n is likely to depend on n

Suppose, for example, we want to know the variance of T_n . The variance of T_n is

$$\int \left(T_n - \int T_n dP \right)^2 dP$$

The sample analog is

$$\int \left(T_n - \int T_n dP_n \right)^2 dP_n$$

The Monte Carlo estimation procedure is

1. take sample $X_1^*, X_2^*, \dots, X_n^*$, of size n from P_n
2. compute T_n^{*1}
3. repeat these steps, obtaining m T_n^{*i} 's, and compute their sample variance.

Parametric Bootstrap

Another way of approaching the problem is to assume a family of distributions for the population. The given sample is then used to determine the specific member of the family of distributions; that is, to estimate the parameters of the distribution.

In this case, P is assumed known up to a finite set of unknown parameters, λ , which includes the parameter of interest. Instead of using P_n we use P with λ replaced by its sample estimates (of some kind).

The procedure is as before, except that the elements $\{X_1^*, X_2^*, \dots, X_n^*\}$ are sampled from a distribution from the assumed family and with parameters that are estimated from the original sample.

The Mechanics of the Nonparametric Bootstrap

The basic bootstrap procedure is to take m random samples each of size n *with replacement* from the given set of data, the original sample X_1, X_2, \dots, X_n ; and for each sample, compute an estimate T_j of the same functional form as the original estimator T .

The distribution of the T_j 's is related to the distribution of T .

The variability of T about Θ can be assessed by the variability of T_j about T ; the bias of T can be assessed by the mean of $T_j - T$.

The Mechanics of the Parametric Bootstrap

In a parametric bootstrap procedure, the first step is to obtain estimates of the parameters that characterize the distribution within the assumed family. After this, the procedure is very similar to that described above: generate m random samples each of size n from the estimated distribution, and for each sample, compute an estimate T_j of the same functional form as the original estimator T .

The sampling is done with replacement, as usual in random number generation.

The distribution of the T_j 's is used to make inferences about the distribution of T .

Bootstrap Estimate of the Variance of an Estimator

For the variance of T the bootstrap estimator is $V(T^*)$; that is, it is the variance of T based on samples of size n taken from P_n .

If T is the sample mean, for example, the bootstrap estimate of the variance $\hat{\sigma}/n$, where $\hat{\sigma}$ is an estimate of the variance of the underlying sample. (This is true no matter what the underlying distribution is.)

There's really no way the bootstrap procedure helps in this situation.

Monte Carlo Bootstrap

Usually, however, we cannot simply write the variance of T in terms of some other simple estimator. We usually have to resort to bootstrap sampling. Example: the standard deviation of the correlation coefficient in the law school data.

Generate m samples of size 15 with replacement from the original data (`ind <- sample(15, replace=T)` each time); compute correlation (`cor(law[c(ind),1],law[c(ind),2])`); and compute variance or standard deviation of these correlation coefficients.

How Large Should the Monte Carlo Bootstrap Sample Be

In the ideal bootstrap, there is no sampling, or equivalently, the sample size is ∞ . In most applications, of course, we do bootstrap sampling. Our bootstrap estimate is $V_m(T^*)$ instead of the ideal $V(T^*)$, or $V_\infty(T^*)$.

How many bootstrap observations should we take?

This depends on the underlying distribution. The variance of the bootstrap estimator is decreasing, but it is not approaching 0!

This also depends on what kind of inference is being made. Setting confidence intervals, for example, usually would suggest larger bootstrap sample sizes than just estimating the variance. This is similar to the situation in ordinary inference; it takes larger samples of the underlying distribution to estimate a probability density than it does to estimate a mean.

Monte Carlo Bootstrap Sample Size

It is a good idea to consider the sample of bootstrap observations as a set of data, rather than just simply computing its variance. (By the way, how do you compute its variance? Use m or $m - 1$? (It doesn't really matter.)) This means plotting a histogram of the bootstrap observations. Is it skewed? How spread out is it? etc.

In practice, bootstrap samples of size a few hundred (200 or so) are usually sufficient for point estimates. For confidence intervals, a few thousand (2000 or so) may be more appropriate. For selecting optimal bandwidth in density estimation, 100 or so may be OK. (Notice I did not say estimating the variance of the density estimate. The bootstrap is often used in selecting parameters in adaptive inference schemes. In such cases, we may get by with smaller bootstrap samples.)

