



Research Report
No. 66

Matched-Block Bootstrap for Long Memory Processes

Tim Hesterberg

Last revised
November 2, 1997

Acknowledgments: This work was supported in part by NSF SBIR DMI-9661424.

MathSoft, Inc.
1700 Westlake Ave. N, Suite 500
Seattle, WA 98109-9891, USA
Tel: (206) 283-8802
FAX: (206) 283-6310

E-mail: timh@statsci.com
ftp: [ftp.statsci.com](ftp://ftp.statsci.com/pub/longmem)
(pub/longmem)

Matched-Block Bootstrap for Long Memory Processes

November 2, 1997

Abstract

The block bootstrap for time series consists in randomly resampling blocks of consecutive values of the given data and aligning these blocks into a bootstrap sample. The matched-block bootstrap (Carlstein et al. (1997)) samples blocks dependently, attempting to follow each block with one that might realistically follow it in the underlying process, to better match the dependence structure of the data. Blocks may be matched using a single value at the end of a block, which is ideal for Markov processes. We investigate other block matching rules, based on linear combinations of observations in the block which forecast the future of the series, and find the new rules to be more suitable for long memory processes.

Key Words: Blocking methods, bootstrap, kernel methods, resampling, time series, variance estimation.

1 Introduction

The bootstrap is a general statistical procedure for statistical inference, including standard error and bias estimation, confidence intervals, and hypothesis tests. In their classical form (Efron (1979)) bootstrap methods are designed for application to samples of independent data. For example, given observations $X_i, i = 1, \dots, n$, which are independent, a standard nonparametric bootstrap sample is obtained as a sample of size n , selected *independently* (with replacement) from the original observations. In order to estimate the standard deviation (standard error) of a statistic such as the sample mean, one generates many bootstrap samples, computes the statistic for each, and uses the standard deviation of those bootstrap statistics to estimate the standard deviation of the original statistic. This procedure is inappropriate for time series because it ignores the correlation between successive observations. An example time series and bootstrap sample of independent observations, are shown in the first two panels of Figure 1. Note that the series of independent observations does not reflect the behavior of the time series.

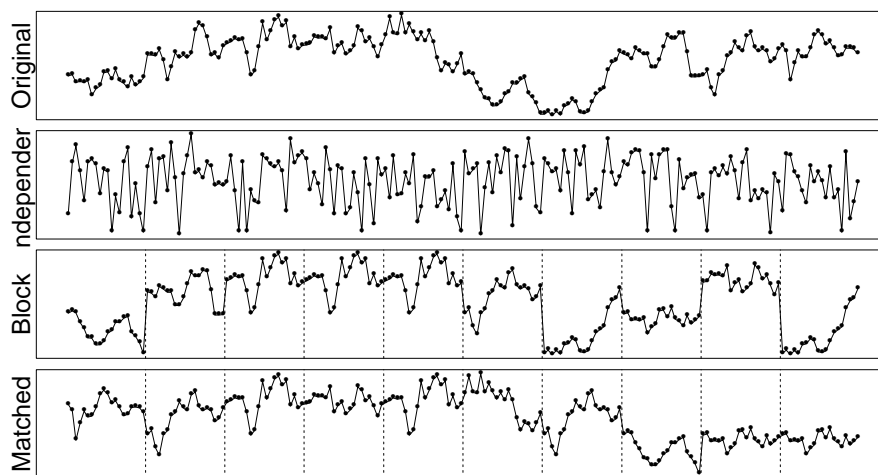


Figure 1: The panels show: (1) the original time series (an AR(1) series), (2) independent observations from the original series, (3) block bootstrap series, and (4) matched block bootstrap series, based on matching the block ends.

Block bootstrap methods for dependent data were introduced by Hall (1985); Carlstein (1986); Künsch (1989); Liu and Singh (1992). These methods are highly adaptive, or non-parametric, in the spirit of bootstrap methods. The method has subsequently been investigated in some detail, e.g. in Shao (1993); Naik-Nimbelkar and Rajarshi (1994); Bühlmann (1994); Bühlmann (1995); Radulovic (1995a); Radulovic (1995b); Radulović (1996); Politis and Romano (1992b); Bühlmann and Künsch (1995). Hall and Jing (1996); Hall et al. (1995); Bühlmann and Künsch (1994) have addressed the issue of block choice and related matters. Politis and Romano (1994); Politis and Romano (1995) study modifications of the

basic procedure. The block bootstrap relies on a compromise between preserving dependence within blocks, but destroying it between blocks. An example is shown in the third panel of Figure 1. Here we observe abrupt changes between the ends of some blocks and the beginnings of the following blocks. Although blocks of data are dependent in the original time series, they are independent (and randomly ordered) in the bootstrap version. For estimating the standard error of the sample mean, this causes estimates which are biased downward, and the bias can be large if the dependence in the data is strong.

The matched-block bootstrap is introduced by Carlstein et al. (1997), who show that it produces standard error estimators that asymptotically are less biased than, and have virtually the same variability as, those based on the ordinary block bootstrap. The procedure uses a block joining rule which favors blocks which were *a priori* more likely to be close to one another. For example, if the end of one block is relatively high then the beginning of the following block is likely to also be relative high (if the series has positive autocorrelation). Blocks are prepared as before, but are placed into bootstrap samples using a Markov chain, using transition probabilities which favor close matching. An example is shown in the fourth panel of Figure 1.

