The Replacement Bootstrap for Dependent Data

Amir Sani, Alessandro Lazaric, Daniil Ryabko

The Replacement Bootstrap for Dependent Data

Amir Sani
INRIA Lille, France

Alessandro Lazaric
INRIA Lille, France

Daniil Ryabko
INRIA Lille, France

***Abstract*—Applications that deal with time-series data often require evaluating complex statistics for which each time series is essentially one data point. When only a few time series are available, bootstrap methods are used to generate additional samples that can be used to evaluate empirically the statistic of interest. In this work a novel bootstrap method is proposed, which is shown to have some asymptotic consistency guarantees under the only assumption that the time series are stationary and ergodic. This contrasts previously available results that impose mixing or finite-memory assumptions on the data. Empirical evaluation on simulated and real data, using a practically relevant and complex extrema statistic is provided.**

I. Introduction

A wide variety of applications, notably in economics, biology and environmental sciences, require estimating complex statistics of highly dependent time-series distributions. In many cases, only a single data sequence generated by the underlying process is available. In the case of dependent time series, such a sequence is essentially one realization (or data point) w.r.t. the computation of a statistic of interest. Meaningful estimates are only possible when several data points are available, and these have to be somehow obtained from the single one given. For some statistics, simple heuristics like chopping the available time series into many may suffice, while for others they clearly do not. Practically meaningful examples of the latter situation include extrema-based statistics, such as maximum drawdown (the difference between the maximum and the minimum attained after it), or even more complex ones such as the performance of a trading algorithm on the given data. Estimating such statistics require having multiple time series of length close to the time series of interest. There are, thus, three problems in estimating such statistics: *complex statistics*, which appear impossible to analyse theoretically, *highly-dependent time series*, and, as a consequence of the latter, *too few, or just a single sample*.

Bootstrap methods attempt to tackle the latter problem by generating new samples from the available ones, in such a way that these new samples are "likely" to be generated by the underlying process distribution. The original Bootstrap [3] was designed for independent and identically distributed (i.i.d.) data; it samples with replacement from the original sequence attempting to improve upon the estimates of simple statistics such as smooth functions of the mean [5, 23]. While generalizations to dependent data are available (some of these are briefly reviewed below), they concern only rather restricted classes of processes, such as Markov chains or geometrically mixing time series. In general, even for simple statistics the traditional bootstrap is concerned with, obtaining convergence guarantees is highly problematic, and the available theoretical results all require additional assumptions on the distributions. Specifically, under assumptions such as moment conditions and several conditions necessary for Edgeworth expansions, the i.i.d. Bootstrap, is a consistent estimator of the sampling distribution for some specific statistics and it is shown to produce asymptotic refinements for problems such as bias reduction and the construction of confidence intervals [5, 23].

In this paper we address the second problem (highly dependent time series), namely, we propose a bootstrap method for time series which have performance guarantees under *the only assumption that the time series are stationary ergodic*. This is vastly more general than the assumptions made in prior works. However, we do not attempt to solve the problem of theoretical analysis of complex statistics; since no theoretical results for such complex statistics as we consider (max drawdown) are available even for the i.i.d. case, there is at present no hope for dependent time series. Thus, the theoretical results we present establish the *consistency of the proposed method for generating new time-series*, and the performance in estimating the statistic of interest is evaluated empirically.

What we propose here is a novel, principally different, approach to generating bootstrap sequences, namely the *replacement bootstrap* (RB). Bootstrap sequences are generated by replacing elements in the original sequence according to their estimated conditional probability distribution (replacement distribution) over the observed sequence around their position. First, R positions in the original sequence are selected at random. Next, the estimated conditional distribution at the selected positions is calculated using the full time series around them. Finally, existing elements in these R positions are replaced with elements drawn from each of the R estimated replacement distributions. The main idea is that, unlike in the i.i.d. or Markov cases, in highly-dependent time series it is impossible to estimate the distribution of the whole sequence. It is, however, possible to estimate conditional distribution of a symbol given its past and future. Thus, instead of generating whole new sequences from scratch, we start with the one available and introduce some changes according to estimated replacement distributions, which we can expect to be reliable. To estimate the replacement distribution, here we use an estimator based on the universal measure $\mathcal{R}$ proposed in [17] (see also [18]). For this estimator we obtain theoretical consistency guarantees that hold for arbitrary stationary ergodic distributions, without relying on finite-memory, mixing or any other assumptions. Further, we study empirically the accuracy of the proposed method on an extrema statistic (max

drawdown) that has been shown to be inconsistent in the traditional bootstrap approaches. Additional details and results are reported in the extended version of this paper [21].

An important difference between the traditional applications of bootstrap and the one considered in this work should be pointed out. Typically, in the literature on bootstrap, one is interested in a statistic $\theta(P)$ which concerns single-dimensional marginals of P. For i.i.d. data, these define the distribution completely. In more general cases, one is interested in estimating a statistic of k-dimensional marginals, where $k \ll n$. The intuition is that these should provide sufficient precision for estimating the distribution P or indeed any statistic thereof, such in the case of m-order Markov (where one can take $k = m + 1$) or mixing processes. However, in our case we are interested in complex statistics $\theta_n(P)$ concerning n-dimensional marginals, where n is the size of the sample available, such as the performance of a (trading, learning, etc.) algorithm or the max drawdown. For stationary ergodic time series, studying these statistics cannot be reduced to studying k-dimensional distributions and thus the object of our study is not exactly the same as in traditional applications of bootstrap.

Prior work. Universal measures, such as $\mathcal{R}$ and those based on data compressors, are used for solving various statistical problems concerning time series, often not directly related to prediction or compression [20, 18], but have not been previously used for bootstrap. Next we briefly review two popular model free Bootstrap approaches to dependent data: block based methods and methods that rely on the Markov property (for a comprehensive review, see, e.g., [8]). Block methods are a direct generalization of the i.i.d. bootstrap to dependent data. Block methods resample contiguous blocks of data from the original sequence to capture the dependence structure defined by a block width. Block width selection determines the bias-variance trade-off: small blocks increase bias, while larger blocks increase variance. As their performance relies on proper width selection, automatically selection methods have been introduced [15]. For relative performance over various construction methods, see [11, 13]. Note that such methods necessarily ignore and break long-range dependence in the sequence. Therefore they are only suitable for processes where it is negligible, e.g., processes with fast mixing. When the Markov property and regularity conditions are met, the Markov bootstrap (MB) of [10] has been shown to be more accurate than block methods [7]. MB sequences are generated by directly estimating a Markov model of a given order from the observed sequence, and then sampling from the model to generate a bootstrap sequence. If the process generating the sequence is not Markov (or is Markov of a higher order) then the MB may perform worse than block-based methods.

II. Preliminaries

A sequence $\boldsymbol{X}_n = (X_1, \dots, X_n)$ is generated by a time-series distribution P over a finite alphabet A. We introduce the notation $\boldsymbol{X}_{<t} = (X_1, \dots, X_{t-1})$, $\boldsymbol{X}_{>t} = (X_{t+1}, \dots, X_n)$, and $\boldsymbol{X}_{t:t'} = (X_t, \dots, X_{t'})$ and we denote $\nu_{\boldsymbol{X}_n}(a_1, \dots, a_m) = \#\{s \leq n : X_s = a_m, \dots, X_{s-m+1} = a_1\}$ the number of occurrences of a word $(a_1, \dots, a_m)$ in $\boldsymbol{X}_n$. We assume that:

Assumption 1. *The process P is* stationary*, i.e. for any m, $\tau \in \mathcal{N}$, and any word $\boldsymbol{v}_m = (a_1, \dots, a_m) \in A^m$*

$$P(X_1 = a_1, \dots, X_m = a_m) = P(X_{1+\tau} = a_1, \dots, X_{m+\tau} = a_m),$$

and ergodic*, i.e., for any word $\boldsymbol{v}_m = (a_1, \dots, a_m)$ the frequency of $\boldsymbol{v}_m$ in a sequence $\boldsymbol{X}_n$ tends to its probability a.s.:* $P(\nu_{\boldsymbol{X}_n}(a_1, \dots, a_m)/n \to P(X_1 = a_1, \dots, X_m = a_m)) = 1$.

