UNIVERSITY OF CALIFORNIA, SAN DIEGO

Model-Based and Model-Free Prediction Techniques for Locally Stationary Time Series and Random Fields

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Intelligent Systems, Robotics and Control)

by

Srinjoy Das

Committee in charge:

Professor Ken Kreutz-Delgado, Chair
Professor Dimitris Politis, Co-Chair
Professor Ery Arias-Castro
Professor Vikash Gilja
Professor Truong Nguyen

2018

The dissertation of Srinjoy Das is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2018

DEDICATION

To my parents.

EPIGRAPH

If I have seen further it is only by standing on the shoulders of giants.

—Isaac Newton

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to wholeheartedly thank Professor Ken Kreutz-Delgado for all his encouragement and guidance which played a key role in my decision to take a break from my professional career and return to school to pursue graduate studies. I consider myself extremely fortunate to have attended his graduate classes at UCSD as these formed the bedrock of my understanding in the area of machine learning algorithms and paved the way for my research focus over the next several years. His exceptional vision and keen multidisciplinary insights were instrumental in driving the early part of my research in the area of Markov Chain Monte Carlo inference on non Von-Neumann processing platforms. I am also blessed to have received his invaluable advice in several other related areas with immense practical applications in this era of rapid developments in Machine Learning and Statistics.

I am extremely thankful to my co-chair Professor Dimitris Politis for introducing me to the novel approach of Model-Free prediction and for giving me an opportunity to work under his supervision on several interesting problems on this subject. Under his guidance I have been able to vastly improve my understanding of how to approach prediction problems in important applications such as time series analysis and how to develop statistical intuition to address the most challenging issues that arise while solving difficult problems in these domains. I am grateful to him for being so patient even during the most difficult times of my research and helping me grow into an independent researcher in this area.

I am deeply grateful to my committee members Professor Ery Arias-Castro, Professor Vikash Gilja and Professor Truong Nguyen for their assistance and invaluable inputs during the course of this work. I am also indebted to Professor Larry Smarr and Professor Tom Defanti at CalIT2 for providing me with the computing infrastructure that was used during

the course of my research. I owe an enormous debt of gratitude to John Graham and Joe Keefe at CalIT2 for helping me with accessing the compute facilities and for coordinating all activities in this area.

This thesis would not have been possible without the constant support and encouragement of my wife Rumpa Giri who shouldered the family responsibilities over these last few years that enabled me to take a break from my career in industry and pursue my mid-life academic ambitions at UCSD. She has always been a source of strength for me and over the course of these years her unwavering support of my career goals has allowed me to follow my dreams. Thanks are also due to my son Sreyan whose infectious smile and playful comments always reminded me to never give up during difficult times. It has been a constant source of joy for me watching him grow up while I charted my exciting new path at UCSD. I am also grateful to my father-in-law Mr. Ratan Chandra Giri and my mother-in-law Mrs. Santi Giri for being a constant source of support for my career and Ph.D. aspirations. I also owe a tremendous debt of gratitude to my former manager and mentor Philip Crary for his encouragement and advice all these years both in my professional and academic careers.

My friends in Mathematics Ashley Chen, Nan Zou, Jeremy Schmitt, Sinan Aksoy, Nish Gurnani among others were a constant source of support over these years. I am thankful to them for participating in so many interesting discussions and for giving me invaluable research feedback. My friends in Engineering Enming Luo, Byeongkeun Kang, Subarna Tripathi, Shebin Parameswaran, Marcela Mendoza, Igor Fedorov, Bruno Pedroni, Tejaswy Paila, Siva Chiluvuri among others have also been tremendously helpful and I am grateful to them for their friendship and support over these years. I have also been extremely fortunate to have known and mentored exceptional undergraduate and masters students including Ojash Neopane, Ian Colbert, Xinyu Zhang and Chih Yin-Kan and for being able to participate in insightful research discussions and projects with such a talented and dedicated group.

This thesis is dedicated to my parents Late Mr. Rabindra Nath Das and Late Mrs. Usashi Das. During the course of their lifetimes, amid severe resource limitations and seemingly insurmountable odds they provided me with exceptional opportunities which helped me pursue my studies in India and subsequently establish my professional career. Their constant encouragement and sacrifices eventually motivated me to return to graduate school and attain my lifelong dream of completing my Ph.D. I am eternally grateful to them and everyone else who helped me succeed in my long journey.

Parts of this dissertation are based on papers I have co-authored with my co-advisor Professor Dimitris Politis.

Chapter 2 is based on the paper ”Nonparametric estimation of the conditional distribution at regression boundary points”, by S. Das and D.N. Politis and has been submitted for publication. The dissertation author was the primary investigator and author of this paper.

Chapter 3 is based on the paper ”Predictive inference for locally stationary time series with an application to climate data”, by S. Das and D.N. Politis and has been submitted for publication. The dissertation author was the primary investigator and author of this paper.

VITA

1995	B.E. (Hons.) Electrical and Electronics Engineering, Birla Institute of Technology and Science, Pilani, Rajasthan, India
1995	M.Sc. (Hons.) Physics, Birla Institute of Technology and Science, Pilani, Rajasthan, India
2005	M.S. Electrical and Computer Engineering, University of California, Irvine
2016	M.S. Statistics, University of California, San Diego
2018	Ph.D. Electrical Engineering (Intelligent Systems, Robotics and Control), University of California, San Diego

PUBLICATIONS

Bruno U. Pedroni, **Srinjoy Das,** Emre Neftci, Kenneth Kreutz-Delgado, and Gert Cauwenberghs. ”Neuromorphic adaptations of restricted boltzmann machines and deep belief networks.” *Neural Networks (IJCNN), The 2013 International Joint Conference on, pp. 1-6. IEEE, 2013.*

Emre Neftci, **Srinjoy Das,** Bruno Pedroni, Kenneth Kreutz-Delgado, and Gert Cauwenberghs. ”Event-driven contrastive divergence for spiking neuromorphic systems.” *Frontiers in neuroscience 7 (2014): 272.*

