Bootstrap Estimating the Long Memory Parameter of Long Memory Time Series

Yuhong Xing*
School of Mathematics and Statistics, Qinghai Normal University, Xining, Qinghai, China.
Email:296613570@qq.com
*Corresponding Author Email: maths@qhnu.edu.cn

Abstract. This paper evaluates the finite sample performances of three bootstrap methods, sieve AR bootstrap (SARB), fractional differencing sieve bootstrap (FDSB) and fractional differencing block bootstrap (FDBB), in estimating the long memory parameter in long memory time series. Extensive simulations show that the FDSB method outperforms others when estimating long memory parameter, and has stable estimated results in most cases.

1. Introduction
Since Efron [6] provide the bootstrap, the method is widely used in various fields of statistics. Bootstrapping techniques can be quite useful as it overcomes the limitations of insufficient data size or unknown theoretical distribution. Most studies about these bootstrap methods such as sieve bootstrap and block bootstrap are concentrating on short memory time series. This leads long memory time series also is an important and well used model in practice. No doubt, studying the performance of sieve bootstrap and block bootstrap in long memory time series is an important issue.

Sieve Bootstrap has been first provided by Buhlmann [3]. The idea of sieve bootstrap is resampling from residuals of a fitted model as opposed to resampling the original data itself. In spite of there are such research of bootstrap for long memory processes, there is less study applying the sieve bootstrap to long memory processes. Hidalgo [7] presented a bootstrap method based on resampling in the frequency and Lazarova [10] indicated it can valid apply to stationary long memory process. The research provided by Andrews et. al. [2] showed that the bootstrap could effectively approximate the probability distribution of covariance parameter estimation for stationary long memory processes. Poskitt [14] pointed out that the SARB could present a superior result of the stationary long memory processes on theoretical analysis. Davision and Rambaccussing [5] provided a FDSB technique be suitable for analysis of non-stationary long memory time series.

In term of block bootstrap, Carlstein [4] first introduced this concept. The block bootstrap method, where the idea is to preserve the dependency structural in the data by resampling from continuous blocks of data as opposed to selecting individual points. Kunsch [9] introduced a moving-block bootstrap, which is an extension of previous research by Efon [6]. Papailias and Kapetanios [12] introduced a FDBB algorithm for long memory time series, and showed that it can provide better bootstrap resamples than the SARB used in Poskitt [14]. However, the SARB [14] does not depend on the long memory parameter estimator while the FDBB needs. Obviously, in this point, compare these two methods is not fair. A parallel sieve bootstrap which needs estimating long memory parameter is FDSB [5]. Does this sieve bootstrap still has worse performance than the FDBB is an interesting topic.

This paper aims to compare the SARB [14], the FDSB [5, 8], and the FDBB [12] under long memory time series. We compare the distribution approximation performance of these three methods.
by estimating the bootstrap sample the long memory parameter. In changing point test problem, the theory of sequence distribution is usually unknown makes the test statistics of asymptotic critical value is not easy to get. By the Bootstrap method to approximate the critical value of test statistics makes the test process is simple. In this paper, numerical simulation was used to compare the approximate effects of three Bootstrap methods under different long-memory parameter assumptions. Three Bootstrap methods were used to approximate the long memory parameter of sequences and compared with the long memory parameters of the original data. The accuracy of these methods in approximating the long memory time series was determined by the degree of difference.

The rest of the paper is organized as follows: Section 2 introduces the research model. In section 3, we illustrate the three bootstrap methods. Extensive simulation experiments will be given in Section 4. Section 5 concludes the paper.

