arXiv:1712.02383v2 [stat.ME] 11 Jun 2018

Predictive inference for locally stationary time series with an application to climate data

Srinjoy Das
Department of Electrical
and Computer Engineering
University of California—San Diego
La Jolla, CA 92093, USA
email: s2das@ucsd.edu

Dimitris N. Politis
Department of Mathematics
University of California—San Diego
La Jolla, CA 92093-0112, USA
email: dpolitis@ucsd.edu

Abstract

The Model-free Prediction Principle of Politis (2015) has been successfully applied to general regression problems, as well as problems involving stationary time series. However, with long time series, e.g. annual temperature measurements spanning over 100 years or daily financial returns spanning several years, it may be unrealistic to assume stationarity throughout the span of the dataset. In the paper at hand, we show how Model-free Prediction can be applied to handle time series that are only locally stationary, i.e., they can be assumed to be as stationary only over short time-windows. Surprisingly there is little literature on point prediction for general locally stationary time series even in model-based setups and there is no literature on the construction of prediction intervals of locally stationary time series. We attempt to fill this gap here as well. Both one-step-ahead point predictors and prediction intervals are constructed, and the performance of model-free is compared to model-based prediction using models that incorporate a trend and/or heteroscedasticity. Both aspects of the paper, model-free and model-based, are novel in the context of time-series that are locally (but not globally) stationary. We also demonstrate the application of our Model-based and Model-free prediction methods to speleothem climate data which exhibits local stationarity and show that our best model-free point prediction results outperform that obtained with the RAMPFIT algorithm previously used for analysis of this data.

Keywords: Kernel smoothing, linear predictor, nonstationary series, prediction intervals.

1 Introduction

Consider a real-valued time series dataset $Y_1, \ldots, Y_n$ spanning a long time interval, e.g. annual temperature measurements spanning over 100 years or daily financial returns spanning several years. It may be unrealistic to assume that the stochastic structure of time series $\{Y_t, t \in \mathbf{Z}\}$ has stayed invariant over such a long stretch of time; hence, we can not assume that $\{Y_t\}$ is stationary. More realistic is to assume a slowly-changing stochastic structure, i.e., a *locally stationary model* – see (Priestley, 1965), (Priestley, 1988), (Dahlhaus et al., 1997) and (Dahlhaus, 2012).

Our objective is predictive inference for the next data point Y_{n+1}, i.e., constructing a point and interval predictor for Y_{n+1}. The usual approach for dealing with nonstationary series is to assume that the data can be decomposed as the sum of three components:

$$\mu(t) + S_t + W_t$$

where $\mu(t)$ is a deterministic trend function, S_t is a seasonal (periodic) time series, and $\{W_t\}$ is (strictly) stationary with mean zero; this is the 'classical' decomposition of a time series to trend, seasonal and stationary components. The seasonal (periodic) component, be it random or deterministic, can be easily estimated and removed; see e.g. (Brockwell & Davis, 1991). Having done that, the 'classical' decomposition simplifies to the following model with additive trend, i.e.,

$$Y_t = \mu(t) + W_t \tag{1}$$

which can be generalized to accomodate a time-changing variance as well, i.e.,

$$Y_t = \mu(t) + \sigma(t) W_t. \tag{2}$$

In both above models, the time series $\{W_t\}$ is assumed to be (strictly) stationary, weakly dependent, e.g. strong mixing, and satisfying $EW_t = 0$; in model (2), it is also assumed that $\mathrm{Var}\,(W_t) = 1$. As usual, the deterministic functions $\mu(\cdot)$ and $\sigma(\cdot)$ are unknown but assumed to belong to a class of functions that is either finite-dimensional (parametric) or not (nonparametric); we will focus on the latter, in which case it is customary to assume that $\mu(\cdot)$ and $\sigma(\cdot)$ possess some degree of smoothness, i.e., that $\mu(t)$ and $\sigma(t)$ change smoothly (and slowly) with t.

Remark 1.1 (Quantifying smoothness) To analyze locally stationary series it is sometimes useful to map the index set $\{1, \ldots, n\}$ onto the interval $[0, 1]$. In that respect, consider two functions $\mu_{[0,1]} : [0, 1] \mapsto \mathbf{R}$ and $\sigma_{[0,1]} : [0, 1] \mapsto (0, \infty)$, and let

$$\mu(t) = \mu_{[0,1]}(a_t) \quad \text{and} \quad \sigma(t) = \sigma_{[0,1]}(a_t) \tag{3}$$

where $a_t = (t-1)/n$ for $t = 1, \ldots, n$. We will assume that $\mu_{_{[0,1]}}(\cdot)$ and $\sigma_{_{[0,1]}}(\cdot)$ are continuous and smooth, i.e., possess k continuous derivatives on $[0, 1]$. To take full advantage of the local linear smoothers of Section 2.2 ideally one would need $k \geq 2$. However, all methods to be discussed here are valid even when $\mu_{_{[0,1]}}(x)$ and $\sigma_{_{[0,1]}}(x)$ are continuous for all $x \in [0, 1]$ but only piecewise smooth.

As far as capturing the first two moments of Y_t, models (1) and (2) are considered general and flexible—especially when $\mu(\cdot)$ and $\sigma(\cdot)$ are not parametrically specified—and have been studied extensively; see e.g. (Zhou & Wu, 2009), (Zhou & Wu, 2010). However, it may be that the skewness and/or kurtosis of Y_t changes with t, in which case centering and studentization alone can not render the problem stationary. To see why, note that under model (2), $EY_t = \mu(t)$ and $\operatorname{Var} Y_t = \sigma^2(t)$; hence,

$$W_t = \frac{Y_t - \mu(t)}{\sigma(t)} \tag{4}$$

cannot be (strictly) stationary unless the skewness and kurtosis of Y_t are constant. Furthermore, it may be the case that the nonstationarity is due to a feature of the m–th dimensional marginal distribution not being constant for some $m \geq 1$, e.g., perhaps the correlation $\operatorname{Corr}(Y_t, Y_{t+1})$ changes smoothly (and slowly) with t. Notably, models (1) and (2) only concern themselves with features of the 1st marginal distribution.

For all the above reasons, it seems valuable to develop a methodology for the statistical analysis of nonstationary time series that does not rely on simple additive models such as (1) and (2). Fortunately, the Model-free Prediction Principle of (Politis, 2013), (Politis, 2015) suggests a way to accomplish Model-free inference—including the construction of prediction intervals—in the general setting of time series that are only locally stationary. The key towards Model-free inference is to be able to construct an invertible transformation $H_n : \underline{Y}_n \mapsto \underline{\epsilon}_n$ where $\underline{\epsilon}_n = (\epsilon_1, \ldots, \epsilon_n)'$ is a random vector with i.i.d. components; the details are given in Section 3. The next section revisits the problem of model-based inference in a locally stationary setting, and develops a bootstrap methodology for the construction of (model-based) prediction intervals. Both approaches, Model-based of Section 2 and Model-free of Section 3, are novel, and they are empirically compared to each other in Section 5 using finite sample experiments. Both synthetic and real-life data are used for this purpose.

The prototype of local (but not global) stationarity is manifested in climate data observed over long periods. In Section 6 we focus on the speleothem climate archive data discussed in (Fleitmann et al., 2003) whose statistical analysis is presented in (Mudelsee, 2014). This dataset which is shown in Figure 1 contains oxygen isotope record obtained from stalagmite Q5 from southern Oman over the past 10,300 years. In this figure delta-O-18 on the Y-axis is a measure of the ratio of stable isotopes oxygen-18 (^{18}O) and oxygen-16

(^{16}O) and Age (a B.P. where B.P. indicates Before Present) on the X-axis denotes time before the present i.e. time increases from right to left. Details of how delta-O-18 is defined can be found on `https://en.wikipedia.org/wiki/%CE%9418O`. Along the growth axis of the nearly 1 meter long speleothem (which is in this case stalagmite), approximately every 0.7 mm about 5 mg material (calcium carbonate) was drilled, thereby yielding n=1345 samples. This carbonate was then analyzed to determine the delta-O-18 values.

The oxygen isotope ratio serves as a proxy variable for the climate variable **monsoon rainfall**. This data can be used for climate analysis applications such as whether there exists solar influences on the variations in monsoon rainfall; here low values of delta-O-18 would indicate a strong monsoon. The full dataset can be referenced at:
`http://manfredmudelsee.com/book/data/1-7.txt`. Previously the RAMPFIT algorithm (Mudelsee, 2000) has been used to fit data that exhibit change points such as the speleothem climate archive. However RAMPFIT was not designed to handle arbitrary locally stationary data which maybe present in climate time series. In Section 6 we focus on a part of the delta-O-18 proxy variable data that contains a linear trend and apply our Model-Free and Model-Based algorithms over this range to estimate the performance of both point prediction and prediction intervals. We then show that our best Model-Free point predictor achieves superior performance in point prediction compared to RAMPFIT; notably, RAMPFIT was not originally designed to estimate prediction intervals.

In Section 4 we also describe techniques for diagnostics which are useful for Model-Free prediction in order to successfully generate both point predictors and prediction intervals. Model-Based and Model-Free algorithms for the construction of prediction intervals are described in detail in Appendix A. The RAMPFIT algorithm used to generate point prediction results for comparison with our model-free and model-based methods is described in Appendix B.

2 Model-based inference

Throughout Section 2, we will assume model (2)—that includes model (1) as a special case—together with a nonparametric assumption on smoothness of $\mu(\cdot)$ and $\sigma(\cdot)$ as described in Remark 1.1.

2.1 Theoretical optimal point prediction

It is well-known that the L_2–optimal predictor of Y_{n+1} given the data $\underline{Y}_n = (Y_1, \ldots, Y_n)'$ is the conditional expectation $E(Y_{n+1}|\underline{Y}_n)$. Furthermore, under model (2), we have

$$E(Y_{n+1}|\underline{Y}_n) = \mu(n+1) + \sigma(n+1)E(W_{n+1}|\underline{Y}_n). \tag{5}$$

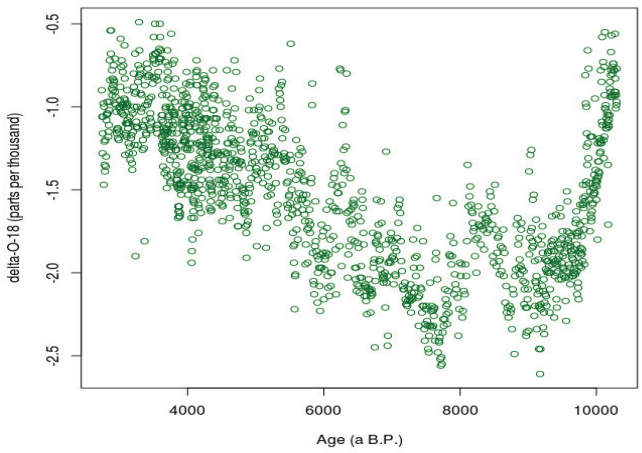

Figure 1: Oxygen Isotope Record from stalagmite Q5 from southern Oman (1345 samples) where B.P. indicates Before Present

For $j < J$, define $\mathcal{F}_j^J(Y)$ to be the *information set* $\{Y_j, Y_{j+1}, \ldots, Y_J\}$, also known as σ–field, and note that the information sets $\mathcal{F}_{-\infty}^t(Y)$ and $\mathcal{F}_{-\infty}^t(W)$ are identical for any t, i.e., knowledge of $\{Y_s \text{ for } s < t\}$ is equivalent to knowledge of $\{W_s \text{ for } s < t\}$; here, $\mu(\cdot)$ and $\sigma(\cdot)$ are assumed known. Hence, for large n, and due to the assumption that W_t is weakly dependent (and therefore the same must be true for Y_t as well), the following large-sample approximation is useful, i.e.,

$$E(W_{n+1}|\underline{Y}_n) \simeq E(W_{n+1}|Y_s, s \leq n) = E(W_{n+1}|W_s, s \leq n) \simeq E(W_{n+1}|\underline{W}_n) \tag{6}$$

where $\underline{W}_n = (W_1, \ldots, W_n)'$.

All that is needed now is to construct an approximation for $E(W_{n+1}|\underline{W}_n)$. Usual approaches involve either assuming that the time series $\{W_t\}$ is Markov of order p as in (Pan & Politis, 2016), or approximating $E(W_{n+1}|\underline{W}_n)$ by a linear function of $\underline{W}_n$ as in (McMurry & Politis, 2015), i.e., contend ourselves with the best linear predictor of W_{n+1} denoted by $\bar{E}(W_{n+1}|\underline{W}_n)$.

Taking the latter approach, the L_2–optimal linear predictor of W_{n+1} based on $\underline{W}_n$ is

$$\bar{E}(W_{n+1}|\underline{W}_n) = \phi_1(n)W_n + \phi_2(n)W_{n-1} + \ldots + \phi_n(n)W_1, \tag{7}$$

where the optimal coefficients $\phi_i(n)$ are computed from the normal equations, i.e., $\phi(n) \equiv (\phi_1(n), \cdots, \phi_n(n))' = \Gamma_n^{-1}\gamma(n)$; here, $\Gamma_n = [\gamma_{|i-j|}]_{i,j=1}^n$ is the autocovariance matrix of the random vector $\underline{W}_n$, and $\gamma(n) = (\gamma_1, \ldots, \gamma_n)'$ where $\gamma_k = EY_jY_{j+k}$. Of course, Γ_n is unknown but can be estimated by any of the positive definite estimators developed in (McMurry & Politis, 2015).

Alternatively, the L_2–optimal linear predictor of W_{n+1} can be obtained by fitting a (causal) AR(p) model to the data $W_1, \ldots, W_n$ with p chosen by minimizing AIC or a related criterion; this would entail fitting the model:

$$W_t = \phi_1 W_{t-1} + \phi_2 W_{t-2} + \cdots + \phi_p W_{t-p} + V_t \tag{8}$$

where V_t is a stationary white noise, i.e., an uncorrelated sequence, with mean zero and variance τ^2. The implication then is that

$$\bar{E}(W_{n+1}|\underline{W}_n) = \phi_1 W_n + \phi_2 W_{n-1} + \cdots + \phi_p W_{n-p+1}. \tag{9}$$

As discussed in the rejoinder to (McMurry & Politis, 2015), the two methods for constructing $\bar{E}(W_{n+1}|\underline{W}_n)$ are closely related; in fact, predictor (7) coincides with the above AR–type predictor if the matrix Γ_n is the one implied by the fitted AR(p) model (8). We will use the AR–type predictor in the sequel because it additionally affords us the possibility of resampling based on model (8).

2.2 Trend estimation and practical prediction

To construct the L_2–optimal predictor (5), we need to estimate the smooth trend $\mu(\cdot)$ and variance $\sigma(\cdot)$ in a nonparametric fashion; this can be easily accomplished via kernel smoothing—see e.g. (Härdle & Vieu, 1992), (Kim & Cox, 1996), (Li & Racine, 2007). When confidence intervals for $\mu(t)$ and $\sigma(t)$ are required, however, matters are more complicated as the asymptotic distribution of the different estimators depends on many unknown parameters; see e.g. (Masry & Tjøstheim, 1995). Even more difficult is the construction of prediction intervals.

Note, furthermore, that the problem of prediction of Y_{n+1} involves estimating the functions $\mu_{[0,1]}(a)$ and $\sigma_{[0,1]}(a)$ described in Remark 1.1 for $a = 1$, i.e., it is essentially a boundary problem. In such cases, it is well-known that local linear fitting has better properties—in particular, smaller bias—than kernel smoothing which is well-known to be tantamount to local constant fitting; (Fan & Gijbels, 1996),(Fan & Yao, 2007), or (Li & Racine, 2007).

Remark 2.1 (One-sided estimation) Since the goal is predictive inference on Y_{n+1}, local constant and/or local linear fitting must be performed in a *one-sided way*. To see why, recall that in predictor (5), the estimands involve $\mu_{[0,1]}(1)$ and $\sigma_{[0,1]}(1)$ as just mentioned. Furthermore to compute $\bar{E}(W_{n+1}|\underline{W}_n)$ in eq. (7) we need access to the stationary data $W_1, \ldots, W_n$ in order to estimate Γ_n. The W_t's are not directly observed, but—much like residuals in a regression—they can be reconstructed by eq. (4) with estimates of $\mu(t)$ and $\sigma(t)$ plugged-in. What is important is that **the way W_t is reconstructed/estimated by (say) $\hat{W}_t$ must remain the same for all t**, otherwise the reconstructed data $\hat{W}_1, \ldots, \hat{W}_n$

can not be considered stationary. Since W_t can only be estimated in a one-sided way for t close to n, the same one-sided way must also be implemented for t in the middle of the dataset even though in that case two-sided estimation is possible.

By analogy to model-based regression as described in (Politis, 2013), the one-sided Nadaraya-Watson (NW) kernel estimators of $\mu(t)$ and $\sigma(t)$ can be defined in two ways. In what follows, the notation $t_k = k$ will be used; this may appear redundant but it makes clear that t_k is the kth design point in the time series regression, and allows for easy extension in the case of missing data. Note that the bandwidth parameter b will be assumed to satisfy

$$b \to \infty \text{ as } n \to \infty \text{ but } b/n \to 0, \tag{10}$$

i.e., b is analogous to the product hn where h is the usual bandwidth in nonparametric regression, see e.g. We will assume throughout that $K(\cdot)$ is a nonnegative, symmetric kernel function.