Srinjoy Das, Bruno Umbria Pedroni, Paul Merolla, John Arthur, Andrew S. Cassidy, Bryan L. Jackson, Dharmendra Modha, Gert Cauwenberghs, and Ken Kreutz-Delgado. ”Gibbs sampling with low-power spiking digital neurons.” *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on, pp. 2704-2707. IEEE, 2015.*

Bruno U. Pedroni, **Srinjoy Das,** John V. Arthur, Paul A. Merolla, Bryan L. Jackson, Dharmendra S. Modha, Kenneth Kreutz-Delgado, and Gert Cauwenberghs. ”Mapping generative models onto a network of digital spiking neurons.” *IEEE transactions on biomedical circuits and systems 10, no. 4 (2016): 837-854.*

Ojash Neopane, **Srinjoy Das,** Ery Arias-Castro, and Ken Kreutz-Delgado. ”A nonparametric framework for quantifying generative inference on neuromorphic systems.” *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on, pp. 1346-1349. IEEE, 2016.*

Srinjoy Das and Dimitris N. Politis. ”Nonparametric estimation of the conditional distribution at regression boundary points.” *arXiv preprint arXiv:1704.00674 (2017).*

Srinjoy Das and Dimitris N. Politis. ”Predictive inference for locally stationary time series with an application to climate data.” *arXiv preprint arXiv:1712.02383 (2018).*

ABSTRACT OF THE DISSERTATION

Model-Based and Model-Free Prediction Techniques for Locally Stationary Time Series and Random Fields

by

Srinjoy Das

Doctor of Philosophy in Electrical Engineering (Intelligent Systems, Robotics and Control)

University of California, San Diego, 2018

Professor Ken Kreutz-Delgado, Chair
Professor Dimitris Politis, Co-Chair

With the recent growth in data owing to ubiquitous internet connectivity, practical problems involving prediction which abound in multiple fields such as healthcare, finance, climate analysis, image processing and other disciplines are now becoming solvable. Traditionally such problems have been addressed by Model-Based approaches; i.e. by employing assumptions about the data generating process (DGP) to construct a model which can then used for predicting future values. However it is well known that such methods may suffer from disadvantages such as model non-robustness, overfitting among others. In this thesis along with Model-Based approaches the alternative paradigm of Model-Free Prediction is explored for problems such as regression, time series and random fields. The Model-Free framework does not assume knowledge of the DGP and is applied directly to the available data to construct estimators for both point prediction and prediction intervals. Both Model-Based and Model-Free prediction methods are applied to synthetic and real-life data and their prediction performances are compared using standard metrics to demonstrate the applicability and usefulness of both approaches.

Chapter 1

Model-Based and Model-Free Prediction

Forecasting and prediction problems abound in a multitude of disciplines including finance, healthcare, climate analysis and image processing. The classical approach to solving such problems utilizes what can be termed as Model-Based and involves constructing a model of the data generating process (DGP) to capture the underlying relationships between variables using the observed data. This model, which is assumed to capture the complexity of the variables and their interactions, is then used to forecast or predict future observations. However, as is well known, model construction, which is based on assuming adequate knowledge of the underlying DGP, has several issues. In particular:

- It is often difficult to accurately characterize the complexity of the model based on which future values are to be generated. This can lead to either an underfitted or overfitted model causing inaccuracies in prediction.

- There may often be outliers in the observed data which can render the constructed model inaccurate thereby again leading to errors in prediction.

Given these and other deficiencies in the typically used Model-Based approach one can ask whether an entirely new prediction paradigm is possible which can overcome these limitations. The Model-Free Prediction approach proposed by (Politis, 2013) and (Politis, 2015) aims to surmount these issues by following a principled procedure which entirely avoids constructing an explicit model and instead follows a purely data-driven methodology whereby a sequence of invertible transformations are created based on the observed data to estimate future values. This approach is briefly outlined below.

Basic Model-Free Approach:

- Objective: Given $\underline{Y}_n = (Y_1, \ldots, Y_n)'$ predict future value Y_{n+1} given the data
- Idea: Find an invertible transformation H_m so that (for all m) the vector $\underline{\varepsilon}_m = H_m(\underline{Y}_m)$ has i.i.d. components ε_k where $\underline{\varepsilon}_m = (\varepsilon_1, \ldots, \varepsilon_m)'$

  $(i)\ \ (Y_1, \ldots, Y_m) \xrightarrow{H_m} (\varepsilon_1, \ldots, \varepsilon_m)$

  $(ii)\ \ (Y_1, \ldots, Y_m) \xleftarrow{H_m^{-1}} (\varepsilon_1, \ldots, \varepsilon_m)$

  (i) implies that $\varepsilon_1, \ldots, \varepsilon_n$ are known given the data $Y_1, \ldots, Y_n$

  (ii) implies that Y_{n+1} is a function of $\varepsilon_1, \ldots, \varepsilon_n$, and ε_{n+1}
- So, given the data $\underline{Y}_n$, Y_{n+1} is a function of ε_{n+1} only, i.e., $Y_{n+1} = \tilde{h}(\varepsilon_{n+1})$
- Suppose $\varepsilon_{n+1} \sim$ cdf $F_{n+1}(\varepsilon)$.
- Note that $F_{n+1}(\varepsilon)$ is consistently estimated by $\varepsilon_1, \ldots, \varepsilon_n \sim$ edf $\hat{F}_n(\varepsilon)$

- Under an L2 criteria this gives:

  $\hat{Y}_{n+1} = \int \tilde{h}(\varepsilon)\, d\hat{F}_n(\varepsilon)$

  $\approx \frac{1}{n} \sum_{i=1}^{n} \tilde{h}(\varepsilon_i)$

  or $\hat{Y}_{n+1} = \text{mean}\ [\tilde{h}(\varepsilon_1), \ldots, \tilde{h}(\varepsilon_n)]$

- Similarly under an L1 criteria we have: $\hat{Y}_{n+1} = \text{median}\ [\tilde{h}(\varepsilon_1), \ldots, \tilde{h}(\varepsilon_n)]$

In addition to constructing the **point predictor** $\hat{Y}_{n+1}$ based on an L1 or L2 criteria, the method of bootstrapping (Efron & Tibshirani, 1993) can also be applied to estimate **prediction intervals**.

