Bootstrapping unit root tests

Stephan Smeekes*

October 16, 2006

Abstract

This paper presents an overview of the application of the bootstrap to unit root testing. We show how a bootstrap unit root test can be set up and discuss several options that have been proposed in the literature. The effects of these options on the performance of the test are analysed.

1 Introduction

In recent years the bootstrap, a statistical method that estimates the distribution of an estimator or test statistic by resampling the data, has become increasingly popular and it has been applied to a wide range of topics, including nonstationary time series. This paper will consider one specific application of the bootstrap involving nonstationary time series, namely unit root testing.

One reason to use the bootstrap instead of asymptotic inference is that often if a statistic is *asymptotically pivotal*, that is if the limiting distribution is free of nuisance parameters, the bootstrap offers *asymptotic refinements*, meaning that the gap between the true distribution of a statistic and the bootstrap distribution declines faster as sample size increases than the gap between the true distribution and the asymptotic distribution. For hypothesis tests this means that the finite sample size of a bootstrap test will be closer to the nominal level than that of an asymptotic test. This is a very attractive property for unit root tests, as unit root tests often have serious size problems.

Another reason is that even if a statistic is not asymptotically pivotal (for example a non-augmented Dickey-Fuller test statistic), the bootstrap can still be used for inference without problems, whereas asymptotic analysis often becomes much more difficult.

A drawback when applying the bootstrap to unit root testing is that it is not automatically valid when certain conditions are fulfilled, as it is when applied to random samples and

---

*The author would like to thank Jeroen van den Berg, Franz Palm and Jean-Pierre Urbain for helpful comments and suggestions.

stationary time series. Therefore validity of a bootstrap test has to be proven separately for every specific application.

In this paper we present an overview of how a bootstrap unit root test can be constructed, in which we discuss several options that been proposed in the literature. Afterwards we analyse what effects these options have on the (finite sample) performance of the tests.

2 A bootstrap unit root test

2.1 The model

Let us consider unit root tests in a univariate time series model. We first define our data generating process. Let y_t be given by

$$y_t = \rho y_{t-1} + u_t, \qquad \text{for } t = 1, \ldots, n.$$

There is a unit root if $\rho = 1$. Different assumptions for the error process u_t are being used. The most simple assumption is that u_t is an i.i.d. process, which is usually not very realistic and interesting. Instead one would prefer to allow for some dependence in the errors. The most common way in the literature to do this is to assume that u_t is a stationary and invertible MA(∞) process, i.e. that $u_t = \Psi(L)\varepsilon_t$, where ε_t is i.i.d., $\Psi(z) = \sum_{j=0}^{\infty} \psi_j z^j$ has roots outside the unit circle and $\sum_{j=0}^{\infty} j|\psi_j| < \infty$. An alternative assumption that is used is that u_t is strong mixing. This is a weaker assumption than the one above, as it does not require linearity.

Furthermore, let T_n be the test statistic of interest. In this paper we will mainly focus on the (augmented) Dickey-Fuller ((A)DF) coefficient and t-tests, but it can be any test statistic used for unit root testing.

2.2 A bootstrap algorithm

A bootstrap unit root test consists of several parts. In order to be able to analyse these separately, we will first describe how a general bootstrap unit root test can be constructed.

Step 1: Construct a stationary series.

As there are no bootstrap methods available that are directly applicable to nonstationary series, one has to create a stationary series first, call this $\hat{u}_t$. Usually this is done by taking first differences of y_t,

$$\hat{u}_t = y_t - y_{t-1},$$

or by taking residuals of a regression of y_t on y_{t-1},

$$\hat{u}_t = y_t - \hat{\rho} y_{t-1}.$$

The $\hat{u}_t$ should always be recentred at this stage (call the recentred residuals $\tilde{u}_t$), as not doing so could induce unwanted drifts in the bootstrap sample.

STEP 2: BOOTSTRAP $\tilde{u}_t$ TO OBTAIN BOOTSTRAP ERRORS u_t^*.
In this step the actual bootstrapping takes place. How to do this depends on the assumptions made about u_t. If u_t is i.i.d., we can simply use the *i.i.d. bootstrap*, which draws n times with replacement from $(\tilde{u}_1, \ldots, \tilde{u}_n)'$ to form the bootstrap errors $u_1^*, \ldots, u_n^*$. However, if the errors are not independent, the i.i.d. bootstrap is not valid anymore. Instead, there are two main alternatives that are valid for time series.

The first option, the *block bootstrap*, is valid under very general assumptions, mixing assumptions are sufficient. The block bootstrap divides the data in blocks and resamples these blocks with replacement. The idea behind it is that inside a block the dependence structure of the original series is preserved. The most common form of the block bootstrap, the moving-blocks bootstrap (Künsch, 1989), constructs blocks of length k starting at very observation, such that block 1 is $(\tilde{u}_1, \ldots, \tilde{u}_k)'$, block 2 is $(\tilde{u}_2, \ldots, \tilde{u}_{k+1})'$, etc. To obtain the bootstrap series, the blocks are resampled with replacement and the blocks are laid end-to-end.

