MS&E 226: "Small" Data

Lecture 13: The bootstrap (v3)

Ramesh Johari
ramesh.johari@stanford.edu

Resampling

Sampling distribution of a statistic

For this lecture:

- There is a population model that gives the distribution (cdf) G of an outcome Y.
- We observe data $\mathbf{Y}$ that comes from this population model.
- We compute a *statistic* $T(\mathbf{Y})$ given the data; an example might be an estimate for a parameter.
- The *sampling distribution* of $T(\mathbf{Y})$ is the distribution of values for the statistic that are obtained over many "parallel universes", where the same process is repeated.

In this lecture we develop a flexible and powerful approach to estimating the sampling distribution of any statistic: the *bootstrap*.

Idea behind the bootstrap

The basic idea behind the bootstrap is straightforward:

- The data $\mathbf{Y}$ are a sample from the population model.

Idea behind the bootstrap

The basic idea behind the bootstrap is straightforward:

- The data $\mathbf{Y}$ are a sample from the population model.
- If we *resample* (with replacement) from $\mathbf{Y}$, this “mimics” sampling from the population model.

Idea behind the bootstrap

The basic idea behind the bootstrap is straightforward:

- The data $\mathbf{Y}$ are a sample from the population model.
- If we *resample* (with replacement) from $\mathbf{Y}$, this "mimics" sampling from the population model.

In the bootstrap, we draw B new samples (of size n) from the original data, and treat each of these as a "parallel universe."

We can then pretend this is the sampling distribution, and compute what we want (e.g., standard errors, confidence intervals, etc.).

Why bootstrap?

We've seen that the sampling distribution of the MLE is asymptotically normal, and we can characterize the mean (it is unbiased) and variance (in terms of Fisher information).

So why do we need the bootstrap?

- Sample sizes might be "small" (asymptopia is not a good assumption).
- Assumptions we made in the population model might have been violated.
- We might want sampling distributions for much more complex estimation procedures, where no closed form expression exists.

Formal definition

The bootstrap algorithm

We are given a sample $\mathbf{Y}$ of n observations.

For $1 \leq b \leq B$:

- Draw n samples uniformly at random, with replacement, from $\mathbf{Y}$. Denote the i'th observation in the b'th sample by $\tilde{Y}_i^{(b)}$.
- Compute the value of the statistic from the b'th sample as $T_b = T(\tilde{\mathbf{Y}}^{(b)})$.

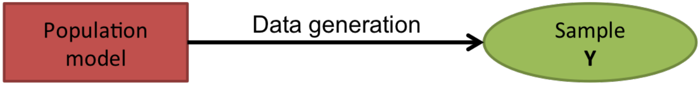
The histogram (i.e., empirical distribution) of $\{T_b, 1 \leq b \leq B\}$ is an estimate of the sampling distribution. We call this the *bootstrap distribution*.

The bootstrap algorithm

A picture:

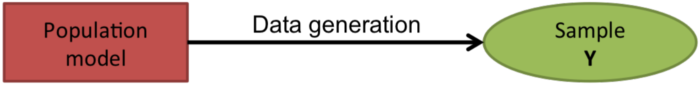
An analogy

The following analogy is helpful to keep in mind. For the sampling distribution we have:

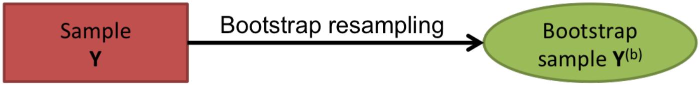


An analogy

The following analogy is helpful to keep in mind. For the sampling distribution we have:



Bootstrapping treats the sample $\mathbf{Y}$ as if it represents the true population model:



When is bootstrap problematic?

From the preceding slides, several things have to happen for the bootstrap distribution to be "close" to the sampling distribution:

- B should be "large enough" that any randomness due to resampling is not consequential in our estimated distribution. (What is large enough? In practice, limited primarily by computational time.)
- The original data sample $\mathbf{Y}$ should be generated as i.i.d. samples from the population model, and $\mathbf{Y}$ should be large enough to be "representative" of the original population model (sufficient sample size, no sampling bias).

The first we can control. The second requires contextual knowledge of how $\mathbf{Y}$ was actually generated.