An aside on “sample” .

In statistics, this word is equivalent to “set” (or “set of indices”). A sample is a set of observations. Each observation is unique, even if it has the same value as another observation. In a bootstrap replication, two pseudo-observations corresponding to the same original observation are considered distinct (so the mathematical object is still a set). The bootstrap sample is a collection of replications. We usually denote the number of replications (the bootstrap sample size) by m .

In some disciplines, especially engineering, it is common to equate “sample” with “observation”. This can lead to lack of precision in the language; whereas using “sample” to mean a collection of “observations” rarely leads to confusion.

Parametric Bootstrap

In the parametric bootstrap, the parameters of the underlying distribution are estimated using the given sample, then m random pseudo samples of size n are generated from that parametric distribution. These m samples are then used for the bootstrap inference.

An aside:

How to generate multivariate normals.

If $x \sim \text{MVN}(0, I)$ and $y = Ax$, the $y \sim \text{MVN}(0, AA^T)$.

In the other direction, suppose we want $\text{MVN}(0, \Sigma)$. Find Σ_C such that $\Sigma = \Sigma_C^T \Sigma_C$, then take $y = \Sigma_C^T x$. The matrix Σ_C is an upper-triangular matrix called the Cholesky factor of Σ . This can be obtained in S-Plus using `chol` for the Cholesky decomposition. (Works for a symmetric positive definite or positive semi-definite matrix.)

Bias Reduction

Find f_t (i.e., t) so that

$$\mathbb{E}(f_t(P, P_n)|P) = 0$$

or

$$\mathbb{E}(T(P_n) - \theta(P) + t|P) = 0.$$

Change the problem to the sample:

$$\mathbb{E}(T(P_n^{(1)}) - T(P_n) + t_1|P_n) = 0,$$

whose solution is

$$t_1 = T(P_n) - \mathbb{E}(T(P_n^{(1)})|P_n),$$

so the bias-reduced estimate is

$$T_1 = 2T(P_n) - \mathbb{E}(T(P_n^{(1)})|P_n).$$

Bias Reduction

We may be able to compute $E(T(P_n)|P_n)$.

If so, do so.

If not, estimate by Monte Carlo.

Resampling to Estimate Bias

Problem: given a random sample (x_1, x_2, \dots, x_n) from an unknown distribution with df P , we want to estimate a parameter, $\theta = \theta(P)$.

One of the most important properties of an estimator,

$$T = t(x_1, x_2, \dots, x_n),$$

is its bias, $E_P(T) - \theta$. A “good” estimator, of course, has a small bias, probably even zero, i.e., the estimator is unbiased.

The plug-in estimator, $T = \theta(P_n)$, may or may not be unbiased, but its bias is often small (e.g., the plug-in estimator of σ^2 ; the bias correction is $n/(n - 1)$).

Of course, without fairly strong assumptions on the underlying distribution, it is unlikely that we would know the bias of an estimator. Resampling methods can be used to estimate the bias.

In particular, the *bootstrap estimate of the bias* is

$$E_{P_n}(T) - \theta(P_n)$$

The Bootstrap Estimate of the Bias

The bootstrap estimate of the bias is the plug-in estimate of $E_P(T) - \theta$. The plug-in step occurs in two places, for estimating $E_P(T)$ and then for estimating θ .

Consider the simple estimators (both plug-in estimators):

Sample mean:

$$E_{P_n}(T) - \theta(P_n) = 0.$$

The sample mean is unbiased for the population mean (if the population mean exists).

Sample second central moment:

$$E_{P_n}(T) - \theta(P_n) = -\sum(x_i - \bar{x})^2/n.$$

This is also what we would want.

This ideal bootstrap estimate must generally be approximated by Monte Carlo simulation.

The Monte Carlo Bootstrap Estimate of the Bias

The ideal estimate is an expected value of a function:

$$E_{P_n}(T) - \theta(P_n);$$

the Monte Carlo estimate of an expected value of a function is the Monte Carlo sample mean of that function.

For the Monte Carlo simulation, we generate a number of bootstrap samples, $(x_1^*, x_2^*, \dots, x_n^*)$, drawn from the empirical distribution resulting from (x_1, x_2, \dots, x_n) .