A practical issue with the block bootstrap is the selection of the block length. The blocks should not be too short, in which case not enough of the dependence structure would be preserved, and not too long, in order to have enough variability in the blocks. Although there exist some methods to select the block length, these methods are not applicable in the unit root setting.

A second issue with the block bootstrap is that the generated sample is not stationary anymore, as there is a kink at the endpoints of the blocks. A variant of the block bootstrap that deals with this issue is the *stationary bootstrap* (Politis and Romano, 1994), which makes the block lengths random (they follow a geometric distribution). This method does produce a stationary bootstrap sample.

The second option, the *sieve bootstrap* (Bühlmann, 1997), is only valid for linear processes. It attempts to “filter” out the dependence of a linear process using a finite-order AR process and then resamples on the basis of the filtered series. Specifically, an AR(p) process is fitted to the series (the estimated $\tilde{u}_t$ in the unit root case), and the residuals of this regression,

$$\hat{\varepsilon}_t = \tilde{u}_t - \sum_{j=1}^{p} \hat{\phi}_j \tilde{u}_{t-j},$$

are recentred and then resampled with replacement to obtain the bootstrap versions ε_t^*. The bootstrap errors u_t^* are then constructed by the recursion

$$u_t^* = \sum_{j=1}^{p} \hat{\phi}_j u_{t-j}^* + \varepsilon_t^*.$$

The role of the lag length here is similar to the role of the block length; p should not be chosen too small, as too much dependence would be left in $\hat{\varepsilon}_t$ to resample it, and not too large, as this would affect the accuracy of the estimates. In contrast with block length selection, lag length selection is not a problem in practice, as there exist well-known and simple methods such as information criteria that can be used.

There exist many other bootstrap methods, as well as many extensions to the methods listed above, but these methods are the ones used most often in the literature.

Step 3: Rebuild the unit root series y_t^*.
Once bootstrap errors u_t^* have been obtained, the bootstrap series y_t^* must be obtained. This is done by the recursion

$$y_t^* = y_{t-1}^* + u_t^*.$$

It is crucial that the null hypothesis of a unit root is imposed. If the recursion $y_t^* = \hat{\rho} y_{t-1}^* + u_t^*$ is used, the limiting distribution of the bootstrap test statistic is random, as shown by Basawa et al. (1991).

Step 4: Calculate the bootstrap test statistic T_n^*.
From the bootstrap sample, the bootstrap test statistic T_n^* can now be calculated. Obviously the bootstrap test statistic should be the same statistic as the original test statistic T_n. As said before, we will focus on DF and ADF coefficient and t-statistics.

Step 5: Repeat the algorithm to find the bootstrap critical value.
The steps above should be repeated B times to obtain the statistics $T_{n,1}^*, \dots, T_{n,B}^*$ which will form the bootstrap distribution. The bootstrap critical value is then determined as the α-quantile (where α is the nominal level) of the bootstrap distribution.

2.3 Currently available tests

Several tests have been proposed in the literature. Here we give a short overview of the ones that fit into the framework described above. Psaradakis (2001) proposes a sieve bootstrap test that is based on differences. The test statistics he considers are the DF coefficient and t-test. The tests proposed by Chang and Park (2003) are very similar, except they consider the ADF test. Paparoditis and Politis (2003) propose a block bootstrap test based on residuals, for both the DF and ADF coefficient test. Swensen (2003a) considers two DF (coefficient and t) bootstrap tests based on differences; one uses the sieve bootstrap, and one the stationary bootstrap. Parker, Paparoditis, and Politis (2006) propose a stationary DF test based on residuals. Paparoditis and Politis (2005) and Palm, Smeekes, and Urbain (2006) propose residual-based sieve bootstrap tests[1] for both DF and ADF statistics. The latter paper also compares the finite sample performance of all the tests mentioned here by simulation.

3 Evaluation

3.1 Validity of the tests

We call a bootstrap procedure asymptotically valid if the bootstrap distribution of a statistic asymptotically correctly mimics the limiting distribution of the statistic. This is of course the basic condition any bootstrap method has to fulfill. For a bootstrap test, one of the implications is that the asymptotic size should be correct. All tests mentioned in the previous section have been shown to be valid. The conditions and assumptions employed differ however, as we have seen above.

Another difference is the use of deterministic components. In the algorithm above deterministic components were left out for simplicity. However, when conducting unit root tests in practice, one will often want to include an intercept and possibly a linear time trend in the regression. Psaradakis (2001), Paparoditis and Politis (2003) and Parker et al. (2006) explicitly discuss deterministic components and show the validity of their tests for these situations.

3.2 The test statistic of interest

Asymptotic refinements in the stationary world occur if the statistic is asymptotically pivotal. Of the tests we consider, only Chang and Park (2003) and the ADF test proposed in Palm et al. (2006) provide asymptotically pivotal statistics. The ADF coefficient test of Paparoditis and Politis (2003) is asymptotically pivotal only for finite AR models. We might expect these tests to have better size properties.

There are however very few theoretical results about asymptotic refinements for bootstrap unit root tests; only Park (2003) shows that for simple cases[2], which are not relevant here, bootstrap tests offer asymptotic refinements.