In this report, we describe different block bootstrap procedures for estimation of the mean of a time series with long-range dependence. We find that the matched block bootstrap procedure is feasible for series with moderate long-range dependence, and that block matching procedures which better take into account long memory characteristics generally have smaller bias and RMSE.

We describe block and matched-block bootstrap in Section 2, and block matching based on predictions of future observations in Section 3. We propose ways to reduce the bias of (matched-)block bootstrap procedures in Section 4, and propose a hybrid method, matched-block bootstrapping of residuals from a long-memory model, in Section 5.

2 Block Bootstrap and Matched-Block Bootstrap

We describe first the basic block bootstrap procedure, then block matching. Given data $\mathcal{X} = \{X_i, 1 \leq i \leq n\}$ from a stationary time series, prepare b blocks $\mathcal{B}_1, \dots, \mathcal{B}_b$ of length l consecutive observations each, i.e. $\mathcal{B}_i = \{X_{i1}, \dots, X_{il}\}$ is a block of length l and X_{ij} is the j th observation in block i . The blocks may be overlapping, with $b = n - l + 1$ and $X_{ij} = X_{i+j-1}$, or non-overlapping, with b the integer part of n/l and $X_{ij} = X_{(i-1)l+j}$. In the block bootstrap, a bootstrap series is generated by randomly sampling blocks of observations, and concatenating them to form a series of length n . If n is not a multiple of l , then the last block is truncated so the series is the right length.

In the matched-block bootstrap, blocks are prepared as before, but are placed into bootstrap samples using a Markov chain, using transition probabilities which favor close matching. An example is shown in the fourth panel of Figure 1.

The matching procedure used here is based on “rank matching”, which we found to be

most accurate and generally satisfactory in previous work (Carlstein et al. (1997)). Let E_i characterize the end of block \mathcal{B}_i ; for now we suppose that E_i is the last observation in the block,

$$E_i = X_{it},$$

but we generalize this later. Let R_i be the rank of E_i , for $i = 1, \dots, b$. If the *current* block (the one most recently appended to the bootstrap series) is \mathcal{B}_i , we randomly *select* one of the $2k + 1$ blocks ranked between $R_i - k$ and $R_i + k$, for some small positive integer k , then *append* the *following* block to the bootstrap series. The appended block becomes the new current block, and the process is repeated until the bootstrap series is the desired length. Only the ends of blocks are used in matching; using the beginnings of following blocks as well would correspond to looking into the future.

Note that by appending a block which follows a block whose end closely matches the current block, this procedure handles negative autocorrelation correctly. Suppose a series is highly negatively autocorrelated, and that the current block ends with a high value. We first search for other blocks whose ends have high ranks and select one of them randomly, but then append the block that originally followed the selected block to the series; the beginning of the appended block is likely to be low, as desired.

Two modifications to the basic matching rule are necessary to handle end cases: extreme ranks, and the beginning and end of the time series. A third modification, to allow a non-integer value of k , is described below in Section 2.1. Some additional notation is useful in explaining these modifications. Let R_i^p be the rank of the end of the “previous” block, the block that preceded \mathcal{B}_i in the original time series by l observations; i.e. \mathcal{B}_{i-l} with overlapping blocks, or \mathcal{B}_{i-1} with non-overlapping blocks. Given that \mathcal{B}_i is the current block, generate U randomly from the integers between $R_i - k$ and $R_i + k$. The basic matching rule says to append the block \mathcal{B}_j for which $R_j^p = U$. The first modification occurs for extreme ranks, if $U < 1$ or $U > b$; in this case we simply “fold back” U into the desired range, using $U' = 2 * b + 1 - U$ if $U > b$ and $U' = 1 - U$ if $U < 1$. This results in a Markov chain with stationary transition probabilities, because all rows and columns of the transition matrix add to 1; this is shown in Figure 2.

The second modification to the basic procedure is necessary because the original series has a beginning and an end—the basic rule of *selecting* a block whose end matches the current end of the series, then *appending* the *following* block to the series results in (a) often trying to append a non-existent block to the series (if the selected block was at the end of the original series), and (b) never appending the initial block(s) from the original series (because they do not have preceding blocks which may be selected). The modification we use is to backcast the series, and use the backcasted and actual observations to define E_1^p, \dots, E_b^p , where E_p^i is the “end” of the block that preceded \mathcal{B}_i in the original series (with backcasted observations prepended), then let R_i^p be the rank of E_i^p .

An alternate approach is to create a circular time series by pasting together the beginning and end of the series, so that if a block selected is near the end of the original series, the

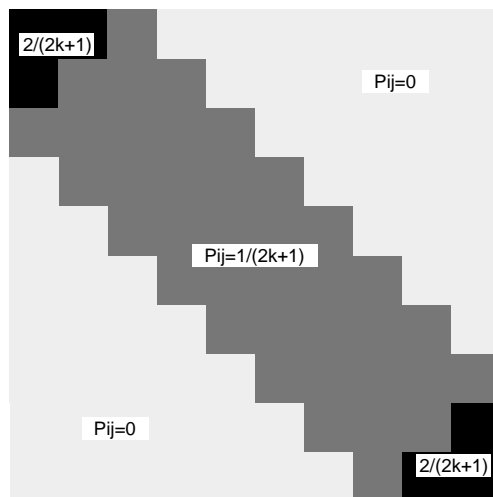


Figure 2: Transition probability matrix for rank matching with $b = 10$ and $k = 2$. Both rows and columns are sorted by ranks. Transition probabilities are zero outside a diagonal band and $(2k + 1)^{-1}$ within the band, except for extreme cases where the probabilities are $2(2k + 1)^{-1}$ due to folding back across the end.

appended block has its end near the beginning of the original series; note that with this approach there are $b = n$ rather than $b = n - l + 1$ overlapping blocks. Circular block bootstrapping was discussed in Politis and Romano (1992a). A related procedure which is likely to perform better for long memory problems would be to reflect the series at both ends, so that $X_i = X_{1-i}$ for $i < 0$ and $X_i = X_{2n+1-i}$ for $i > n$.