1. **NW–Regular fitting:** Let $t \in [b+1, n]$, and define

$$\hat{\mu}(t) = \sum_{i=1}^{t} Y_i \; \hat{K}\left(\frac{t-t_i}{b}\right) \quad \text{and} \quad \hat{M}(t) = \sum_{i=1}^{t} Y_i^2 \; \hat{K}(\frac{t-t_i}{b}) \tag{11}$$

   where

$$\hat{\sigma}(t) = \sqrt{\hat{M}_t - \hat{\mu}(t)^2} \quad \text{and} \quad \hat{K}\left(\frac{t-t_i}{b}\right) = \frac{K(\frac{t-t_i}{b})}{\sum_{k=1}^{t} K(\frac{t-t_k}{b})}. \tag{12}$$

   Using $\hat{\mu}(t)$ and $\hat{\sigma}(t)$ we can now define the *fitted* residuals by

$$\hat{W}_t = \frac{Y_t - \hat{\mu}(t)}{\hat{\sigma}(t)} \quad \text{for} \quad t = b+1, \ldots, n. \tag{13}$$

2. **NW–Predictive fitting (delete-1):** Let

$$\tilde{\mu}(t) = \sum_{i=1}^{t-1} Y_i \; \tilde{K}\left(\frac{t-t_i}{b}\right) \quad \text{and} \quad \tilde{M}(t) = \sum_{i=1}^{t-1} Y_i^2 \; \tilde{K}(\frac{t-t_i}{b}) \tag{14}$$

   where

$$\tilde{\sigma}(t) = \sqrt{\tilde{M}_t - \tilde{\mu}(t)^2} \quad \text{and} \quad \tilde{K}\left(\frac{t-t_i}{b}\right) = \frac{K(\frac{t-t_i}{b})}{\sum_{k=1}^{t-1} K(\frac{t-t_k}{b})}. \tag{15}$$

   Using $\tilde{\mu}(t)$ and $\tilde{\sigma}(t)$ we now define the *predictive* residuals by

$$\tilde{W}_t = \frac{Y_t - \tilde{\mu}(t)}{\tilde{\sigma}(t)} \quad \text{for} \quad t = b+1, \ldots, n. \tag{16}$$

Similarly, the one-sided local linear (LL) fitting estimators of $\mu(t)$ and $\sigma(t)$ can be defined in two ways.

1. **LL–Regular fitting:** Let $t \in [b+1, n]$, and define

$$\hat{\mu}(t) = \frac{\sum_{j=1}^{t} w_j Y_j}{\sum_{j=1}^{t} w_j + n^{-2}} \quad \text{and} \quad \hat{M}(t) = \frac{\sum_{j=1}^{t} w_j Y_j^2}{\sum_{j=1}^{t} w_j + n^{-2}} \tag{17}$$

   where

$$w_j = K(\frac{t-t_j}{b}) \left[s_{t,2} - (t-t_j) s_{t,1} \right], \tag{18}$$

   and $s_{t,k} = \sum_{j=1}^{t} K(\frac{t-t_j}{b})(t-t_j)^k$ for $k = 0, 1, 2$. The term n^{-2} in eq. (17) is just to ensure the denominator is not zero; see Fan (1993). Eq. (12) then yields $\hat{\sigma}(t)$, and eq. (13) yields $\hat{W}_t$.

2. **LL–Predictive fitting (delete-1):** Let

$$\tilde{\mu}(t) = \frac{\sum_{j=1}^{t-1} w_j Y_j}{\sum_{j=1}^{t-1} w_j + n^{-2}} \quad \text{and} \quad \tilde{M}(t) = \frac{\sum_{j=1}^{t-1} w_j Y_j^2}{\sum_{j=1}^{t-1} w_j + n^{-2}} \tag{19}$$

   where

$$w_j = K(\frac{t-t_j}{b}) \left[s_{t-1,2} - (t-t_j) s_{t-1,1} \right]. \tag{20}$$

   Eq. (15) then yields $\tilde{\sigma}(t)$, and eq. (16) yields $\tilde{W}_t$.

Using one of the above four methods (NW vs. LL, regular vs. predictive) gives estimates of the quantities needed to compute the L_2–optimal predictor (5). In order to approximate $E(W_{n+1}|\underline{Y}_n)$, one would treat the proxies $\hat{W}_t$ or $\tilde{W}_t$ as if they were the true W_t, and proceed as outlined in Section 2.1.

Remark 2.2 (Predictive vs. regular fitting) In order to estimate $\mu(n+1)$ and $\sigma(n+1)$, the predictive fits $\tilde{\mu}(n+1)$ and $\tilde{\sigma}(n+1)$ are constructed in a straightforward manner. However, the formula giving $\hat{\mu}(t)$ and $\hat{\sigma}(t)$ changes when t becomes greater than n; this is due to an effective change in kernel shape since part of the kernel is not used when $t > n$. Focusing momentarily on the trend estimators, what happens is that the formulas for $\tilde{\mu}(t)$ and $\hat{\mu}(t)$—although different when $t \leq n$—become identical when $t > n$ except for the difference in kernel shape. Traditional model-fitting ignores these issues, i.e., proceeds with using different formulas for estimation of $\mu(t)$ according to whether $t \leq n$ or $t > n$. However, in trying to predict the new, unobserved W_{n+1} we need to first capture its statistical characteristics, and for this reason we need a sample of W_t's. But the residual from the model at $t = n+1$ looks like $\tilde{W}_{n+1}$ from *either* regular or predictive approach,

since $\tilde{\mu}(t)$ and $\hat{\mu}(t)$ become the same when $t = n+1$; it is apparent that traditional model-fitting tries to capture the statistical characteristics of $\tilde{W}_{n+1}$ from a sample of $\hat{W}_t$'s, i.e., comparing apples to oranges. Herein lies the problem which is analogous to the discussion on prediction using fitted vs. predictive residuals in nonparametric regression as discussed in (Politis, 2013). Therefore, our preference is to use the predictive quantities $\tilde{\mu}(t)$, $\tilde{\sigma}(t)$, and $\tilde{W}_t$ throughout the predictive modeling.

Remark 2.3 (Time series cross-validation) To choose the bandwidth b for either of the above methods, predictive cross-validation may be used but it must be adapted to the time series prediction setting, i.e., always one-step-ahead. To elaborate, let $k < n$, and suppose only subseries $Y_1, \ldots, Y_k$ has been observed. Denote $\hat{Y}_{k+1}$ the best predictor of Y_{k+1} based on the data $Y_1, \ldots, Y_k$ constructed according to the above methodology and some choice of b. However, since Y_{k+1} is known, the quality of the predictor can be assessed. So, for each value of b over a reasonable range, we can form either $PRESS(b) = \sum_{k=k_o}^{n-1} (\hat{Y}_{k+1} - Y_{k+1})^2$ or $PRESAR(b) = \sum_{k=k_o}^{n-1} |\hat{Y}_{k+1} - Y_{k+1}|$; here k_o should be big enough so that estimation is accurate, e.g., k_o can be of the order of $\sqrt{n}$. The cross-validated bandwidth choice would then be the b that minimizes $PRESS(b)$; alternatively, we can choose to minimize $PRESAR(b)$ if an L_1 measure of loss is preferred. Finally, note that a quick-and-easy (albeit suboptimal) version of the above is to use the (supoptimal) predictor $\hat{Y}_{k+1} \simeq \hat{\mu}(k+1)$ and base $PRESS(b)$ or $PRESAR(b)$ on this approximation.

2.3 Model-based prediction intervals

To go from point prediction to prediction intervals, some form of resampling is required. Since model (2) is driven by the stationary sequence $\{W_t\}$, a model-based bootstrap can then be concocted in which $\{W_t\}$ is resampled, giving rise to the bootstrap pseudo-series $\{W_t^*\}$, which in turn gives rise to bootstrap pseudo-data $\{Y_t^*\}$ via a fitted version of model (2). To generate a stationary bootstrap pseudo-series $\{W_t^*\}$, two popular time series resampling methods are (a) the stationary bootstrap of (Politis & Romano, 1994) and (b) the AR bootstrap which entails treating the V_t appearing in eq. (8) as if they were i.i.d., performing an i.i.d. bootstrap on them, and then generating $\{W_t^*\}$ via the recursion (8) driven by the bootstrapped innovations. We will use the latter in the sequel because it ties in well with the AR-type predictor of W_{n+1} developed at the end of Section 2.1, and it is more amenable to the construction of prediction intervals as discussed in (Pan & Politis, 2016). In addition, (Kreiss, Paparoditis, & Politis, 2011) have recently shown that the AR bootstrap—also known as AR-sieve bootstrap since p is allowed to grow with n—can be valid under some conditions even if the V_t of eq. (8) are not trully i.i.d.

We will now develop an algorithm for the construction of model-based prediction intervals; this is a 'forward' bootstrap algorithm in the terminology of (Pan & Politis, 2016)

although a 'backward' bootstrap algorithm can also be concocted. To describe it in general, let $\check{\mu}(\cdot)$ and $\check{\sigma}(\cdot)$ be our chosen estimates of $\mu(\cdot)$ and $\sigma(\cdot)$ according to one of the abovementioned four methods (NW vs. LL, regular vs. predictive); also let $\check{W}_t$ denote the resulting proxies for the unobserved W_t for $t = 1, \ldots, n$. Hence, our approximation to the L_2–optimal point predictor of Y_{n+1} is

$$\Pi = \check{\mu}(n+1) + \check{\sigma}(n+1)\left[\hat{\phi}_1 \check{W}_n + \cdots + \hat{\phi}_p \check{W}_{n-p+1}\right] \tag{21}$$

where $\hat{\phi}_1, \ldots, \hat{\phi}_p$ are the Yule-Walker estimators of $\phi_1, \ldots, \phi_p$ appearing in eq. (8).

As discussed in Chapter 2 of (Politis, 2015) the construction of prediction intervals will be based on approximating the distribution of the *predictive root*: $Y_{n+1} - \Pi$ by that of the bootstrap predictive root: $Y^*_{n+1} - \Pi^*$ where the quantities Y^*_{n+1} and Π^* are formally defined in the Model-based (MB) bootstrap algorithm outlined below.

Algorithm 2.1 Model-based bootstrap for prediction intervals for Y_{n+1}

1. *Based on the data $Y_1, \ldots, Y_n$, calculate the estimators $\check{\mu}(\cdot)$ and $\check{\sigma}(\cdot)$, and the 'residuals' $\check{W}_1, \ldots, \check{W}_n$ using model* (2).

2. *Fit the AR(p) model* (8) *to the series $\check{W}_1, \ldots, \check{W}_n$ (with p selected by AIC minimization), and obtain the Yule-Walker estimators $\hat{\phi}_1, \ldots, \hat{\phi}_p$, and the error proxies*

$$\check{V}_t = \check{W}_t - \hat{\phi}_1 \check{W}_{t-1} - \cdots - \hat{\phi}_p \check{W}_{t-p} \quad \textit{for} \quad t = p+b+1, \ldots, n.$$

   Here b is the bandwidth determined by the cross-validation procedure of Remark 2.3.

3. (a) *Let $\check{V}^*_t$ for $t = 1, \ldots, n, n+1$ be drawn randomly with replacement from the set $\{\check{\check{V}}_t$ for $t = p+b+1, \ldots, n\}$ where $\check{\check{V}}_t = \check{V}_t - (n-p-b)^{-1}\sum_{i=p+b+1}^{n} \check{V}_i$. Let I be a random variable drawn from a discrete uniform distribution on the values $\{p+b, p+b+1, \ldots, n\}$, and define the bootstrap initial conditions $\check{W}^*_t = \check{W}_{t+I}$ for $t = -p+1, \ldots, 0$. Then, create the bootstrap data $\check{W}^*_1, \ldots, \check{W}^*_n$ via the AR recursion*

$$\check{W}^*_t = \hat{\phi}_1 \check{W}^*_{t-1} + \cdots + \hat{\phi}_p \check{W}^*_{t-p} + \check{V}^*_t \quad \textit{for} \quad t = 1, \ldots, n.$$

   (b) *Create the bootstrap pseudo-series $Y^*_1, \ldots, Y^*_n$ by the formula*

$$Y^*_t = \check{\mu}(t) + \check{\sigma}(t)\check{W}^*_t \quad \textit{for} \quad t = 1, \ldots, n.$$

   (c) *Re-calculate the estimators $\check{\mu}^*(\cdot)$ and $\check{\sigma}^*(\cdot)$ from the bootstrap data $Y^*_1, \ldots, Y^*_n$. This gives rises to new bootstrap 'residuals'* [1] *on which an AR(p) model is again fitted yielding the bootstrap Yule-Walker estimators $\hat{\phi}^*_1, \ldots, \hat{\phi}^*_p$.*

[1] The bootstrap estimators $\check{\mu}^*(\cdot)$ and $\check{\sigma}^*(\cdot)$ are based on bandwidth b' determined by Algorithm A.3 given in Appendix A. This may be different from the bandwidth b found using model-based cross-validation.

(d) Calculate the bootstrap predictor

$$\Pi^* = \check{\mu}^*(n+1) + \check{\sigma}^*(n+1)\left[\hat{\phi}_1^*\check{W}_n + \ldots + \hat{\phi}_p^*\check{W}_{n-p+1}\right].$$

[Note that in calculating the bootstrap conditional expectation of $\check{W}^*_{n+1}$ given its p–past, we have re-defined the values $(\check{W}^*_n, \ldots, \check{W}^*_{n-p+1})$ to make them match the original $(\check{W}_n, \ldots, \check{W}_{n-p+1})$; this is an important part of the 'forward' bootstrap procedure for prediction intervals as discussed in (Pan & Politis, 2016)].

(e) Calculate a bootstrap future value

$$Y^*_{n+1} = \check{\mu}(n+1) + \check{\sigma}(n+1)\check{W}^*_{n+1}$$

where again $\check{W}^*_{n+1} = \hat{\phi}_1\check{W}_n + \cdots + \hat{\phi}_p\check{W}_{n-p+1} + \check{V}^*_{n+1}$ uses the original values $(\check{W}_n, \ldots, \check{W}_{n-p+1})$; recall that $\check{V}^*_{n+1}$ has already been generated in step (a) above.

(f) Calculate the bootstrap root replicate $Y^*_{n+1} - \Pi^*$.

4. *Steps (a)—(f) in the above are repeated a large number of times (say B times), and the B bootstrap root replicates are collected in the form of an empirical distribution whose α–quantile is denoted by $q(\alpha)$.*

5. *Finally, a $(1-\alpha)100\%$ equal-tailed prediction interval for Y_{n+1} is given by*

$$[\Pi + q(\alpha/2),\ \Pi + q(1-\alpha/2)]. \tag{22}$$

It is easy to see that prediction interval (22) is asymptotically valid (conditionally on $Y_1, \ldots, Y_n$) provided: (i) estimators $\check{\mu}(n+1)$ and $\check{\sigma}(n+1)$ are consistent for their respective targets $\mu_{[0,1]}(1)$ and $\sigma_{[0,1]}(1)$, and (ii) the AR(p) approximation is consistent allowing for the possibility that p grows as $n \to \infty$. If $\check{\mu}(\cdot)$ and $\check{\sigma}(\cdot)$ correspond to one of the above mentioned four methods (NW vs. LL, regular vs. predictive), then provision (i) is satisfied under standard conditions including the bandwidth condition (10). Provision (ii) is also easy to satisfy as long as the spectral density of the series $\{W_t\}$ is continuous and bounded away from zero; see e.g. Lemma 2.2 of (Kreiss et al., 2011).

Although desirable, asymptotic validity does not tell the whole story. A prediction interval can be thought to be successful if it also manages to capture the finite-sample variability of the estimated quantities such as $\check{\mu}(\cdot)$, $\check{\sigma}(\cdot)$ and $\hat{\phi}_1, \hat{\phi}_2, \ldots$. Since this finite-sample variability vanishes asymptotically, the performance of a prediction interval such as (22) must be gauged by finite-sample simulations. Results of these simulations are shown in Section 5.

3 Model-free inference

Model (2) is a flexible way to account for a time-changing mean and variance of Y_t. However, nothing precludes that the time series $\{Y_t \text{ for } t \in \mathbf{Z}\}$ has a nonstationarity in its third (or higher moment), and/or in some other feature of its mth marginal distribution. A way to address this difficulty, and at the same time give a fresh perspective to the problem, is provided by the Model-Free Prediction Principle of Politis (2013, 2015).

The key towards Model-free inference is to be able to construct an invertible transformation $H_n : \underline{Y}_n \mapsto \underline{\epsilon}_n$ where $\underline{\epsilon}_n = (\epsilon_1, \ldots, \epsilon_n)'$ is a random vector with i.i.d. components. In order to do this in our context, let some $m \geq 1$, and denote by $\mathcal{L}(Y_t, Y_{t-1}, \ldots, Y_{t-m+1})$ the mth marginal of the time series Y_t , i.e. the joint probability law of the vector $(Y_t, Y_{t-1}, \ldots, Y_{t-m+1})'$. Although we abandon model (2) in what follows, we still want to employ nonparametric smoothing for estimation; thus, we must assume that $\mathcal{L}(Y_t, Y_{t-1}, \ldots, Y_{t-m+1})$ changes smoothly (and slowly) with t.