Nevertheless, the simulations reported in Palm et al. (2006) indicate that the assertions about asymptotic refinements are valid for unit root testing as well, as the ADF tests considered in the paper all have considerably smaller size distortions than the DF tests.

---

[1] In such a test, Step 1 and Step 2 are combined: residuals are directly resampled from an ADF regression.

[2] Park (2003) works with the assumption that u_t is a finite AR process, with order known in advance.

3.3 The time series bootstrap method

Based on the research done for stationary time series, one can expect the sieve bootstrap to have better size properties than the block bootstrap. This indeed shows in the simulations of Palm et al. (2006), who have an ARMA(1,1) process as DGP for u_t. The empirical size of the block bootstrap tests is seen to depend heavily on the values of the AR parameter and the MA parameter, whereas this sensitivity is much less for the sieve bootstrap tests. When the block bootstrap is combined with an ADF test statistic, the sensitivity is quite reduced.

For a DGP where u_t does not belong to the class of linear processes, the sieve bootstrap would not be valid and the block bootstrap should hence perform better. This has not been asserted yet in the literature on bootstrap unit root tests.

3.4 Residual- or difference-based tests

Power issues are related to the choice in Step 1. Intuitively, one might argue that while first differences should perform better when the null hypothesis is true (if there is indeed a unit root, imposing it leads to the best result), residuals might perform better under the alternative, as imposing a unit root would then be false.

Swensen (2003b) shows that power functions are the same for both cases if the errors u_t are i.i.d. However, Paparoditis and Politis (2003, 2005) show that difference-based tests have poorer power properties than residual-based tests, which holds both for sieve and for block bootstrap tests.

In the simulations conducted by Palm et al. (2006), a difference in power is indeed found. However, this difference in power is often accompanied by a difference in size. In general, the difference-based tests exhibit less size distortions than the residual-based tests, so a large part of the higher power of the block bootstrap test is in fact due to its size. The exception to this is the residual-based sieve ADF test, which has size and power comparable to its difference-based counterpart.

4 Conclusion

From the evaluation in the previous section we can draw several conclusions. The most important result is that an ADF test should always be preferred to a DF test in the setting we have been looking at. Furthermore, the sieve bootstrap performs better than the block bootstrap, and differences perform somewhat better than residuals, although not in all cases. Hence for practical purposes, if one believes the errors u_t can be represented as a linear process, the best options are the ADF sieve bootstrap tests by Chang and Park (2003) and Palm et al. (2006).

Note however that this is far from a complete overview. We have mainly looked at tests and DGPs that fit into the Dickey-Fuller framework. Problems that are faced in practice, such as heteroscedasticity and structural breaks, are very interesting to consider as well for bootstrap testing. Another major lack is, as said before, the lack of results about the asymptotic refinements that may be present when testing for unit roots using different bootstrap methods.

5 About the author

Stephan Smeekes is a PhD student at the Department of Quantitative Economics of the Universiteit Maastricht.

Email: `s.smeekes@ke.unimaas.nl`

Home page: `www.personeel.unimaas.nl/s.smeekes`

References

Basawa, I. V., A. K. Mallik, W. P. McCormick, J. H. Reeves, and R. L. Taylor (1991). Bootstrapping unstable first-order autoregressive processes. *Annals of Statistics 19*, 1098–1101.

Bühlmann, P. (1997). Sieve bootstrap for time series. *Bernoulli 3*, 123–148.

Chang, Y. and J. Y. Park (2003). A sieve bootstrap for the test of a unit root. *Journal of Time Series Analysis 24*, 379–400.

Künsch, H. R. (1989). The jackknife and the bootstrap for general stationary observations. *Annals of Statistics 17*, 1217–1241.

Palm, F. C., S. Smeekes, and J.-P. Urbain (2006). Bootstrap unit root tests: comparison and extensions. METEOR Research Memorandum 06/015, Universiteit Maastricht.

Paparoditis, E. and D. N. Politis (2003). Residual-based block bootstrap for unit root testing. *Econometrica 71*, 813–855.

Paparoditis, E. and D. N. Politis (2005). Bootstrapping unit root tests for autoregressive time series. *Journal of the American Statistical Association 100*, 545–553.

Park, J. Y. (2003). Bootstrap unit root tests. *Econometrica 71*, 1845–1895.

Parker, C., E. Paparoditis, and D. N. Politis (2006). Unit root testing via the stationary bootstrap. *Journal of Econometrics 133*, 601–638.

Politis, D. N. and J. P. Romano (1994). The stationary bootstrap. *Journal of the American Statistical Association 89*, 1303–1313.

Psaradakis, Z. (2001). Bootstrap tests for an autoregressive unit root in the presence of weakly dependent errors. *Journal of Time Series Analysis 22*, 577–594.

Swensen, A. R. (2003a). Bootstrapping unit root tests for integrated processes. *Journal of Time Series Analysis 24*, 99–126.

Swensen, A. R. (2003b). A note on the power of bootstrap unit root tests. *Econometric Theory 19*, 32–48.