ECONOMETRICS II, FALL 2020

Parametric Bootstrap

Consider, for example, a linear model (although you more often would use the bootstrap for more complicated models):

$$y_i = X_i\alpha + u_i\,,$$

where we assume the OLS assumptions are satisfied and u_i follows a known distribution for $i = 1, ..., N$. Let us assume Normal but with unknown variance σ^2. You would then do OLS and obtain $\hat{\alpha}$ and $\hat{\sigma}^2$.

Now you want to examine the distribution of some test statistic, say, a t-test of one of the parameters being 0. More precisely, you would often look for the critical values for a test; say, a 5% one-sided test (just to cut down on notation, 2-sided would be similar).

Now you set a number of bootstrap replications, say 10,000. Then for $j = 1, ..., 10,000$ you draw a set of random normal variables $e_i^j \sim N(0, \hat{\sigma}^2)$ for $i = 1, ..., N$. Then you construct artificial dependent variables keeping X constant:

$$y_i^j = X_i\hat{\alpha} + e_i^j\,,$$

for $i = 1, ..., N$. Now you have a new dataset and you can run a regression and find $\hat{\alpha}^j$ and $\hat{t}^j$ where the latter refers to the t-statistic for the test you are interested in. (You don't actually need to keep $\hat{\alpha}^j$, although we often take a look at its distribution.)

You do this pulling of N error terms, constructing dependent variables, running the regression, and calculating the t-stat 10,000 times.

Now sort the $\hat{t}^j$ values. These are randomly generated and we get a distribution of those, but if you just care about the critical values, as is often the case, it is approximately the value taken by 95th percentile; i.e., observation number 9,500 in the sorted vector. This is the critical value generated by your parametric Monte Carlo simulation and you can now we test if the original $\hat{\alpha}$ (or its deviation from some non-zero value) is larger than that critical value.

COMMENTS:

1. If you think the data normal and the model linear, you would not do this, because we know the distribution is t-distributed and we can look up the critical value in a table. But if the model is non-linear and you, for example, rely on the Delta-rule (which you almost always do in the non-linear case, if you use a Wald-test), you might want to simulate to see if the first-order Taylor series approximation implicitly involved in using the Delta-rule works well. (When would we think it works well?)

2. The number of Monte Carlo studies that you need can be as low as say 1,000 or as high as 1,000,000 or more...if you have errors from a heavy-tailed distribution (meaning that only rarely will there be outliers, but they may be large), you will need a large number.

3. If you have lagged y's on the right-hand side, then you need to use the time-series model for y. For example, if y follows an AR(k), you do

this recursively, using the error terms that you draw in each iteration.

4. You can do this for *any* non-linear or whatever model. But you need to assume a distribution of the innovations.

Non-Parametric Bootstrap

Here you estimate the model as before. Now you may worry that the errors are not Normal, or more generally you just don't know the distribution of the innovation, so you then do not know the finite sample distribution of the t-statistic (maybe we should then say "t-statistic" in quotes, as it is not t-distributed). You could make up your own test-statistic, but usually we use the standard t- and Wald (ξ-square) test statistics because we know that they have good properties for Normal innovations.

You do the simulations as before. You do not draw from a Normal distribution but from the empirical distribution of the error terms from the original estimation, $\hat{u}_i$. You do this by selecting $e_i^j = \hat{u}_k$, where for each i (in each iteration j) you chose k randomly, with equal probability, from $1, ..., N$. Often people draw some Z from a uniform distribution on $[0, N]$ and set $k = l$ if $l - 1 < Z \leq l$ etc. All other steps are similar to the parametric case.

Wild Bootstrap

You follow the previous methods but select $e_i^j = +/ - \hat{u}_i$ or more precisely

$e_i^j = \xi_i \hat{u}_i$, where $\xi_i = 1$ or $\xi_i = -1$, each with probability 0.5. Notice, that if the errors are correlated with x, the heteroskedasticity will be inherited by the u-draws, but maybe not by the "$+/-u$" draws. An improvement can often be achieved by instead have the variable ξ take the values $\frac{1+\sqrt{5}}{2}$ or $\frac{1-\sqrt{5}}{2}$ with probabilities $\frac{\sqrt{5}-1}{2\sqrt{5}}$ and $\frac{\sqrt{5}+1}{2\sqrt{5}}$, respectively. I have not yet worked out the intuition for this.

Block Bootstrap

If the error terms are correlated, the bootstrap draws have to retain the correlation structure. Think of the error terms e_t following an AR or MA model. The simplest case is that of a panel with large N. Run the regression $y_{it} = X_{it}\alpha + u_{it}$ and obtain residuals $\hat{u}_{it}$. Now you draw the T-vector $\{e_{i1}, ..., e_{iT}\} = \{\hat{u}_{k1}, ..., \hat{u}_{kT}\}$, where for each i (in each iteration j) you chose k randomly, with equal probability, from $1, ..., N$ and proceed as before.

If you have a time series think the error terms follow an MA(p) process, you will for each t draw $\{e_t, ..., e_{t+p}\} = \{\hat{u}_k, ..., \hat{u}_{k+p}\}$, where for each t (in each iteration j) you chose k randomly, with equal probability, from $1, ..., T-p.$; i.e., a block (overlapping) so that the variables in each block are of low correlation on average. If the error terms are not following an MA(p) process, you use the MA process as an approximation and do the same. For such block bootstraps there are many permutations about how long the blocks should be (that is, how large a p) and I am not an expert on this. If you were to use this, I strongly suggest first trying to figure out the type of time series model that captures the residuals and then doing a parametric Monte Carlo study for this model to see how things work for your model.