(The latter definition of ergodicity is equivalent to the usual one involving shift-invariant sets [4].) This assumption is vastly more general than those used in the literature on nonparametric (model free) bootstrap methods for time series, and allows us to focus on what is arguably the most general class of processes for which it is still possible to define meaningful statistical inference problems. We recall the definition of the Kullback-Leibler (KL) divergence, used to measure the accuracy of estimated distributions. Given two distributions P and Q over A, the KL divergence between P and Q is

$$\mathrm{KL}(P; Q) = \sum_{a \in A} P(a) \log(P(a)/Q(a)). \tag{1}$$

We use h for the Shannon entropy, $h_k(P)$ for the kth order entropy of the distribution P, and $h_\infty(P)$ for its *entropy rate*.

A bootstrap algorithm is a *random* mapping $\mathcal{B}_n : A^n \rightsquigarrow A^n$, such that given $\boldsymbol{X}_n$, $\mathcal{B}_n(\boldsymbol{X}_n)$ returns a (random) bootstrap sequence $\boldsymbol{b}_n$ of length n. The intuition behind the bootstrap approach is that a sequence $\boldsymbol{X}_n$ might provide enough information about P to generate additional sequences which are *likely* to be generated by P. Thus, given a single sequence $\boldsymbol{X}_n$, a bootstrap algorithm $\mathcal{B}_n$ repeatedly generates bootstrap sequences $\boldsymbol{b}_n^1, \dots, \boldsymbol{b}_n^B$. These sequence, together with the original $\boldsymbol{X}_n$ are then used to estimated the statistic of interest.

III. The Replacement Bootstrap

The main idea of the replacement bootstrap (RB) is to choose a set of R random positions $1 \leq t_1, \dots, t_R \leq n$ in the original sequence $\boldsymbol{X}_n$ and replace symbols $(X_{t_1}, \dots, X_{t_R})$ with symbols $(b_{t_1}, \dots, b_{t_R})$, drawn from the distribution $\hat{P}(a_1, \dots, a_R | \boldsymbol{X}_n - \{X_{t_1}, X_{t_2}, \dots, X_{t_R}\})$, where $\hat{P}$ is an estimation of the distribution P generating the data. The RB preserves part of the original sequence's temporal structure, while simultaneously exploiting the *full* sequence to determine replacements through an estimate of the conditional distribution $P(a_1, \dots, a_R | \boldsymbol{X}_n - \{X_{t_1}, X_{t_2}, \dots, X_{t_R}\})$. It is important to note that the estimated distribution is conditional both on the past and the future, and not just on the past; thus the bootstrap is not directly reduced to the problem of predicting the next symbol of a time series. In general, it is possible to construct examples where if the length of the *past* $\boldsymbol{X}_{<t}$ and the *future* $\boldsymbol{X}_{>t}$ tends to infinity, the entropy of the process conditioned on $\boldsymbol{X}_{<t}, \boldsymbol{X}_{>t}$ tends to zero; whereas the process itself has a non-zero entropy. This shows that replacements that take into account the structure of the *whole* sequence could replicate more accurately the unknown structure of the process.

INPUT: Seq. $\boldsymbol{X}_n$, replacements R, pattern size K_n
Compute counts $\nu_{\boldsymbol{X}_n}(a|\boldsymbol{v}_m)$, $a \in A$, $\boldsymbol{v}_m \in A^m$, $m \in \{0..K_n\}$
Set $\boldsymbol{z}_n^0 = \boldsymbol{X}_n$
for $r = 0, \dots, R$ **do**
  Draw the replacement point $t_r \sim \mathcal{U}([1, n])$
  Draw $b_{t_r} \sim \mathcal{R}_{\boldsymbol{X}_n}(\cdot|\boldsymbol{z}_{<t_r}^{r-1}; \boldsymbol{z}_{>t_r}^{r-1})$
  Set $\boldsymbol{z}_n^r = (\boldsymbol{z}_{<t_r}^{r-1}, b_{t_r}, \boldsymbol{z}_{>t_r}^{r-1})$
end for
OUTPUT: Synthetic sequence $\boldsymbol{b}_n = \boldsymbol{z}_n^R$

Fig. 1. Pseudo-code of the $\mathcal{R}$-Boot algorithm.

A critical aspect of this bootstrap scheme is to compute an accurate estimate of the conditional distribution of replacements. A direct implementation of the RB requires estimating the probability $P(a_1, \dots, a_R|\boldsymbol{X}_n - \{X_{t_1}, X_{t_2}, \dots, X_{t_R}\})$, which corresponds to the probability of a word $\boldsymbol{v} = (a_1, \dots, a_R)$ in R (random) locations $t_1, \dots, t_R$, conditioned on the rest of the sequence. Unfortunately, this requires estimating probabilities for an alphabet of size $|A|^R$, conditional on the rest of the sequence, which would rapidly become infeasible as R increases. For this reason, we propose to implement the RB through a sequential process based on the estimation of the one-symbol replacement probability $P(\cdot|\boldsymbol{X}_{<t}, \boldsymbol{X}_{>t})$. One way of doing it is by adapting a universal predictor. Universal predictors estimate conditional probabilities of the future outcomes given the past and they are asymptotically consistent for all stationary ergodic distributions.

Definition 1. *A measure ρ is called universal (or a universal predictor) if for any stationary and ergodic process P we have*

$$\frac{1}{n}\sum_{t=1}^{n}\mathbb{E}_{\boldsymbol{X}_{<t}}\Big[KL\big(P(\cdot|\boldsymbol{X}_{<t}); \rho(\cdot|\boldsymbol{X}_{<t})\big)\Big] \to 0. \tag{2}$$

Several predictors with this property are known. Here we use the universal measure $\mathcal{R}$ from [17] (see also [18]). While other measures can be used, $\mathcal{R}$ appears to be easier to implement efficiently than compression-based predictors (see e.g., [16, 19]); it is also not wasteful of data, unlike the Ornstein predictor [14]. In order to define the measure $\mathcal{R}$, we first introduce the finite-memory Krichevsky predictors [9]:

Definition 2. *For any $m \geq 0$, the Krichevsky predictor of order m estimates $P(X_t = a|\boldsymbol{X}_{<t})$ as*

$$\mathcal{K}^m(X_t = a|X_{t-m} = v_1, \dots, X_{t-1} = v_m) \tag{3}$$
$$= \begin{cases} \frac{\nu_{\boldsymbol{X}_n}(v_1,\dots,v_m,a)+\frac{1}{2}}{\sum_{c\in A}\nu_{\boldsymbol{X}_n}(v_1,\dots,v_m,c)+\frac{|A|}{2}}, & t > m, \\ \frac{1}{|A|}, & t \leq m. \end{cases}$$

Originally, in the Krichevsky predictor the frequencies are only counted up to the time t where the forecast is made; in the definition above, while the conditioning is only on the past, the counters $\nu_{\boldsymbol{X}_n}$ are computed based on both the past and the future. The original Krichevsky predictor is optimal (w.r.t. expected KL divergence)[9] for any fixed-length sequence for the set of (k-order) Markov processes. Finally, the measure $\mathcal{R}$ is defined as follows.

Definition 3 ([17])**.** *For any t, the measure $\mathcal{R}$ is defined as*

$$\mathcal{R}(X_1, \dots, X_n) = \sum_{m=0}^{\infty}\omega_{m+1}\mathcal{K}^m(X_1, \dots, X_n), \tag{4}$$

with weights $\omega_m = (\log(m+1))^{-1} - (\log(m+2))^{-1}$.

