Notes on Bootstrapping in Linear Models with i.i.d. Disturbances

The bootstrap is a method of re-sampling. It is very easy to use, and can be a very useful way of conducting small sample inference. (That is, when the sample is not sufficiently large to rely on the asymptotic distribution theory).

We will discuss how to use the bootstap to construct empirical distribution functions for estimated parameters.

First, consider the classical linear model:

$$y_t = x_t\beta + \varepsilon_t$$

Recall the standard assumptions here: the x's are fixed, and are of dimension T x N, and the disturbance term is an i.i.d. process.

Suppose we estimate this model using OLS. We get:

$$\hat{\beta} = (x_t'x_t)^{-1}(x_t'y_t)$$

The empirical counterpart to the disturbance term is:

$$\hat{\varepsilon}_t = y_t - x_t\hat{\beta}$$

Now, let's use the bootstrap to construct the empirical distribution of the OLS estimator, $\hat{\beta}$.

We first use the *parameteric bootstrap.*

To do this, we make some parametric assumptions about the process that ε is drawn from. What type of assumptions should we make? First, we know that the process is mean zero by assumption. For the second moment, we can use the sample variance estimate. What about the type of distribution? Since the normality assumption is so common, one can first test whether the empirical residuals are normal. There are a number of tests available for testing for normality, and many econometric routines provide these tests. Typically, these tests evaluate whether there is excess skewness and kurtosis in the series. Let's assume for the time being that the empirical residuals pass the normality test. This suggests we can use the normal distribution function.

(1) Generate a sequence of length "T" random, normal, variables that are mean zero, and have the same variance as that of the empirical residuals. Call this series e^1.(In matlab, this is done using the "randn" function).

(2) Construct y^1 as:

$$y_t^1 \equiv x_t \hat{\beta} + e_t^1$$

(3) Re-estimate $\hat{\beta}$, and call this $\hat{\beta}^1$:

$$\hat{\beta}^1 = (x_t' x_t)^{-1} (x_t' y_t^1)$$

(4) Repeat steps 1-3 "N-1" times, and index the estimated β by i

Now, we have "N" estimates of β : $\{\hat{\beta}^i\}_{i=1}^n$. We can use the distribution of the β to construct confidence intervals, form empirical densities, calculate variances and covariances, etc.

Now, let's use the *non-parametric* bootstrap. The only difference is how we generate new series of residuals that we use to re-estimate the parameters.

(1) Take the empirical residuals $\{\hat{\varepsilon}_t\}_{t=1}^T$. Now, let's re-order these residuals randomly. To do this, use a uniform random number generator (in matlab, this is the "rand" function). Generate a length "T" series of uniform random numbers between 0 and 1. Suppose T is 5, and the random numbers generated are .6, .4, .2, .5, .1. Since .6 is the largest number, the first empirical residual would become the last one. Since .4 is the third largest number, the second empirical residual would become the third one, etc.

(2) Call this re-shuffled set of residuals $\{\hat{\varepsilon}_t^1\}_{t=1}^T$. Now, Construct y^1 as:

$$y_t^1 \equiv x_t \hat{\beta} + \hat{\varepsilon}_t^1$$

(3) Re-estimate $\hat{\beta}$, and call this $\hat{\beta}^1$:

$$\hat{\beta}^1 = (x_t' x_t)^{-1} (x_t' y_t^1)$$

(4) Repeat steps 1-3 ”N-1” times, and index the estimated β by i

Now, we have ”N” estimates of β : $\{\hat{\beta}^i\}_{i=1}^n$. We can use the distribution of the β to construct confidence intervals, form empirical densities, calculate variances and covariances, etc.