Some results from our earlier work are shown in Figure 2 (part of this is included in Carlstein et al. (1997)). This figure indicates that:

1. matched block bootstrapping has much less bias than simple block bootstrapping,
2. overlapping blocks (curves with overlaid circles) have about the same bias but a smaller mean squared error than non-overlapping blocks,
3. the optimal block length is shorter when using matched blocks (because blocks do not need to be as long to capture dependence structure).

In the sequel we use overlapping blocks.

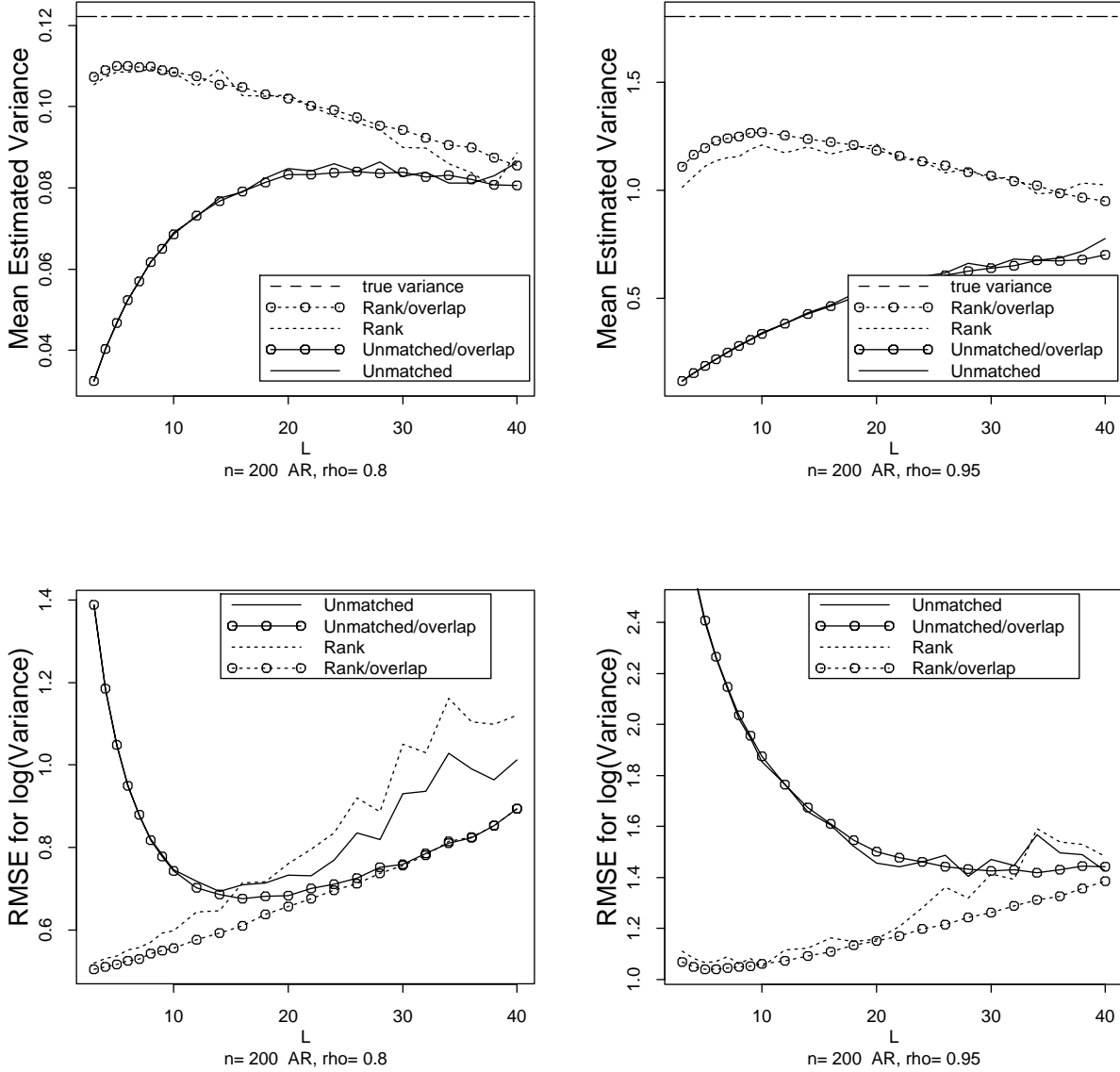


Figure 3: Average, and root mean square error, of bootstrap estimates of the variance of the sample mean (i.e. square of the standard error), for series of length 200 from AR(1) processes with autocorrelation 0.8 and 0.95. “Unmatched” indicates simple block bootstrapping, “rank” indicates block matching, using rank matching based on the last observation in a block, and “/overlap” indicates the use of overlapping blocks. L is the length of a single block. Results are across 500 simulated series, with 500 bootstrap series generated from each simulated series.