The Model-Free principle outlined above therefore follows a transformation based approach to predictive inference (Politis, 2013) whose goal is to create a sequence of invertible transformations which can generate i.i.d. data. For problems such as regression on boundary points, one-step ahead prediction of time-series or random fields the first step in this process involves estimating the conditional distribution $F_x(y) = P(Y \leq y | X = x)$ where x is a boundary point. Estimating this conditional distribution function at such a regressor point is immediate via standard kernel methods but problems ensue if local linear methods are to be used. In particular, the distribution function estimator is not guaranteed to be monotone increasing, and the quantile curves can "cross". In Chapter 2, a simple method of correcting the local linear distribution estimator for monotonicity is proposed, and its good performance is demonstrated via simulations and real data examples.

The Model-Free Prediction Principle 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 Chapter 3, 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 Chapter 3, 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 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.

In Chapter 4 we demonstrate the use of the Model-Free principle for inference on Random Fields defined on the non symmetric half-plane (NSHP), applications of which abound in image processing, satellite data analysis, radiography and other areas. We show how both Model-Based and Model-Free Prediction can be applied to such Random Fields that are only locally stationary, i.e. fields that can be assumed to be stationary over short 'spatial' regions. One-step ahead point predictors are constructed for the Model-Based and Model-Free case and their performance is compared using a model that incorporates a trend.

Chapter 2

Nonparametric estimation of the conditional distribution at regression boundary points

2.1 Introduction

Nonparametric regression via kernel smoothing is a standard statistical tool with increased importance in the Big Data era; see e.g. (Wand & Jones, 1994), (Yu & Jones, 1998), (Yu, Lu, & Stander, 2003), (Koenker, 2005) or (Schucany, 2004) for reviews. The fundamental nonparametric regression problem is estimating the regression function $\mu(x) = E(Y|X = x)$ from data $(Y_1, x_1), \ldots, (Y_n, x_n)$ under the sole assumption that the function $\mu(\cdot)$ belongs to some smoothness class, e.g., that it possesses a certain number of continuous derivatives. Here, Y_i is the real-valued response associated with the regressor X taking a value of x_i. Either by design or via the conditioning, the regressor values $x_1, \ldots, x_n$ are treated as nonrandom. For simplicity of exposition, we will assume that the regressor X is univariate but extension to the multivariate case is straightforward.

A common approach to nonparametric regression starts with assuming that the data were generated by an additive model such as

$$Y_i = \mu(x_i) + \sigma(x_i)\varepsilon_i \text{ for } i = 1,2,\ldots,n \tag{2.1}$$

where the errors ε_i are assumed to be independent, identically distributed (i.i.d.) with mean zero and variance one, and $\sigma(\cdot)$ is another unknown function.

Nevertheless, standard kernel smoothing methods are applicable in a Model-Free context as well, i.e., without assuming an equation such as (2.1). An important example is the Nadaraya-Watson kernel estimator defined as

$$\hat{\mu}(x) = \frac{\sum_{i=1}^{n} \tilde{K}_{i,x} Y_i}{\sum_{i=1}^{n} \tilde{K}_{i,x}} \tag{2.2}$$

where $b > 0$ is the bandwidth, $K(x)$ is a nonnegative kernel function satisfying $\int K(x)dx = 1$, and

$$\tilde{K}_{i,x} = \frac{1}{b} K\left(\frac{x - x_i}{b}\right).$$

Estimator $\hat{\mu}(x)$ enjoys favorable properties such as consistency and asymptotic normality under standard regularity conditions in a Model-Free context, e.g. assuming the pairs $(Y_1, X_1), \ldots, (Y_n, X_n)$ are i.i.d. (Li & Racine, 2007).

The rationale behind the Nadaraya-Watson estimator (2.2) is approximating the unknown function $\mu(x)$ by a constant over a window of "width" b; this is made clearer if a rectangular function is chosen as the kernel K, e.g. letting $K(x) = \mathbf{1}\{|x| < 1/2\}$ where $\mathbf{1}_A$ is the indicator of set A, in which case $\hat{\mu}(x)$ is just the average of the Y's whose x value fell in the window. Going from a local constant to a local linear approximation for $\mu(x)$, i.e., a first-order Taylor expansion, motivates the local linear estimator

$$\hat{\mu}^{LL}(x) = \frac{\sum_{i=1}^{n} w_i Y_i}{\sum_{i=1}^{n} w_i} \tag{2.3}$$

where

$$w_i = \tilde{K}_{i,x}\left(1 - \hat{\beta}(x - x_i)\right) \text{ and } \hat{\beta} = \frac{\sum_{i=1}^{n} \tilde{K}_{i,x}(x - x_i)}{\sum_{i=1}^{n} \tilde{K}_{i,x}(x - x_i)^2}. \tag{2.4}$$

If the design points x_j are (approximately) uniformly distributed over an interval $[a_1, a_2]$, then $\hat{\mu}^{LL}(x)$ is typically indistinguishable from the Nadaraya-Watson estimator $\hat{\mu}(x)$ when x is in the 'interior', i.e., when $x \in [a_1 + b/2, a_2 - b/2]$. The local linear estimator $\hat{\mu}^{LL}(x)$ offers an advantage when the design points x_j are non-uniformly distributed, e.g., when there are gaps in the design points, and/or when x is a *boundary* point, i.e., when $x = a_1$ or $x = a_2$ (plus or minus $b/2$); see (Fan & Gijbels, 1996) for details.