Remark 3.1 (Quantifying smoothness–model-free case) As in Remark 1.1, we can formally quantify smoothness by mapping the index set $\{1, \ldots, n\}$ onto the interval $[0, 1]$. Let $\underline{s} = (s_0, s_1, \ldots, s_{m-1})'$, and define the distribution function of the mth marginal by

$$D_t^{(m)}(\underline{s}) = P\{Y_t \leq s_0, Y_{t-1} \leq s_1, \ldots, Y_{t-m+1} \leq s_{m-1}\}.$$

Let $a_t = (t-1)/n$ as before, and assume that we can write

$$D_t^{(m)}(\underline{s}) = D_{a_t}^{[0,1]}(\underline{s}) \text{ for } t = 1, \ldots, n. \tag{23}$$

We can now quantify smoothness by assuming that, for each fixed $\underline{s}$, the function $D_x^{[0,1]}(\underline{s})$ is continuous and smooth in $x \in [0, 1]$, i.e., possesses k continuous derivatives. As in Remark 1.1, here as well it seems to be sufficient that $D_x^{[0,1]}(\underline{s})$ is continuous in x but only piecewise smooth.

A convenient way to ensure both the smoothness and data-based consistent estimation of $\mathcal{L}(Y_t, Y_{t-1}, \ldots, Y_{t-m+1})$ is to assume that, for all t,

$$Y_t = \mathbf{f}_t(W_t, W_{t-1}, \ldots, W_{t-m+1}) \tag{24}$$

for some function $\mathbf{f}_t(w)$ that is smooth in both arguments t and w, and some strictly stationary and weakly dependent, univariate time series W_t; without loss of generality, we may assume that W_t is a Gaussian time series. In fact, Eq. (24) with $\mathbf{f}_t(\cdot)$ not depending on t is a familiar assumption in studying non-Gaussian and/or long-range dependent stationary processes—see e.g. (Samorodnitsky & Taqqu, 1994). By allowing $\mathbf{f}_t(\cdot)$ to vary smoothly

(and slowly) with t, Eq. (24) can be used to describe a rather general class of locally stationary processes. Note that model (2) is a special case of Eq. (24) with $m = 1$, and the function $\mathbf{f}_t(w)$ being affine/linear in w. Thus, for concreteness and easy comparison with the model-based case of Eq. (2), we will focus in the sequel on the case $m = 1$. Section 3.10 discusses how to handle the case $m > 1$.

3.1 Constructing the theoretical transformation

Hereafter, adopt the setup of Eq. (24) with $m = 1$, and let

$$D_t(y) = P\{Y_t \leq y\}$$

denote the 1st marginal distribution of time series $\{Y_t\}$. Throughout Section 3, the default assumption will be that $D_t(y)$ is (absolutely) continuous in y for all t; however, a departure from this assumption will be discussed in Section 3.8.

We now define new variables via the probability integral transform, i.e., let

$$U_t = D_t(Y_t) \quad \text{for } t = 1, \ldots, n; \tag{25}$$

the assumed continuity of $D_t(y)$ in y implies that $U_1, \ldots, U_n$ are random variables having distribution Uniform $(0, 1)$. However, $U_1, \ldots, U_n$ are dependent; to transform them to independence, a preliminary transformation towards Gaussianity is helpful as discussed in (Politis, 2013). Letting Φ denote the cumulative distribution function (cdf) of the standard normal distribution, we define

$$Z_t = \Phi^{-1}(U_t) \quad \text{for } t = 1, \ldots, n; \tag{26}$$

it then follows that $Z_1, \ldots, Z_n$ are standard normal—albeit correlated—random variables.

Let Γ_n denote the $n \times n$ covariance matrix of the random vector $\underline{Z}_n = (Z_1, \ldots, Z_n)'$. Under standard assumptions, e.g. that the spectral density of the series $\{Z_t\}$ is continuous and bounded away from zero,[2] the matrix Γ_n is invertible when n is large enough. Consider the Cholesky decomposition $\Gamma_n = C_n C_n'$ where C_n is (lower) triangular, and construct the *whitening* transformation:

$$\underline{\epsilon}_n = C_n^{-1} \underline{Z}_n. \tag{27}$$

It then follows that the entries of $\underline{\epsilon}_n = (\epsilon_1, \ldots, \epsilon_n)'$ are uncorrelated standard normal. Assuming that the random variables $Z_1, \ldots, Z_n$ were *jointly* normal, this can be strenghtened

[2] If the spectral density is equal to zero over an interval—however small—then the time series $\{Z_t\}$ is perfectly predictable based on its infinite past, and the same would be true for the time series $\{Y_t\}$; see Brockwell and Davis (1991, Theorem 5.8.1) on Kolmogorov's formula.

to claim that $\epsilon_1, \ldots, \epsilon_n$ are i.i.d. $N(0,1)$; see Section 3.10 for further discussion. Consequently, the transformation of the dataset $\underline{Y}_n = (Y_1, \ldots, Y_n)'$ to the vector $\underline{\epsilon}_n$ with i.i.d. components has been achieved as required in premise (a) of the Model-free Prediction Principle. Note that all the steps in the transformation, i.e., eqs. (25), (26) and (27), are invertible; hence, the composite transformation $H_n : \underline{Y}_n \mapsto \underline{\epsilon}_n$ is invertible as well.

3.2 Kernel estimation of the 'uniformizing' transformation

We first focus on estimating the 'uniformizing' part of the transformation, i.e., eq. (25). Recall that the Model-free setup implies that the function $D_t(\cdot)$ changes smoothly (and slowly) with t; hence, local constant and/or local linear fitting can be used to estimate it. Using local constant, i.e., kernel estimation, a consistent estimator of the marginal distribution $D_t(y)$ is given by:

$$\hat{D}_t(y) = \sum_{i=1}^{T} \mathbf{1}\{Y_{t_i} \leq y\} \tilde{K}(\frac{t - t_i}{b}) \tag{28}$$

where $\tilde{K}(\frac{t-t_i}{b}) = K(\frac{t-t_i}{b}) / \sum_{j=1}^{T} K(\frac{t-t_j}{b})$. Note that the kernel estimator (28) is *one-sided* for the same reasons discussed in Remark 2.1. Since $\hat{D}_t(y)$ is a step function in y, a smooth estimator can be defined as:

$$\bar{D}_t(y) = \sum_{i=1}^{T} \Lambda(\frac{y - Y_{t_i}}{h_0}) \tilde{K}(\frac{t - t_i}{b}) \tag{29}$$

where h_0 is a secondary bandwidth. Furthermore, as in Section 2.2, we can let $T = t$ or $T = t - 1$ leading to a **fitted vs. predictive** way to estimate $D_t(y)$ by either $\hat{D}_t(y)$ or $\bar{D}_t(y)$. Cross-validation is used to determine the bandwidths h_0 and b ; details are described in Section 3.5.

3.3 Local linear estimation of the 'uniformizing' transformation

Note that the kernel estimator $\hat{D}_t(y)$ defined in eq. (28) is just the Nadaraya-Watson smoother, i.e., local average, of the variables $u_1, \ldots, u_n$ where $u_i = \mathbf{1}\{Y_i \leq y\}$. Similarly, $\bar{D}_t(y)$ defined in eq. (29) is just the Nadaraya-Watson smoother of the variables $v_1, \ldots, v_n$ where $v_i = \Lambda(\frac{y - Y_i}{h_0})$. In either case, it is only natural to try to consider a local linear smoother as an alternative to Nadaraya-Watson especially since, once again, our interest lies on the boundary, i.e., the case $t = n$.

Let $\hat{D}_t^{LL}(y)$ and $\bar{D}_t^{LL}(y)$ denote the local linear estimators of $D_t(y)$ based on either the indicator variables $\mathbf{1}\{Y_i \leq y\}$ or the smoothed variables $\Lambda(\frac{y - Y_i}{h_0})$ respectively. Keeping y fixed, $\hat{D}_t^{LL}(y)$ and $\bar{D}_t^{LL}(y)$ exhibit good behavior for estimation at the boundary, e.g.

smaller bias than either $\hat{D}_t(y)$ and $\bar{D}_t(y)$ respectively. However, there is no guarantee that these will be proper distribution functions as a function of y, i.e., being nondecreasing in y with a left limit of 0 and a right limit of 1; see (Li & Racine, 2007) for a discussion.

There have been several proposals in the literature to address this issue. An interesting one is the adjusted Nadaraya-Watson estimator of (Hall, Wolff, & Yao, 1999) which, however, is tailored towards nonparametric autoregression estimation rather than our setting where Y_t is regressed on t. Coupled with the fact that we are interested in the boundary case $t = n$, the equation yielding the adjusted Nadaraya-Watson weights do not always admit a solution.

One proposed solution put forward by (Hansen, 2004) involves a straightforward adjustment to the local linear estimator of a conditional distribution function that maintains its favorable asymptotic properties. The local linear versions of $\hat{D}_t(y)$ and $\bar{D}_t(y)$ adjusted via Hansen's (2004) proposal are given as follows:

$$\hat{D}_t^{LLH}(y) = \frac{\sum_{i=1}^{T} w_i^{\diamond} \mathbf{1}(Y_i \leq y)}{\sum_{i=1}^{T} w_i^{\diamond}} \quad \text{and} \quad \bar{D}_t^{LLH}(y) = \frac{\sum_{i=1}^{T} w_i^{\diamond} \Lambda(\frac{y-Y_i}{h_0})}{\sum_{i=1}^{T} w_i^{\diamond}}. \tag{30}$$

The weights $w_i^{\diamond}$ are defined by

$$w_i^{\diamond} = \begin{cases} 0 & \text{when } \hat{\beta}(t-t_i) > 1 \\ w_i(1-\hat{\beta}(t-t_i)) & \text{when } \hat{\beta}(t-t_i) \leq 1 \end{cases} \tag{31}$$

where

$$w_i = \frac{1}{b}\, K(\frac{t-t_i}{b}) \quad \text{and} \quad \hat{\beta} = \frac{\sum_{i=1}^{T}\ w_i(t-t_i)}{\sum_{i=1}^{T}\ w_i(t-t_i)^2}. \tag{32}$$

As with eq. (28)and (29), we can let $T = t$ or $T = t-1$ in the above, leading to a **fitted vs. predictive** local linear estimators of $D_t(y)$, by either $\hat{D}_t^{LLH}(y)$ or $\bar{D}_t^{LLH}(y)$.

3.4 Uniformization using Monotone Local Linear Distribution Estimation

Hansen's (2004) proposal replaces negative weights by zeros, and then renormalizes the nonzero weights. The problem here is that if estimation is performed on the boundary (as in the case with one-step ahead prediction of time-series), negative weights are crucially needed in order to ensure the extrapolation takes place with minimal bias. A recent proposal by (Das & Politis, 2017) addresses this issue by modifying the original, possibly nonmonotonic local linear distribution estimator $\bar{D}_t^{LL}(y)$ to construct a monotonic version denoted by $\bar{D}_t^{LLM}(y)$.

The Monotone Local Linear Distribution Estimator $\bar{D}_t^{LLM}(y)$ can be constructed by Algorithm 3.1 given below.

Algorithm 3.1 Monotone Local Linear Distribution Estimation

1. *Recall that the derivative of $\bar{D}_t^{LL}(y)$ with respect to y is given by*

$$\bar{d}_t^{LL}(y) = \frac{\frac{1}{h_0}\sum_{j=1}^{n} w_j \lambda(\frac{y-Y_j}{h_0})}{\sum_{j=1}^{n} w_j}$$

   where $\lambda(y)$ is the derivative of $\Lambda(y)$.

2. *Define a nonnegative version of $\bar{d}_t^{LL}(y)$ as $\bar{d}_t^{LL+}(y) = \max(\bar{d}_t^{LL}(y), 0)$.*

3. *To make the above a proper density function, renormalize it to area one, i.e., let*

$$\bar{d}_t^{LLM}(y) = \frac{\bar{d}_t^{LL+}(y)}{\int_{-\infty}^{\infty} \bar{d}_t^{LL+}(s)ds}. \tag{33}$$

4. *Finally, define $\bar{D}_t^{LLM}(y) = \int_{-\infty}^{y} \bar{d}_t^{LLM}(s)ds$.*

The above modification of the local linear estimator allows one to maintain monotonicity while retaining the negative weights that are helpful in problems which involve estimation at the boundary. As with eq. (28)and (29), we can let $T = t$ or $T = t - 1$ in the above, leading to a **fitted vs. predictive** local linear estimators of $D_t(y)$ that are monotone.

Different algorithms could also be employed for performing monotonicity correction on the original estimator $\bar{D}_t^{LL}(y)$; these are discussed in detail in (Das & Politis, 2017). In practice, Algorithm 3.1 is preferable because it is the fastest in term of implementation; notably, density estimates can be obtained in a fast way (using the Fast Fourier Transform) using standard functions in statistical software such as R. Computational speed is particularly important in constructing bootstrap prediction intervals since a large number of estimates of $\bar{D}_t^{LLM}(y)$ must be computed; the same is true for cross-validation implementation which is addressed next.

3.5 Cross-validation Bandwidth Choice for Model-Free Inference

There are two bandwidths, b and h_0, required to construct the estimators $\bar{D}_t(y)$, $\bar{D}_t^{LLH}(y)$ and $\bar{D}_t^{LLM}(y)$. This discussion first focuses on choice of b as it is the most crucial of the two. The following steps are recommended:

Algorithm 3.2 *BANDWIDTH DETERMINATION FOR MODEL-FREE INFERENCE*

1. *Perform the uniformizing transform described in* (25) *over the given time-series dataset* $Y_1, \ldots, Y_n$ *using either of the estimators* $\bar{D}_t(y)$, $\bar{D}_t^{LLH}(y)$ *or* $\bar{D}_t^{LLM}(y)$ *over* q *pre-defined bandwidths that span an interval of possible values.*

2. *Calculate the value of the Kolmogorov-Smirnov (KS) test statistic using the uniform distribution* $U[0,1]$ *as reference for each of these* q *cases.*

3. *From the full list of* q *values given in step (1) above pick a pre-defined number of bandwidths, say this is* p*, whose corresponding KS test statistic values are minimum. These represent the bandwidths which achieved the best transformation to 'uniformity' using* $\bar{D}_t(y)$, $\bar{D}_t^{LLH}(y)$ *or* $\bar{D}_t^{LLM}(y)$.

4. *Obtain the best bandwidth* b *among these* p *values by using one-sided cross-validation in a similar manner as described for the Model-Based case in Section 2.2. For this purpose let* $k < n$*, and suppose only subseries* $Y_1, \ldots, Y_k$ *has been observed. Denote* $\hat{Y}_{k+1}$ *the best predictor of* Y_{k+1} *based on the data* $Y_1, \ldots, Y_k$ *constructed using* $\bar{D}_t(y)$, $\bar{D}_t^{LLH}(y)$ *or* $\bar{D}_t^{LLM}(y)$ *and a value of* b *selected among the* p *values obtained above. Since* Y_{k+1} *is known, the quality of the predictor can be assessed. So, for each value of* b *we can form either* $PRESS(b) = \sum_{k=k_o}^{n-1} (\hat{Y}_{k+1} - Y_{k+1})^2$ *or* $PRESAR(b) = \sum_{k=k_o}^{n-1} |\hat{Y}_{k+1} - Y_{k+1}|$*; here* k_o *should be big enough so that estimation is accurate, e.g.,* k_o *can be of the order of* $\sqrt{n}$*. We then select the bandwidth* b *that minimizes* $PRESS(b)$*; alternatively, we can choose to minimize* $PRESAR(b)$ *if an* L_1 *measure of loss is preferred.*

5. *Coming back to the problem of selecting* h_0*, as in (Politis, 2013), our final choice is* $h_0 = h^2$ *where* $h = b/n$*. Note that an initial choice of* h_0 *needed (to perform uniformization, KS statistic generation and cross-validation to determine the optimal bandwidth* b*) can be set by any plug-in rule; the effect of choosing an initial value of* h_0 *is minimal.*

The above algorithm needs large data sizes in order to work well. In the case of smaller data sizes of, say, a hundred or so data points, it is recommended to omit steps (1)–(3) and directly perform steps (4) and (5) using the full range of q pre-defined bandwidths.

3.6 Estimation of the whitening transformation

To implement the whitening transformation (27), it is necessary to estimate Γ_n, i.e., the $n \times n$ covariance matrix of the random vector $\underline{Z}_n = (Z_1, \ldots, Z_n)'$ where the Z_t are the normal random variables defined in eq. (26).

As discussed in the analogous model-based problem in Section 2.1, there are two approaches towards positive definite estimation of Γ_n based on the sample $Z_1, \ldots, Z_n$. They are both based on the sample autocovariance defined as $\breve{\gamma}_k = n^{-1} \sum_{t=1}^{n-|k|} Z_t Z_{t+|k|}$ for $|k| < n$; for $|k| \geq n$, we define $\breve{\gamma}_k = 0$.

A. Fit a causal AR(p) model to the data $Z_1, \ldots, Z_n$ with p obtained via AIC minimization. Then, let $\hat{\Gamma}_n^{AR}$ be the $n \times n$ covariance matrix associated with the fitted AR model. Let $\hat{\gamma}_{|i-j|}^{AR}$ denote the i, j element of the Toeplitz matrix $\hat{\Gamma}_n^{AR}$. Using the Yule-Walker equations to fit the AR model implies that $\hat{\gamma}_k^{AR} = \breve{\gamma}_k$ for $k = 0, 1, \ldots, p$. For $k > p$, $\hat{\gamma}_k^{AR}$ can be found by solving (or just iterating) the difference equation that characterizes the (fitted) AR model; R automates this process via the `ARMAacf()` function.