2.1 Number of neighbors for rank matching

Our rule for choosing the number of neighbors k for rank matching is based on Carlstein et al. (1997). There, based on minimizing asymptotic mean square error, the choice $h = O(n^{-1/5})$ is found to be optimal, where h is the bandwidth for kernel matching (kernel matching is described in Carlstein et al. (1997)). The choice involves a tradeoff between bias and variance. This suggests the choice $h = cn^{-1/5}$, for some c . The optimal choice of c is an open question, but in simulations done in support of Carlstein et al. (1997) (but not reported there), we found that the choice $c = 1$ was reasonable.

For rank matching, we choose the number of neighbors by the rule

$$2k + 1 = 1 + .84bh \tag{1}$$

so that the effective bandwidth of the rank matching procedure is roughly equal overall to that of the kernel matching procedure. The equality can only be rough, because the methods are substantively different—this choice makes the effective bandwidth of the rank matching procedure greater (less) than that of kernel matching in the tails (center) of the distribution of the block ends, respectively; a plot of the transition probability matrix for kernel smoothing, similar to that shown for rank matching in Figure 2, would show a roughly diagonal pattern but with the width of the diagonal band wider in the center than at either tail. As in the case of kernel matching, we found in simulation work that the choice of $c = 1$ was reasonable.

The use of (1) leads to a non-integer value of k . Rather than round k to the nearest integer, we modify the matching rule. Given that \mathcal{B}_i is the current block, generate U randomly with a uniform distribution between $R_i \pm (2k + 1)/2$, then round U to the nearest integer, and proceed as described earlier.

3 Matching based on predictions

Ranking blocks according to the *single* last value in the block is well suited for Markov processes, in which the last value in a block contains all the information contained in the block for predicting future observations. Matching procedures which are more suitable for bootstrapping the mean of a long memory process are based on ranks for E_i which are *linear combinations of the observations in the block*. We propose that these combinations be chosen to forecast the future of the series, for example where E_i is defined as the prediction of the next observation, or the mean of the next 100 observations. We consider five procedures here:

1. block bootstrapping without matching (“no matching”),
2. matched-block based on the last observation in the block,

3. matched-block based on predictions for the next observation following each block,
4. matched-block based on predictions for the average of the next 100 observations following each block, with predictions based on observations in a block, (“predict 100”) and
5. matched-block based on predictions for the average of the next 100 observations following each block, with predictions based on observations preceding and within a block, (“predict 100m”).

Simulation results for these procedures are shown in Figures 3 and 3 for estimating the variance of the sample mean for long-memory processes (fractional ARIMA (FARIMA) with $d = 0.1$ or $d = 0.25$). Note that this problem is substantially more difficult than the same problem for a short-memory process. For instance, even the procedure that performed the best in Figure 2—block matching using the ends of blocks, with overlapping blocks—substantially underestimates the variance. In practice this would cause confidence intervals to be too narrow.

The new procedures, based on predictions, have smaller bias than both bootstrapping without matching and matching based on block ends. The best results are obtained with the last procedure, which respects the long-memory aspect of these processes in two ways: forecasts are for the average of the next 100 observations, not just the next observation, and forecasts are based on the current block and all preceding blocks, not just the observations in the current block—the extra observations have a significant impact when working with short blocks.

The optimum value of the block length L is smaller with the new procedures. Because the optimal block length involves a tradeoff between bias caused by incorrectly modeling autocorrelations (which is reduced by longer blocks) and variability (which is larger if there are fewer blocks), this provides further evidence that the new procedures do a better job of modeling autocorrelations.

The results should be considered preliminary. In particular, the predictions were based on the true FARIMA model (with coefficients estimated from the data), but in practice predictions must be made without that knowledge. In addition, the last procedure seems to have relatively high variability in the case $d = 0.1$. We do not yet know why this is the case.

The new procedures provide less-biased estimates, but even the best procedure substantially underestimate the variance. In the next section we discuss ways to further reduce the bias. We also note that results are reasonable for the case $d = 0.1$, which is important in practice, because mild long-range dependency is the hardest for practitioners to diagnose. Furthermore, the series simulated here are FARIMA processes with no AR or MA components. If those components were nonzero (for short-term lags), both matched and unmatched block bootstraps would capture a higher fraction of the total variation than here.