2. Model
We consider the following long memory process

\[ Y_t = \mu + X_t, \quad \Phi(L)(1 - L)^d X_t = \Psi(L)\epsilon_t, \quad t = 1,2,\ldots,n. \]  

where \( \mu = \text{E}(Y_t) \) is a deterministic component, \( n \) is the sample size. Random component \( X_t \) follows an ARFIMA \( (p, d, q) \) process, in which \( \epsilon_t \) are i.i.d random variables with mean zero and variance \( \sigma^2 \), and \( L \) is the lag operator. The AR- and MA- polynomials \( \Phi(L) \) and \( \Psi(L) \) are assumed to have all roots outside the unit circle. The long memory parameter \( d \) is restricted to \( 0 \leq d \leq 1 \). Note that the process \( Y_t \) is a stationary long memory process if \( 0 \leq d < 1/2 \), and the process \( Y_t \) is a non-stationary long memory process if \( 1/2 < d \leq 1 \). The fractional differencing operator defined by the binomial expansion can be decomposed as \( (1 - L)^d = \sum_{j=0}^{\infty} \frac{\Gamma(d+j)}{\Gamma(d)\Gamma(1+j)} \), where \( \Gamma(\cdot) \) denotes the gamma function. We are interested in checking how well the available three bootstrap methods in the next section when estimating mean \( \mu \) of model (1).

3. Bootstrap Methods
Suppose samples \( y_1, \ldots, y_n \) are realization from model (1). In this section, we show three popular bootstrap methods to approximate the probability distribution or the critical value of statistic \( T_y \) based on the samples \( y_1, \ldots, y_n \).

The SARB algorithm is as follows:

Step1. Fit an autoregressive model AR(\( p \)) based on the OLS residuals \( \hat{\epsilon} = y - \hat{\mu} \), i.e., \( \hat{x}_t = \beta_1 \hat{x}_{t-1} + \beta_2 \hat{x}_{t-2} + \ldots + \beta_p \hat{x}_{t-p}(n) \) with fixed \( p(n) = 10 \log_{10}(n) \), and then choose an optimal \( p \) using the AIC or BIC criterion. Then, we obtain the estimated coefficients \( \hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_p \) and residuals \( \hat{\epsilon}_{p+1, \ldots, \hat{\epsilon}_m} \) via Yule-Walker equation or other estimation.

Step2. Generate SARB observations \( y_t^* \) according to \( y_t^* = \hat{\mu} + x_t^* \hat{x}_t = \hat{\beta}_1 \hat{x}_{t-1}^* + \ldots + \hat{\beta}_p \hat{x}_{t-p}^* + \hat{\epsilon}_t^* \), \( t = 1,2,\ldots,n \), where \( \epsilon_t^* \) are randomly selected with replacement from the residuals \( \hat{\epsilon}_{p+1, \ldots, \hat{\epsilon}_m} \), and the starting \( p \) observations \( x_{p-1}^*, \ldots, x_{0}^* \) can be set equal to \( \bar{x} = \frac{1}{n} \sum_{t=1}^{n} x_t \).

Step3. Calculate the statistic \( T_{y^*} \), which has same definition as \( T_y \), based on the SARB samples \( y_1^*, \ldots, y_n^* \).

Step4. Repeat Step 2 to Step 3 \( B \) times, approximate the probability distribution function of \( T_y \) and it’s critical values by the empirical distribution function and empirical quantiles of \( T_{y^*} \).

The FDSB algorithm is as follows:

Step1. Having observed the samples \( y_1, \ldots, y_n \), we start computing the OLS residuals \( \hat{x}_t = y_t - \hat{\mu}, \) \( t = 1,2,\ldots,n \), and then estimating long memory parameter \( d \) based \( \hat{x}_t = y_t - \hat{\mu}, \) \( t = 1,2,\ldots,n \), through some estimation. The estimator writes as \( \hat{d} \).

Step2. Fit an autoregressive model AR(\( p \)) based on the \( \hat{d} \)-order fractional differencing data \( \hat{\epsilon}_t = (1 - B)^{\hat{d}} \hat{x}_t \), i.e. \( \hat{\epsilon}_t = \beta_1 \hat{x}_{t-1} + \beta_2 \hat{x}_{t-2} + \ldots + \beta_p \hat{x}_{t-p}(n) \) with fixed \( p(n) = 10 \log_{10} n \), and
then chose the optimal \( p \) using the AIC or BIC criterion. Then, we obtain the estimated coefficients \( \hat{\beta}_1, \ldots, \hat{\beta}_p \) and residuals \( \hat{\epsilon}_{p+1}, \ldots, \hat{\epsilon}_n \) via Yule-Walker equation or other estimation.