Letting $(x_1^{*j}, x_2^{*j}, \dots, x_n^{*j})$ represent the j^{th} bootstrap sample, the Monte Carlo estimate of the bootstrap estimate of the bias is

$$\sum t(x_1^{*j}, x_2^{*j}, \dots, x_n^{*j})/m - \theta(P_n)$$

For the Monte Carlo estimate we can define a *resampling vector*, P^* , corresponding to each bootstrap sample as the vector of proportions of the elements of the original sample in the given bootstrap sample.

The Resampling Vector

If the bootstrap sample $(x_1^*, x_2^*, x_3^*, x_4^*)$ is really the sample (x_2, x_2, x_4, x_3) , the resampling vector P^* is

$$(0, 1/2, 1/4, 1/4)$$

The resampling vector has random components that sum to 1.

The bootstrap replication of the estimator T is a function of P^* .

The Monte Carlo estimate of the bootstrap estimate of the bias can be improved if the estimator whose bias is being estimated is a plug-in estimator.

The Bootstrap Estimate of the Bias of a Plug-In Estimator

Consider the resampling vector, $P^0 = (1/n, 1/n, \dots, 1/n)$.

Such a resampling vector corresponds to a permutation of the original sample. If the estimator is a plug-in estimator, then its value is invariant to permutations of the sample; and, in fact,

$$\theta(P^0) = \theta(P_n),$$

so the Monte Carlo estimate of the bootstrap estimate of the bias,

$$\sum t(x_1^{*j}, x_2^{*j}, \dots, x_n^{*j})/m - \theta(P_n),$$

can be written as

$$\sum t(x_1^{*j}, x_2^{*j}, \dots, x_n^{*j})/m - \theta(P^0).$$

Instead of using $\theta(P^0)$, we can increase the precision of the Monte Carlo estimate by using the individual P^* 's actually obtained:

$$\sum t(x_1^{*j}, x_2^{*j}, \dots, x_n^{*j})/m - \theta(\sum P^{*j}/m),$$

that is, by using the mean of the resampling vectors. Notice that for an unbiased plug-in estimator, e.g., the sample mean, this quantity is 0.

Variance Reduction in Monte Carlo

The use of $\theta(\bar{P}^*)$ is a type of *variance reduction* in a Monte Carlo procedure.

Remember that a Monte Carlo procedure is estimating an integral. Suppose the integral is

$$\int (f(x) + g(x)) dx = \int f(x) dx + \int g(x) dx,$$

and suppose we know $\int g(x) dx$. What is the best way to do the Monte Carlo, to do the integral of the sum, or just to do the integral of $f(x)$ and add on the known value of the integral of $g(x)$?

What makes one estimator better than another?

Is the variance of $f(X) + g(X)$ smaller than the variance of $f(X)$?

Remember that the variance of the Monte Carlo estimator, \hat{I} , of an integral, $\int h(x) dx$, is generally proportional to the variance of $h(X)$, where X is a random variable. (There are, of course, different ways of using random variables in the Monte Carlo estimation of the integral.)

Variance Reduction in Monte Carlo

If the objective in Monte Carlo experimentation is to estimate some quantity, just as in any estimation procedure, we want to reduce the variance of our estimator (while preserving its other good qualities).

The basic idea is usually to reduce the problem analytically as far as possible, and then to Monte Carlo what is left.

Beyond that general reduction principle, in Monte Carlo experimentation, there are several possibilities for reducing the variance.

Variance Reduction in Monte Carlo

- judicious use of an auxiliary variable
 - control variates (any correlated variable, either positively or negatively correlated)
 - antithetic variates (in the basic uniform generator)
 - regression methods
- use of probability sampling
 - discrete: stratified sampling
 - continuous: importance sampling

Balanced Resampling

Another way of reducing the variance in Monte Carlo experimentation is to constrain the sampling so that some aspects of the samples reflect precisely some aspects of the population.

What about constraining \bar{P}^* so as to equal P^0 ? This makes $\theta(\bar{P}^*) = \theta(P^0)$, and hopefully makes $\sum t(x_1^{*j}, x_2^{*j}, \dots, x_n^{*j})/m$ closer to its expected value, while preserving its correlation with $\theta(\bar{P}^*)$.

This is called *balanced resampling*.

Hall (1990) has shown that the balanced resampling Monte Carlo estimator of the bootstrap estimator has a bias $O(m^{-1})$, but that the reduced variance generally more than makes up for it.