Instead of focusing on the conditional moment $\mu(x) = E(Y|X = x)$, one may consider estimating the conditional distribution $F_x(y) = P(Y \leq y|X = x)$ at some fixed point y. Note that $F_x(y) = E(W|X = x)$ where $W = \mathbf{1}\{Y \leq y\}$. Hence, estimating $F_x(y)$ can be easily done via local constant or local linear estimation of the conditional mean from the new dataset $(W_1, x_1) \ldots, (W_n, x_n)$ where $W_i = \mathbf{1}\{Y_i \leq y\}$. To elaborate, the local constant and the local linear estimators of $F_x(y)$ are respectively given by

$$\hat{F}_x(y) = \frac{\sum_{i=1}^{n} \tilde{K}_{i,x}\mathbf{1}\{Y_i \leq y\}}{\sum_{i=1}^{n} \tilde{K}_{i,x}}, \text{ and } \hat{F}_x^{LL}(y) = \frac{\sum_{j=1}^{n} w_j \mathbf{1}\{Y_j \leq y\}}{\sum_{j=1}^{n} w_j} \tag{2.5}$$

where the local linear weights w_j are given by eq. (2.4).

Viewed as a function of y, $\hat{F}_x(y)$ is a well-defined distribution function; however, being a local constant estimator, it often has poor performance at boundary points. As already discussed, $\hat{F}_x^{LL}(y)$ has better performance at boundary points. Unfortunately, $\hat{F}_x^{LL}(y)$ is neither guaranteed to be in $[0,1]$ nor is it guaranteed to be nondecreasing as a function of y; this is due to some of the weights w_j potentially being negative.

The problem with non-monotonicity of $\hat{F}_x^{LL}(y)$ and the associated quantile curves potentially "crossing" is well-known; see (Hall, Wolff, & Yao, 1999) for the former issue, and (Yu & Jones, 1998) for the latter, as well as the reviews on quantile regression by (Yu et al., 2003) and (Koenker, 2005). In case of quantile crossing, non-monotonicity of the estimated conditional distribution function may lead to erroneous results where the estimated 95% (say) percentile turns out to be smaller than the 90% estimated percentile.

In case of the adjusted Nadaraya-Watson estimator described in (Hall et al., 1999) the predictive distribution of an observation Y_i, given $\mathbf{X} = \mathbf{x}$ which may denote a vector of past observed values of $\mathbf{Y}$ is estimated by:

$$P(Y_i \leq y|\mathbf{X_i} = \mathbf{x}) \equiv \tilde{\pi}(y|x) = \frac{\Sigma_{i=1}^n I(Y_i \leq y) p_i(x) K_h(X_i - x)}{\Sigma_{i=1}^n p_i(x) K_h(X_i - x)} \tag{2.6}$$

The weights p_i satisfy the conditions $p_i \geq 0$, $\Sigma_i p_i = 1$ and

$$\Sigma_{i=1}^n p_i(x)(X_i - x) K_h(X_i - x) = 0 \tag{2.7}$$

The estimated CDF $\tilde{\pi}(y|x)$ therefore lies between 0 and 1 as required for a distribution function. However the bias equation for the adjusted Nadaraya-Watson estimator (2.7) cannot be satisfied for boundary points i.e. when x is close to either max x_i or min x_i. Therefore in general this method cannot be used in place of the local constant kernel estimator of the conditional distribution function. In case of the local linear estimators proposed by (Yu &

Jones, 1998) the distribution function estimators are not constrained either to lie between 0 and 1 or to be monotone increasing. Given the shortcomings of these previously proposed techniques we propose a new method in this Chapter for constructing a monotone local linear distribution estimator.

In the next section, a simple method of correcting the local linear distribution estimator for monotonicity is proposed; its good performance is demonstrated via simulations and real data examples in Section 3. It should be noted here that while we focus on the monotonicity correction for local linear estimators of the conditional distribution, the monotonicity correction idea can equally be applied to other distribution estimators constructed via different nonparametric methods, e.g. wavelets.

2.2 Local Linear Estimation of smooth conditional distributions

2.2.1 Some issues with current methods

The good performance of local constant and local linear estimators (2.5) hinges on $F_x(\cdot)$ being smooth, e.g. continuous, as a function of x. In all that follows, we will further assume that $F_x(y)$ is also continuous in y for all x. Since the estimators (2.5) are discontinuous (step functions) in y, it is customary to smooth them, i.e., define

$$\bar{F}_x(y) = \frac{\sum_{i=1}^{n} \tilde{K}_{i,x} \Lambda(\frac{y-Y_i}{h_0})}{\sum_{i=1}^{n} \tilde{K}_{i,x}}, \text{ and } \bar{F}_x^{LL}(y) = \frac{\sum_{j=1}^{n} w_j \Lambda(\frac{y-Y_j}{h_0})}{\sum_{j=1}^{n} w_j} \tag{2.8}$$

where $\Lambda(y)$ is some smooth distribution function which is strictly increasing with density $\lambda(y) > 0$, i.e., $\Lambda(y) = \int_{-\infty}^{y} \lambda(s)ds$. Here again the local linear weights w_j are given by eq. (2.4), and $h_0 > 0$ is a secondary bandwidth whose choice is not as important as the choice of b; see Section 2.2.5 for some concrete suggestions on picking b and h_0 in practice.

Under standard conditions, both estimators appearing in eq. (2.8) are asymptotically consistent, and preferable to the respective estimators appearing in eq. (2.5), i.e., replacing $\mathbf{1}\{Y_j \leq y\}$ by $\Lambda(\frac{y-Y_j}{h_0})$ is advantageous; see Ch. 6 of (Li & Racine, 2007). Furthermore, as discussed in the Introduction, we expect that $\bar{F}_x^{LL}(y)$ would be a better estimator than $\bar{F}_x(y)$ when x is a boundary point and/or the design is not uniform, while $\bar{F}_x^{LL}(y)$ and $\bar{F}_x(y)$ would be practically equivalent when x is an interior point and the design is (approximately) uniform. Hence, all in all, $\bar{F}_x^{LL}(y)$ would be preferable to $\bar{F}_x(y)$ as an estimator of $F_x(y)$ for any fixed y. The problem again is that $\bar{F}_x^{LL}(y)$ is not guaranteed to be a proper distribution viewed as a function of y by analogy to $\hat{F}_x^{LL}(y)$.