B. Let $\hat{\Gamma}_n = \left[\hat{\gamma}_{|i-j|}\right]_{i,j=1}^n$ be the matrix estimator of (McMurry & Politis, 2010) where $\hat{\gamma}_s = \kappa(|s|/l)\breve{\gamma}_s$. Here, $\kappa(\cdot)$ can be any member of the *flat-top* family of compactly supported functions defined in (Politis, 2001) the simplest choice—that has been shown to work well in practice—is the trapezoidal, i.e.., $\kappa(x) = (\max\{1, 2 - |x|\})^+$ where $(y)^+ = \max\{y, 0\}$ is the positive part function, (Politis & Romano, 1994). Our final estimator of Γ_n will be $\hat{\Gamma}_n^\star$ which is a a positive definite version of $\hat{\Gamma}_n$ that is banded and Toeplitz; for example, $\hat{\Gamma}_n^\star$ may be obtained by shrinking $\hat{\Gamma}_n$ towards white noise or towards a second order estimator as described in McMurry and Politis (2015).

Estimating the 'uniformizing' transformation $D_t(\cdot)$ and the whitening trasformation based on Γ_n allows us to estimate the transformation $H_n : \underline{Y}_n \mapsto \underline{\epsilon}_n$. However, in order to put the Model-Free Prediction Principle to work, we also need to estimate the transformation H_{n+1} (and its inverse). To do so, we need a positive definite estimator for the matrix Γ_{n+1}; this can be accomplished by either of the two ways discussed in the above.

A′. Let $\hat{\Gamma}_{n+1}^{AR}$ be the $(n+1) \times (n+1)$ covariance matrix associated with the fitted AR(p) model.

B′. Denote by $\hat{\gamma}_{|i-j|}^\star$ the i, j element of $\hat{\Gamma}_n^\star$ for $i, j = 1, \ldots, n$. Then, define $\hat{\Gamma}_{n+1}^\star$ to be the symmetric, banded Toeplitz $(n+1) \times (n+1)$ matrix with ij element given by $\hat{\gamma}_{|i-j|}^\star$ when $|i - j| < n$. Recall that $\hat{\Gamma}_n^\star$ is banded with banding parameter l as discussed in (McMurry & Politis, 2015), so it is only natural to assign zeros to the two ij elements of $\hat{\Gamma}_{n+1}^\star$ that satisfy $|i - j| = n$, i.e., the bottom left and the top right.

Consider the 'augmented' vectors $\underline{Y}_{n+1} = (Y_1, \ldots, Y_n, Y_{n+1})'$, $\underline{Z}_{n+1} = (Z_1, \ldots, Z_n, Z_{n+1})'$ and $\underline{\epsilon}_{n+1} = (\epsilon_1, \ldots, \epsilon_n, \epsilon_{n+1})'$ where the values Y_{n+1}, Z_{n+1} and ϵ_{n+1} are yet unobserved. We now show how to obtain the inverse transformation $H_{n+1}^{-1} : \underline{\epsilon}_{n+1} \mapsto \underline{Y}_{n+1}$. Recall that $\underline{\epsilon}_n$

and $\underline{Y}_n$ are related in a one-to-one way via transformation H_n, so the values $Y_1, \ldots, Y_n$ are obtainable by $\underline{Y}_n = H_n^{-1}(\epsilon_n)$. Hence, we just need to show how to create the unobserved Y_{n+1} from $\underline{\epsilon}_{n+1}$; this is done in the following three steps.

Algorithm 3.3 *GENERATION OF UNOBSERVED DATAPOINT FROM FUTURE INNOVATIONS*

i. Let

$$\underline{Z}_{n+1} = C_{n+1}\underline{\epsilon}_{n+1} \tag{34}$$

where C_{n+1} is the (lower) triangular Cholesky factor of (our positive definite estimate of) Γ_{n+1}. From the above, it follows that

$$Z_{n+1} = \underline{c}_{n+1}\underline{\epsilon}_{n+1} \tag{35}$$

where $\underline{c}_{n+1} = (c_1, \ldots, c_n, c_{n+1})$ is a row vector consisting of the last row of matrix C_{n+1}.

ii. Create the uniform random variable

$$U_{n+1} = \Phi(Z_{n+1}). \tag{36}$$

iii. Finally, define

$$Y_{n+1} = D_{n+1}^{-1}(U_{n+1}); \tag{37}$$

of course, in practice, the above will be based on an estimate of $D_{n+1}^{-1}(\cdot)$.

Since $\underline{Y}_n$ has already been created using (the first n coordinates of) $\underline{\epsilon}_{n+1}$, the above completes the construction of $\underline{Y}_{n+1}$ based on $\underline{\epsilon}_{n+1}$, i.e., the mapping $H_{n+1}^{-1} : \underline{\epsilon}_{n+1} \mapsto \underline{Y}_{n+1}$.

3.7 Model-free predictors and prediction intervals

In the previous sections, it was shown how the construct the transformation $H_n : \underline{Y}_n \mapsto \underline{\epsilon}_n$ and its inverse $H_{n+1}^{-1} : \underline{\epsilon}_{n+1} \mapsto \underline{Y}_{n+1}$, where the random variables $\epsilon_1, \epsilon_2, \ldots,$ are i.i.d. Note that by combining eq. (35), (36) and (37) we can write the formula:

$$Y_{n+1} = D_{n+1}^{-1}\left(\Phi(\ \underline{c}_{n+1}\underline{\epsilon}_{n+1})\right).$$

Recall that $\underline{c}_{n+1}\underline{\epsilon}_{n+1} = \sum_{i=1}^{n} c_i\epsilon_i + c_{n+1}\epsilon_{n+1}$; hence, the above can be compactly denoted as

$$Y_{n+1} = g_{n+1}(\epsilon_{n+1}) \ \text{ where } \ g_{n+1}(x) = D_{n+1}^{-1}\left(\Phi\left(\sum_{i=1}^{n} c_i\epsilon_i + c_{n+1}x\right)\right). \tag{38}$$

Eq. (38) is the predictive equation required in the Model-free Prediction Principle; conditionally on $\underline{Y}_n$, it can be used like a model equation in computing the L_2– and L_1–optimal point predictors of Y_{n+1}. We will give these in detail as part of the general algorithms for the construction of Model-free predictors and prediction intervals.

Algorithm 3.4 MODEL-FREE (MF) PREDICTORS AND PREDICTION INTERVALS FOR Y_{n+1}

1. *Construct $U_1, \ldots, U_n$ by eq. (25) with $D_t(\cdot)$ estimated by either $\bar{D}_t(\cdot)$, $\bar{D}_t^{LLH}(\cdot)$ or $\bar{D}_t^{LLM}(\cdot)$; for all the 3 types of estimators, use the respective formulas with $T = t$.*

2. *Construct $Z_1, \ldots, Z_n$ by eq. (26), and use the methods of Section 3.6 to estimate Γ_n by either $\hat{\Gamma}_n^{AR}$ or $\hat{\Gamma}_n^{\star}$.*

3. *Construct $\epsilon_1, \ldots, \epsilon_n$ by eq. (27), and let $\hat{F}_n$ denote their empirical distribution.*

4. *The Model-free L_2–optimal point predictor of Y_{n+1} is then*

$$\hat{Y}_{n+1} = \int g_{n+1}(x) dF_n(x) = \frac{1}{n} \sum_{i=1}^{n} g_{n+1}(\epsilon_i)$$

   where the function g_{n+1} is defined in the predictive equation (38) *with $D_{n+1}(\cdot)$ being again estimated by either $\bar{D}_{n+1}(\cdot)$, $\bar{D}_{n+1}^{LLH}(\cdot)$ or $\bar{D}_{n+1}^{LLM}(\cdot)$ all with $T = t$.*

5. *The Model-free L_1–optimal point predictor of Y_{n+1} is given by the median of the set $\{g_{n+1}(\epsilon_i)$ for $i = 1, \ldots, n\}$.*

6. *Prediction intervals for Y_{n+1} with prespecified coverage probability can be constructed via the Model-free Boootstrap of Algorithm A.1 based on either the L_2– or L_1–optimal point predictor.*

Algorithm 3.4 used the construction of $\bar{D}_t(\cdot)$, $\bar{D}_t^{LLH}(\cdot)$ or $\bar{D}_t^{LLM}(\cdot)$ with $T = t$; using $T = t - 1$ instead, leads to the *predictive* version of the algorithm.

Algorithm 3.5 PREDICTIVE MODEL-FREE (PMF) PREDICTORS AND PREDICTION INTERVALS FOR Y_{n+1}

The algorithm is identical to Algorithm 3.5 except for using $T = t - 1$ instead of $T = t$ in the construction of $\bar{D}_t(\cdot)$, $\bar{D}_t^{LLH}(\cdot)$ and $\bar{D}_t^{LLM}(\cdot)$.

Remark 3.2 Under a model-free setup of a locally stationary time series, (Paparoditis & Politis, 2002) proposed the Local Block Bootstrap (LBB) in order to generate pseudo-series

$Y_1^*, \ldots, Y_n^*$ whose probability structure mimics that of the observed data $Y_1, \ldots, Y_n$. The Local Block Bootstrap has been found useful for the construction of confidence intervals; see (Dowla A. & Politis D.N, 2003) and (Dowla, Paparoditis, & Politis, 2013). However, it is unclear if/how the LBB can be employed for the construction of predictors and prediction intervals for Y_{n+1}.

Recall that when the theoretical transformation H_n is employed, the variables $\epsilon_1, \ldots, \epsilon_n$ are i.i.d. $N(0,1)$. Due to the fact that features of H_n are unknown and must be estimated from the data, the practically available variables $\epsilon_1, \ldots, \epsilon_n$ are only approximately i.i.d. $N(0,1)$. However, their empirical distribution of $\hat{F}_n$ converges to $F = \Phi$ as $n \to \infty$. Hence, it is possible to use the limit distribution $F = \Phi$ in instead of $\hat{F}_n$ in both the construction of point predictors and the prediction intervals; this is an application of the Limit Model-Free (LMF) approach as discussed in (Politis, 2015).

The LMF Algorithm is simpler than Algorithm 3.5 as the first three steps of the latter can be omitted. As a matter of fact, the LMF Algorithm is totally based on the inverse transformation $H_{n+1}^{-1} : \underline{\epsilon}_{n+1} \mapsto \underline{Y}_{n+1}$; the forward transformation $H_n : \underline{Y}_n \mapsto \underline{\epsilon}_n$ is not needed at all. But for the inverse transformation it is sufficient to estimate $D_t(y)$ by the step functions $\hat{D}_t(y)$, $\hat{D}_t^{LLH}(y)$ or $\hat{D}_t^{LLM}(y)$ with the understanding that their inverse must be a *quantile* inverse; recall that the quantile inverse of a distribution $D(y)$ is defined as $D^{-1}(\beta) = \inf\{y \text{ such that } D(y) \geq \beta\}$.

Algorithm 3.6 Limit Model-free (LMF) predictors and prediction intervals for Y_{n+1}

1. *The LMF L_2–optimal point predictor of Y_{n+1} is*

$$\hat{Y}_{n+1} = \int g_{n+1}(x) d\Phi(x) \tag{39}$$

   where the function g_{n+1} is defined in the predictive equation (38) *where $D_{n+1}(\cdot)$ is estimated by either $\hat{D}_{n+1}(\cdot)$, $\hat{D}_{n+1}^{LLH}(\cdot)$ or $\hat{D}_{n+1}^{LLM}(\cdot)$ all with $T = t - 1$.*

2. *In practice, the integral* (39) *can be approximated by Monte Carlo, i.e.,*

$$\int g_{n+1}(x) d\Phi(x) \simeq \frac{1}{M} \sum_{i=1}^{M} g_{n+1}(x_i)$$

   where $x_1, \ldots, x_M$ are generated as i.i.d. $N(0,1)$, and M is some large integer.

3. *Using the above Monte Carlo framework, the LMF L_1–optimal point predictor of Y_{n+1} can be approximated by the median of the set $\{g_{n+1}(x_i)$ for $i = 1, \ldots, M\}$.*

4. *Prediction intervals for Y_{n+1} with prespecified coverage probability can be constructed via the LMF Booostrap of Algorithm A.2 based on either the L_2– or L_1–optimal point predictor.*

Remark 3.3 Interestingly, there is a closed-form solution for the LMF L_1–optimal point predictor of Y_{n+1} that can also be used in Step 5 of Algorithm 3.4. To elaborate, first note that under the assumed weak dependence, e.g. strong mixing, of the series $\{Y_t\}$ (and therefore also of $\{Z_t\}$), we have the following approximations (for large n), namely:

$$Median\left(Z_{n+1}|\mathcal{F}_1^n(Z)\right) \simeq Median\left(Z_{n+1}|\mathcal{F}_{-\infty}^n(Z)\right)$$

$$= Median\left(Z_{n+1}|\mathcal{F}_{-\infty}^n(Y)\right) \simeq Median\left(Z_{n+1}|\mathcal{F}_1^n(Y)\right).$$

Now eq. (36) and (37) imply that $Y_{n+1} = D_{n+1}^{-1}\left(\Phi(Z_{n+1})\right)$. Since $D_{n+1}(\cdot)$ and $\Phi(\cdot)$ are strictly increasing functions, it follows that the Model-free L_1–optimal predictor of Y_{n+1} equals

$$Median\left(Y_{n+1}|\mathcal{F}_1^n(Y)\right) = D_{n+1}^{-1}\left(\Phi\left(Median\left(Z_{n+1}|\mathcal{F}_1^n(Y)\right)\right)\right)$$

$$\simeq D_{n+1}^{-1}\left(\Phi\left(Median\left(Z_{n+1}|\mathcal{F}_1^n(Z)\right)\right)\right) = D_{n+1}^{-1}\left(\Phi\left(E\left(Z_{n+1}|\mathcal{F}_1^n(Z)\right)\right)\right), \tag{40}$$

the latter being due to the symmetry of the normal distribution of Z_{n+1} given $\mathcal{F}_1^n(Z)$. But, as in eq. (7), we have $E\left(Z_{n+1}|\mathcal{F}_1^n(Z)\right) = \phi_1(n)Z_n + \phi_2(n)Z_{n-1} + \ldots + \phi_n(n)Z_1$ where $(\phi_1(n), \cdots, \phi_n(n))' = \Gamma_n^{-1}\gamma(n)$. Plugging-in either $\bar{D}_{n+1}(\cdot)$, $\bar{D}_{n+1}^{LLH}(\cdot)$ or $\bar{D}_{n+1}^{LLM}(\cdot)$ in place of $D_{n+1}(\cdot)$ in eq. (40), and also employing consistent estimates of Γ_n and $\gamma(n)$ completes the calculation. As discussed in Section 3.6, Γ_n can be estimated by either $\hat{\Gamma}_n^{AR}$ or by the positive definite banded estimator $\hat{\Gamma}_n^{\star}$ with a corresponding estimator for $\gamma(n)$; see (McMurry & Politis, 2015) for details.

Remark 3.4 (Robustness of LMF approach) The LMF approach focuses completely on the predictive equation (38) for which an estimate of (the inverse of) $D_{n+1}(\cdot)$ must be provided; interestingly, estimating $D_t(y)$ for $t \neq n+1$ is nowhere used in Algorithm 3.6. In the usual case where the kernel $K(\cdot)$ is chosen to have compact support, estimating $D_{n+1}(\cdot)$ is only based on the last b data values $Y_{n-b+1}, \ldots, Y_n$. Hence, in order for the LMF Algorithm 3.6 to be valid, the sole requirement is that the subseries $Y_{n-b+1}, \ldots, Y_n, Y_{n+1}$ is approximately stationary. In other words, the first (and biggest) part of the data, namely $Y_1, \ldots, Y_{n-b}$, can suffer from arbitrary nonstationarities, change points, outliers, etc. *without the LMF predictive inference for Y_{n+1} being affected;* this robustness of the LMF approach is highly advantageous.

3.8 Discrete-valued time series

Untill now, it has been assumed that $D_t(y)$ is (absolutely) continuous in y for all t; in this subsection, we briefly discuss a departure from this assumption.

Throughout subsection 3.8 we will assume that the locally stationary time series $\{Y_t\}$ takes values in a countable set $S \subset \mathbf{R}$; as an example, consider the case of a finite state Markov chain whose first marginal changes smooth (and smoothly) with time. It is apparent that $D_t(y)$ is a step function; hence, step function estimators such as $\hat{D}_t(y)$, $\hat{D}_t^{LLH}(y)$ or $\hat{D}_t^{LLM}(y)$ are preferable to their smoothed counterparts $\bar{D}_t(y)$, $\bar{D}_t^{LLH}(y)$ or $\bar{D}_t^{LLM}(y)$ since the latter assign positive probabilities to values $y \notin S$.