The measure $\mathcal{R}$ is a fully nonparametric estimator constructed directly from $\boldsymbol{X}_n$, it does not rely on any parametric assumptions and is proved to be universal in [17]. Here we propose the following method of using $\mathcal{R}$ to generate bootstrap sequences. Let $t \leq n$ be an arbitrary point in the original sequence. We replace the original symbol X_t with a new symbol b_t drawn from an estimate of the replacement distribution used to generate X_t, i.e., $P(\cdot|\boldsymbol{X}_{<t}, \boldsymbol{X}_{>t})$, computed using $\mathcal{R}$ as

$$\mathcal{R}_{\boldsymbol{X}_n}(X_t = a|\boldsymbol{X}_{<t}, \boldsymbol{X}_{>t}) = \frac{\mathcal{R}_{\boldsymbol{X}_n}(\boldsymbol{X}_{<t}; a; \boldsymbol{X}_{>t})}{\sum_{c\in A}\mathcal{R}_{\boldsymbol{X}_n}(\boldsymbol{X}_{<t}; c; \boldsymbol{X}_{>t})}. \tag{5}$$

Once the replacement probability is calculated, $\mathcal{R}$-Boot substitutes X_t in the original sequence with b_t drawn from $\mathcal{R}_{\boldsymbol{X}_n}(\cdot|\boldsymbol{X}_{<t}, \boldsymbol{X}_{>t})$, thus obtaining the new sequence $\boldsymbol{z}_n^1 = (\boldsymbol{X}_{<t}, b_t, \boldsymbol{X}_{>t})$. Here $\mathcal{R}_{\boldsymbol{X}_n}$ refers to the measure $\mathcal{R}$ with frequency counts $\nu(\cdot)$ in (2) computed only once from the sequence $\boldsymbol{X}_n$. This indicates that once X_t is replaced by b_t and $\boldsymbol{z}_n^1$ is obtained, the counts are not recomputed on $\boldsymbol{z}_n^1$ and $\mathcal{R}_{\boldsymbol{X}_n}$ is still used as an estimate of replacement distribution for all the following replacements. This process is iterated R times; such that, at each step r, a random point t_r is chosen and the new symbol b_{t_r} is drawn from $\mathcal{R}_{\boldsymbol{X}_n}(\cdot|\boldsymbol{z}_{<t_r}^{r-1}, \boldsymbol{z}_{>t_r}^{r-1})$. Finally, the sequence $\boldsymbol{b}_n = \boldsymbol{z}_n^R$ is returned. The pseudo-code of $\mathcal{R}$-Boot is reported in Fig. 1. Although the measure $\mathcal{R}$ combines an infinite number of predictors, it can be computed in polynomial time. First of all, note that replacing infinity with any K_n that increases to infinity with n does not affect the asymptotic convergence properties of the measure. A practically and theoretically meaningful choice for K_n is $O(\log n)$. Indeed the frequency estimates in $\mathcal{K}^m$ for $m \gg \log n$ are not consistent, so they only add to the noise of the estimate. This meaningful choice of K_n is also efficiently computable. If K_n is of order $O(\log n)$, then the computational complexity of $\mathcal{R}$-Boot to generate a bootstrap sequence with R replacements is $O((n + R\log n)\log^2(n)|A|^2)$. More details about Eq. 5 and the implementation of $\mathcal{R}$-Boot are reported in [21].

The parameter R retains an equivalent expected impact on performance as in the general RB. For small values, most of the original temporal structure of the sequence is preserved, but bootstrap sequences may not reproduce enough variability from the original process. For large values, $\mathcal{R}$-Boot has higher variance, but the error in estimating the replacement distribution may introduce significant errors in the bootstrap sequences. In general, the incremental process of $\mathcal{R}$-Boot requires large R to compensate for the iterative (versus simultaneous) nature of replacements. Since replacement points are random, the same location may be repeatedly selected through the execution of $\mathcal{R}$-Boot. Furthermore, at step r, $\mathcal{R}$-Boot uses the current sequence, $\boldsymbol{z}_n^{r-1}$, to define the conditional probability $\mathcal{R}_{\boldsymbol{X}_n}(\cdot|\boldsymbol{z}_n^{r-1}, \boldsymbol{z}_n^{r-1})$, for X_{t_r}; thus

replacements in one location (i.e., changes to the symbols $X_{t_1}, \dots, X_{t_{r-1}}$ in the original sequence) may later trigger changes in other locations. Thus, $\mathcal{R}$-Boot is an incremental approximation to the simultaneous replacement of R symbols in RB. The results below are limited to this approximation. However, introducing another parameter, it is possible to define an intermediate approach where contiguous *blocks* of size d are incrementally replaced. At location t, instead of replacing only symbol X_t, we may replace an entire block $\boldsymbol{X}_{t:t+d}$ with a word $\boldsymbol{v} = (a_1, \dots, a_d)$ drawn from an estimation of $P(a_1, \dots, a_d | \boldsymbol{X}_{<t}, \boldsymbol{X}_{>t+d})$ and repeat over multiple iterations. Although this requires estimating probabilities over larger alphabets ($|A|^d$), working on contiguous blocks may improve the one-symbol replacements of $\mathcal{R}$-Boot.

IV. Theoretical Guarantees

A desirable property for a bootstrap method is to produce an accurate estimate of the distribution over the whole sequence of length n, $P(X_1, \dots, X_n)$, given just one data point (i.e., a sequence $\boldsymbol{X}_n$). Unfortunately, this is not possible in the case of stationary-ergodic processes. This is in stark contrast with the classes of processes considered in prior works, where estimating $P(X_1, \dots, X_m)$ for a critical $m \ll n$ (e.g., $m = k + 1$ in the case of k-order Markov processes) results in a sufficiently accurate estimate of $P(X_1, \dots, X_n)$. While considering the general case of stationary ergodic distributions significantly increases the applicability of the bootstrap, it prevents us from providing theoretical guarantees for the bootstrap estimate of $P(X_1, \dots, X_n)$. Moreover, in this setting it is provably impossible to establish any nontrivial rates of convergence. As a result we focus on asymptotic consistency of the individual replacement step in $\mathcal{R}$-Boot and we show that if the sequence length of at least one side of the replacement is sufficiently long, then on average the probability distribution for the inserted symbol converges to the distribution of the symbol in that position given the past and the future. Moreover, when the size of the sequence both before and after the replacement goes to infinity, the probability distribution over the inserted symbol approaches the double-sided entropy rate.

We introduce additional notation. A stationary distribution over one-way infinite sequences $X_1, X_2, \dots$ can be uniquely extended to a distribution over two-way infinite sequences $\dots, X_{-1}, X_0, X_1, \dots$. We assume this extension whenever necessary. For stationary processes, the k-order entropy is $h_k(P) = \mathbb{E}_{\boldsymbol{X}_{-k:-1}}[h(X_0 | \boldsymbol{X}_{-k:-1})]$. Similar to the *entropy rate*, one can define the two-sided entropy $h_{k,m} = \mathbb{E}_{\boldsymbol{X}_{-k:-1}, \boldsymbol{X}_{1:m}} h(X_0 | \boldsymbol{X}_{-k:-1}, \boldsymbol{X}_{1:m})$, which is non-decreasing with k and m, and whose limit $\lim_{k,m\to\infty} h_{k,m}$ we denote $h_{\infty\times\infty}$, where $\lim_{k,m\to\infty}$ is an abbreviation for "for every pair of increasing sequences $(k_l)_{l\in\mathcal{N}}, (m_l)_{l\in\mathcal{N}}$, $\lim_{l\to\infty}$." Obviously $h_{\infty\times\infty} \leq h_\infty(P)$ and it is easy to construct examples when the inequality is strict. We also denote the KL divergence between process measures P and $\mathcal{R}$ as $\delta(X_t | \boldsymbol{X}_{<t}) = \mathrm{KL}(P(X_t | \boldsymbol{X}_{<t}); \mathcal{R}(X_t | \boldsymbol{X}_{<t}))$, where the capital letter indicates that X_t and $\boldsymbol{X}_{<t}$ are random variables. Finally, in $\mathbb{E}[\delta(X_t | \boldsymbol{X}_{<t})]$, the expectation is over $\boldsymbol{X}_{<t}$. The proof of the following theorem is reported in [21] and it builds on the consistency of the $\mathcal{R}$ measure as a predictor [17].