\[
\text{Step 3.} \quad \text{Generate the bootstrap samples } \hat{y}_t^* \text{ according to } \hat{y}_t^* = \hat{\mu} + \hat{\epsilon}_t^*, \quad (1 - B)^d \hat{\epsilon}_t^* = \hat{\epsilon}_t^* - \sum_{j=1}^{p} \hat{\beta}_j \hat{\epsilon}_{t-j}^* + \hat{\epsilon}_t, \quad t = 1, 2, \ldots, T, \text{ where } \hat{\epsilon}_t^* \text{ are randomly selected with replacement from the residuals } \hat{\epsilon}_{p+1}, \ldots, \hat{\epsilon}_n, \text{ and starting } p \text{ observations } \hat{\epsilon}_1^*, \ldots, \hat{\epsilon}_0^* \text{ can be set equal to } \frac{1}{n-p} \sum_{i=p+1}^{n} \hat{\epsilon}_i.
\]

\[
\text{Step 4.} \quad \text{Calculate the statistic } T_{Y^*}, \text{ which has same definition as } T_Y, \text{ based on the FDSB samples } y_1^*, \ldots, y_T^* \text{ write as } T_{y^*}.
\]

\[
\text{Step 5.} \quad \text{Repeat Step 3 to Step 4 } B \text{ times, approximate the probability distribution function of } T_Y \text{ and it’s critical values by the empirical distribution function and empirical quantiles of } T_{Y_1^*}, \ldots, T_{Y_B^*}.
\]

The FDBB algorithm is as follows:

\[
\text{Step 1.} \quad \text{Having observed the samples } y_1, \ldots, y_n, \text{ we start computing the OLS residuals } \hat{x}_t = y_t - \hat{\mu}, \quad t = 1, 2, \ldots, n, \text{ and then estimating long memory parameter } d \text{ based on } \hat{x}_t, \quad t = 1, 2, \ldots, n, \text{ through some estimation. The estimator writes as } \hat{d}.
\]

\[
\text{Step 2.} \quad \text{Dividing the } \hat{d}\text{-order fractional differencing data } \hat{\epsilon}_t = (1 - B)^d \hat{x}_t \text{ into } M = \left[ \frac{n}{b} \right] \text{ disjoint blocks according to a given block size } b, \text{ we write the } k\text{-th block as } \hat{\epsilon}_k = \{\hat{\epsilon}_{(k-1)b+1}, \ldots, \hat{\epsilon}_{kb}\}.
\]

\[
\text{Step 3.} \quad \text{We randomly choose } M \text{ blocks by sampling with replacement from } \{\hat{\epsilon}_k, k = 1, \ldots, M\}, \text{ and obtain block bootstrap sample } \{\hat{\epsilon}_k^*, k = 1, \ldots, M\}. \text{ Then, we generate FDBB samples according to } \hat{y}_t^* = \hat{\mu} + \hat{\epsilon}_t^*, \quad (1 - B)^d \hat{\epsilon}_t^* = \hat{\epsilon}_t^*.
\]

\[
\text{Step 4.} \quad \text{Calculate the statistic } T_{Y^*}, \text{ which has same definition as } T_Y, \text{ based on the FDBB samples } y_1^*, \ldots, y_T^* \text{ write as } T_{y^*}.
\]

\[
\text{Step 5.} \quad \text{Repeat Step 3 to Step 4 } B \text{ times, approximate the probability distribution function of } T_Y \text{ and it’s critical values by the empirical distribution function and empirical quantiles of } T_{Y_1^*}, \ldots, T_{Y_B^*}.
\]

4. Simulation Studies

In this section, we perform simulation for a comparison of SARB, FDSB and FDBB methods via two sets of experiments. We compare the distribution approximation performance by estimating the long memory parameter. We use ARFIMA(p,d,q) model (1) to generate data by set \( \mu = 1 \), and consider AR coefficients \( \varphi_i \) and MA coefficients \( \theta_j \) set to be: \( (\varphi_1, \theta_1) = (0,0), (\varphi_1, \theta_1) = (0.5,0), (\varphi_1, \theta_1) = (-0.3,0.4), \) and \( (\varphi_1, \varphi_2, \theta_1) = (0.2,0.5,0) \), and the error term is standard normal in all cases. The sample size is varying among \{100,400\}. All simulations are obtained via 1000 repeats. We evaluate these methods when approximating long memory parameter \( d \) although the extended local whittle (ELW) estimator proposed by Abadir [1], and the number of bootstrap repeat B=199. The long memory parameter \( d \) varying among \{0,0.4,1\}.