Fortunately, the LMF methodology of Algorithm 3.6 can be employed based on just the step function estimators $\hat{D}_t(y)$, $\hat{D}_t^{LLH}(y)$ or $\hat{D}_t^{LLM}(y)$. Note that with discrete data, predicting Y_{n+1} by a conditional mean or median makes little sense since the latter will likely not be in the set S; it is more appropriate to adopt a 0-1 loss function and predict Y_{n+1} by the *mode* of the conditional distribution. A prediction interval is not appropriate either unless the set S is of lattice form—and even then, problems ensue regarding non-attainable α–levels. It is thus more informative to present an estimate of the conditional distribution instead of summarizing the latter into a prediction interval.

A version of the LMF algorithm for discrete valued data is given below; (for details see (Politis, 2015).

Algorithm 3.7 LMF BOOTSTRAP FOR PREDICTIVE DISTRIBUTION OF DISCRETE-VALUED Y_{n+1}

1. *Based on the data $\underline{Y}_n$, estimate the inverse transformation H_n^{-1} by $\hat{H}_n^{-1}$ (say). In addition, estimate g_{n+1} by $\hat{g}_{n+1}$.*

2. *(a) Generate bootstrap pseudo-data $\varepsilon_1^*, ..., \varepsilon_n^*$ as i.i.d. from $F = \Phi$.*
   (b) Use the inverse transformation $\hat{H}_n^{-1}$ to create pseudo-data in the Y domain, i.e., let $\underline{Y}_n^* = (Y_1^*, ..., Y_n^*)' = \hat{H}_n^{-1}(\varepsilon_1^*, ..., \varepsilon_n^*)$.
   (c) Based on the bootstrap pseudo-data $\underline{Y}_n^*$, re-estimate the transformation H_n and its inverse H_n^{-1} by $\hat{H}_n^*$ and $\hat{H}_n^{-1*}$ respectively. In addition, re-estimate g_{n+1} by $\hat{g}_{n+1}^*$.
   (d) Calculate a bootstrap pseudo-value Y_{n+1}^{**} as the point $\hat{g}_{n+1}^*(\underline{Y}_n, \varepsilon)$ where ε is generated from $F = \Phi$.

3. *Steps (a)—(d) in the above should be repeated B times (for some large B), and the B bootstrap replicates of the pseudo-values Y_{n+1}^{**} are collected in the form of an empirical distribution which is our Model-free estimate of the predictive distribution of Y_{n+1}; the <u>mode</u> of this distribution is the LMF optimal predictor of Y_{n+1} under 0-1 loss.*

3.9 Special case: strictly stationary data

It is interesting to consider what happens if/when the data $Y_1, \ldots, Y_n$ are a stretch of a strictly stationary time series $\{Y_t\}$. Of course, a time series that is strictly stationary is a *a fortiori* locally stationary; so all the aforementioned procedures should work *verbatim*. Nevertheless, one could take advantage of the stationarity to obtain better estimators; effectively, one can take the bandwidth b to be comparable to n, i.e., employ global—as opposed to local—estimators.

To elaborate, in the stationary case the distribution $D_t(y)$ does not depend on t at all. Hence, for the purposes of the LMF Algorithm 3.6—as well as the discrete data Algorithm 3.7—we can estimate $D_t(y)$ by the regular (non-local) empirical distribution

$$\hat{D}(y) = n^{-1} \sum_{t=1}^{n} \mathbf{1}\{Y_t \leq y\}.$$

Furthermore, for the purposes of Algorithm 3.4 we can estimate the (assumed smooth) $D_t(y)$ by the smoothed empirical distribution

$$\bar{D}(y) = n^{-1} \sum_{t=1}^{n} \Lambda(\frac{y - Y_t}{h_0})$$

where h_0 is a positive bandwidth parameter satisfying $h_0 \to 0$ as $n \to \infty$. As mentioned in Section 3.5, the optimal rate is $h_0 \sim n^{-2/5}$ when the estimand $D_t(y)$ is sufficiently smooth in y.

3.10 Local stationarity in a higher-dimensional marginal

The success of the theoretical transformation of Section 3.1 in transforming the data vector $\underline{Y}_n$ to the vector of i.i.d. components $\underline{\epsilon_n}$ hinges on two conditions: (a) the nonstationarity of $\{Y_t\}$ is only due to nonstationarity in its first marginal $D_t(\cdot)$, and (b) the instantaneous transformation to Gaussianity also manages to create a Gaussian random vector, i.e., all its finite-dimensional marginals are Gaussian. Both of these conditions can be empirically checked. For example, condition (a) can be checked by looking at some features of interest of the mth (say) marginal, e.g., looking at the autocorrelation $\mathrm{Corr}(Y_t, Y_{t+m})$ estimated over different subsamples of the data, and checking whether it depends on t. Condition (b) can be checked by performing a normality test, e.g., Shapiro-Wilk test, or other diagnostics, e.g., quantile plot, on selected linear combinations of m consecutive components of the random vector.

Interestingly, if either condition (a) or (b) seem to fail, there is a single solution to address the problem, namely blocking the time series. To elaborate, one would then create

blocks of data by defining $B_t = (Y_t, \ldots, Y_{t+m-1})'$ for $t = 1, \ldots, q$ with $q = n - m + 1$. Now focus on the multivariate time series dataset $\{B_1, \ldots, B_q\}$, and let $D_t^{(m)}(\cdot)$ denote the distribution function of vector B_t which will be assumed to vary smoothly (and slowly) with t as in Remark 3.1.

Using the (Rosenblatt, 1952) transformation, we can now map B_t to a random vector V_t that has components[3] i.i.d. Uniform (0,1), and then do the Gaussian transformation and whitening as required by the Model-Free Principle. Thus, when the time series $\{Y_t\}$ is locally stationary in its mth marginal, the algorithm to transform the dataset $\underline{Y}_n = (Y_1, \ldots, Y_n)'$ to an i.i.d. dataset goes as follows.

1. From the dataset $\underline{Y}_n = (Y_1, \ldots, Y_n)'$, create blocks/vectors $B_t = (Y_t, \ldots, Y_{t+m-1})'$ for $t = 1, \ldots, q$ with $q = n - m + 1$.

2. Use the Rosenblatt transformation to map the multivariate dataset $\{B_1, \ldots, B_q\}$ to the dataset $\{V_1, \ldots, V_q\}$; here $V_t = (V_t^{(1)}, \ldots, V_t^{(m)})'$ is a random vector having components that are i.i.d. Uniform (0,1).

3. Let $Z_t^{(j)} = \Phi^{-1}(V_t^{(j)})$ for $j = 1, \ldots, m$, and $t = 1, \ldots, q$ where Φ is the cdf of a standard normal. Note that, for each t, the variables $Z_t^{(1)}, \ldots, Z_t^{(m)}$ are i.i.d. $N(0, 1)$.

4. Define the vector time series $Z_t = (Z_t^{(1)}, \ldots, Z_t^{(m)})'$ that is multivariate Gaussian. Estimate the (matrix) autocovariance sequence $\mathrm{Cov}(Z_t, Z_{t+k})$ for $k = 0, 1, \ldots$, and use it to 'whiten' the sequence $Z_1, \ldots, Z_q$, i.e., to map it (in a one-to-one way) to the i.i.d. sequence $\zeta_1, \ldots, \zeta_q$; here, $\zeta_t \in \mathbf{R}^m$ is a random vector having components that are i.i.d. $N(0, 1)$.

In Step 2 above, the mth dimensional Rosenblatt transformation can be estimated in practice using a local average or local linear estimator, i.e., a multivariate analog of $\bar{D}_t(\cdot)$, $\bar{D}_t^{LLH}(\cdot)$ or $\bar{D}_t^{LLM}(\cdot)$. Regarding Step 4, standard methods exist to estimate the (matrix) autocovariance of Z_t with Z_{t+k}; see e.g. (Jentsch & Politis, 2015). Finally, note that the map $H_n : \underline{Y}_n \mapsto (\zeta_1, \ldots, \zeta_q)'$ is invertible since all four steps given above are one-to-one. Hence, Model-free prediction can take place based on a multivariate version of the Model-free Prediction Principle of (Politis, 2013); the details are straightforward.

[3]Recall that the (Rosenblatt, 1952) transformation maps an arbitrary random vector $\underline{Y}_m = (Y_1, \ldots, Y_m)'$ having absolutely continuous joint distribution onto a random vector $\underline{V}_m = (V_1, \ldots, V_m)'$ whose entries are i.i.d. Uniform(0,1); this is done via the probability integral transform based on conditional distributions. To elaborate, for $k > 1$ define the conditional distributions $D_k(y_k|y_{k-1}, \ldots, y_1) = P\{Y_k \leq y_k|Y_{k-1} = y_{k-1}, \ldots, Y_1 = y_1\}$, and let $D_1(y_1) = P\{Y_1 \leq y_1\}$. Then, the (Rosenblatt, 1952) transformation amounts to letting $V_1 = D_1(Y_1), V_2 = D_2(Y_2|Y_1), V_3 = D_3(Y_3|Y_2, Y_1), \ldots$, and $V_m = D_m(Y_m|Y_{m-1}, \ldots, Y_2, Y_1)$.

4 Diagnostics for Model-Free Inference

The steps outlined in Section 3.1 for Model-Free inference involve generating samples from both uniform $U[0,1]$ and standard normal distributions. Careful analysis is necessary to ensure that the samples generated are from the correct distributions failing which the Model-Free point and interval predictors will be inaccurate. The following discussion serves as an aid to the practitioner to ensure realization of optimal performance for both point prediction and prediction interval generation using the Model-Free methodology.

4.1 QQ-plots after uniformization

The success of the uniformization step outlined in Section 3.1 can be visually verified using QQ-plots of the obtained uniform samples versus samples obtained from an ideal uniform distribution which is available in standard statistical software such as R. Any deviations in these curves from linearity should be closely investigated for possible issues wrt choice of bandwidth during cross-validation as it can impact both point prediction and prediction interval generation.

4.2 Shapiro-Wilk test for joint normality

The random vector $\underline{Z}_n = (Z_1, \ldots, Z_n)'$ from Section 3.6 should be tested for normality in order to ensure that the described whitening transformation successfully produces i.i.d. normal samples. Marginal normality of the data Z can be verified by gauging linearity of QQ-plots versus the standard normal distribution. Furthermore the Cramer-Wold theorem states that any linear combination of jointly normal variables is univariate normal. This can be used to empirically verify whether the joint normality requirement is violated by taking any linear combination i.e. for example a pair or triplet of variables from the set $\underline{Z}_n = (Z_1, \ldots, Z_n)'$ and verify their normality using the Shapiro-Wilk test. An example of this is provided in Figure 2 where for a given λ we form the linear combination $(1-\lambda)Z_i + \lambda Z_{i+1}$ over all obtained values $\underline{Z}_n = (Z_1, \ldots, Z_n)'$ and calculate the mean value of the Shapiro-Wilk test statistic. This is done over a range of λ values. As can be seen from the plot sufficiently high values of the test statistic are obtained which indicates that from this particular test we cannot conclude that joint normality has been violated. Further tests can be done by forming linear combinations over pairs of non-successive values of Z.

4.3 Kolomogorov-Smirnov test for i.i.d. standard normal samples

Provided that the inputs are jointly normal the whitening transformation described in Section 3.6 produces i.i.d. standard normal variables. The covariance matrix used in this

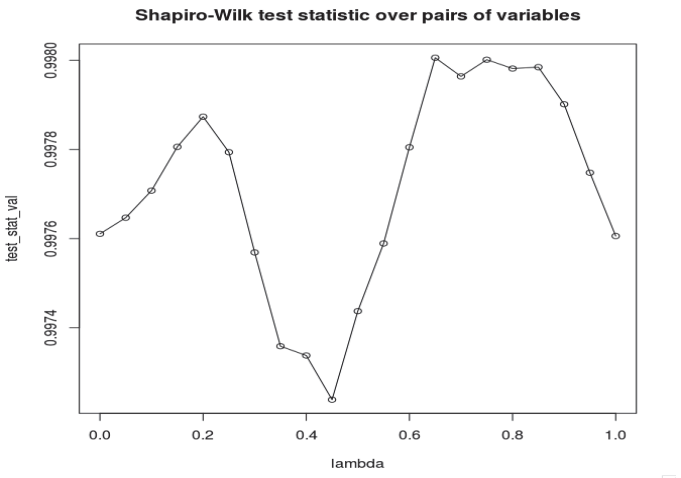

Figure 2: Values of Shapiro-Wilk test statistic for joint normality test. Note that corresponding p-values range from 0.09 to 0.29.

step can be derived either by fitting a causal AR(p) model to $\underline{Z}_n = (Z_1, \ldots, Z_n)'$ or using the flat-top kernel banded, tapered estimator outlined in (McMurry & Politis, 2010). To verify that the data generated after whitening are standard normal a Kolmogorov-Smirnov test can be used with the reference distribution as $N[0, 1]$.

4.4 Independence test of standard normal samples

The success of the Model-Free procedure involves the ability to produce i.i.d. data after a series of invertible transformations. In the case of Locally Stationary Time Series independence of the data produced at the final step after applying the whitening transformation can be verified visually using an autocorrelation function (ACF) plot as the data are approximately standard normal. An example of this is given in Figure 3 where it can be noticed from the ACF plot that the Model-Free transformations were successful in producing decorrelated and therefore i.i.d. (normal) data.

5 Model-Free vs. Model-Based Inference: empirical comparisons

The performance of the Model-Free and Model-Based predictors described above are empirically compared using both simulated and real-life datasets based on point prediction and also calculation of prediction intervals. The Model-Based local constant and local linear methods are denoted as MB-LC and MB-LL respectively. Model-Based predictors MB-LC

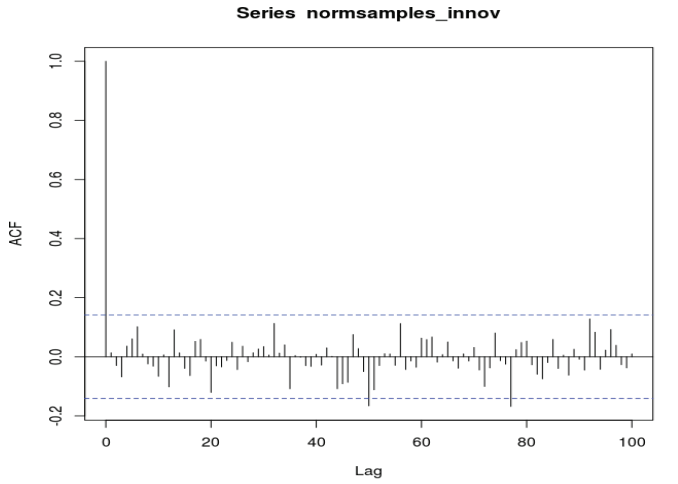

Figure 3: Autocorrelation plot showing decorrelation/independence of data after whitening transformation

and MB-LL are described in Section 2. The Model-Free methods using local constant, local linear (Hansen) and local linear (Monotone) using the flat-top tapered covariance estimator are denoted as MF-LC, MF-LLH, MF-LLM. Model-Free methods using local constant, local linear (Hansen) and local linear (Monotone) using the covariance estimator obtained from fitting a causal AR(p) model are denoted as MF-LC-ARMA, MF-LLH-ARMA, MF-LLM-ARMA. Model-Free predictors are described in Section 3. The covariance estimators using the flat-top tapered kernel and fitting an AR(p) model are discussed in Section 3.6. Results are also shown for the LMF counterparts of these methods which are denoted as LMF-LC, LMF-LLH, LMF-LLM and LMF-LC-ARMA, LMF-LLH-ARMA, LMF-LLM-ARMA respectively. Results for all methods are given for both fitted (F) and predictive (P) residuals. Following metrics are used to compare the estimators:

1. Point prediction performance as indicated by Bias and Mean Squared Error (MSE) on simulated and real-life datasets using all Model-Based and Model-Free methods listed above.

2. Bootstrap performance as indicated by coverage probability (CVR), mean length of prediction intervals and standard deviation (sd) of length of prediction intervals. All prediction interval metrics given in the following tables have been generated based on a nominal coverage of 90%.

5.1 Simulation: Additive model with stationary AR(5) errors

Data Y_i for $t = 1, \ldots, 1000$ were simulated as per model (1) with trend as in eq. (3), i.e., $\mu(t) = \mu_{[0,1]}(a_t)$ with $a_t = (t-1)/n$ and $\mu_{[0,1]}(x) = \sin(2\pi x)$. The series W_t is constructed via an AR(5) model driven by errors V_t that are i.i.d. $N(0, \tau^2)$; with $\tau = 0.14$. The AR(5) coefficients are set to 0.5, 0.1, 0.1, 0.1, 0.1. Sample size n is set to 1000. Point prediction and prediction intervals are measured for boundary point $n = 1000$. Bandwidths for estimating the trend are calculated using the cross-validation techniques for Model-Based and Model-Free cases described in Sections 2.2 and 3.5 respectively.

Results for point prediction including bias and mean square error (MSE) over all MB and MF methods are shown in Table 1 below. A total of 500 realizations of the dataset were used for measuring point prediction performance.

Results for prediction intervals including CVR, length and standard deviation of the predicted intervals over all MB and MF methods are shown in Table 2 below. A total of 250 realizations were used for measuring prediction interval performance. The number of bootstrap replications B was set to 250.