There have been several proposals in the literature to address this issue. An interesting one is the adjusted Nadaraya-Watson estimator of (Hall et al., 1999) that is a linear function of the Y's with weights being selected by an appropriate optimization procedure. The adjusted Nadaraya-Watson estimator is much like a local linear estimator in that it has reduced bias (by an order of magnitude) compared to the regular Nadaraya-Watson local constant estimator. Unfortunately, the adjusted Nadaraya-Watson estimator does not work well when x is a boundary point as the required optimization procedure typically does not admit a solution.

Noting that the problems with $\bar{F}_x^{LL}(y)$ and $\hat{F}_x^{LL}(y)$ arise due to potentially negative weights w_j computed by eq. (2.4), Hansen proposed a straightforward adjustment to the local linear estimator that maintains its favorable asymptotic properties (Hansen, 2004) .

The local linear versions of $\hat{F}_x(y)$ and $\bar{F}_x(y)$ adjusted via Hansen's proposal are given as follows:

$$\hat{F}_x^{LLH}(y) = \frac{\sum_{i=1}^n w_i^\diamond \mathbf{1}(Y_i \leq y)}{\sum_{i=1}^n w_i^\diamond} \text{ and } \bar{F}_{x_m}^{LLH}(y) = \frac{\sum_{i=1}^n w_i^\diamond \Lambda(\frac{y-Y_i}{h_0})}{\sum_{i=1}^n w_i^\diamond} \tag{2.9}$$

where

$$w_i^\diamond = \begin{cases} 0 & \text{when } \hat{\beta}(x-x_i) > 1 \\ \tilde{K}_{i,x}\left(1-\hat{\beta}(x-x_i)\right) & \text{when } \hat{\beta}(x-x_i) \leq 1. \end{cases} \tag{2.10}$$

Essentially, Hansen's proposal replaces negative weights by zeros, and then renormalizes the nonzero weights. The problem here is that if x is on the boundary, negative weights are crucially needed in order to ensure an extrapolation takes place with minimal bias; this is further elaborated upon in the following subsection.

2.2.2 Extrapolation vs. interpolation

In order to illustrate the need for negative weights consider the simple case of $n = 2$, i.e., two data points (Y_1,x_1) and (Y_2,x_2). The question is to predict a future response Y_3 associated with a regressor value of x_3; assuming finite second moments, the L_2–optimal predictor of Y_3 is $\mu(x_3)$ where $\mu(x) = E(Y|X = x)$ as before.

If x_3 is an interior point as depicted in Figure 2.1, the problem is one of *interpolation*. If x_3 is a boundary point, and in particular if x_3 is outside the convex hull of the design points as in Figure 2.2, the problem is one of *extrapolation*. Let $\hat{\mu}^{LL}(x)$ denote the local linear estimator of $\mu(x)$ as before. With $n = 2$, $\hat{\mu}^{LL}(x)$ reduces to just finding the line that passes through the two data points (Y_1,x_1) and (Y_2,x_2). In other words, $\hat{\mu}^{LL}(x)$ reduces to a convex combination of Y_1 and Y_2, i.e., $\hat{\mu}^{LL}(x) = \omega_x Y_1 + (1-\omega_x)Y_2$ where $\omega_x = \frac{x_2-x}{x_2-x_1}$ where

$x_1 < x < x_2$ for interior points and $x_1 < x_2 < x$ for boundary points. Note that $\omega_x \in [0,1]$ if x is an interior point, whereas $\omega_x \notin [0,1]$ if x is outside the convex hull of the design points. Hence, negative weights are a *sine qua non* for effective linear extrapolation.

For example, assume we are in the setup of Figure 2.2 where $x_1 < x_2 < x_3$. In this case, ω_{x_3} is negative. Hansen's proposal (Hansen, 2004) would replace ω_{x_3} by zero and renormalize the coefficients, leading to $\hat{\mu}^{LLH}(x_3) = Y_2$; it is apparent that this does not give the desired linear extrapolation effect.

To generalize the above setup, suppose that now n is an arbitrary even number, and Y_i represents the average of $n/2$ responses associated with regressor value x_i for $i = 1$ or 2. Thus, we have a *bona fide* n–dimensional scatterplot that is supported on two design points. Interestingly, the formula for $\hat{\mu}^{LL}(x)$ is exactly as given above, and so is the argument requiring negative weights for linear extrapolation. Of course, we cannot expect a general scatterplot to be supported on just two design points. Nonetheless, in a nonparametric situation one performs a linear regression *locally*, i.e., using a local subset of the data. Typically, there is a scarcity of design points near the boundary, and the general situation is qualitatively similar to the case of two design points.

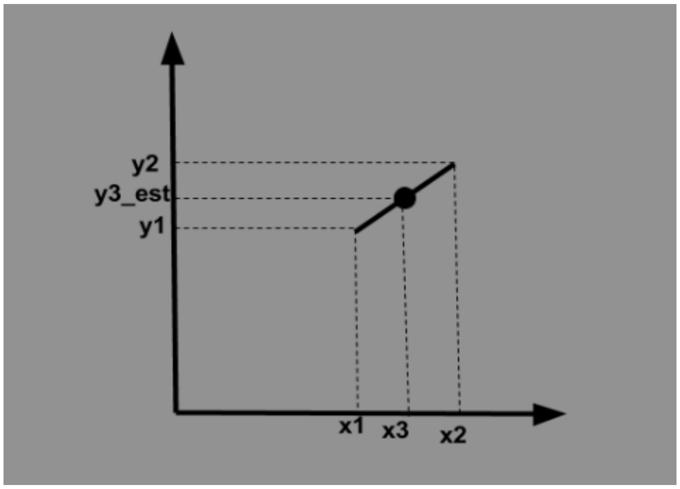

Figure 2.1: Interpolation: prediction of Y_3 when x_3 is an interior point; $\hat{Y}_3$ is a convex combination of Y_1 and Y_2 with nonnegative weights.