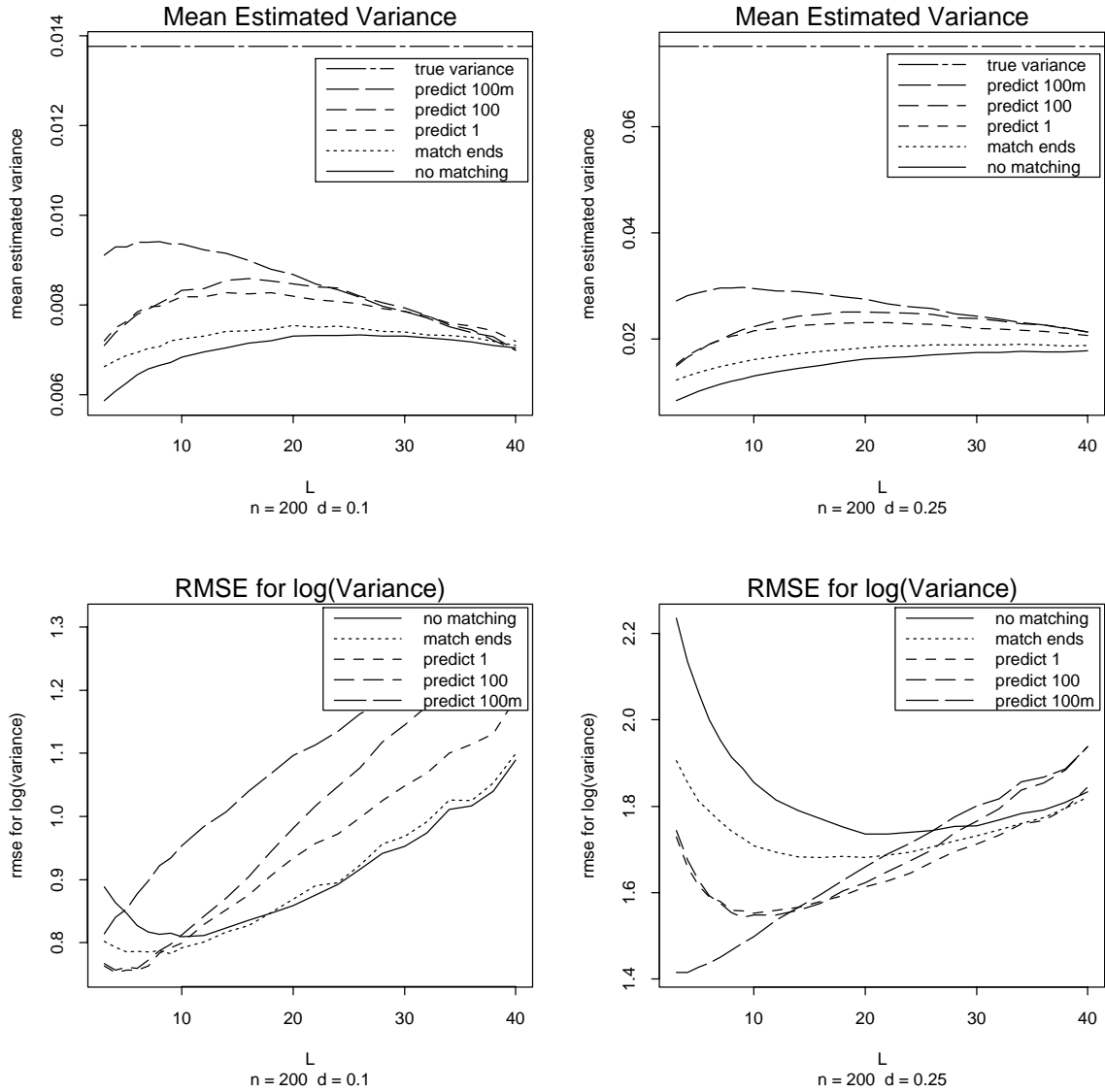


Figure 4: Root mean square error, and average, of bootstrap estimates of the variance of the sample mean (i.e. square of the standard error), for series of length 200 from FARIMA processes with $d = 0.1$ and $d = 0.25$. The bootstrap methods are: (1) simple block bootstrapping, without block matching, (2) block matching, using the last observation in a block, (3) block matching, using predictions of the observation following a block, (4) block matching, using predictions of the average of the 100 observation following a block. (5) like (4), but with predictions based on preceding blocks in addition to the current block. L is the length of a single block. Results are across 500 simulated series, with 500 bootstrap series generated from each simulated series.