Theorem 1. *For all $m \in \mathcal{N}$ we have*

$$(i) \lim_{N\to\infty} \mathbb{E}\Big[\sum_{n=m}^{N} \frac{\delta(X_{n-m+1} | \boldsymbol{X}_{0:n-m}, \boldsymbol{X}_{n-m+2:n})}{N-m} \Big] = 0,$$

$$(ii) \lim_{N\to\infty} \mathbb{E} \sum_{n=m}^{N} \frac{\delta(X_{-n+m-1} | \boldsymbol{X}_{-n:-n+m}, \boldsymbol{X}_{-n+m-2:0})}{N-m} = 0,$$

$$(iii) \lim_{m\to\infty} \lim_{N\to\infty} -\mathbb{E} \sum_{n=m}^{N} \frac{\log \mathcal{R}(X_{n-m+1} | \boldsymbol{X}_{0:n-m}, \boldsymbol{X}_{n-m+2:n})}{N-m} = h_{\infty\times\infty},$$

$$(iv) \lim_{N\to\infty} \lim_{m\to\infty} -\mathbb{E} \sum_{n=m}^{N} \frac{\log \mathcal{R}(X_{n-m+1} | \boldsymbol{X}_{0:n-m}, \boldsymbol{X}_{n-m+2:n})}{N-m} = h_{\infty\times\infty}.$$

One could wish for a stronger consistency statement than those established in Theorem 1. For example, a statement we would like to demonstrate is the following $\lim_{m,n\to\infty} -\mathbb{E} \log \mathcal{R}(X_0 | \boldsymbol{X}_{-n:-1}, \boldsymbol{X}_{1:m}) = h_{\infty\times\infty}$. There are two differences with respect to (iii) and (iv): first, the limits are taken simultaneously, and, second, there is no averaging over time. We conjecture that this statement is possible to prove. The reason for this conjecture is that it is possible to prove this for some other predictors (other than $\mathcal{R}$). Namely, the consistency proof for the Ornstein predictor [14], as well as those of its improvements and generalizations in [12] can be extended to our case. These predictors are constructed by taking frequencies of events based on growing segments of the past; however, unlike $\mathcal{R}$, they are very wasteful of data, which is perhaps the reason they have never been used (to the best of our knowledge) beyond theoretical analysis. Another possible extension is to prove "almost sure" analogues of the statements of the theorem. Note that the a.s. consistency holds for $\mathcal{R}$ as a predictor; however, in this case time-averaging is essential, as is also established in [17]. Finally, notice that for standard bootstrap methods it is often possible to derive asymptotic convergence rates based on the central limit theorem and Edgeworth expansions under relatively strong assumptions about the generative process (e.g., exponentially mixing). However, the class of all stationary ergodic processes is so large that this type of analysis is provably impossible; further, finite-time error bounds are also impossible for this setting, since the convergence rates of any non-trivial estimate can be arbitrary slow [22]. Thus, a direct theoretical comparison between $\mathcal{R}$-Boot and other bootstrap methods is not possible and we rely on the empirical investigation to evaluate their differences.

V. Empirical Evaluation

An empirical comparison of $\mathcal{R}$-Boot is presented against the circular block bootstrap (CBB) and the Markov bootstrap (MB) on simulated data for the estimation of the maximum drawdown (MDD) statistic, a challenging statistic used in optimization, finance and economics to characterize the "adverse excursion" risk. Synthetic sequences are simulated using a real-valued mean-reverting fractional Brownian motion (FBM) process P with mean $\mu = 0$, standard deviation $\sigma = 1$ and Hurst exponent $H = 0.25$. We sample 10^4 sequences of length $n = 1001$ from P and difference them into stationary increments. From this real-valued sequence

we construct a binary sequence by thresholding at 0: $X_t = -1$ for negative increments and $X_t = 1$ for positive. In practice, adaptive quantization schemes are often used, but these could introduce confounding effects in the results so we do not use them. From $\boldsymbol{X}_n$, the corresponding *cumulative sequence* $\boldsymbol{y}_n$ ($y_t = \sum_{s=1}^{t} X_t$) is computed (representing, e.g., a price sequence) and the MDD is defined as $\hat{f}(\boldsymbol{X}_n) = \max_{t=1,\dots,n} \left(\max_{s=1,\dots,t} y_s - y_t \right)$, while the MDD θ_n of the P is obtained by averaging the raw estimates $\hat{\theta}_n = \hat{f}(\boldsymbol{X}_n)$ over 10^6 sequences. The MDD is a complex extrema statistic for which reliable estimates are not available yet not even for i.i.d. data [6]. Note that we are interested in estimating of the statistic itself, which is different from the typical applications of bootstrap where estimating the statistic is easy and bootstrap is used to construct confidence intervals.

As θ_n is an increasing function of n, we normalize it by its rate of growth n and we compute the estimation error as $\text{MSE}(\mathcal{B}) = \mathbb{E}\left[\frac{(\theta_n - \tilde{\theta}_n^{\mathcal{B}})^2}{n}\right]$, with $\tilde{\theta}_n^{\mathcal{B}} = \frac{1}{B}\sum_j \hat{f}(\boldsymbol{b}_n^j)$ and $\boldsymbol{b}_n^j = \mathcal{B}(\boldsymbol{X}_n)$. We run $B = 10^3$ bootstraps for each method. For CBB, theoretical guidelines are provided in the literature (see e.g., [15]) suggesting the block width should be $O(n^{\frac{1}{3}})$, while for MB any tuning of the order would require knowledge about the process. In the following we report results for the best parameter choice (in *hindsight*) for each value of n separately, where CBB is optimized in the range of block width $[1, 20]$ (thus always including the theoretical value up to $2n^{\frac{1}{3}}$)[1] and the order of MB is optimized in $[1, 20]$. Notice that such tuning is not possible in practice since only one sequence is available and the true statistics of the process are obviously unknown. These *best* parameters for CBB and MB are intended to upper bound the performance that can be achieved by these methods. Furthermore, the best order for MB also represents an upper bound for any other method using a mixture of Markov models with different orders, such as a direct use of the $\mathcal{R}$ measure in Def. 3 to generate sequences sampling from $P(X_t|\boldsymbol{X}_{<t})$ or the sieve bootstrap [2] that automatically selects the order.

Results. $\mathcal{R}$-Boot, CBB and MB are compared on FBM data in Fig. 2. CBB is run with its best block width, while MB is run with its best model size. $\mathcal{R}$-Boot is run with $K_n = \lfloor 1.5 \log(n) \rfloor$ and two values for R, $0.75n$ and $3.5n$. $\mathcal{R}$-Boot $R = 0.75n$ achieves better performance than CBB and raw estimator $\hat{\theta}_n = \hat{f}(\boldsymbol{X}_n)$ (single sequence), demonstrating that approximated replacement distributions are accurate enough to guarantee bootstraps which resemble the original process. With this value of R RB still does not outperform MB with the best (in hindsight) order. This is perhaps because $R = 0.75n$ corresponds to approximately 30% replacements to the original sequence which is not enough to generate sufficiently different sequences. In comparison, MB generates bootstrapped sequences from scratch. With $R = 3.5n$ the replacements increase to approximately 140% and $\mathcal{R}$-Boot significantly outperforms MB for all values of n. This demonstrates that, while sub-optimal value of the parameters result large difference of performance of both CBB and MB, $\mathcal{R}$-Boot is quite robust, meaning with values of R in a wide range it consistently outperforms CBB and MB by a significant margin.

[1]The optimal constant in $O(n^{1/3})$ depends on the (unknown) autocovariance and spectral density functions of the process [15].

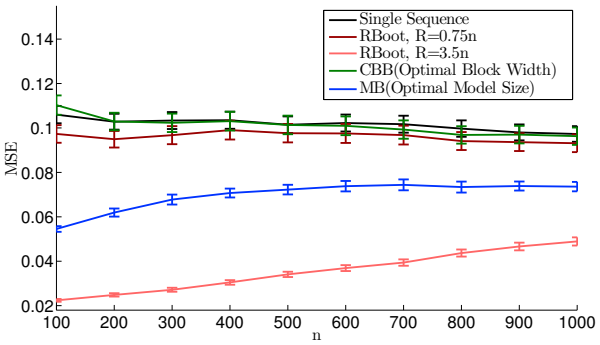

Fig. 2. MDD MSE (with standard errors) on FBM sequences.

REFERENCES

[1] K. Bassler, J. McCauley, and G. Gunaratne. Nonstationary increments, scaling distributions, and variable diffusion processes in financial markets. *Proc. National Academy of Sciences*, 104(44):17287–17290, 2007.

[2] Peter Bühlmann et al. Sieve bootstrap for time series. *Bernoulli*, 3(2):123–148, 1997.

[3] Bradley Efron. Bootstrap methods: Another look at the Jackknife. *The Annals of Statistics*, 1979.

[4] R. Gray. *Probability, Random Processes, and Ergodic Properties*. Springer Verlag, 1988.

[5] Peter Hall. *The bootstrap and Edgeworth expansion*. Springer, 1992.

[6] Joel L Horowitz. The bootstrap. *Handbook of econometrics*, 5:3159–3228, 2001.

[7] Joel L Horowitz. Bootstrap methods for markov processes. *Econometrica*, 71(4):1049–1082, 2003.

[8] J.-P. Kreiss and S. Lahiri. Bootstrap methods for time series. *Handbook of Statistics: Time Series Analysis: Methods and Applications*, 30, 2012.

[9] R. Krichevsky. A relation between the plausibility of information about a source and encoding redundancy. *Probl.Inf.Trans.*, 4(3):48–57, 1968.