From point-prediction results on this dataset it can be seen that one of the best predictors is MB-LL; this is expected since the LL regression estimator is great for extrapolation, and the innovations are generated using an AR model which is directly employed in the MB-LL estimator. Nevertheless, predictors MF-LLM and MF-LLM-ARMA appear equally as good which is re-assuring and surprising at the same time; it appears that—as with the case of regression with independent errors (Das & Politis, 2017)—the monotonicity correction in the LLM distribution estimator has minimal effect on the center of the distribution that is used for point prediction. The MF-ARMA and LMF-ARMA outperform their respective MF and LMF counterparts for point prediction; this is consistent with that fact that the data is generated by an AR process and therefore the covariance estimator using AR(p) estimation outperforms its flat-top tapered counterpart. However the MF-LLM, LMF-LLM, MF-LLM-ARMA and LMF-LLM-ARMA estimators give the best prediction intervals when both coverage probabilities and mean interval lengths are considered. This is a somewhat surprising result given the fact that the data was generated using an AR(5) model, and one would expect that the model-based estimator MB-LL would perform comparably with its MF counterparts, i.e., MF-LLM and MF-LLM-ARMA, in terms of prediction intervals.

Among the MF estimators it is the MF-LLM, LMF-LLM, MF-LLM-ARMA and LMF-LLM-ARMA methods that perform better than their LC and LLH counterparts both for the flat-top tapered and AR(p) based covariance estimators. This improvement can be attributed to using negative weights for estimation at the boundary with the Monotone Local Linear Distribution estimator i.e. the LLM methods.

As before prediction interval coverage is enhanced using predictive as compared to fit-

ted residuals which is consistent with the results of interval coverage using both types of residuals as discussed for the regression case in (Politis, 2013).

5.2 Simulation: Additive model with nonlinearly generated errors

Data Y_i for $t = 1, \ldots, 1000$ were simulated from model (1) with trend as in eq. (3), i.e., $\mu(t) = \mu_{[0,1]}(a_t)$ with $a_t = (t-1)/n$ and $\mu_{[0,1]}(x) = 5 * \sin(2\pi x)$. The series W_t is now constructed via the nonlinear model given below:

$$W_t = \begin{cases} 1 + \alpha W_{t-1} + e_t & if\ W_{t-1} \leq r \\ -1 + \beta W_{t-1} + \gamma e_t & if\ W_{t-1} > r \end{cases} \tag{41}$$

where the errors e_t are assumed i.i.d. $N(0, \tau^2)$. Eq. (41) describes a TAR(1) model, i.e., Threshold Autoregression of order 1; see (Tong, 2011) and the references therein. For our implementation, we chose $\tau = 0.4$, $\alpha = 0.5$, $\beta = -0.6$, $r = 0.6$, $\gamma = 1$; the initial value of W_t is set to 0, and $n = 1000$. A scatterplot showing W_t versus W_{t-1} is shown in Figure 4. The process of eq. (41) is not zero-mean; however its mean is removed during detrending either with Model-Based or Model-Free methods. Point prediction and prediction intervals are measured for boundary point $n = 1000$. Bandwidths for estimating the trend are calculated using the cross-validation techniques for Model-Based and Model-Free cases described in Sections 2.2 and 3.5 respectively.

Results for point prediction including bias and mean square error (MSE) over all MB and MF methods are shown in Table 3 below. A total of 500 realizations of the dataset were used for measuring point prediction performance.

Results for prediction intervals including CVR, length and standard deviation of the predicted intervals over all MB and MF methods are shown in Table 4 below. A total of 250 realizations were used for measuring prediction interval performance. The number of bootstrap replications B was set to 250.

From point-prediction results on this dataset it can be seen that the MF-LLM-ARMA and LMF-LLM-ARMA estimators give the best performance. The MF-ARMA and LMF-ARMA outperform their respective MF and LMF counterparts for point prediction. This is consistent with that fact that the data is not generated by an MA process and therefore the covariance estimator using AR(p) estimation outperforms its flat-top tapered counterpart which assumes an MA model. The MF-LLM, LMF-LLM, MF-LLM-ARMA and LMF-LLM-ARMA estimators give the best prediction intervals when both coverage probabilities and mean interval lengths are considered. These results are somewhat expected since the

Table 1: Point Prediction performance for AR(5) dataset

Prediction Method	Residual Type	Bias	MSE
MB-LC	P	-2.899e-02	2.878e-02
	F	-3.310e-02	2.923e-02
MB-LL	P	-3.031e-03	2.848e-02
	F	-7.315e-03	2.841e-02
MF-LC	P	-3.910e-02	2.955e-02
	F	4.327e-02	2.949e-02
MF-LLH	P	-3.591e-02	2.996e-02
	F	-4.177e-02	3.000e-02
MF-LLM	P	-2.716e-02	2.832e-02
	F	-3.599e-02	2.909e-02
LMF-LC	P	-3.915e-02	2.961e-02
	F	-4.349e-02	2.953e-02
LMF-LLH	P	-3.691e-02	2.996e-02
	F	-4.224e-02	3.010e-02
LMF-LLM	P	-2.753e-02	2.855e-02
	F	-3.614e-02	2.915e-02
MF-LC-ARMA	P	-3.418e-02	2.929e-02
	F	-3.932e-02	2.920e-02
MF-LLH-ARMA	P	-3.067e-02	2.941e-02
	F	-3.766e-02	2.917e-02
MF-LLM-ARMA	P	-2.226e-02	2.829e-02
	F	-3.219e-02	2.876e-02
LMF-LC-ARMA	P	-3.452e-02	2.957e-02
	F	-3.968e-02	2.942e-02
LMF-LLH-ARMA	P	-3.141e-02	2.942e-02
	F	-3.776e-02	2.927e-02
LMF-LLM-ARMA	P	-2.229e-02	2.824e-02
	F	-3.300e-02	2.893e-02

Table 2: Interval estimation performance using bootstrap for AR(5) dataset

Prediction Method	Residual Type	CVR	Mean Length	SD Length
MB-LC	P	0.88	7.001e-01	1.781e-01
	F	0.83	5.598e-01	2.013e-01
MB-LL	P	0.92	7.802e-01	1.718e-01
	F	0.88	7.039e-01	1.725e-01
MF-LC	P	0.85	7.443e-01	1.500e-01
	F	0.83	6.362e-01	1.709e-01
MF-LLH	P	0.88	7.489e-01	1.422e-01
	F	0.84	6.495e-01	1.234e-01
MF-LLM	P	0.89	7.343e-01	1.386e-01
	F	0.88	6.422e-01	1.229e-01
LMF-LC	P	0.86	7.424e-01	1.515e-01
	F	0.83	6.373e-01	1.492e-01
LMF-LLH	P	0.88	7.582e-01	1.386e-01
	F	0.85	6.534e-01	1.275e-01
LMF-LLM	P	0.89	7.423e-01	1.401e-01
	F	0.88	6.460e-01	1.278e-01
MF-LC-ARMA	P	0.85	7.452e-01	1.485e-01
	F	0.80	6.317e-01	1.421e-01
MF-LLH-ARMA	P	0.85	7.474e-01	1.416e-01
	F	0.84	6.569e-01	1.286e-01
MF-LLM-ARMA	P	0.88	7.362e-01	1.442e-01
	F	0.87	6.502e-01	1.264e-01
LMF-LC-ARMA	P	0.85	7.437e-01	1.485e-01
	F	0.82	6.382e-01	1.452e-01
LMF-LLH-ARMA	P	0.86	7.428e-01	1.389e-01
	F	0.85	6.564e-01	1.254e-01
LMF-LLM-ARMA	P	0.88	7.422e-01	1.423e-01
	F	0.87	6.519e-01	1.278e-01

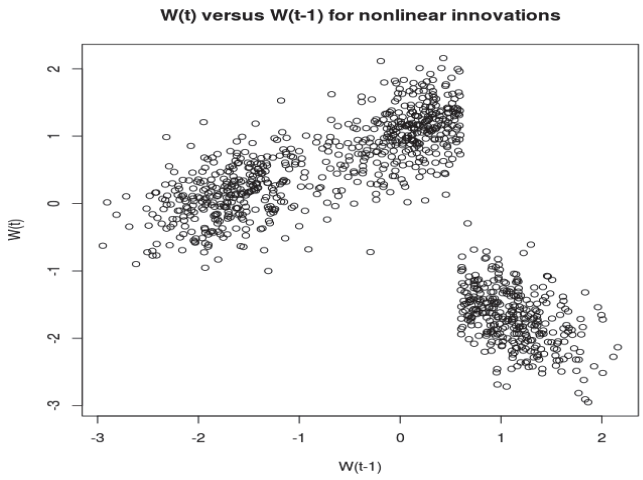

Figure 4: Nonlinear time series scatterplot of W_t versus W_{t-1}.

innovations are generated using a nonlinear model and the MB methods use a linear predictor. Therefore MF-LLM and LMF-LLM estimators perform better than their model-based counterparts i.e. the MB-LL methods. However it is striking to see a Model-Free method outperform the Model-Based ones when the additive model is true.

It can also be seen that for most cases prediction interval coverage is enhanced using predictive as compared to fitted residuals which is consistent with the results of interval coverage using both types of residuals as discussed for the regression case in (Politis, 2013).

6 Real-life example: Speleothem data

The Speleothem dataset first discussed in (Fleitmann et al., 2003) and further analyzed in (Mudelsee, 2014) is an interesting real-life example to compare metrics of point prediction and prediction intervals for all MB and MF estimators described before. This dataset which is shown in Figure 1 contains oxygen isotope record (the ratio of ^{18}O to ^{16}O) from stalagmite Q5 from southern Oman over the past 10,300 years. The oxygen isotope ratio obtained from the speleothem climate archive serves as a proxy variable for the actual climate variable **monsoon rainfall**. The full dataset has Y_i for $t = 1, \ldots, 1345$ points which are in general obtained with unequal spacing. The following points should be noted in the context of our analysis of the speleothem proxy dataset:

1. One important application of proxy data obtained from climate archives is prediction of the unobserved climate variable values. This prediction is based on known values

Table 3: Point Prediction performance for nonlinear dataset

Prediction Method	Residual Type	Bias	MSE
MB-LC	P	-1.894e-01	8.420e-01
	F	-1.897e-01	8.542e-01
MB-LL	P	-1.109e-01	8.003e-01
	F	-1.082e-01	8.048e-01
MF-LC	P	-1.697e-01	8.616e-01
	F	-1.937e-01	8.407e-01
MF-LLH	P	-1.134e-01	8.345e-01
	F	-1.193e-01	8.137e-01
MF-LLM	P	-2.418e-02	8.208e-01
	F	-1.770e-02	7.886e-01
LMF-LC	P	-1.631e-01	8.671e-01
	F	-1.858e-01	8.456e-01
LMF-LLH	P	-1.004e-01	8.338e-01
	F	-1.108e-01	8.420e-01
LMF-LLM	P	-1.339e-02	8.287e-01
	F	-8.603e-03	7.941e-01
MF-LC-ARMA	P	-1.151e-01	8.233e-01
	F	-1.308e-01	8.003e-01
MF-LLH-ARMA	P	-1.346e-01	8.075e-01
	F	-1.370e-01	7.945e-01
MF-LLM-ARMA	P	-9.632e-03	7.861e-01
	F	-5.183e-03	7.849e-01
LMF-LC-ARMA	P	-1.214e-01	8.290e-01
	F	-1.390e-01	8.140e-01
LMF-LLH-ARMA	P	-1.274e-01	8.225e-01
	F	-1.340e-01	8.008e-01
LMF-LLM-ARMA	P	-4.025e-03	7.945e-01
	F	2.181e-03	7.966e-01

Table 4: Interval estimation performance using bootstrap for nonlinear dataset

Prediction Method	Residual Type	CVR	Mean Length	SD Length
MB-LC	P	0.86	3.265	3.864e-01
	F	0.81	2.837	3.841e-01
MB-LL	P	0.85	3.123	3.383e-01
	F	0.81	2.780	3.466e-01
MF-LC	P	0.88	3.999	5.874e-01
	F	0.90	2.954	4.272e-01
MF-LLH	P	0.88	4.051	6.745e-01
	F	0.84	2.732	4.605e-01
MF-LLM	P	0.89	3.891	6.956e-01
	F	0.86	2.657	4.726e-01
LMF-LC	P	0.87	3.987	6.052e-01
	F	0.88	2.942	4.133e-01
LMF-LLH	P	0.88	4.042	6.797e-01
	F	0.84	2.723	4.373e-01
LMF-LLM	P	0.88	3.946	6.620e-01
	F	0.84	2.661	4.558e-01
MF-LC-ARMA	P	0.86	3.850	5.307e-01
	F	0.89	2.896	4.343e-01
MF-LLH-ARMA	P	0.89	3.917	6.602e-01
	F	0.88	2.694	4.719e-01
MF-LLM-ARMA	P	0.86	3.794	6.319e-01
	F	0.85	2.614	4.766e-01
LMF-LC-ARMA	P	0.88	3.981e	5.723e-01
	F	0.89	2.966	4.423e-01
LMF-LLH-ARMA	P	0.90	4.022	6.889e-01
	F	0.86	2.764	4.451e-01
LMF-LLM-ARMA	P	0.88	3.948	6.556e-01
	F	0.86	2.659	4.844e-01

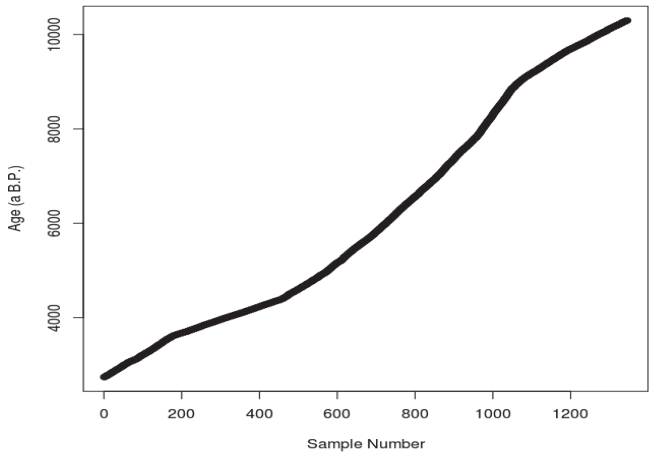

Figure 5: Age (a B.P.) of delta-O-18 versus sample number

of proxy and climate variables which in this case are the oxygen isotope ratio and monsoon rainfall respectively. Proxy data are also useful for construction of confidence intervals for parameter estimates of the proxy variable model. In our case we use a part of the proxy variable dataset which contains a linear trend for estimating the performance of Model-Based and Model-Free predictors for the proxy variable delta-O-18.

2. Proxy data obtained from climate archives may be obtained over either even or uneven time spacing. In case of the speleothem dataset under consideration as shown in Figure 5 the spacing variations are small in general and definitely negligible over the part of the dataset (last 62 points) where we perform prediction; see Figure 5 that depicts the age versus sample number. Hence we will assume even time spacing in our analysis. No interpolation is applied i.e. the number of time-points assumed with even spacing is the same as the number of time points which are present with slightly uneven spacing in the original dataset. It is to be noted that several other techniques such as Singular Spectrum Analysis, Principal Component Analysis and Wavelet Analysis also assume even spacing for time-series analysis. Extension of our methods to incorporate uneven time spacing will be the focus of future work.

We consider the dataset over the last 270 points as shown in Figure 6. This dataset is divided into 2 parts: the first part is used to determine the bandwidths for the MB and MF estimators using methods outlined in Sections 2.2 and 3.5 respectively; the last 62 points are used to calculate point prediction and prediction intervals. It can be noticed from Figures 1 and 6 that this last part of the data appears to have a linear trend. A moving window

method is adopted for cross-validation i.e. for point Y_t (whose metrics for point prediction and prediction intervals are calculated) we use points $[Y_{t-w}, Y_{t-1}]$ for cross-validation. Here the value of w is set to 189. Note also that since this dataset contains a smaller number of points, cross-validation was done over a range of bandwidths using only the last 2 steps of Algorithm 3.2.

Results for point prediction including bias and mean square error (MSE) over all MB and MF methods are shown in Table 5 below.

Results for prediction intervals including CVR, length and standard deviation of the predicted intervals over all MB and MF methods are shown in Table 6 below. The number of bootstrap replications B was set to 1000.

From point-prediction results on this dataset it can be seen that the MF-LLM and LMF-LLM estimators give the best performance. The MF-LLM and LMF-LLM estimators also have the highest coverage probabilities for prediction interval estimation among all estimators that are considered here. For comparison purposes we have listed the performance of point prediction using the RAMPFIT algorithm outlined in (Mudelsee, 2000) and also used for the speleothem dataset in (Fleitmann et al., 2003).

RAMPFIT introduced by (Mudelsee, 2000) is a popular algorithm used to fit climate data which show transitions such as the speleothem dataset. This algorithm was designed to handle change points in climate time-series and to the best of our knowledge cannot handle arbitrary local stationarity which may be present in data. Hence we chose to use RAMPFIT to compare performance of point prediction versus that obtained using our MB and MF point predictors. The MF-LLM-ARMA and LMF-LLM-ARMA estimators outperform RAMPFIT for point prediction as shown in Table 5. We attribute the superior results of MF-LLM-ARMA and LMF-LLM-ARMA for point prediction and prediction intervals to the most likely reason that the data is not compatible with the assumption of an additive model. RAMPFIT was not originally designed to generate prediction interval estimates hence comparisons of these interval metrics versus those obtained using our MB and MF methods are not provided. The RAMPFIT algorithm is described in Appendix B.