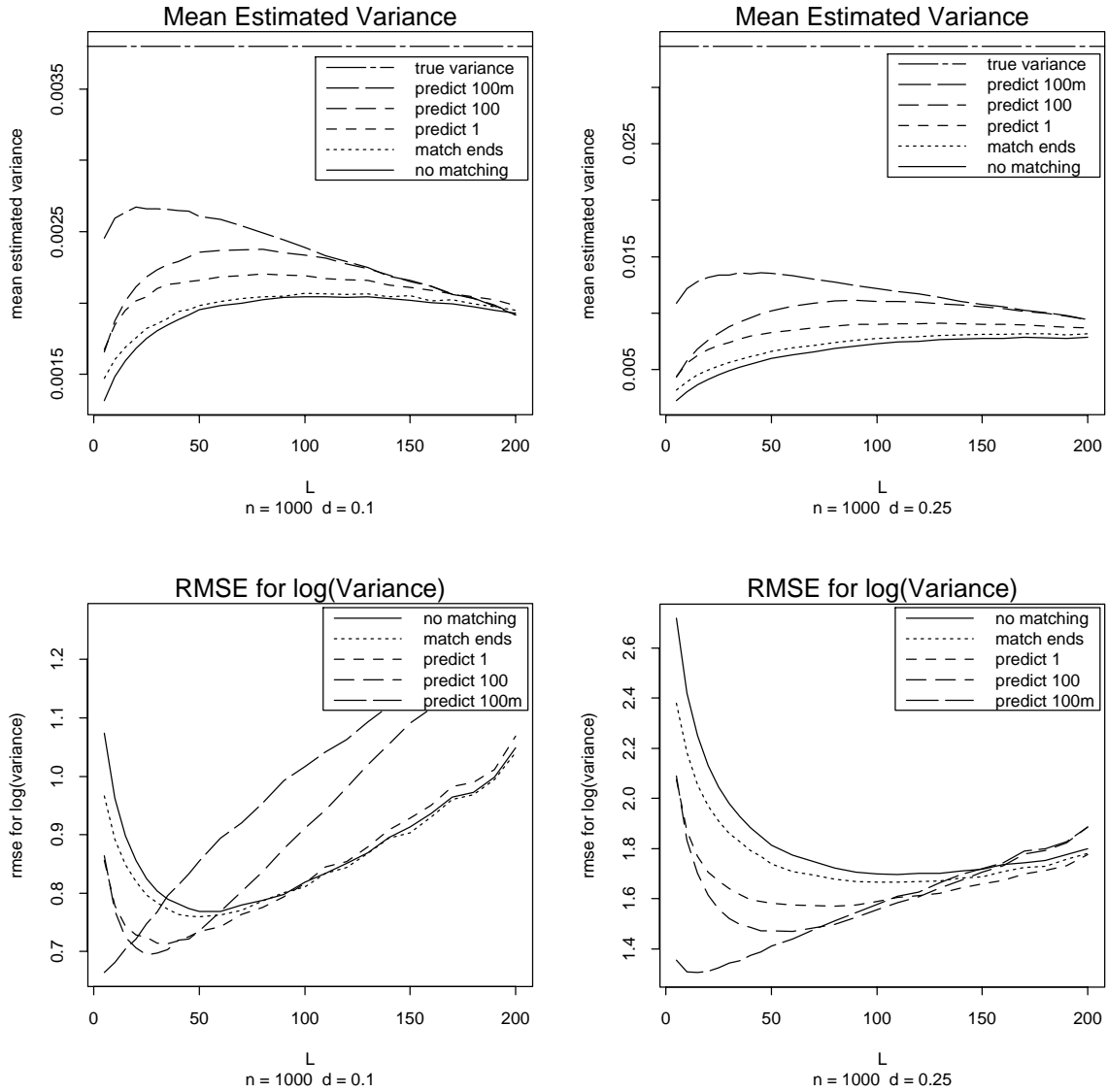


Figure 5: Root mean square error, and average, of bootstrap estimates of the variance of the sample mean (i.e. square of the standard error), for series of length 1000 from FARIMA processes with $d = 0.1$ and $d = 0.25$. The bootstrap methods are: (1) simple block bootstrapping, without block matching, (2) block matching, using the last observation in a block, (3) block matching, using predictions of the observation following a block, (4) block matching, using predictions of the average of the 100 observation following a block. (5) like (4), but with predictions based on preceding blocks in addition to the current block. L is the length of a single block. Results are across 500 simulated series, with 500 bootstrap series generated from each simulated series.

4 Reducing bias

It appears from these results that matched-block bootstrapping is feasible for obtaining standard errors for the sample mean, for series with short-range dependence and moderate long-range dependence, but that there is still significant bias. There are a number of ways that the bias might be reduced, including:

- rescaling: this is similar to dividing by $(n - 1)$ rather than n in the common definition of sample variance, but the factor would take into account the estimated dependence between observations,
- calibration: use of the iterated bootstrap to estimate the bias in the first-order bootstrap procedure, and correcting estimates for the bias,
- forecasting some number other than 100 of future observations,
- dynamic matching: letting forecasts of future observations based on the current and previous blocks use previous blocks in a bootstrap sample rather than previous blocks from the original series,
- data-based methods for choosing the block length and the bandwidth k used for rank matching,
- non-uniform probabilities for rank matching.

We discuss these refinements in turn.

Rescaling: one reason for underestimating variance is that the empirical distribution underestimates the variance in the underlying distribution: $E(n^{-1} \sum_{i=1}^n (X_i - \bar{X})^2) < \sigma^2$. If the variance of individual observations is too small, then bootstrap estimates of variance of a sample mean also tend to be too small. For independent samples the usual remedy is to multiple the quantity inside the expectation by a factor $n/(n - 1)$ to obtain an unbiased estimate of the variance. For a time series the corresponding factor would be

$$\frac{\sigma^2}{\sigma^2 - \text{var}(\bar{X})} \quad (2)$$

where σ^2 is the variance of a single observation. For long-memory processes this factor can be substantial.

Another approach to rescaling focuses on block means, rather than individual observations. The empirical distribution of block means underestimates the underlying variance of block means: $E(b^{-1} \sum_{i=1}^b (\bar{X}_i - \bar{X})^2) < \text{var}(\bar{X}_i)$, where \bar{X}_i is the mean of \mathcal{X}_i . The corresponding factor would be

$$\frac{\sigma_l^2}{\sigma_l^2 - \text{var}(\bar{X})} \quad (3)$$

where σ_l^2 is the variance of a block mean of length l .

Either factor (2, 3) can be estimated from the data, then multiplied by the current estimate of $\text{var}(\bar{X})$ to obtain a new estimate. Repeating the process using the new estimate

as the current estimate, and iterating until convergence, corresponds to solving

$$\widehat{\text{var}}(\bar{X}) = \frac{s^2}{s^2 - \widehat{\text{var}}(\bar{X})} \widehat{\text{var}}_{(0)}(\bar{X}) \quad (4)$$

where $\widehat{\text{var}}_{(0)}(\bar{X})$ is the initial (un-rescaled) matched-block bootstrap estimate of the variance and s^2 is the usual sample variance. Replace s^2 with the sample variance of block means s_l^2 to use factor (3).

Calibration: It may be possible to use the iterated bootstrap (bootstrapping from bootstrap samples) to estimate the bias in the bootstrap estimate of the variance. However, this is may not be effective for that part of the bias which is caused by failing to capture long-range dependence. Iterative bootstrapping is also computationally expensive.