[10] RJ Kulperger and BLS Prakasa Rao. Bootstrapping a finite state markov chain. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 178–191, 1989.

[11] Soumendra N Lahiri. Theoretical comparisons of block bootstrap methods. *Annals of Statistics*, pages 386–404, 1999.

[12] G. Morvai, S. Yakowitz, and L. Györfi. Nonparametric inference for ergodic, stationary time series. *Ann. Stat.*, 24(1):370–379, 1996.

[13] Daniel J Nordman et al. A note on the stationary bootstrap's variance. *The Annals of Statistics*, 37(1):359–370, 2009.

[14] Donald S Ornstein. Guessing the next output of a stationary process. *Israel Journal of Mathematics*, 30(3):292–296, 1978.

[15] Dimitris N Politis and Halbert White. Automatic block-length selection for the dependent bootstrap. *Econometric Reviews*, 23(1):53–70, 2004.

[16] B. Ryabko and V. Monarev. Experimental investigation of forecasting methods based on data compression algorithms. *Problems of Information Transmission*, 41(1):65–69, 2005.

[17] Boris Ryabko. Prediction of random sequences and universal coding. *Problems of Information Transmission*, 24:87–96, 1988.

[18] Boris Ryabko. Applications of Kolmogorov complexity and universal codes to nonparametric estimation of characteristics of time series. *Fundamenta Informaticae*, 83(06):1–20, 2008.

[19] Boris Ryabko. Compression-based methods for nonparametric prediction and estimation of some characteristics of time series. *IEEE Transactions on Information Theory*, 55:4309–4315, 2009.

[20] Boris Ryabko. Applications of universal source coding to statistical analysis of time series. *Selected Topics in Information and Coding Theory, World Scientific Publishing*, pages 289–338, 2010.

[21] A. Sani, A. Lazaric, and D. Ryabko. The replacement bootstrap for dependent data. Technical Report hal-01144547, INRIA, 2015.

[22] P. Shields. *The Ergodic Theory of Discrete Sample Paths*. AMS, 1996.

[23] Kesar Singh. On the asymptotic accuracy of Efron's bootstrap. *The Annals of Statistics*, pages 1187–1195, 1981.

APPENDIX A
COMPUTATION OF THE REPLACEMENT DISTRIBUTION

In this section we report more details on how to actually compute the Replacement Bootstrap in Eq. 5 and we discuss its overall computational complexity.

Dealing with numerical issues. As illustrated in Eq. 5, the replacement probability that a (new) sequence has a symbol $a \in A$ in position t given that the rest of the sequence is as in the original $\boldsymbol{X}_n$ is calculated by estimating $P(X_t = a|\boldsymbol{X}_{<t}, \boldsymbol{X}_{>t})$ as

$$\mathcal{R}_{\boldsymbol{X}_n}(X_t = a|\boldsymbol{X}_{<t}, \boldsymbol{X}_{>t}) = \frac{\mathcal{R}_{\boldsymbol{X}_n}(\boldsymbol{X}_{<t}; a; \boldsymbol{X}_{>t})}{\sum_{c\in A} \mathcal{R}_{\boldsymbol{X}_n}(\boldsymbol{X}_{<t}; c; \boldsymbol{X}_{>t})} = \frac{\sum_{i=0}^{\infty} \omega_i \mathcal{K}^i_{\boldsymbol{X}_n}(\boldsymbol{X}_{<t}; a; \boldsymbol{X}_{>t})}{\sum_{c\in A}\sum_{j=0}^{\infty} \omega_j \mathcal{K}^j_{\boldsymbol{X}_n}(\boldsymbol{X}_{<t}; c; \boldsymbol{X}_{>t})}$$
$$= \sum_{i=0}^{\infty} \frac{\omega_i \mathcal{K}^i_{\boldsymbol{X}_n}(\boldsymbol{X}_{<t}; a; \boldsymbol{X}_{>t})}{\sum_{c\in A}\sum_{j=0}^{\infty} w_j \mathcal{K}^j_{\boldsymbol{X}_n}(\boldsymbol{X}_{<t}; c; \boldsymbol{X}_{>t})} = \sum_{i=0}^{\infty} \left[\sum_{c\in A}\sum_{j=0}^{\infty} \frac{\omega_j}{\omega_i} \frac{\mathcal{K}^j_{\boldsymbol{X}_n}(\boldsymbol{X}_{<t}; c; \boldsymbol{X}_{>t})}{\mathcal{K}^i_{\boldsymbol{X}_n}(\boldsymbol{X}_{<t}; a; \boldsymbol{X}_{>t})}\right]^{-1}. \quad (6)$$

Although the previous expression could be computed directly by using the definition of the Krichevsky predictors, the values returned by $\mathcal{K}^j_{\boldsymbol{X}_n}(\boldsymbol{X}_{<t}; c; \boldsymbol{X}_{>t})$ and $\mathcal{K}^i_{\boldsymbol{X}_n}(\boldsymbol{X}_{<t}; a; \boldsymbol{X}_{>t})$ would rapidly fall below the precision threshold as n increases, thus introducing significant numerical approximations. Therefore, we need to further elaborate the previous expression. For any $i, j \neq 0$, let $\boldsymbol{Y}^c = \{\boldsymbol{X}_{<t}; c; \boldsymbol{X}_{>t}\}$, $\boldsymbol{Y}^a = \{\boldsymbol{X}_{<t}; a; \boldsymbol{X}_{>t}\}$ be exactly as the original sequence with only the symbol at position t replaced by symbol a and $c \in A$ respectively, and let $t'_{i,j} = t + \max\{i, j\}$. In the following we exploit the fact that, conditionally on the frequency counts, the Krichevsky predictors of order m only uses the m symbols before to predict the current symbol, so we obtain[2]

$$\begin{aligned}
\frac{\mathcal{K}^j(\boldsymbol{X}_{<t}; c; \boldsymbol{X}_{>t})}{\mathcal{K}^i(\boldsymbol{X}_{<t}; a; \boldsymbol{X}_{>t})} &= \prod_{s=1}^{n} \frac{\mathcal{K}^j(\boldsymbol{Y}^c_s|\boldsymbol{Y}^c_{<s})}{\mathcal{K}^i(\boldsymbol{Y}^a_s|\boldsymbol{Y}^a_{<s})} \\
&= \prod_{s=1}^{t-1} \frac{\mathcal{K}^j(Y^c_s|\boldsymbol{Y}^c_{<s})}{\mathcal{K}^i(Y^a_s|\boldsymbol{Y}^a_{<s})} \prod_{s=t}^{n} \frac{\mathcal{K}^j(Y^c_s|\boldsymbol{Y}^c_{<s})}{\mathcal{K}^i(Y^a_s|\boldsymbol{Y}^a_{<s})} \\
&= \prod_{s=1}^{t-1} \frac{\mathcal{K}^j(X_s|\boldsymbol{X}_{<s})}{\mathcal{K}^i(X_s|\boldsymbol{X}_{<s})} \prod_{s=t}^{n} \frac{\mathcal{K}^j(Y^c_s|\boldsymbol{Y}^c_{<s})}{\mathcal{K}^i(Y^a_s|\boldsymbol{Y}^a_{<s})} \\
&= \prod_{s=1}^{t-1} \frac{\mathcal{K}^j(X_s|X_{s-j}, \ldots, X_{s-1})}{\mathcal{K}^i(X_s|X_{s-i}, \ldots, X_{s-1})} \prod_{s=t}^{n} \frac{\mathcal{K}^j(Y^c_s|Y^c_{s-j}, \ldots, Y^c_{s-1})}{\mathcal{K}^i(Y^a_s|Y^a_{s-i}, \ldots, Y^a_{s-1})} \\
&= \prod_{s=1}^{t-1} \frac{\mathcal{K}^j(X_s|X_{s-j}, \ldots, X_{s-1})}{\mathcal{K}^i(X_s|X_{s-i}, \ldots, X_{s-1})} \prod_{s=t}^{t'_{i,j}} \frac{\mathcal{K}^j(Y^c_s|Y^c_{s-j}, \ldots, Y^c_{s-1})}{\mathcal{K}^i(Y^a_s|Y^a_{s-i}, \ldots, Y^a_{s-1})} \prod_{s=t'_{i,j}+1}^{n} \frac{\mathcal{K}^j(Y^c_s|Y^c_{s-j}, \ldots, Y^c_{s-1})}{\mathcal{K}^i(Y^a_s|Y^a_{s-i}, \ldots, Y^a_{s-1})} \\
&= \prod_{s=1}^{t-1} \frac{\mathcal{K}^j(X_s|X_{s-j}, \ldots, X_{s-1})}{\mathcal{K}^i(X_s|X_{s-i}, \ldots, X_{s-1})} \prod_{s=t}^{t'_{i,j}} \frac{\mathcal{K}^j(Y^c_s|Y^c_{s-j}, \ldots, Y^c_{s-1})}{\mathcal{K}^i(Y^a_s|Y^a_{s-i}, \ldots, Y^a_{s-1})} \prod_{s=t'_{i,j}+1}^{n} \frac{\mathcal{K}^j(X_s|X_{s-j}, \ldots, X_{s-1})}{\mathcal{K}^i(X_s|X_{s-i}, \ldots, X_{s-1})} \\
&= \pi_{i,j,1:t-1} \prod_{s=t}^{t'_{i,j}} \frac{\mathcal{K}^j(Y^c_s|Y^c_{s-j}, \ldots, Y^c_{s-1})}{\mathcal{K}^i(Y^a_s|Y^a_{s-i}, \ldots, Y^a_{s-1})} \pi_{i,j,t'_{i,j}+1:n},
\end{aligned}$$

