Resampling & the Bootstrap

Ethan Ligon

April 20, 2022

The Real World Data-Generating Process

Suppose that at any particular moment in time t, we can describe the *state* of our world by a variable $s_t \in S$, and the history of previous states up to t by by $s^t \in S^t$.

Observed Data

Given a particular history s^t, different economic agents observe (possibly different) sets of reported measurements (which censor, select, and may add error):

$$d_t = \mathcal{R}(s^t)$$

The Real World Data-Generating Process

Decisions

Given a particular history s^t, economic agents take actions $y_t = \mathcal{M}(\mathcal{R}(s^t))$. The realization y_t becomes part of the next period's state.

History's Evolution

The state of the world in the subsequent period depends on a law of motion:

$$s_{t+1} = \mathcal{F}(s^t, y_t)$$

The RWDGP

The RWDGP can thus be described by a triple $(\mathcal{M}, \mathcal{R}, \mathcal{F})$. Initialize with the history to time zero, s^0, and it returns a corresponding dataset.

Interpretation

- So far the RWDGP has produced all the data available to us. This dataset d^t is finite, but depends on the particular history s^t realized up to this point.
- A different history $\tilde{s}^t$ would have produced a different finite dataset.

Our Monte Carlo Data-Generating Process

So: we've discussed the creation of a DGP that can be described as a triple $(\mathcal{M}, \mathcal{R}, \mathcal{F})$. Feed in a initial state s^0 (e.g., a seed to a pseudo-random number generator) and it returns a dataset $d = (y, X, Z)$.

Interpretation

- We have the god-like power of resampling from our DGP; Each draw from our DGP produces a different finite dataset. Call these Monte Carlo draws.
- There's no limit on the number of draws we can make from the dataset. As our draws $m \to \infty$ we may be able to draw increasingly accurate inferences about $(\mathcal{M}, \mathcal{R}, \mathcal{F})$ (this is a question of identification).

Our Monte Carlo Experiment

One particular experiment involved repeated draws to explore the finite sample properties of a linear IV estimator. We found circumstances under which the limiting distribution of b_N was very different from the estimated empirical distribution.

Three different possible takeaways

1. We need a different estimator with better finite sample properties. (Explore this in discussion question).
2. We need more data. Or;
3. We could use the estimated empirical distribution for inference & hypothesis testing. Call this the *Empirical Monte Carlo* process.

After you collect your data

Use the empirical MC distribution, and assume that the MC DGP is close enough to the actual real-world DGP that the empirical distribution of β can serve for testing & inference.

Issue

May requires a lot of confidence in the MC DGP. And if you have this much confidence you may want to use the MC DGP to actually help *estimate* the parameters.

Estimating parameters (Indirect Inference)

Idea: Choose "truth" parameters to make simulated data from the Monte Carlo DGP (in this setting called the 'auxiliary model') match moments or distributions observed in the real-world data. Often used when economic model involves parameters which are complicated functions of the data.

Examples

- Method of Simulated Moments (MSM/SMM)
- Maximum Simulated Likelihood (MSL/SML)
- Monte Carlo Integration

Before you collect data

Standard power calculations usually assume a normal model with very limited forms of dependence. But what if your estimated coefficients aren't normally distributed?

- Typically wind up collecting too little data and being under-powered.
- Use MC distribution instead, where the experiment is actually measuring the finite sample properties of the estimator you'll use when you write your dissertation.
- How big a sample do you really need to achieve a given level of power in your MC experiment?

Issue with Monte Carlo is that we have to construct a model to build estimates. This will often require us to assume more than we wish to about the Real World DGP.

Alternative

Use the RWDGP! We begin by observing a sample of N observations X_j once; say D_N. If these are independent (they're identically distributed by construction) we just need to figure out how to repeat this draw.

Sampling

Since D_N is comprised of N iid observations we can use this sample to construct an empirical distribution function of X, say $\hat{F}$. Then think of simply drawing samples from this empirical distribution.

Non-parametric estimator of empirical distribution function

$$\hat{F}(x) = \frac{1}{N} \sum_j \mathbb{1}(X_j \leq x)$$

Simplification

Since the probability of drawing a particular X from $\hat{F}$ is proportional to the frequency with which X appears in D_N, there's an trivial simplification: instead of constructing $\hat{F}$ just:

1. Draw X_j from D_N.
2. Repeat until you have the sample size you want; often (usually?) this will be N, the size of the original sample. Call the resulting "bootstrap" sample D_N^1.