When is bootstrap problematic?

Bootstrap estimation is a very powerful technique for approximating the sampling distribution, because it makes very few assumptions about the nature of the population model.

That said, it is not immune to yielding inaccurate results; for example, for "extreme value" statistics (e.g., when T is the maximum or minimum of the data), the bootstrap can yield badly biased estimates.

In practice, for estimating things like standard errors of estimators, the bootstrap is a fairly reliable technique (assuming the two conditions on the preceding slide hold).

Standard errors

We use the bootstrap distribution just like we use the sampling distribution.

For example, the bootstrap estimate of the standard error $\widehat{\mathsf{SE}}_{\mathsf{BS}}$ is the standard deviation of the bootstrap distribution.

Confidence intervals

Since we have an estimate of the sampling distribution, we can use it to construct confidence intervals. Two approaches to building 95% confidence intervals:

Confidence intervals

Since we have an estimate of the sampling distribution, we can use it to construct confidence intervals. Two approaches to building 95% confidence intervals:

- *The normal interval*: $[T(\mathbf{Y}) - 1.96\widehat{\mathsf{SE}}_{\mathsf{BS}}, T(\mathbf{Y}) + 1.96\widehat{\mathsf{SE}}_{\mathsf{BS}}]$. This approach assumes that the sampling distribution of $T(\mathbf{Y})$ is normal, and uses the standard error accordingly.

Confidence intervals

Since we have an estimate of the sampling distribution, we can use it to construct confidence intervals. Two approaches to building 95% confidence intervals:

- *The normal interval*: $[T(\mathbf{Y}) - 1.96\widehat{\mathsf{SE}}_{\mathsf{BS}}, T(\mathbf{Y}) + 1.96\widehat{\mathsf{SE}}_{\mathsf{BS}}]$. This approach assumes that the sampling distribution of $T(\mathbf{Y})$ is normal, and uses the standard error accordingly.
- *The percentile interval*: Let T_q be the q'th quantile of the bootstrap distribution. Then the 95th percentile bootstrap interval is: $[T_{0.025}, T_{0.975}]$.

  In general the percentile interval works well when the sampling distribution is symmetric, but not necessarily normal. (Many other types of intervals as well...)

Example 1: Mean of in-class midterm scores

The mean of the score on the 226 in-class midterm was $\hat{\mu} = 33.8$ (with $n = 108$). What is the standard error?

By the central limit theorem, the sample mean has standard error approximately $\hat{\sigma}/\sqrt{n} = 0.60$, where $\hat{\sigma} = 6.25$ is the sample standard deviation.

What does the bootstrap suggest?

Example 1: Mean of in-class midterm scores

Here is how we can directly code the bootstrap in R for the sample mean:

```
n_reps = 10000
n = 79

# Test scores are in "scores"
boot.out = sapply(1:n_reps,
  function(i) {
    mean(sample(scores, n, replace = TRUE))
  })
```

Now `boot.out` contains `n_reps` replicates of the sample mean.

Example 1: Mean of in-class midterm scores

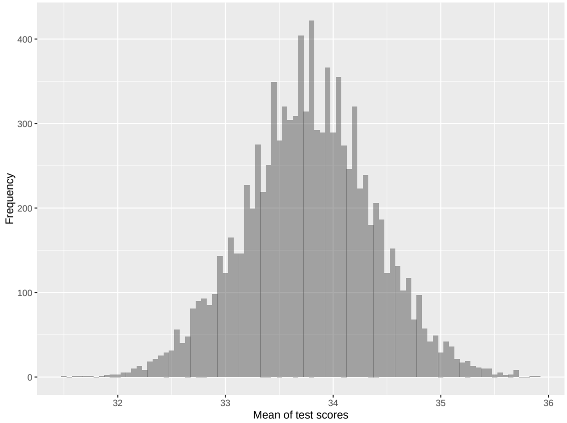
Code to run bootstrap in R (using `boot` library):

```
# create function for mean
mean.boot = function(data, indices) {
  d = data[indices]
  return(mean(d))
}
# Test scores are in "scores"
boot.out = boot(scores, mean.boot, 10000)
```

`boot.out$t` contains the means across parallel universes. Use `boot.ci` for confidence intervals.