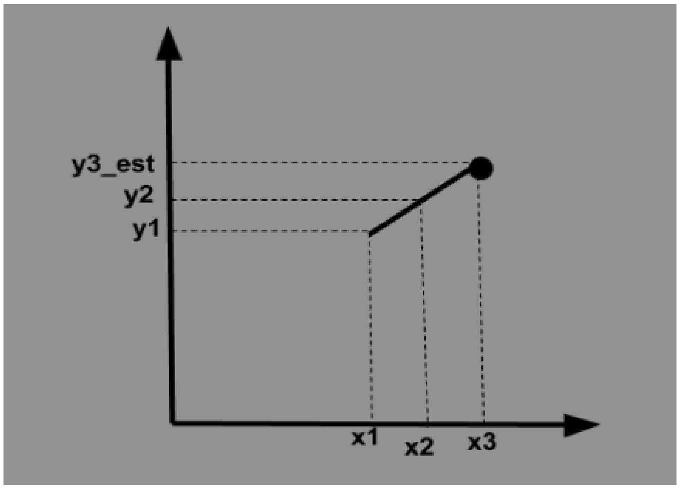

Figure 2.2: Extrapolation: prediction of Y_3 when x_3 is outside the convex hull of the design points; $\hat{Y}_3$ is a linear combination of Y_1 and Y_2 with one positive and one negative weight.

2.2.3 Monotone Local Linear Distribution Estimation

The estimator $\hat{F}_x^{LL}(y)$ from eq. (2.5) is discontinuous as a function of y therefore we will focus our attention on $\bar{F}_x^{LL}(y)$ described in eq. (2.8) from here on. It seems that with this double-smoothed estimator $\bar{F}_x^{LL}(y)$ we can "have our cake and eat it too", i.e., modify it towards monotonicity while retaining (some of) the negative weights that are helpful in the extrapolation problem as discussed in the last subsection. We are thus led to define a new estimator denoted by $\bar{F}_x^{LLM}(y)$ which is a monotone version of $\bar{F}_x^{LL}(y)$; we will refer to $\bar{F}_x^{LLM}(y)$ as the *Monotone Local Linear Distribution Estimator*.

One way of constructing the estimator $\bar{F}_x^{LLM}(y)$ is by first constructing its associated density function denoted by $\bar{f}_x^{LLM}(y)$ which will be called the *Monotone Local Linear Density Estimator*. This algorithm goes as follows.

Algorithm 1 (A1)

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

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

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

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

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

$$\bar{f}_x^{LLM}(y) = \frac{\bar{f}_x^{LL+}(y)}{\int_{-\infty}^{\infty} \bar{f}_x^{LL+}(s)ds}. \tag{2.11}$$

4. Finally, define $\bar{F}_x^{LLM}(y) = \int_{-\infty}^{y} \bar{f}_x^{LLM}(s)ds$.

To implement the above one would again need to divide the range of the y variable using a grid of size ε in order to construct the maximum function in step 2 of Algorithm 1. The same grid can by used to provide Riemann-sum approximations to the two integrals appearing in steps 3 and 4.

Algorithm 1 works by computing the density function in Step 1 using local linear estimation with both positive and negative weights. In general this obtained ”density” may not be positive for all values of y and thus in Step 2 a correction is made to make it assume only zero or positive values at all points. The obtained value are then normalized in Step 3 so that it integrates to 1 and therefore represents a legitimate density function. Finally the monotone distribution function $\bar{F}_x^{LLM}(y)$ is obtained in Step 4.

An alternative way to correct the monotonicity of $\bar{F}_x^{LL}(y)$ is via a direct construction as follows.

Algorithm 2 (A2)

1. Compute the unconstrained double-smoothed estimator $\bar{F}_x^{LL}(y)$.

2. Choose a small enough $\varepsilon > 0$ and divide the range of the y variable using a uniform grid of size ε. Let the gridded values be denoted by y_i where $y_i = y_{i-1} + \varepsilon$. Let $y^* =$ min $\{y : \bar{F}_x^{LL}(y) = 1\}$ and k_* be the minimum integer such that $y_{k_*} \geq y^*$. Similarly, let $y_* =$ max $\{y : \bar{F}_x^{LL}(y) = 0\}$ and k^* be the maximum integer such that $y_{k^*} \leq y_*$.