Basic Bootstrap estimation

Suppose we want to estimate a vector of parameters β. We can construct an estimate of this using the original sample, say b_N. But we may not know much about the distribution of this estimator.

Procedure

1. Choose some positive tolerance ϵ.
2. Having drawn a bootstrap sample D_N^1, use it to estimate b_N^1.
3. Draw a new sample D_N^2, and compute b_N^2
4. ...Repeat 30 times...
5. Calculate the sample covariance matrix of the estimates of β,

$$\hat{V}_N^{30} = \frac{1}{30}\sum_m (b_N^m - \bar{b}_N)(b_N^m - \bar{b}_N)^\top$$

6. Repeat: compute additional bootstrap samples until

$$\|\hat{V}_N^M - \hat{V}_N^{M-1}\| < \epsilon$$

Use

We've just described the construction of a covariance matrix for the estimator b_N via the bootstrap, so this can be used for testing and inference in the usual way. But note that the "usual way" assumes that the distribution of b_N is normal.

Non-normal distributions

In finite samples our distributions may be decidedly *non*-normal. But we have an estimate of the distribution! Just construct the empirical distribution of the M bootstrapped estimates of β.

- Tests of normality available
- Simple construction of confidence intervals

When Sample isn't Simple Random

Or, what's an observation? What *is* selected randomly?

Panel data

We often work with longitudinal *panels* comprising, say, N households observed over T periods.

Stratified samples

Suppose we're interested in the effects of an experimental intervention on both men & women. It may make sense to *stratify* the sample so that we're powered to detect effects for both sexes.

Clustered samples

Surveys of households are often *clustered* geographically, with randomization conducted in two stages: (i) geographical locations (clusters) are randomly selected; then (ii) households who live within a cluster are randomly sampled.

Bootstrapping when a sample isn't simple random

The basic idea is for your bootstrap samples to mimic the randomness used to construct the original sample. So:

Panel data

Resample *households* and their entire histories, not household-periods.

Stratified samples

Think of each strata as it's own random sample, and resample within each strata.

Clustered samples

Resample in two stages: (i) clusters (with replacement); then (ii) households within clusters.

Latent variables

Suppose there are some sets $\{L_i\}$ that an randomly selected observation may belong to (e.g., male and female), and we think membership in these sets is important for determining some outcome.

Then we might have, e.g.,

$$y_j = \sum_i \alpha_i \mathbb{1}(j \in L_i) + \beta^\top X_j + u_j$$

Here α_i is interpreted as something like the mean of y conditional on being in the set L_i.

Suppose the sample is simple random. How should you construct a bootstrap estimator?

Residual Bootstrap

One solution is to hold fixed observables X. Then:

1. Use full dataset to estimate, e.g.,
$$y = X\beta + u,$$
obtaining some estimate $b^{(1)}$ of β.
2. Construct residuals
$$e^{(1)} = y - Xb^{(1)}.$$
3. Now, instead of resampling (y, X) just resample the residuals $e^{(1)}$ obtaining $\tilde{e}^{(1)}$, and construct
$$y^{(1)} = Xb^{(1)} + \tilde{e}^{(1)}$$
4. Re-estimate
$$y^{(1)} = X\beta + \tilde{u},$$
obtaining an estimate $b^{(2)}$.
5. Repeat until convergence.

Wild Bootstrap

The residual bootstrap relies on the disturbances being homoskedastic. But what if $\mathbb{E}(u^2|X)$ is a function of X?

Wild Bootstrap

One idea: generate an auxiliary random variable π_j which takes values $\{-1, 1\}$ with equal probability. Then modify the residual bootstrap algorithm:

1. Use full dataset to estimate, e.g.,
$$y = X\beta + u,$$
obtaining some estimate $b^{(1)}$ of β.
2. Construct residuals
$$e^{(1)} = y - Xb^{(1)}.$$
3. Now, instead of resampling (y, X) or e, hold (X, e) fixed and just draw realizations π_j, $j = 1, \ldots, N$, and construct
$$y_n = X\hat{\beta} + \pi_n e$$
4. Re-estimate
$$y_n = Xb_n + u_n$$
5. Repeat until convergence.