Example 1: Mean of in-class midterm scores

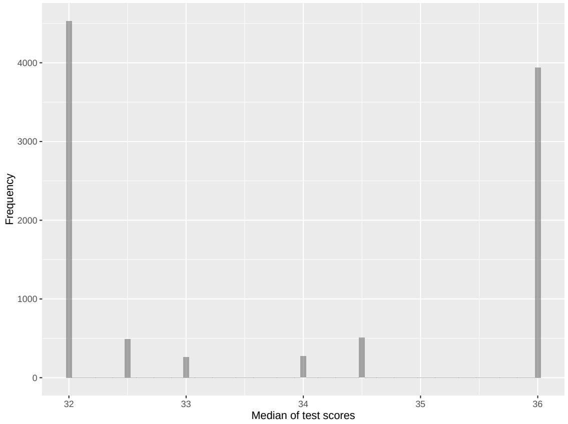
Results:



$\widehat{\text{SE}}_{\text{BS}} = 0.60$, matching the normal approximation (because the central limit theorem is pretty good here).

Example 2: Median of in-class midterm scores

For the median, we don't have an easy way to approximate the standard error; so we turn to the bootstrap. Results:



$\widehat{\mathsf{SE}}_{\mathsf{BS}} = 1.87$. Note that the actual median in the class was 32.5.

Example 3: Funnels (survival analysis)

Suppose as an online retailer you have a three stage checkout flow: customers (1) add an item to their cart, (2) enter their credit card info, and (3) hit "Purchase".

In practice, customers might *abandon* before completing these activities.

Example 3: Funnels (survival analysis)

Suppose as an online retailer you have a three stage checkout flow: customers (1) add an item to their cart, (2) enter their credit card info, and (3) hit "Purchase".

In practice, customers might *abandon* before completing these activities.

Suppose you've collected data on n customers, where $Y_i \in \{1, 2, 3\}$ denotes the latest stage the customer completed.

Example 3: Funnels (survival analysis)

Suppose as an online retailer you have a three stage checkout flow: customers (1) add an item to their cart, (2) enter their credit card info, and (3) hit “Purchase”.

In practice, customers might *abandon* before completing these activities.

Suppose you've collected data on n customers, where $Y_i \in \{1, 2, 3\}$ denotes the latest stage the customer completed.

Let γ_i be the probability that a customer that completes stage s will also complete stage $s + 1$, for $s = 1, 2$. We estimate these as follows:

$$\hat{\gamma}_1 = \frac{|\{i : Y_i \geq 2\}|}{|\{i : Y_i \geq 1\}|}; \; \hat{\gamma}_2 = \frac{|\{i : Y_i = 3\}|}{|\{i : Y_i \geq 2\}|}.$$

Example 3: Funnels (survival analysis)

Suppose as an online retailer you have a three stage checkout flow: customers (1) add an item to their cart, (2) enter their credit card info, and (3) hit "Purchase".

In practice, customers might *abandon* before completing these activities.

Suppose you've collected data on n customers, where $Y_i \in \{1, 2, 3\}$ denotes the latest stage the customer completed.

Let γ_i be the probability that a customer that completes stage s will also complete stage $s + 1$, for $s = 1, 2$. We estimate these as follows:

$$\hat{\gamma}_1 = \frac{|\{i : Y_i \geq 2\}|}{|\{i : Y_i \geq 1\}|}; \; \hat{\gamma}_2 = \frac{|\{i : Y_i = 3\}|}{|\{i : Y_i \geq 2\}|}.$$

But standard errors are not easy to compute, since these are quotients; the bootstrap is an easy approach to get standard errors.

Example 4: Clustered data

This example came up when working on a project with Netflix; it is a common example of computing standard errors correctly when you have *clustered observations*.

It is also a useful example to see how you can sometimes apply the bootstrap even in settings where the data is not perfectly i.i.d.

Example 4: Clustered data

Suppose we collect data on viewing behavior of $n = 100$ users. Each user i has k_i *sessions*; and the average *bit rate* of session j of user i is r_{ij}.

So our data is $\{r_{ij}, 1 \leq i \leq 1000, 1 \leq j \leq k_i\}$.