| n   | d = 0   | d = 0.4 | d = 1   | d = 0   | d = 0.4 | d = 1   |
|-----|---------|---------|---------|---------|---------|---------|
|     | Mean    | St. D.  | Mean    | St. D.  | Mean    | St. D.  | Mean    | St. D.  | Mean    | St. D.  | Mean    | St. D.  |
| 100 | SM      | -0.016  | 0.411   | 0.143   | 0.969   | 0.152   | 0.571   | 0.172   | 0.981   | 0.17    | 1.458   | 0.08    |
|     | SAR     | -0.019  | 0.158   | 0.224   | 0.198   | 0.807   | 0.156   | 0.277   | 0.136   | 0.689   | 0.139   | 1.275   | 0.131   |
|     | FDSB    | -0.003  | 0.192   | 0.417   | 0.187   | 0.969   | 0.198   | 0.38    | 0.138   | 0.752   | 0.131   | 1.334   | 0.097   |
|     | FDBB    | -0.052  | 0.201   | 0.395   | 0.205   | 0.92    | 0.208   | 0.563   | 0.159   | 0.951   | 0.162   | 1.426   | 0.085   |
| 400 | SM      | -0.007  | 0.082   | 0.415   | 0.092   | 0.986   | 0.086   | 0.45    | 0.089   | 0.852   | 0.098   | 1.424   | 0.073   |
|     | SAR     | -0.005  | 0.083   | 0.327   | 0.121   | 0.923   | 0.085   | 0.268   | 0.008   | 0.714   | 0.086   | 1.332   | 0.088   |
|     | FDSB    | -0.001  | 0.105   | 0.421   | 0.111   | 0.99    | 0.106   | 0.485   | 0.068   | 0.887   | 0.069   | 1.444   | 0.076   |
|     | FDBB    | -0.023  | 0.118   | 0.413   | 0.117   | 0.967   | 0.112   | 0.463   | 0.085   | 0.858   | 0.095   | 1.424   | 0.087   |
we found the FD y s Values for Estimating the Variance of a General Statistic. 

and MA part of model ARFIMA(p,d,q) use the AR part S 

This paper has studied finite sample performances of three popular used bootstrap methods in long memory and MA part in this process are not very significant to the model. ARFIMA (1, d, 1) model, the estimations are quite well. We think this mainly beca

Table 1-2 report the mean and St.d of estimated d in four different data generating cases. From table 1 we found that SM method outperforms other three method, and the FDSB is the winner among remaining methods in the most cases. Especially, SARB always tends to give a smaller estimation. Since we directly estimate long memory parameter regardless the AR part and MA part of model ARFIMA(p,d,q), which leads our estimated results are very poor for all four methods. These results in table 2 indicate that we have to determine the order of AR part and MA part of model ARFIMA(p,d,q) before estimating d. However, how to determine these orders, especially in non-stationary ARFIMA (p, d, q) model, is a challenge work. Although the data generating process in table 2 also is an ARFIMA (1, d, 1) model, the estimations are quite well. We think this mainly because the AR part and MA part in this process are not very significant to the model.

5. Conclusions
This paper has studied finite sample performances of three popular used bootstrap methods in long memory time series via simulation experiments. Three Bootstrap methods were used to approximate the long memory parameter of the samples and compared with the real distribution of the samples. By the degree of difference, we found the FDSB methodis accurate enough to estimate the long memory parameters of the approximate sample, that is, the FDSB method is superior to the other two methods in approximating the long memory parameter.

6. Acknowledgements
This work was supported by the National Natural Science Foundation of China (No.11661067), Natural Science Foundation of Qinghai Province (No. 2019-ZJ-920).