3. Define a second function G_2 with the property that $G_2(y) = 0$ for all $y \leq y_*$.

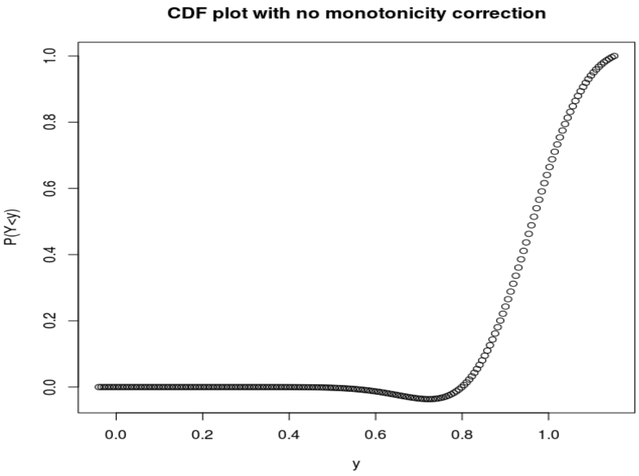

Figure 2.3: Nonmonotonicity in CDF with negative weights

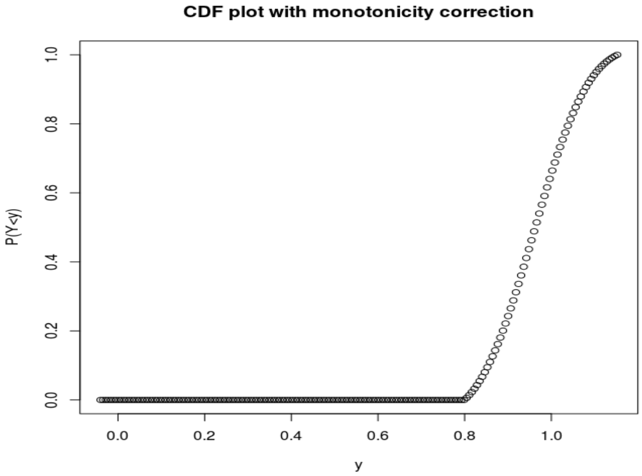

Figure 2.4: Monotone CDF

Now let $G_2(y) = \max\ (\ \bar{F}_x^{LL}(y_{i+1}),\ G_2(y_i)\ )$ for $y \in (y_i, y_{i+1}]$.

Repeat this procedure and define $G_2(y)$ for $i = k^*, k^*+1, \ldots, k_*$.

4. Define $\bar{F}_x^{LLM}(y) = G_2(y)/L$ where $L = \lim_{y\to\infty} G_2(y)$.

Algorithm 2 works by first estimating the distribution function in Step 1 using local linear estimation with both positive and negative weights. In general this obtained "distribution" may not be monotonic and thus in Step 3 monotonicity corrections are effected in the direction of increasing y values. Finally the range of values obtained are scaled in Step 4 so that the final values of $\bar{F}_x^{LLM}(y)$ lie between 0 and 1 and therefore represent a legitimate distribution function.

Illustrations of monotonicity corrections using Algorithm A2 are given in the following figures. In Figure 2.3 after applying positive and negative local linear weights a nonmonotonic distribution function is obtained. This is corrected for monotonicity with appropriate scaling of values to lie between $[0,1]$ using Algorithm 2 as shown in Figure 2.4.

It should also be noted that the monotonicity correction proposal in Step 2 of Algorithm 2 is not unique. Algorithm 2 works from left to right thereby forcing the estimated distribution to be an increasing function. One alternative to this procedure would be to first calculate the generally nonmonotonic function $\bar{F}_x^{LL}(y)$ and then proceed from right to left thereby achieving monotonicity correction by forcing the estimated distribution to be decreasing when y decreases. This is described in Algorithm 3 below.

Algorithm 3 (A3)

1. Compute the unconstrained double-smoothed estimator $\bar{F}_x^{LL}(y)$.

2. Choose a small enough $\varepsilon > 0$ and divide the range of the y variable using a uniform grid of size ε. Let the gridded values be denoted by y_i where $y_i = y_{i-1} + \varepsilon$. Let $y^* =$ min $\{y : \bar{F}_x^{LL}(y) = 1\}$ and k_* be the minimum integer such that $y_{k_*} \geq y^*$. Similarly, let $y_* =$ max $\{y : \bar{F}_x^{LL}(y) = 0\}$ and k^* be the maximum integer such that $y_{k^*} \leq y_*$.

3. Define a second function G_2 with the property that $G_2(y) = 1$ for all $y \geq y^*$.

Now let $G_2(y) = \min\ (\ \bar{F}_x^{LL}(y_{i-1}),\ G_2(y_i)\ )$ for $y \in [y_{i-1}, y_i)$.

Repeat this procedure and define $G_2(y)$ for $i = k_*, k_* - 1, \ldots, k^*$.

4. Denote $l = \lim_{y \to -\infty} G_2(y)$ and define a function $G_3(y) = G_2(y) - l$.

5. Define $\bar{F}_x^{LLM}(y) = G_3(y)/L$ where $L = \lim_{y \to \infty} G_3(y)$.

Algorithm 3 is similar in style to Algorithm 2 the key difference being that it runs over grid points from right to left (decreasing index) whereas the latter runs over grid points from left to right (increasing index).

Yet another method of monotonicity correction can be based on the important notion due to (Brunk, 1955), namely the Pool Adjacent Violators Algorithm (PAVA). Algorithm 4 shows how we can implement PAVA in order to correct $\bar{F}_x^{LL}(y)$ for monotonicity.

Algorithm 4 (A4)

1. Compute the unconstrained double-smoothed estimator $\bar{F}_x^{LL}(y)$.

2. Choose a small enough $\varepsilon > 0$ and divide the range of the y variable using a uniform grid of size ε. Let the gridded values be denoted by y_i where $y_i = y_{i-1} + \varepsilon$. Let $Z_i = \bar{F}_x^{LL}(y_i)$.

3. The PAVA algorithm runs over these gridded points as follows:

   (a) Consider values Z_{k-1} and Z_k with frequencies of occurrence f_{k-1} and f_k respectively.

   (b) If $Z_{k-1} > Z_k$ then define:

$W_{k-1} = W_k = \frac{f_{k-1}Z_{k-1} + f_k Z_k}{f_{k-1} + f_k}$

Set $f_{k-1} = f_k = (f_{k-1} + f_k)$

else define:

$W_{k-1} = Z_{k-1}$ and $W_k = Z_k$

f_{k-1}, f_k remain unchanged

(c) Repeat the above steps (a) and (b) in decreasing order of indices until all gridded points y_i are covered. Define a new function $G_2(y)$ by letting $G_2(y_i) = W_i \; \forall i$ and if $y \in (y_i, y_{i+1})$ let $G_2(y) = G_2(y_i)$ which are the monotonicity corrected updated values from $\bar{F}_x^{LL}(y)$ obtained in the iterations above.