For point prediction there is a difference in performance between fitted and predictive residuals which is not the case with the simulation datasets discussed before. This is due to finite sample effects as we use only a small part of the whole speleothem dataset to illustrate the performance differences between the various estimators. Prediction interval coverage is better using predictive as compared to fitted residuals which is consistent with the results associated with i.i.d. regression (Politis, 2013).

As a final point, we consider the practical problem of out-of-sample prediction of the next data point i.e. prediction of Y_{1346} using RAMPFIT and our best predictor (MF-LLM-ARMA) chosen based on in-sample performance. The predicted values using RAMPFIT

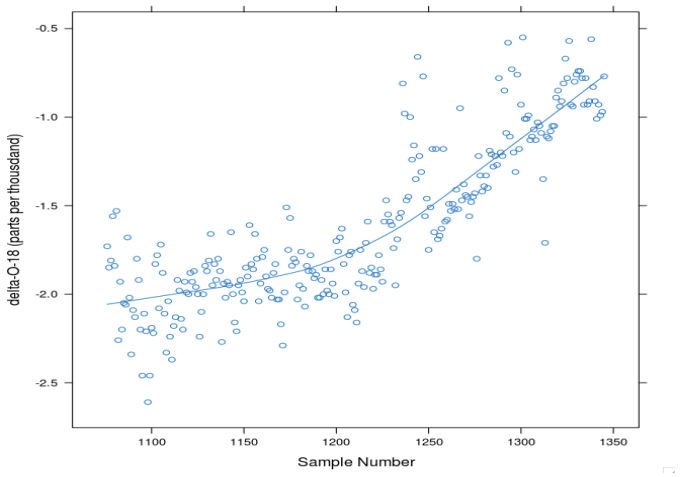

Figure 6: Speleothem data segment used for cross-validation and prediction

and MF-LLM-ARMA are nearly the same (which is reassuring), and approximately equal to -0.81. The 90% prediction interval using MF-LLM is $(-1.165, -0.513)$; as previously mentioned, RAMPFIT cannot be used to generate a prediction interval.

A Appendix: Basic Model-free Bootstrap and Double Bootstrap Algorithms

This section describes in detail algorithms A.1 and A.2 for the construction of Model-Free and Limit Model-Free algorithms as described in (Politis, 2015). However note that we also present new algorithms A.3 and A.4 to determine bandwidth inside the bootstrap loop for the Model-Based and Model-Free cases.

Define the *predictive root* to be the error in prediction, i.e.,

$$Y_{n+1} - \Pi(\hat{g}_{n+1}, \underline{Y}_n, \hat{F}_n) \tag{42}$$

where $\Pi(\hat{g}_{n+1}, \underline{Y}_n, \hat{F}_n)$ is our chosen point predictor of Y_{n+1}, and $\hat{g}_{n+1}$ is our estimate of function g_{n+1} based on the data $\underline{Y}_n$.

Given bootstrap data $\underline{Y}^*_n$ and Y^*_{n+1}, the bootstrap predictive root is the error in prediction in the bootstrap world, i.e.,

$$Y^*_{n+1} - \Pi(\hat{g}^*_{n+1}, \underline{Y}_n, \hat{F}_n) \tag{43}$$

where $\hat{g}^*_{n+1}$ is our estimate of function g_{n+1} based on the bootstrap data $\underline{Y}^*_n$.

Table 5: Point Prediction performance for speleothem dataset

Prediction Method	Residual Type	Bias	MSE
MB-LC	P	-5.800e-03	4.248e-02
	F	-1.845e-02	4.081e-02
MB-LL	P	1.219e-02	4.205e-02
	F	1.227e-03	3.891e-02
MF-LC	P	-2.755e-02	4.006e-02
	F	-1.535e-02	3.805e-02
MF-LLH	P	-2.762e-02	3.683e-02
	F	-2.141e-02	3.925e-02
MF-LLM	P	-3.776e-03	3.513e-02
	F	-2.593e-02	3.730e-02
LMF-LC	P	-2.602e-02	3.959e-02
	F	-1.524e-02	3.815e-02
LMF-LLH	P	-2.672e-02	3.682e-02
	F	-2.060e-02	4.011e-02
LMF-LLM	P	5.724e-03	3.494e-02
	F	-2.702e-02	3.643e-02
MF-LC-ARMA	P	-2.999e-02	4.171e-02
	F	-2.058e-02	3.874e-02
MF-LLH-ARMA	P	-1.8842e-02	4.242e-02
	F	-1.299e-02	3.894e-02
MF-LLM-ARMA	P	-3.235e-03	3.645e-02
	F	-2.077e-02	3.427e-02
LMF-LC-ARMA	P	-2.718e-02	4.143e-02
	F	-2.388e-02	3.953e-02
LMF-LLH-ARMA	P	-1.461e-02	4.550e-02
	F	-1.355e-02	4.095e-02
LMF-LLM-ARMA	P	3.538e-03	3.721e-02
	F	-2.174e-02	3.550e-02
RAMPFIT	Not Applicable	1.781e-02	3.913e-02

Table 6: Interval estimation performance using bootstrap for speleothem dataset

Prediction Method	Residual Type	CVR	Mean Length	SD Length
MB-LC	P	0.82	7.812e-01	2.178e-01
	F	0.78	5.46e-01	1.885e-01
MB-LL	P	0.87	8.731e-01	1.970e-01
	F	0.84	7.254e-01	1.689e-01
MF-LC	P	0.94	7.963e-01	1.631e-01
	F	0.84	5.076e-01	1.525e-01
MF-LLH	P	0.87	7.252e-01	1.372e-01
	F	0.84	5.868e-01	1.747e-01
MF-LLM	P	0.90	7.230e-01	1.914e-01
	F	0.89	5.788e-01	1.774e-01
LMF-LC	P	0.95	7.855e-01	1.804e-01
	F	0.84	5.010e-01	1.454e-01
LMF-LLH	P	0.89	7.284e-01	1.396e-01
	F	0.81	5.568e-01	1.613e-01
LMF-LLM	P	0.90	7.397e-01	1.946e-01
	F	0.89	6.145e-01	1.814e-01
MF-LC-ARMA	P	0.90	8.088e-01	1.535e-01
	F	0.86	5.754e-01	1.665e-01
MF-LLH-ARMA	P	0.86	7.701e-01	1.588e-01
	F	0.80	5.759e-01	1.911e-01
MF-LLM-ARMA	P	0.89	7.427e-01	1.715e-01
	F	0.86	5.819e-01	1.973e-01
LMF-LC-ARMA	P	0.89	8.213e-01	1.721e-01
	F	0.84	5.690e-01	1.599e-01
LMF-LLH-ARMA	P	0.87	7.783e-01	1.527e-01
	F	0.78	5.772e-01	1.916e-01
LMF-LLM-ARMA	P	0.91	7.780e-01	1.818e-01
	F	0.87	6.234e-01	2.096e-01

Remark A.1 Note that eq. (43) depends on the bootstrap data $\underline{Y}_n^*$ only through the estimated function $\hat{g}_{n+1}^*$; both the predictor $\Pi(\hat{g}_{n+1}^*, \underline{Y}_n, \hat{F}_n)$ and the construction of future value Y_{n+1}^* in the sequel are based on the true dataset $\underline{Y}_n$ in order to give validity to the prediction intervals *conditionally* on the data $\underline{Y}_n$.

Algorithm A.1 Model-free bootstrap for prediction intervals for Y_{n+1}

1. *Based on the data $\underline{Y}_n$, estimate the transformation H_n and its inverse H_n^{-1} by $\hat{H}_n$ and $\hat{H}_n^{-1}$ respectively. In addition, estimate g_{n+1} by $\hat{g}_{n+1}$.*
2. *Use $\hat{H}_n$ to obtain the transformed data, i.e., $(\varepsilon_1^{(n)}, ..., \varepsilon_n^{(n)})' = \hat{H}_n(\underline{Y}_n)$. By construction, the variables $\varepsilon_1^{(n)}, ..., \varepsilon_n^{(n)}$ are approximately i.i.d.; let $\hat{F}_n$ denote their empirical distribution.*
   (a) *Sample randomly (with replacement) the data $\varepsilon_1^{(n)}, ..., \varepsilon_n^{(n)}$ to create the bootstrap pseudo-data $\varepsilon_1^*, ..., \varepsilon_n^*$.*
   (b) *Use the inverse transformation $\hat{H}_n^{-1}$ to create pseudo-data in the Y domain, i.e., let $\underline{Y}_n^* = (Y_1^*, ..., Y_n^*)' = \hat{H}_n^{-1}(\varepsilon_1^*, ..., \varepsilon_n^*)$.*
   (c) *Calculate a bootstrap pseudo-response Y_{n+1}^* as the point $\hat{g}_{n+1}(\underline{Y}_n, \varepsilon)$ where ε is drawn randomly from the set $(\varepsilon_1^{(n)}, ..., \varepsilon_n^{(n)})$.*
   (d) *Based on the pseudo-data $\underline{Y}_n^*$, estimate the function g_{n+1} by $\hat{g}_{n+1}^*$ respectively.*
   (e) *Calculate a bootstrap root replicate using eq. (43).*
3. *Steps (a)—(e) in the above should be repeated a large number of times (say B times), and the B bootstrap root replicates should be collected in the form of an empirical distribution whose α—quantile is denoted by $q(\alpha)$.*
4. *A $(1-\alpha)100\%$ equal-tailed prediction interval for Y_{n+1} is given by*

$$[\Pi + q(\alpha/2),\ \Pi + q(1-\alpha/2)] \tag{44}$$

where Π is short-hand for $\Pi(\hat{g}_{n+1}, \underline{Y}_n, \hat{F}_n)$.

Sometimes, the empirical distribution $\hat{F}_n$ converges to a limit distribution F that is of known form (perhaps after estimating a finite-dimensional parameter). Using it instead of the empirical $\hat{F}_n$ results into the Limit Model-Free (LMF) resampling algorithm that is given below. Note that now the point predictor Π is no more a function of $\hat{F}_n$ but of F. Hence, the LMF predictive root is denoted by

$$Y_{n+1} - \Pi(\hat{g}_{n+1}, \underline{Y}_n, F) \tag{45}$$

whose distribution can be approximated by that of the LMF bootstrap predictive root

$$Y^*_{n+1} - \Pi(\hat{g}^*_{n+1}, \underline{Y}_n, F). \tag{46}$$

Algorithm A.2 Limit Model-free (LMF) bootstrap for prediction intervals for Y_{n+1}

1. *Based on the data $\underline{Y}_n$, estimate the transformation H_n and its inverse H_n^{-1} by $\hat{H}_n$ and $\hat{H}_n^{-1}$ respectively. In addition, estimate g_{n+1} by $\hat{g}_{n+1}$.*

2. *(a) Generate bootstrap pseudo-data $\varepsilon^*_1, ..., \varepsilon^*_n$ in an i.i.d. manner from F.*

   (b) Use the inverse transformation $\hat{H}_n^{-1}$ to create pseudo-data in the Y domain, i.e., let $\underline{Y}^*_n = (Y^*_1, ..., Y^*_n)' = \hat{H}_n^{-1}(\varepsilon^*_1, ..., \varepsilon^*_n)$.

   (c) Calculate a bootstrap pseudo-response Y^*_{n+1} as the point $\hat{g}_{n+1}(\underline{Y}_n, \varepsilon)$ where ε is a random draw from distribution F.

   (d) Based on the pseudo-data $\underline{Y}^*_n$, estimate the function g_{n+1} by $\hat{g}^*_{n+1}$ respectively.

   (e) Calculate a bootstrap root replicate using eq. (46).

3. *Steps (a)—(e) in the above should be repeated a large number of times (say B times), and the B bootstrap root replicates should be collected in the form of an empirical distribution whose α—quantile is denoted by $q(\alpha)$.*

4. *A $(1-\alpha)100\%$ equal-tailed prediction interval for Y_{n+1} is given by*

$$[\Pi + q(\alpha/2),\ \Pi + q(1-\alpha/2)] \tag{47}$$

   where Π is short-hand for $\Pi(\hat{g}_{n+1}, \underline{Y}_n, F)$.

Both Model-Based and Model-Free bootstrap algorithms enable the construction of prediction intervals for a pre-determined nominal coverage level. Point-prediction can use the bandwidth b determined by the respective cross-validation procedures outlined for the MB and MF cases in Sections 2.2 and 3.5 respectively. However to prevent under or overcoverage with respect to the nominal level during calculation of prediction intervals we recommend a double bootstrap procedure to accurately set the bandwidth b' inside the bootstrap loop which uses the resampled residuals from point prediction in both the MB and MF cases. The algorithms A.3 and A.4 below enable the determination of this adjusted bandwidth b'.

Algorithm A.3 MB DOUBLE BOOTSTRAP FOR BANDWIDTH IN BOOTSTRAP LOOP

1. *Based on the data* $Y_1, \ldots, Y_n$ *and the bandwidth* b *based on model-based cross-validation, calculate the estimators* $\check{\mu}(\cdot)$ *and* $\check{\sigma}(\cdot)$*, and the 'residuals'* $\check{W}_1, \ldots, \check{W}_n$ *using model* (2)*.*

2. *Fit the AR(p) model* (8) *to the series* $\check{W}_1, \ldots, \check{W}_n$ *(with* p *selected by AIC minimization), and obtain the Yule-Walker estimators* $\hat{\phi}_1, \ldots, \hat{\phi}_p$*, and the error proxies*

$$\check{V}_t = \check{W}_t - \hat{\phi}_1 \check{W}_{t-1} - \cdots - \hat{\phi}_p \check{W}_{t-p} \quad \text{for} \quad t = p+b+1, \ldots, n.$$

3. *Let* $\check{V}_t^*$ *for* $t = 1, \ldots, n, n+1$ *be drawn randomly with replacement from the set* $\{\check{\check{V}}_t$ *for* $t = p+b+1, \ldots, n\}$ *where* $\check{\check{V}}_t = \check{V}_t - (n-p-b)^{-1} \sum_{i=p+b+1}^{n} \check{V}_i$*. Let* I *be a random variable drawn from a discrete uniform distribution on the values* $p+b, p+b+1, \ldots, n$ *, and define the bootstrap initial conditions* $\check{W}_t^* = \check{W}_{t+I}$ *for* $t = -p+1, \ldots, 0$*. Then, create the bootstrap data* $\check{W}_1^*, \ldots, \check{W}_n^*$ *via the AR recursion*

$$\check{W}_t^* = \hat{\phi}_1 \check{W}_{t-1}^* + \cdots + \hat{\phi}_p \check{W}_{t-p}^* + \check{V}_t^* \quad \text{for} \quad t = 1, \ldots, (n+1).$$

   This is the first bootstrap loop.

4. *Create the bootstrap pseudo-series* $Y_1^*, \ldots, Y_{n+1}^*$ *by the formula*

$$Y_t^* = \check{\mu}(t) + \check{\sigma}(t) \check{W}_t^* \quad \text{for} \quad t = 1, \ldots, (n+1).$$

5. *Based on the data* $Y_1^*, \ldots, Y_n^*$ *(first* n *values only) and the bandwidth* b *based on model-based cross-validation, calculate the estimators* $\check{\mu}(\cdot)^*$ *and* $\check{\sigma}(\cdot)^*$*, and the 'residuals'* $W_1^*, \ldots, W_n^*$ *using model* (2)*.*

6. *Fit the AR(p) model* (8) *to the series* $W_1^*, \ldots, W_n^*$ *(with* p *selected by AIC minimization), and obtain the Yule-Walker estimators* $\hat{\phi}_1^*, \ldots, \hat{\phi}_p^*$*, and the error proxies*

$$\check{V}_t^* = W_t^* - \hat{\phi}_1^* W_{t-1}^* - \cdots - \hat{\phi}_p^* W_{t-p}^* \quad \text{for} \quad t = p+b+1, \ldots, n.$$

7. (a) *Let* $\check{V}_t^{**}$ *for* $t = 1, \ldots, n, n+1$ *be drawn randomly with replacement from the set* $\{\check{\check{V}}_t^*$ *for* $t = p+b+1, \ldots, n\}$ *where* $\check{\check{V}}_t^* = \check{V}_t^* - (n-p-b)^{-1} \sum_{i=p+b+1}^{n} \check{V}_i^*$*. Let* I *be a random variable drawn from a discrete uniform distribution on the values*

$p+b, p+b+1, \ldots, n$, *and define the bootstrap initial conditions* $\check{W}_t^{**} = W_{t+I}^*$ *for* $t = -p+1, \ldots, 0$. *Then, create the bootstrap data* $\check{W}_1^{**}, \ldots, \check{W}_n^{**}$ *via the AR recursion*

$$\check{W}_t^{**} = \hat{\phi}_1 W_{t-1}^{**} + \cdots + \hat{\phi}_p W_{t-p}^{**} + \check{V}_t^{**} \quad \textit{for} \quad t = 1, \ldots, (n+1).$$

This is the second bootstrap loop.