I generated synthetic data where each user generated 10 sessions; for each i, user i's session had mean rate that is $\mu_i \sim \text{Exp}(1/1000)$; and each of the k_i sessions of user i are of rate $\mathcal{N}(\mu_i, 1)$.

We estimate the average delivered bit rate as:

$$\hat{\mu} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{k_i} r_{ij}}{\sum_{i=1}^{n} k_i}.$$

This estimate is 908.0.

Example 4: Clustered data

What is the standard error of this estimate? Naive approach: take sample standard deviation of the per session bit rates, divided by square root of number of sessions. For my synthetic dataset this gave $\widehat{\mathsf{SE}} = 9.22$.

This is not quite right however, because sessions are not independent: sessions belonging to the same user are likely to have similar bit rate.

How to adjust for this correlation?

Example 4: Clustered data

Recreate the data generating process with a *block bootstrap*:

- Sample *users* ($n = 1000$) with replacement.
- Each time a user is sampled, include *all* of their sessions.
- The bootstrap distribution is the resulting histogram of $\hat{\mu}^{(b)}$, over each bootstrap sample b.

Using this approach gave a standard error of $\widehat{\mathsf{SE}}_{\mathsf{BS}} = 91.6$. (Why is this ten times larger than the naive $\widehat{\mathsf{SE}}$?)

Bootstrap for regression modeling

Resampling the data

As usual for regression, we assume we are given n data points, with observations $\mathbf{Y}$ and covariates $\mathbf{X}$.

There are two ways to bootstrap in this case:

- Resample new bootstrap observations $\tilde{\mathbf{Y}}^{(b)}$, holding $\mathbf{X}$ fixed.
- Resample new data $\mathbf{X}$ and $\mathbf{Y}$.

We'll focus on the second (also called *case resampling*), since it's both more likely to be relevant to you in practice, and it is more robust to modeling errors (though it can be more inefficient).

Linear regression via OLS

Suppose we want bootstrap standard errors for OLS coefficients; we can get these as follows.

For $1 \leq b \leq B$:

- Draw n samples uniformly at random, with replacement, from $(\mathbf{X}, \mathbf{Y})$. Denote the i'th outcome (resp. covariate) in the b'th sample by $\tilde{Y}_i^{(b)}$ (resp., $\tilde{\mathbf{X}}_i^{(b)}$).
- Given the resulting data $(\tilde{\mathbf{X}}^{(b)}, \tilde{\mathbf{Y}}^{(b)})$ in the b'th sample, run OLS and compute the resulting coefficient vector $\hat{\boldsymbol{\beta}}^{(b)}$.

This gives the bootstrap distribution of $\hat{\boldsymbol{\beta}}$, and we can use it to, e.g., compute standard errors or confidence intervals (or even bias) for each of the coefficients.

Example: Heteroskedastic data

Suppose for $1 \leq i \leq 100$, $X_i \sim \mathcal{N}(0, 1)$, and $Y_i = X_i + \varepsilon_i$, where $\varepsilon_i \sim \mathcal{N}(0, X_i^4)$. This data exhibits *heteroskedasticity*: the error variance is not constant.

In this case the linear normal model assumptions are violated. What is the effect on the standard error that R reports?

We use the bootstrap to find out.

Example: Heteroskedastic data

Here is the code to run the bootstrap:

```
df = data.frame(X,Y)
coef.boot = function(data, indices) {
  fm = lm(data = data[indices,], Y ~ 1 + X)
  return(coef(fm))
}
boot.out = boot(df, coef.boot, 50000)
```

Example: Heteroskedastic data

If we run `lm(data = df, Y ~ 1 + X)`, then R reports a standard error of 0.26 on the coefficient on X, which is 1.42.

But using the bootstrap approach, we compute a standard error of 0.62!

Because of the failure of the regression assumption, the bootstrap reports a standard error which is higher than the (optimistic) standard error reported using the assumption of constant variance.

Other applications

Note that you can use the bootstrap in a similar way to also compute standard errors, confidence intervals, etc., for regularized regression (ridge, lasso).

The bootstrap is an incredibly flexible tool, with many variations developed to make it applicable in a wide range of settings; in fact, it can even be used to estimate test error in a prediction setting (instead of cross validation).