Forecasting: The number of future observations forecast in the above work, 100, was chosen arbitrarily. Some other number may be better. In fact, it should be particularly effective to forecast as many observations as still need to be placed into the bootstrap sample. However, this would require using different sets of ends and ranks, E_i, E_i^p, R_i, R_i^p for $i = 1, \dots, b$ after every block, which would be computationally more expensive.

Dynamic matching: in the fifth method in the simulations, 100 future observations were forecast following each block using observations in the block together with preceding observations (which were backcasted when necessary). The ends E_j and E_j^p were computed as the average of the 100 forecasts, for $j = 1, \dots, b$. The forecasts were precomputed, using observations as ordered in the original series. An alternative would be to replace E_i (when the current block is \mathcal{B}_i) with E computed “on the fly” using observations already in the bootstrap vector. This procedure would be more computationally expensive, and in the form just described might be even more biased downward for estimating variances, partly because the stationary probabilities on the blocks would not be uniform. However, combining this with a post-hoc control variate adjustment, to control for nonuniform frequencies of block selection, may be effective.

Block length and bandwidth: there may be more effective rules for selecting the bandwidth (number of neighbors used for rank matching) than we have used to date. It would also be desirable to have a rule for selecting the block length L , rather than repeating the bootstrap process for many values of L . The optimal choices would depend on the underlying covariance structure, which may be estimated from the data.

Non-uniform probabilities for rank matching: in Section 2.1 we described a procedure for use when the number of neighbors k is not an integer. One result is that the next block is chosen with unequal probabilities from the candidates, with smaller probabilities for the two candidates which are “on the edge”. Extending this idea, we might generate the random variate U from a bell-shaped distribution rather than a uniform distribution. This may result in a slightly more favorable bias/variance tradeoff, though experience with different shapes of kernels in kernel regression or density estimation suggests that the improvement may be minor. For computational efficiency it would be helpful if U has a bounded distribution,

e.g. generated as the convolution of four uniformly-distributed random deviates.

4.1 Computational Efficiency

Computational efficiency can be an important consideration in bootstrap procedures. We discuss here some computational considerations in matched-block bootstrapping.

The matched-block procedures we have used to date are relatively efficient computationally. The block ends E_i and E_i^p , and corresponding ranks, are all computed once, prior to doing any bootstrap sampling. We sort the blocks and the corresponding E_i by the values of E_i^p prior to bootstrap sampling; this makes it easy during the bootstrap sampling to find the block with the desired rank which is to be appended.

Furthermore, for the special case of bootstrapping the sample mean, the block sums can be precomputed, so the actual bootstrap time series need never be formed; instead the sum of the observations in the series can be calculated using just the block sums. This method could also be used for other statistics that depend solely on sample moments, such as variances and autocorrelations (for a small, fixed, number of lags).

Some of the alternate procedures described above would be more computationally expensive. Iterated bootstrapping, in particular, is very expensive. Changing the number of observations to be forecasted, depending on how many observations have been placed, may be expensive, as would dynamic matching.

In other cases it may be possible to reduce the computational burden. The most expensive procedures we used involve forecasting 100 future observations; we did this one at a time, with each forecast based on the actual plus some number (0 to 99) of forecasted observations. This process could probably be sped up, by expressing the sum of the next 100 observations as a linear combination of the actual observations (any linear combination of actual observations and observations which are forecasted using a linear combination of the actual observations can be re-expressed as a linear combination of the actual observations). This might be done numerically or (in some cases) analytically.

In addition, there are numerous “variance reduction techniques” which have been proposed in the bootstrap literature (including our own publications), of which we judge two to be particularly promising for use in bootstrapping time series – control variates (Davison et al. (1986); Efron (1990); Hesterberg (1996)) and our unpublished bootstrap-specific variation of quasi-random sampling (see Do and Hall (1991); Niederreiter (1992); Kocis and Whiten (1997) for information on quasi-random sampling). These may reduce the number of bootstrap samples necessary to reduce the bootstrap simulation variance to acceptable levels (note that this affects variance only due to random bootstrap sampling).

5 Matched-block bootstrap of residuals

The primary focus of this report is on block bootstrap methods. A competing approach for bootstrapping time series, bootstrapping residuals (Efron and Tibshirani (1986); Holbert and Son (1986)), involves postulating a parametric model, estimating the parameters for the model and calculating the residuals $\hat{\epsilon}_i = X_i - \hat{X}_i$ where \hat{X}_i is the predicted value for X_i using previous observations (e.g. a linear combination of p previous observations in an AR(p) model). Then a bootstrap sample is generated one observation at a time by combining model-based predictions with innovations obtained by sampling with replacement from the estimated residuals. The accuracy of this procedure depends on how well the model approximates reality.

In future work we will combine this model-based approach with matched-block bootstrapping, by generating the sequence of innovations using a matched-block bootstrap of the residuals from the original model fit. We envision using parametric models which capture long-range dependence, but not necessarily the shorter-range dependence structure. The residuals from this model would have shorter-range dependence, to be captured by block bootstrapping. This shares advantages of both methods – capturing long-range dependence using a model-based approach, while reducing errors caused by differences between the model and reality.