(b) Create the bootstrap pseudo-series $Y_1^{**}, \ldots, Y_n^{**}$ *by the formula*

$$Y_t^{**} = \check{\mu}(t)^* + \check{\sigma}(t)^* \check{W}_t^{**} \quad \textit{for} \quad t = 1, \ldots, n.$$

(c) Re-calculate the estimators $\check{\mu}^{**}(\cdot)$ *and* $\check{\sigma}^{**}(\cdot)$ *from the bootstrap data* $Y_1^*, \ldots, Y_n^*$. *The bootstrap estimators* $\check{\mu}^{**}(\cdot)$ *and* $\check{\sigma}^{**}(\cdot)$ *are based on a bandwidth value* b' *which is different from the bandwidth* b *obtained by model-based cross-validation. This gives rises to new bootstrap residuals* $\check{W}_1^{**}, \ldots, \check{W}_n^{**}$ *on which an AR(p) model is again fitted yielding the bootstrap Yule-Walker estimators* $\hat{\phi}_1^{**}, \ldots, \hat{\phi}_p^{**}$.

(d) Calculate the bootstrap predictor

$$\Pi^{**} = \check{\mu}^{**}(n+1) + \check{\sigma}^{**}(n+1) \left[\hat{\phi}_1^{**} W_n^* + \ldots + \hat{\phi}_p^{**} W_{n-p+1}^* \right].$$

(e) Calculate a bootstrap future value

$$Y_{n+1}^{**} = \check{\mu}^*(n+1) + \check{\sigma}^*(n+1) W_{n+1}^{**}$$

where again $W_{n+1}^{**} = \hat{\phi}_1^* W_n^* + \cdots + \hat{\phi}_p^* W_{n-p+1}^* + \check{V}_{n+1}^{**}$ *uses the original values* $(W_n^*, \ldots, W_{n-p+1}^*)$*; recall that* $\check{V}_{n+1}^{**}$ *has already been generated in step (a) above.*

(f) Calculate the bootstrap root replicate $Y_{n+1}^{**} - \Pi^{**}$.

8. *Steps (a)—(f) in the above are repeated a large number of times (say* C *times), and the* C *bootstrap root replicates are collected in the form of an empirical distribution whose* α*–quantile is denoted by* $q(\alpha)$.

9. *Finally, a* $(1-\alpha)100\%$ *equal-tailed prediction interval for* Y_{n+1}^* *(nth value of* $\underline{Y}_{n+1}^*$*) is given by*

$$[\Pi^* + q(\alpha/2),\ \Pi^* + q(1-\alpha/2)]. \tag{48}$$

Here Π^* *is given by:*

$$\Pi^* = \check{\mu}^*(n+1) + \check{\sigma}^*(n+1)\left[\hat{\phi}_1^* W_n^* + \cdots + \hat{\phi}_p^* W_{n-p+1}^*\right] \tag{49}$$

where $\hat{\phi}_1^*, \ldots, \hat{\phi}_p^*$ *are the Yule-Walker estimators of* $\phi_1, \ldots, \phi_p$ *appearing in eq.* (8).

10. *Steps (3)–(9) in the above should be repeated a large number of times (say* B *times) to obtain* B *values of* Y_{n+1}^* *and their corresponding* $(1-\alpha)100\%$ *equal-tailed prediction intervals as outlined by Step (9) above. This can then be used to calculate a coverage probability (CVR) for various values of the second bootstrap loop (*C *iterations) bandwidth* b' *while keeping the bandwidth* b *of the outer bootstrap loop (*B *iterations) fixed to what was obtained from cross-validation. The value of* b' *that gives the target CVR can be used as the bandwidth for the bootstrap loop in Algorithm 2.1.*

Algorithm A.4 MF DOUBLE BOOTSTRAP FOR BANDWIDTH IN BOOTSTRAP LOOP

1. *Based on the data* $\underline{Y}_n$ *and the bandwidth* b *obtained from model-free cross-validation, estimate the transformation* H_n *and its inverse* H_n^{-1} *by* $\hat{H}_n$ *and* $\hat{H}_n^{-1}$ *respectively. In addition, estimate* g_{n+1} *by* $\hat{g}_{n+1}$.

2. *Use* $\hat{H}_n$ *to obtain the transformed data, i.e.,* $(\varepsilon_1^{(n)}, ..., \varepsilon_n^{(n)})' = \hat{H}_n(\underline{Y}_n)$. *By construction, the variables* $\varepsilon_1^{(n)}, ..., \varepsilon_n^{(n)}$ *are approximately i.i.d.*

3. *Sample randomly (with replacement) the data* $\varepsilon_1^{(n)}, ..., \varepsilon_n^{(n)}$ *to create the bootstrap pseudo-data* $\varepsilon_1^*, ..., \varepsilon_{n+1}^*$. *This is the first bootstrap loop.*

4. *Use the inverse transformation* $\hat{H}_n^{-1}$ *and the bandwidth* b *from model-free cross-validation to create pseudo-data in the* Y *domain, i.e., let* $\underline{Y}_{n+1}^* = (Y_1^*, ..., Y_{n+1}^*)' = \hat{H}_n^{-1}(\varepsilon_1^*, ..., \varepsilon_{n+1}^*)$.

5. *Based on the data* $\underline{Y}_n^*$ *(first* n *values only) and the bandwidth* b *obtained from model-free cross-validation, estimate the transformation* H_n^* *and its inverse* H_n^{*-1} *by* $\hat{H}_n^*$ *and* $\hat{H}_n^{*-1}$ *respectively. In addition, estimate* g_{n+1} *by* $\hat{g}_{n+1}^*$.

6. *Use* $\hat{H}_n^*$ *to obtain the transformed data, i.e.,* $(\varepsilon_1^{*(n)}, ..., \varepsilon_n^{*(n)})' = \hat{H}_n^*(\underline{Y}_n^*)$. *By construction, the variables* $\varepsilon_1^{*(n)}, ..., \varepsilon_n^{*(n)}$ *are approximately i.i.d; let* $\hat{F}_n^*$ *denote their empirical distribution.*

   (a) *Sample randomly (with replacement) the data* $\varepsilon_1^{*(n)}, ..., \varepsilon_n^{*(n)}$ *to create the bootstrap pseudo-data* $\varepsilon_1^{**(n)}, ..., \varepsilon_n^{**(n)}$. *This is the second bootstrap loop.*

   (b) *Use the inverse transformation* $\hat{H}_n^{*-1}$ *and a bandwidth* b' *(different from* b *found from model-free cross-validation) to create pseudo-data in the* Y *domain, i.e., let* $\underline{Y}_n^{**} = (Y_1^{**}, ..., Y_n^{**})' = \hat{H}_n^{*-1}(\varepsilon_1^{**}, ..., \varepsilon_n^{**})$.

(c) *Calculate a bootstrap pseudo-response* Y^{**}_{n+1} *as the point* $\hat{g}^*_{n+1}(\underline{Y}^*_n, \varepsilon^*)$ *where* ε^* *is drawn randomly from the set* $(\varepsilon_1^{*(n)}, ..., \varepsilon_n^{*(n)})$.

(d) *Based on the pseudo-data* $\underline{Y}^{**}_n$ *and bandwidth* b', *estimate the function* g_{n+1} *by* $\hat{g}^{**}_{n+1}$ *respectively.*

(e) *Calculate a bootstrap root replicate using*

$$Y^{**}_{n+1} - \Pi(\hat{g}^{**}_{n+1}, \underline{Y}^*_n, \hat{F}^*_n). \tag{50}$$

7. *Steps (a)—(e) in the above should be repeated a large number of times (say* C *times), and the* C *bootstrap root replicates should be collected in the form of an empirical distribution whose* α*—quantile is denoted by* $q(\alpha)$.

8. *A* $(1-\alpha)100\%$ *equal-tailed prediction interval for* Y^*_{n+1} *(nth value of* $\underline{Y}^*_{n+1}$*) is given by*

$$[\Pi^* + q(\alpha/2),\ \Pi^* + q(1-\alpha/2)] \tag{51}$$

where Π^* *is short-hand for* $\Pi(\hat{g}^*_{n+1}, \underline{Y}^*_n, \hat{F}^*_n)$.

9. *Steps (3)–(8) in the above should be repeated a large number of times (say* B *times) to obtain* B *values of* Y^*_{n+1} *and their corresponding* $(1-\alpha)100\%$ *equal-tailed prediction intervals as outlined by Step (8) above. This can then be used to calculate a coverage probability (CVR) for various values of the second bootstrap loop (*C *iterations) bandwidth* b' *while keeping the bandwidth* b *of the outer bootstrap loop (*B *iterations) fixed to what was obtained from cross-validation. The value of* b' *that gives the target CVR can be used as the bandwidth for the bootstrap loop in Algorithms A.1 and A.2.*

B Appendix: RAMPFIT algorithm for analyzing climate data with transitions

The RAMPFIT algorithm which can handle uneven time-spacing in observations was proposed by (Mudelsee, 2000) for performing regression on climate data which shows transitions such as the speleothem dataset considered in this paper. However RAMPFIT was not originally designed to handle arbitrary local stationarity which may be present in data. Here we briefly outline the steps in RAMPFIT used to obtain point prediction estimates which are used for comparison with their Model-Based and Model-Free counterparts.

Define $x(i) = X(t(i))$ where $(X_t, t \in \mathbf{R})$ is an underlying continuous-time stochastic process. For a time series $x(i)$ measured at times $t(i), i = 1, \ldots, n$, the model under consideration is (Mudelsee, 2000):

$$x(i) = x_{fit}(i) + \epsilon(i) \tag{52}$$

It is assumed that the errors $\epsilon(i)$ are heteroskedastic and are distributed as $N(0, \sigma(i)^2)$. The fitted model is a ramp function as defined below:

$$x_{fit}(t) = \begin{cases} x1, & for\ t \leq t1, \\ x1 + (t - t1)(x2 - x1)/(t2 - t1), & for\ t1 \leq t \leq t2, \\ x2, & for\ t \geq t2 \end{cases} \tag{53}$$

Here $t1$ and $t2$ denote the start and end of the ramp and $x1$, $x2$ denote the corresponding values at those points. The regression model is fitted to data $\{t(i),\, x(i)\}_{i=1}^{n}$ by minimizing the weighted sum of squares as given below:

$$SSQW(t1, x1, t2, x2) = \sum_{i=1}^{n} \frac{[x(i) - x_{fit}(i)]^2}{\sigma(i)^2} \tag{54}$$

Owing to the non-differentiabilities at $t1$ and $t2$, RAMPFIT does a search over a range of values supplied for these 2 values and chooses the values $(\hat{t}1, \hat{x}1, \hat{t}2, \hat{x}2)$ for which the $SSQW$ is minimum. In addition since $\sigma(i)$ is not known an initial guess of this is supplied to the algorithm following which the $\sigma(i)$ values are recalculated from the obtained residuals. The estimates $(\hat{t}1, \hat{x}1, \hat{t}2, \hat{x}2)$ are then regenerated. These steps are repeated till MSE values of point prediction converge.

The full algorithm is described below:

Algorithm B.1 *RAMPFIT REGRESSION*

1. *Set initial estimate of* $\sigma(i) = i$ *with* $i = 1, \ldots, n$
2. *Set search ranges* $[t1_{min},\ t1_{max}]$ *and* $[t2_{min}, t2_{max}]$ *for values of* $t1$ *and* $t2$
3. *Calculate SSQW using (53) and (54) over this grid of* $t1$ *and* $t2$ *values; denote a typical point in this grid as* $(\bar{t}1,\ \bar{t}2)$
4. *Determine* $(\hat{t}1, \hat{x}1, \hat{t}2, \hat{x}2)$ = *argmin* $[SSQW(\bar{t}1, \hat{x}1, \bar{t}2, \hat{x}2)]$ *and obtain* x_{fit}
5. *Calculate residuals* $e(i) = x(t(i)) - x_{fit}(t(i))$
6. *Re-estimate the variance* $\sigma(i)$ *from* $e(i)$ *using k-nearest-neighbour smoothing*
7. *Repeat steps (2) to (6) above till MSE values converge.*

Acknowledgements

This research was partially supported by NSF grants DMS 12-23137 and DMS 16-13026. The authors would like to acknowledge the Pacific Research Platform, NSF Project ACI-1541349 and Larry Smarr (PI, Calit2 at UCSD) for providing the computing infrastructure used in this project. Many thanks are also due to Richard Davis and Stathis Paparoditis for their helpful comments. The authors would also like to thank Dr. Manfred Mudelsee for kindly sharing the FORTRAN code for the RAMPFIT algorithm used for analyzing the speleothem dataset used in this paper.

References

Brockwell, P. J., & Davis, R. A. (1991). Time series: theory and methods (Second ed.). Springer, New York.

Dahlhaus, R. (2012). Locally stationary processes. In T. S. Rao et al. (Eds.), Handbook of statistics (Vol. 30, p. 351-412). Elsevier.

Dahlhaus, R., et al. (1997). Fitting time series models to nonstationary processes. The Annals of Statistics, 25(1), 1–37.

Das, S., & Politis, D. N. (2017). Nonparametric estimation of the conditional distribution at regression boundary points. arXiv preprint arXiv:1704.00674.

Dowla, A., Paparoditis, E., & Politis, D. N. (2013). Local block bootstrap inference for trending time series. Metrika, 76(6), 733–764.

Dowla A., Paparoditis E., & Politis D.N. (2003). Locally stationary processes and the local block bootstrap. In M. G. Akritas & D. N. Politis (Eds.), Recent advances and trends in nonparametric statistics (p. 437-444). Elsevier.

Fan, J., & Gijbels, I. (1996). Local polynomial modelling and its applications: monographs on statistics and applied probability (Vol. 66). CRC Press, Boca Raton.

Fan, J., & Yao, Q. (2007). Nonlinear time series: nonparametric and parametric methods. Springer, New York.

Fleitmann, D., Burns, S. J., Mudelsee, M., Neff, U., Kramers, J., Mangini, A., & Matter, A. (2003). Holocene forcing of the indian monsoon recorded in a stalagmite from southern oman. Science, 300(5626), 1737–1739.

Hall, P., Wolff, R. C., & Yao, Q. (1999). Methods for estimating a conditional distribution function. Journal of the American Statistical Association, 94(445), 154–163.

Hansen, B. E. (2004). Nonparametric estimation of smooth conditional distributions. Unpublished paper: Department of Economics, University of Wisconsin.

Härdle, W., & Vieu, P. (1992). Kernel regression smoothing of time series. Journal of Time Series Analysis, 13(3), 209–232.

Jentsch, C., & Politis, D. N. (2015). Covariance matrix estimation and linear process bootstrap for multivariate time series of possibly increasing dimension. The Annals of Statistics, 43(3), 1117–1140.

Kim, T. Y., & Cox, D. D. (1996). Bandwidth selection in kernel smoothing of time series. Journal of Time Series Analysis, 17(1), 49–63.

Kreiss, J.-P., Paparoditis, E., & Politis, D. N. (2011). On the range of validity of the autoregressive sieve bootstrap. The Annals of Statistics, 39(4), 2103–2130.

Li, Q., & Racine, J. S. (2007). Nonparametric econometrics: theory and practice. Princeton University Press, Princeton.

Masry, E., & Tjøstheim, D. (1995). Nonparametric estimation and identification of nonlinear arch time series strong convergence and asymptotic normality: Strong convergence and asymptotic normality. Econometric theory, 11(2), 258–289.

McMurry, T. L., & Politis, D. N. (2010). Banded and tapered estimates for autocovariance matrices and the linear process bootstrap. Journal of Time Series Analysis, 31(6), 471–482.

McMurry, T. L., & Politis, D. N. (2015). High-dimensional autocovariance matrices and optimal linear prediction. Electronic Journal of Statistics, 9(1), 753–788.

Mudelsee, M. (2000). Ramp function regression: a tool for quantifying climate transitions. Computers & Geosciences, 26(3), 293–307.

Mudelsee, M. (2014). Climate time series analysis: classical statistical and bootstrap methods. Springer Science and Business Media.

Pan, L., & Politis, D. N. (2016). Bootstrap prediction intervals for markov processes. Computational Statistics & Data Analysis, 100, 467–494.

Paparoditis, E., & Politis, D. N. (2002). Local block bootstrap. Comptes Rendus Mathematiques, 335(11), 959–962.

Politis, D. N. (2001). On nonparametric function estimation with infinite-order flat-top kernels. In Ch. A. Charalambides et al. (Eds.), Probability and statistical models with applications (p. 469-483). Chapman and Hall/CRC: Boca Raton.

Politis, D. N. (2013). Model-free model-fitting and predictive distributions. Test, 22(2), 183–221.

Politis, D. N. (2015). Model-free prediction and regression. Springer, New York.

Politis, D. N., & Romano, J. P. (1994). The stationary bootstrap. Journal of the American Statistical association, 89(428), 1303–1313.

Priestley, M. B. (1965). Evolutionary spectra and non-stationary processes. Journal of the Royal Statistical Society. Series B (Methodological), 204–237.

Priestley, M. B. (1988). Non-linear and non-stationary time series analysis. Academic Press, London.

Rosenblatt, M. (1952). Remarks on a multivariate transformation. The Annals of Mathematical Statistics, 23(3), 470–472.

Samorodnitsky, G., & Taqqu, M. S. (1994). Stable non-gaussian random processes: Stochastic models with infinite variance (stochastic modeling series). Chapman and Hall/CRC Press.

Tong, H. (2011). Threshold models in time series analysis—30 years on. Statistics and its Interface, 4(2), 107–118.

Zhou, Z., & Wu, W. B. (2009). Local linear quantile estimation for nonstationary time series. The Annals of Statistics, 37(5B), 2696–2729.

Zhou, Z., & Wu, W. B. (2010). Simultaneous inference of linear models with time varying coefficients. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(4), 513–531.