where for any t_1 and t_2, we define

$$\pi_{i,j,t_1:t_2} = \prod_{s=t_1}^{t_2} \frac{\mathcal{K}^j(X_s|X_{s-j}, \ldots, X_{s-1})}{\mathcal{K}^i(X_s|X_{s-i}, \ldots, X_{s-1})}.$$

Notice that the previous expression is still valid for either $i = 0$ or $j = 0$ with the convention that $\mathcal{K}^0(Y^b_s|\boldsymbol{Y}^b_{<s}) = \mathcal{K}^0(Y^b_s)$. Thus, we obtain that Eq. 6 can be conveniently rewritten as

$$\mathcal{R}_{\boldsymbol{X}_n}(X_t = a|\boldsymbol{X}_{<t}, \boldsymbol{X}_{>t}) = \sum_{i=0}^{K_n} \left[\sum_{j=0}^{K_n} \frac{\omega_j}{\omega_i} \pi_{i,j,1:t-1} \pi_{i,j,t'_{i,j}+1:n} \left(\sum_{c\in A} \prod_{s=t}^{t'_{i,j}} \frac{\mathcal{K}^j_{\boldsymbol{X}_n}(Y^c_s|Y^c_{s-j}, \ldots, Y^c_{s-1})}{\mathcal{K}^i_{\boldsymbol{X}_n}(Y^a_s|Y^a_{s-i}, \ldots, Y^a_{s-1})}\right)\right]^{-1}. \quad (7)$$

[2]In the following we drop the dependency on $\boldsymbol{X}_n$.

We are now left with computing the expressions $\mathcal{K}^i(Y_s|Y_{s-i},\dots,Y_{s-1})$ (for $\boldsymbol{Y}=\boldsymbol{Y}^a$ and $\boldsymbol{Y}=\boldsymbol{Y}^c$) over the sequence of values dependent on the replacement at t. As an illustrative case, let $Y_s=a$ and $Y_{s-i},\dots,Y_{s-1}=\boldsymbol{V}^i$, where $\boldsymbol{V}^i$ is some word of length i, then

$$\mathcal{K}^i(Y_s|Y_{s-i},\dots,Y_{s-1})=\frac{\nu_{\boldsymbol{X}_n}(a|\boldsymbol{V}^i)+1/2}{\sum_{c\in A}\nu_{\boldsymbol{X}_n}(c|\boldsymbol{V}^i)+|A|/2}=\frac{\nu_{\boldsymbol{X}_n}(a|\boldsymbol{V}^i)+1/2}{\nu_{\boldsymbol{X}_n}(\boldsymbol{V}^i)+|A|/2}. \tag{8}$$

Each term in the previous expression is computable with no major numerical error, since it stays in the range 2^{-i} (and i should never exceed K_n), and so the ratios between $\mathcal{K}$ values in Eq. 7 for each s and their products continue to be well conditioned. Even in the case that for some pairs i and j the numbers become too small, they are just added in the summation over c and j in Eq. 7, so this does not pose any numerical approximation problem anymore.

Implementation details for Eq. 7 over successive replacements. In order to efficiently implement Eq. 7 over multiple replacements, it is useful to store a series of intermediate values that will allow a small incremental computational cost from one replacement to the other. Let K be a matrix of dimension $(K_n+1)\times n$ such that for any $i=0,\dots,K_n$ and $s=1,\dots,n$,

$$K_{i,s}=\mathcal{K}^i(X_s|X_{s-i},\dots,X_{s-1}).$$

We also define the multi-dimensional matrix R of dimensions $(K_n+1)\times(K_n+1)\times n$ such that for any $i,j=0,\dots,K_n$ and $s=1,\dots,n$,

$$R_{i,j,s}=\frac{K_{j,s}}{K_{i,s}}.$$

Then we compute π_{prev} and π_{post} as

$$\pi_{i,j,\text{prev}}=\prod_{s=1}^{t-1}R_{i,j,s},\quad \pi_{i,j,\text{post}}=\prod_{s=t'_{i,j}+1}^{n}R_{i,j,s}.$$

Similarly we define the multi-dimensional matrix K' of dimensions $(K_n+1)\times|A|\times(t_{\text{end}}-t)$ where $t_{\text{end}}=\min\{t'_{i,j},n\}$ and

$$K_{i,a,s}=\mathcal{K}^i(y^a_s|y^a_{s-i},\dots,y^a_{s-1}).$$

Thus the matrix R' of dimensions $(K_n+1)\times(K_n+1)\times|A|\times|A|\times(t_{\text{end}}-t)$ is

$$R'_{i,j,a,b,s}=\frac{K_{j,b,s}}{K_{i,a,s}}.$$

Once these structures are computed, the final probability of replacement of a letter a in position t is computed as

$$\mathcal{R}(X_t=a|\boldsymbol{X}_{<t},\boldsymbol{X}_{>t})=\sum_{i=0}^{K_n}\left[\sum_{j=0}^{K_n}\frac{w_j}{w_i}\pi_{i,j,\text{prev}}\pi_{i,j,\text{post}}\left(\sum_{b\in A}\prod_{s=t}^{t'_{i,j}}R'_{i,j,a,b,s}\right)\right]^{-1}. \tag{9}$$

As a result, the computational complexity for the first symbol replacement requires initializing the counts ν and coefficients π after a full scan of the entire sequence $\boldsymbol{X}_n$ and the overall complexity is $O(nK_n^2A^2)$. Nonetheless, it is possible to dramatically reduce the computational complexity of the other steps.

We use $l=1,\dots,L$ as an index for the replacement points, $t(l)$ as the time index of the l-th replacement point, $\boldsymbol{Z}_n(l)$ as the sequence obtained after l replacement points (i.e., $\boldsymbol{Z}_n(0)=\boldsymbol{X}_n$), and $\pi_{i,j,\text{prev}}(l)$ and $\pi_{i,j,\text{post}}(l)$ as the products of probabilities computed on the l-th sequence. Furthermore, we assume that the position of the replacement points is known in advance (i.e., $t(l)$ is chosen for all $l=1,\dots,L$ at the beginning). Beside $\pi_{i,j,\text{prev}}(0)$ and $\pi_{i,j,\text{post}}(0)$ at the first iteration $l=0$, we also compute the additional structures

$$\pi_{i,j,\text{prod}}(0)=\pi_{i,j,\text{prev}}(0)\times\pi_{i,j,\text{post}}(0),\quad \pi_{i,j,\text{next}}(0)=\prod_{s=t(1)}^{t'_{i,j}(1)}R_{i,j,s}(0).$$