6 Conclusion

The primary result in this article is that matched-block bootstrapping has potential for estimating the variance of the sample mean of a time series. For ARIMA processes, and fractional ARIMA processes with mild long-range dependence, the results are reasonable. For fractional ARIMA processes with moderate to high long-range dependence, even our best procedures substantially underestimate the bias. Various refinements of the existing procedures may improve this situation. These refinements need further study, to compare their effectiveness, computational efficiency, and (in some cases) sensitivity to differences between models used for forecasts and the underlying true model.

Future work should involve bootstrapping statistics other than the sample mean. In the case of bootstrapping the sample mean effective matching procedures involve forecasting the *mean* of some number of future observations, but for other statistics other matching procedures would be appropriate. In particular, linearizing the statistic (finding a nonlinear transformation of the data such that the desired statistic is approximately equal to the mean of the transformed series, for both the original data and bootstrap series) would often be useful. Then rank matching would be based on forecasts obtained as a linear combination of values in the linearized series. However, the actual bootstrap series would still use the original data.

References

- Bühlmann, P. (1994). Blockwise bootstrapped empirical processes for stationary sequences. *Ann. Statist.*, 22:995–1012.
- Bühlmann, P. (1995). The blockwise bootstrap for general empirical processes of stationary sequences. *Stoch. Proc. Appl.*, 58:247–265.
- Bühlmann, P. and Künsch, H. R. (1994). Block length selection in the bootstrap for time series. Research Report 72, Seminar für Statistik, ETH Zurich.
- Bühlmann, P. and Künsch, H. R. (1995). The blockwise bootstrap for general parameters of a stationary time series. *Scand. J. Statist.*, 22:1985–2007.
- Carlstein, E. (1986). The use of subseries values for estimating the variance of a general statistic from a stationary sequence. *Ann. Statist.*, 14:1171–1179.
- Carlstein, E., Do, K., Hall, P., Hesterberg, T. C., and Künsch, H. R. (1997). Matched-Block Bootstrap for Dependent Data. *Bernoulli*. in press.
- Davison, A. C., Hinkley, D. V., and Schechtman, E. (1986). Efficient Bootstrap Simulation. *Biometrika*, 73:555 – 566.
- Do, K. A. and Hall, P. (1991). Quasi-random sampling for the bootstrap. *Statistics and Computing*, 1(1):13–22.
- Efron, B. (1979). Bootstrap methods: another look at the jackknife. (with discussion). *Annals of Statistics*, 7:1–26.
- Efron, B. (1990). More Efficient Bootstrap Computations. *Journal of the American Statistical Association*, 85(409):79 – 89.
- Efron, B. and Tibshirani, R. J. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science*, 1:54–77.
- Hall, P. (1985). Resampling a coverage pattern. *Stoch. Proc. Appl.*, 20:231–246.
- Hall, P., Horowitz, J. L., and Jing, B. (1995). On blocking rules for the block bootstrap with dependent data. *Biometrika*, 82:561–574.
- Hall, P. and Jing, B. (1996). On sample reuse methods for dependent data. *Journal of the Royal Statistical Society, Series B*, 58:727–737.
- Hesterberg, T. C. (1996). Control Variates and Importance Sampling for Efficient Bootstrap Simulations. *Statistics and Computing*, 6(2):147–157.

- Holbert, D. and Son, M. S. (1986). Bootstrapping a time series model: Some empirical results. *Communications in Statistics, A.*, 15:3669–3691.
- Kocis, L. and Whiten, W. J. (1997). Computational Investigations of Low Discrepancy Sequences. *ACM Trans. Math. Software*, to appear. available as unpublished technical report, Julius Kruttschnitt Mineral Research Centre, University of Queensland.
- Künsch, H. R. (1989). The jackknife and the bootstrap for general stationary observations. *Annals of Statistics*, 17:1217–1241.
- Liu, R. and Singh, K. (1992). Moving blocks jackknife and bootstrap capture weak dependence. In LePage, R. and Billard, L., editors, *Exploring the Limits of the Bootstrap*, pages 225–248, New York. Wiley.
- Naik-Nimbelkar, U. V. and Rajarshi, M. B. (1994). Validity of blockwise bootstrap for empirical processes with stationary observations. *Ann. Statist.*, 22:980–994.
- Niederreiter, H. (1992). *Random number generation and Quasi-Monte Carlo methods*. Capital City Press: Montpelier, Vermont.
- Politis, D. N. and Romano, J. P. (1992a). A Circular Block-Resampling Procedure for Stationary Data. In LePage, R. and Billard, L., editors, *Exploring the Limits of Bootstrap*, pages 263–270. Wiley, New York.
- Politis, D. N. and Romano, J. P. (1992b). A general resampling scheme for triangular arrays of α -mixing random variables with application to the problem of spectral density estimation. *Ann. Statist.*, 20:1985–2007.
- Politis, D. N. and Romano, J. P. (1994). The stationary bootstrap. *Journal of the American Statistical Association*, 89:1303–1313.
- Politis, D. N. and Romano, J. P. (1995). Bias-corrected nonparametric spectral estimation. *J. Time Series Anal.*, 16:67–103.
- Radulovic, D. (1995a). The bootstrap for empirical processes based on stationary observations. Preprint, University of Connecticut.
- Radulovic, D. (1995b). The bootstrap of empirical processes for α -mixing sequences. Preprint, University of Connecticut.
- Radulović, D. (1996). The bootstrap of the mean for strong mixing sequences under minimal conditions. *Statistics & Probability Letters*, 28:65–72.
- Shao, J. (1993). Linear model selection by cross-validation. *Journal of the American Statistical Association*, 88:486–494.