7. References
[1] Abadir, K.M., Distaso, W., Giraitis, L., 2007. Nonstationarity-extended local Whittle estimation. Journal of Econometrics 141, 1353-1384.
[2] Andrews D.W.K., Lieberman O. and Marmer V. Higher-order improvements of the parametric bootstrap for long-memory Gaussian processes. Journal of Econometrics, 2006, 133(2):673-702.
[3] Buhlmann P. Sieve bootstrap for time series[J]. Bernoulli, 1997, 3(2): 123-148.
[4] Carlstein E. The Use of Subseries Values for Estimating the Variance of a General Statistic from a Stationary Sequence[J]. Annals of Statistics, 1986, 14(3):1171-1179.
[5] Davidson J., Rambacussing D. 2015. A Test of the Long Memory Hypothesis Based on. Self-Similarity. Journal of Time Series Econometrics, 2015, 7(2): 115-141.
[6] Efron B. Bootstrap Methods: Another Look at the Jackknife, Annals of Statistics, 1979, 7:1-26.
[7] HidalgoJ. Analternativebootstrapmovingblocksfortimeseriesregressionmodels. Journal of Econometrics, 2003, 117(2):369-399.
[8] Kapetanios G.A. generalization of a sieve bootstrap invariance principle to long memory. processes. Quantitative and Qualitative Analysis in Social Sciences, 2010, 4: 19-40.

Table 2. Mean and St.d of estimated d

| Mean | St. D. | Mean | St. D. | Mean | St. D. | Mean | St. D. | Mean | St. D. |
|------|--------|------|--------|------|--------|------|--------|------|--------|
| d = 0 | 0.026 | 0.155 | 0.39 | 0.16 | 0.965 | 0.158 | 0.237 | 0.171 | 0.618 | 0.17 | 1.2 | 0.159 |
| d = 0.4 | 0.022 | 0.157 | 0.181 | 0.198 | 0.794 | 0.166 | 0.072 | 0.094 | 0.459 | 0.141 | 1.025 | 0.139 |
| d = 1 | 0.038 | 0.187 | 0.378 | 0.203 | 0.933 | 0.199 | 0.344 | 0.135 | 0.715 | 0.124 | 1.276 | 0.119 |
| d = 0 | 0.066 | 0.212 | 0.373 | 0.206 | 0.925 | 0.21 | 0.219 | 0.172 | 0.614 | 0.151 | 1.168 | 0.158 |
| d = 0 | 0.004 | 0.086 | 0.407 | 0.092 | 0.987 | 0.084 | 0.11 | 0.086 | 0.514 | 0.08 | 1.103 | 0.088 |
| d = 0.4 | 0.008 | 0.085 | 0.304 | 0.131 | 0.917 | 0.085 | 0.041 | 0.05 | 0.46 | 0.081 | 1.043 | 0.085 |
| d = 1 | 0.021 | 0.109 | 0.401 | 0.11 | 0.97 | 0.106 | 0.197 | 0.093 | 0.597 | 0.092 | 1.18 | 0.128 |
| d = 0 | 0.022 | 0.116 | 0.407 | 0.12 | 0.97 | 0.118 | 0.106 | 0.323 | 0.072 | 0.709 | 1.097 | 0.123 |
[9] Kunsch H.R. The Jackknife and the Bootstrap for General Stationary Observations. Annals of Statistics, 1989, 17(3):1217-1241.
[10] Lazarova S. Testing for structural change in regression with long memory processes. Journal of Econometrics, 2004, 129(1):329-372.
[11] Murphy A., Izzeldin M. Bootstrapping long memory tests: Some Monte Carlo results. Computational Statistics and Data Analysis, 2009, 53: 2325-2334.
[12] Papailias F., Kapetanios G. Block Bootstrap and Long Memory. 2011, University of London. Queen Mary School of Economics and Finance Working Paper No. 679.
[13] Politis D N, Romano J P. A circular block-resampling procedure for stationary data. Exploring the limits of bootstrap. 1992.
[14] Poskitt D.S. Properties of the sieve bootstrap for fractionally integrated and noninvertible processes. Journal of Time Series Analysis, 2008, 29(2): 224-250.