After the first replacement, only the coefficients $\pi_{i,j,t_1:t_2}$ directly affected by the replacement of X_t by b_t need to be recomputed. In particular, we notice that for any $i=1,\dots,K_n$ and for any s such that $s<t(l)$ or $s>t(l)+i$ then $K_{i,s}(l+1)=K_{i,s}(l)$. Thus for any i we only need to recompute $K_{i,s}$ for $t(l)\le s\le t(l)+i$. As a result, once the replacement is done and $\boldsymbol{Z}^1_n$ is computed, the $R_{i,j,s}$ affected by the change are recomputed and at the beginning of the new iteration we use them to compute

$$\pi_{i,j,\text{cur}}(1)=\prod_{s=t(0)}^{t'_{i,j}(0)}R_{i,j,s}(1),$$

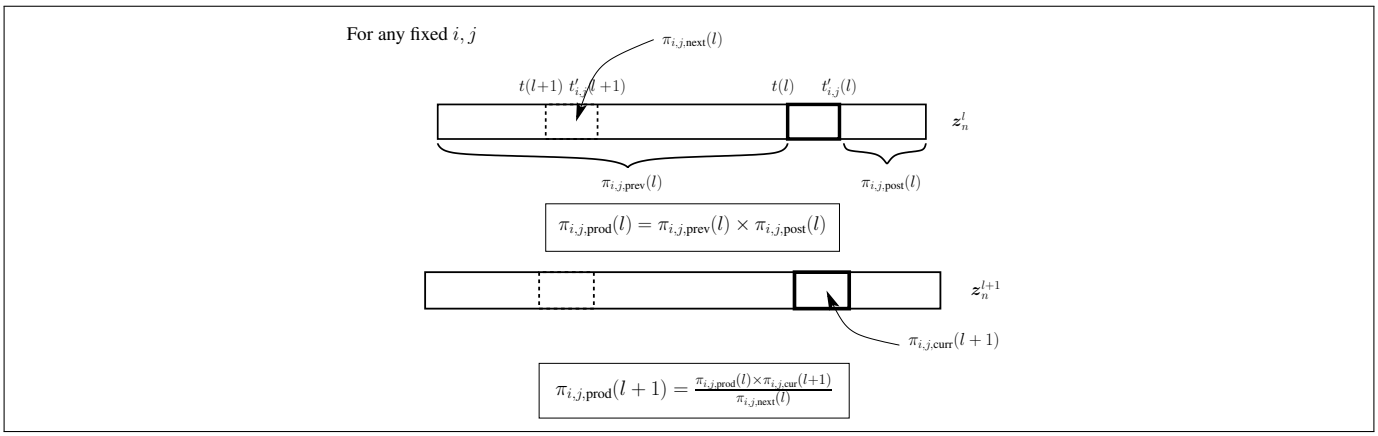

Fig. 3. Computation of other replacement points (i.e., $l \geq 1$).

and this allows to compute the π_{prod} terms of the next iteration as

$$\pi_{i,j,\text{prod}}(1) = \frac{\pi_{i,j,\text{prod}}(0)\pi_{i,j,\text{cur}}(1)}{\pi_{i,j,\text{next}}(0)},$$

In general, after the first iteration, at any iteration $l \geq 1$, the computation of the previous elements can be done iteratively as (see also Fig. 3 for an illustration of these quantities)

$$\pi_{i,j,\text{prod}}(l) = \frac{\pi_{i,j,\text{prod}}(l-1)\pi_{i,j,\text{cur}}(l)}{\pi_{i,j,\text{next}}(l-1)},$$

where

$$\pi_{i,j,\text{next}}(l) = \prod_{s=t(l+1)}^{t'_{i,j}(l+1)} R_{i,j,s}(l), \;\; \pi_{i,j,\text{cur}}(l) = \prod_{s=t(l-1)}^{t'_{i,j}(l-1)} R_{i,j,s}(l).$$

As a result, while the complexity of recomputing the coefficients $K_{i,s}$ for $t(l) \leq s \leq t(l)+i$, is limited to $O(K_n^2)$ computations, the overall computation for the replacement at each $l > 0$ is of order $O(K_n^3 A^2)$.

APPENDIX B
PROOF OF THEOREM 1.

Proof. For the first statement, first note that,

$$\mathbb{E}\left[\delta(X_{n-m+1}|\boldsymbol{X}_{0:n-m}, \boldsymbol{X}_{n-m+2:n})\right] \leq \mathbb{E}\left[\delta(\boldsymbol{X}_{n-m+1:n}|\boldsymbol{X}_{0:n-m}) - \delta(\boldsymbol{X}_{n-m+2:n}|\boldsymbol{X}_{0:n-m})\right] \leq \mathbb{E}\left[\delta(\boldsymbol{X}_{n-m+1:n}|\boldsymbol{X}_{0:n-m})\right]$$

where the inequality follows from the fact that the KL divergence is non-negative. The statement follows from the consistency of $\mathcal{R}$ as a predictor: for every $m \in \mathcal{N}$ we have (see [17, 18]).

$$\lim_{N\to\infty} \frac{1}{N}\mathbb{E}\sum_{n=m}^{N} \delta(\boldsymbol{X}_{n-m+1:n}|\boldsymbol{X}_{0:n-m}) \to 0. \tag{10}$$

The proof of the second statement is analogous to that of the first, except that we need the consistency of $\mathcal{R}$ as a predictor "backwards," that is, when the sequence extends to the past rather than to the future. The proof of this consistency is analogous to the proof of the usual (forward) consistency (10). Since it is important for exposing some further ideas, we give it here. We consider the case $m = 1$. The general case follows by replacing $Y_i = X_{i:i+m}$ for every i and noting that if the distribution of X_i is stationary ergodic then so is the distribution of Y_i. We have

$$\begin{aligned}\mathbb{E}\sum_{n=0}^{N} \delta(X_{-n}|\boldsymbol{X}_{0:-n+1}) &= -\sum_{n=0}^{N} \mathbb{E}_{\boldsymbol{X}_{0:-n+1}}\mathbb{E}_{X_{-n}} \log\left(\frac{\mathcal{R}(X_{-n}|\boldsymbol{X}_{0:-n+1})}{P(X_{-n}|\boldsymbol{X}_{0:-n+1})}\right) \\ &= -\mathbb{E}_{\boldsymbol{X}_{0:-N}} \log\left(\frac{\mathcal{R}(\boldsymbol{X}_{-N:0})}{P(\boldsymbol{X}_{-N:0})}\right) = -\mathbb{E}\log\mathcal{R}(\boldsymbol{X}_{-N:0}) + \mathbb{E}\log P(\boldsymbol{X}_{-N:0}).\end{aligned}$$

Since $\lim -\frac{1}{N}\mathbb{E}\log P(\boldsymbol{X}_{-N:0}) = h_\infty(P)$, to establish the consistency statement it is enough to show that $\lim -\frac{1}{N}\mathbb{E}\log \mathcal{R}(\boldsymbol{X}_{-N:0}) = h_\infty(P)$. For every $k \in \mathcal{N}$

$$\begin{aligned}-\mathbb{E}\log \mathcal{R}(\boldsymbol{X}_{-N:0}) - Nh_\infty(P) &= -\mathbb{E}\log \sum_{i=1}^{\infty} w_{i+1}\mathcal{K}^i(\boldsymbol{X}_{-N:0}) - Nh_k(P) + Nh_k(P) - Nh_\infty(P) \\ &\leq -\mathbb{E}\log \mathcal{K}^k(\boldsymbol{X}_{-N:0}) - Nh_k(P) - \log w_{k+1} + N\epsilon_k = o(N) + N\epsilon_k\end{aligned}$$

where $\epsilon_k = h_k(P) - h_\infty(P)$ and the last equality follows from the consistency of $\mathcal{K}^k$. Since the statement holds for each $k \in \mathcal{N}$, it remains to notice that $\epsilon_k \to 0$.

The third statement follows from the first by noting that

$$\mathbb{E}\delta(X_{n-m+1}|\boldsymbol{X}_{0:n-m}, \boldsymbol{X}_{n-m+2:n}) = -\mathbb{E}\log \mathcal{R}(\boldsymbol{X}_{n-m+1}|\boldsymbol{X}_{0:n-m}, \boldsymbol{X}_{n-m+2:n}) + h_{n,m},$$

and that by definition $\lim_{n,m\to\infty} h_{n,m} = h_{\infty\times\infty}$. Analogously, the fourth statement follows from the second, where we additionally need the stationarity of P to shift the time-index by n to the right. □

APPENDIX C
ADDITIONAL EMPIRICAL RESULTS

A. Fractional Brownian Motion

Here we report additional experiments on the FBM model, a synthetic process commonly used to model complicated time series.

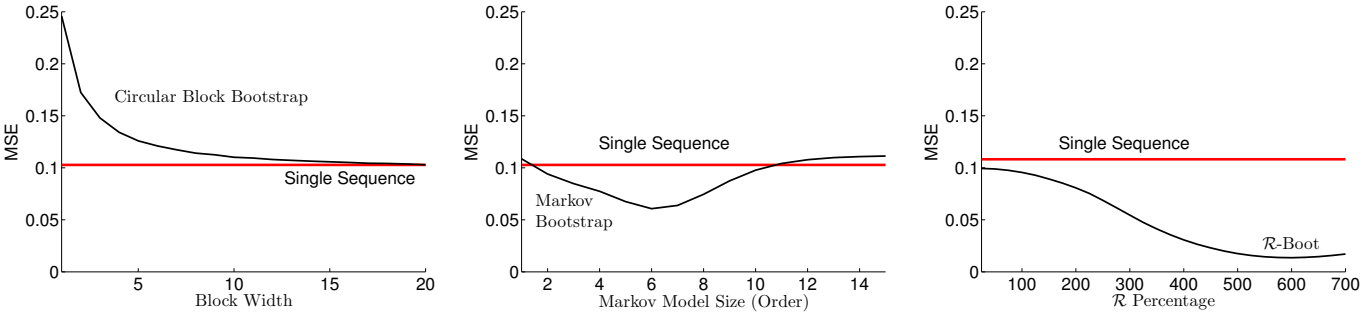

Fig. 4. Sensitivity analysis of CBB, MB, and $\mathcal{R}$-Boot with respect to their parameters (block width, order, and number of replacements R) in the FBM experiment for $n = 200$ (notice the difference in scale for CBB).

Sensitivity analysis. In the results reported in the main text, we considered two values of the parameter R to show that $\mathcal{R}$-Boot is competitive w.r.t. the optimally tuned CBB and MB. In this section we explore the parameter sensitivity of these three methods, showing that advantage of $\mathcal{R}$-Boot is potentially very important. In Fig. 4 we report the MSE performance of each method for a full range of parameters. CBB shows a very poor performance for block sizes that are too small while an increasing block width improves performance. However, as noted in Section V above, CBB fails to demonstrate decisively improved performance even against the single sequence estimator. On the other hand, MB significantly outperforms CBB and the single sequence estimator. Fig. 4 illustrates the dependence of the performance of MB on the model size specified (the parameter corresponding to the order of the Markov source). Small orders introduce too much bias (i.e., the process cannot be described accurately enough with 1 or 2-Markov models), while large orders suffer from large variance, where model sizes above the optimal order overfit noise. As expected for a non-Markovian source, the best model size increases with n. Thus we can conclude that properly tuning the order parameter for MB is challenging and a poor parameter significantly impacts performance.

Finally, we report the performance of $\mathcal{R}$-Boot w.r.t. the number of replacements. As discussed in Sect. III, R corresponds to the number of *attempted* replacements. The need for large values is due to $\mathcal{R}$-Boot's sequential nature. In fact, as the sequence $\boldsymbol{Z}_n^r$ changes, the replacements in a specific position t may have different outcomes because of the conditioning in computing $\mathcal{R}_{\boldsymbol{X}_n}(\cdot|\boldsymbol{Z}_{<t_r}^{r-1}, \boldsymbol{Z}_{>t_r}^{r-1})$. As a result, we need R to be large enough to allow for a sufficient number of actual changes in the original sequence to generate a consistent bootstrap sequence. In order to provide an intuition about the *actual* replacements, let R' be the number of times the value $z_{t_r}^{r-1}$ is changed across R iterations. For $R = 0.75n$, we obtain on average $R' \approx 0.30n$, meaning that less than 30% of the original sequence is actually changed. Similar results are observed from different values of R in both FBM and the real datasets. As illustrated in Fig. 4, both choices of R used in the experiments are suboptimal,

since the performance further improves for larger R (before deteriorating as the choice of R grows too large). Nonetheless, the change in performance is quite smooth and $\mathcal{R}$-Boot outperforms both optimal block width CBB and optimal model size MB for a large range of R values.

B. Currency Data

In this section we report the result of testing $\mathcal{R}$-Boot on differenced high-frequency 1-minute currency pair data. Currency pairs are relative value comparisons between two currencies. When differenced, these data can be considered close to being stationary and ergodic [1]. From the application point of view, these data are suitable for testing analysis methods due to their wide availability and since the underlying financial process is characterized by high liquidity, 24-hour nature and minimal gaps in the data due to non-trading times. This final feature is important because it reduces the noise caused by activities during non-trading hours, such as stock or economic news.

Estimator	*USDCHF*	*EURUSD*	*GBPUSD*	*USDJPY*
Asymptotic Estimator (single sequence)	93.0018	66.6727	72.8849	119.1314
Circular Block (optimal block width)	93.2946 ± 4.1490	66.9138 ± 3.5688	73.1951 ± 3.3746	119.2367 ± 6.6380
$\mathcal{R}$-Boot (100% actual replacements)	44.6137 ± 5.1640	31.7459 ± 3.3220	37.0970 ± 4.5164	50.5639 ± 5.1623
Markov Bootstrap (optimal model size)	43.7268 ± 4.6376	29.4964 ± 3.0739	36.5849 ± 4.2520	47.5460 ± 4.7423

Fig. 5. MDD MSE (with 2X standard errors) on multiple high-frequency currency data.

We estimate the maximum drawdown statistic using bootstrap methods on four pairs of currencies considering the ratio of the first currency over the second currency. For example, the British Pound to U.S. Dollar cross is calculated as GBP/USD. We work with the 1-minute closing price. One minute data constitute a compressed representation of the actual sequence, which includes four data points for each 1 minute time interval: the open, high, low and close prices. The data[3] used in this paper were segmented into 1-day blocks of $n = 1440$ minutes. Days which were partially open due to holidays or announced closures were removed from the data set for consistency. A total of 300 days (1.5 years excluding holidays and weekends) of daily sequential samples from the underlying daily generative process were used for each currency pair.

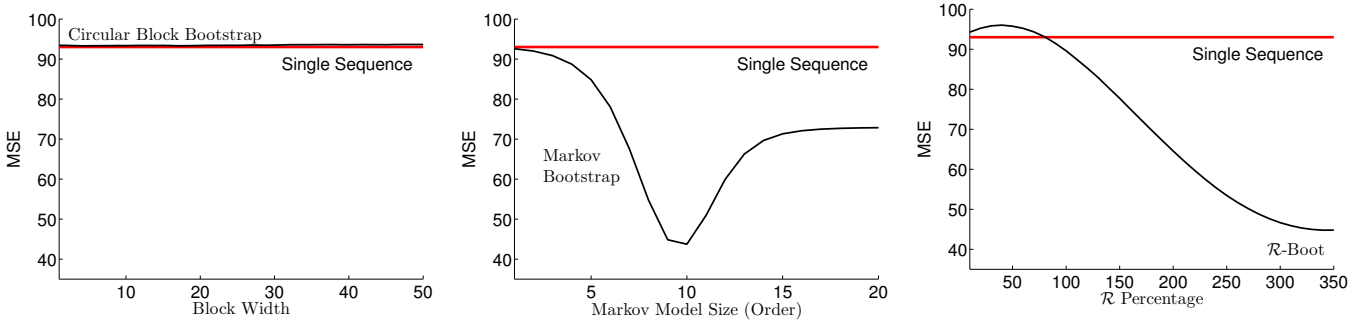

Fig. 6. Sensitivity to parameters for different bootstrap methods for the USDCHF currency (notice the difference in scale for CBB).

In Figure 5, we report estimation performance across all the currency datasets for MB with the best-performing order (with model size selected in $[1, 20]$), best block width CBB (with width selected in $[1, 20]$), and $\mathcal{R}$-Boot with $R = 3.5n$ (chosen to match the value used in the FBM experiments). As we noted in the FBM analysis, a large value of R does not necessarily correspond to a large value of replacements. In fact, we observed that the actual number of replacements in the FBM sequences with $R = 3.5n$ were approximately 140% replacements. On these datasets, CBB exhibits constant behavior and can not beat the simple baseline estimator (single sequence). On the other hand, $\mathcal{R}$-Boot and MB significantly improve the maximum drawdown estimation and they always perform similarly across different currencies (notice that the small advantage of MB is

[3]We downloaded the 1-minute historical data from http://www.forexhistorydatabase.com/

never statistically significant). The full range of parameters for each of these methods is reported in Fig. 6. These results mostly confirm the analysis in Section C-A, where MB achieves very good performance for a specific range of model sizes, while performance rapidly degrades in model sizes that are too large and too small. Finally, $\mathcal{R}$-Boot confirms a similar behavior, as in the FBM, with a performance which gracefully improves as R increases with a gradual degradation thereafter.