4. Denote $l = \lim_{y \to -\infty} G_2(y)$ and define a function $G_3(y) = G_2(y) - l$.

5. Define $\bar{F}_x^{LLM}(y) = G_3(y)/L$ where $L = \lim_{y \to \infty} G_3(y)$.

Remark 2.2.1 (Performance and runtime comparison of A1,A2,A3,A4) *It should be noted that the novel aspect of our proposal is to keep negative weights when required and then make a monotonicity correction so that one can obtain well-defined estimates of the distribution function $\bar{F}_x^{LLM}(y)$. Either of the 4 algorithms above can be applied to obtain the monotone distribution function. Indeed, other algorithms could also be devised for this purpose. Notably Algorithms 1,2,3,4 perform similarly on data be it real or simulated. However Algorithm 1 is preferable as it can be implemented much faster than Algorithms 2,3,4. The reason is that density estimates can be obtained very rapidly using built-in functions in statistical software such as R. This is especially helpful during bootstrap when* *a large number of estimates of $\bar{F}_x^{LLM}(y)$ are required to obtain confidence intervals as in Section 2.2.4. A performance comparison of Algorithms 1,2,3,4 based on a simulation study over a dataset with i.i.d. errors is given in Tables 2.5 and 2.10. Based on point prediction results over this dataset it is observed that in our current implementation A1 runs approximately* 9 *times faster than A2, A3 and* 6 *times faster than A4.*

Remark 2.2.2 (Comparison with Isotonic Regression) *In addition we would also like to point out that there have been several methods proposed in the literature for "isotonic" regression i.e. monotonic estimation of a regression function $\mu(x) = E(Y|X = x)$ such as the ones described in (Brunk, 1955), (Hall & Huang, 2001) and (Dette, Neumeyer, Pilz, et al., 2006). While these are valid methods for constructing a monotone estimate of $E(Y|X = x)$ as a function of x in our case we require an estimate of $F_x(y) = E(W|X = x)$ where $W = \mathbf{1}\{Y \leq y\}$ that is monotone with respect to y and not with respect to x.*

2.2.4 Standard Error of the Monotone Local Linear Estimator

Under standard conditions, the local linear estimator $\sqrt{nb}\bar{F}_x^{LL}(y)$ is asymptotically normal with a variance $V_{x,y}^2$ that depends on the design; for details, see Ch. 6 of (Li & Racine, 2007). In addition, the bias of $\sqrt{nb}\bar{F}_x^{LL}(y)$ is asymptotically vanishing if $b = o(n^{1/5})$. Hence, letting $b \sim n^{\alpha}$ for some $\alpha \in (0, 1/5)$, $\bar{F}_x^{LL}(y)$ will be consistent for $F_x(y)$, and approximate 95% confidence intervals for $F_x(y)$ can be constructed as $\bar{F}_x^{LL}(y) \pm 1.96\frac{V_{x,y}}{nb}$.

The consistency of $\bar{F}_x^{LL}(\cdot)$ towards $F_x(\cdot)$ implies that the monotonicity corrections described in the previous subsection will be asymptotically negligible. To see why, fix a point x of interest, and assume that $F_x(y)$ is absolutely continuous with density $f_x(y)$ that is strictly positive over its support. The consistency of $\bar{f}_x^{LL}(y)$ to a positive target implies that

$\bar{f}_x^{LL}(y)$ will eventually become (and stay) positive as n increases. Hence, the monotonicity correction eventually vanishes, and $\bar{F}_x^{LLM}(y)$ is asymptotically equivalent to $\bar{F}_x^{LL}(y)$.

Regardless, it is not advisable to use the aforementioned asymptotic distribution and variance of $\bar{F}_x^{LL}(y)$ to approximate those of $\bar{F}_x^{LLM}(y)$ for practical work since, in finite samples, $\bar{F}_x^{LLM}(y)$ and $\bar{F}_x^{LL}(y)$ can be quite different. Our recommendation is to use some form of bootstrap in order to approximate the distribution and/or standard error of $\bar{F}_x^{LLM}(y)$ directly. In particular, the Model-Free bootstrap (Politis, 2015) in its many forms is immediately applicable in the present context. For instance, the "Limit Model-Free" (LMF) bootstrap would go as follows:

LMF Bootstrap Algorithm

1. Generate $U_1, \ldots, U_n$ i.i.d. Uniform(0,1).

2. Define $Y_i^* = G_{x_i}^{-1}(U_i)$ for $i = 1, \ldots, n$ where $G_{x_i}^{-1}(\cdot)$ is the quantile inverse of $\bar{F}_{x_i}^{LLM}(\cdot)$, i.e., $G_{x_i}^{-1}(u) = \inf\{y : \bar{F}_{x_i}^{LLM}(y) \geq u\}$.

3. For the points x and y of interest, construct the pseudo-statistic $\bar{F}_x^{LLM*}(y)$ which is computed by applying estimator $\bar{F}_x^{LLM}(y)$ to the bootstrap dataset $(Y_1^*, x_1) \ldots, (Y_n^*, x_n)$.

4. Repeat steps 1–3 a large number (say B) times. Plot the B pseudo-replicates $\bar{F}_x^{LLM*}(y)$ in a histogram that will serve as an approximation of the distribution of $\bar{F}_x^{LLM}(y)$. In addition, the sample variance of the B pseudo-replicates $\bar{F}_x^{LLM*}(y)$ is the bootstrap estimator of the variance of $\bar{F}_x^{LLM}(y)$.

Our focus is on point estimation of $F_x(y)$ so we will not elaborate further on the construction of interval estimates.

2.2.5 Bandwidth Choice

There are two bandwidths, b and h_0, required to construct estimator $\bar{F}_x^{LLM}(y)$ and its relatives $\bar{F}_x(y)$ and $\bar{F}_x^{LLH}(y)$. We will now focus on choice of the bandwidth b which is the most crucial of the two, and is often picked via leave-one-out cross-validation.

In this Chapter we are mostly concerned with estimation and prediction at boundary points. Since often boundary problems present their own peculiarities, we are strongly recommending carrying out the cross-validation procedure 'locally', i.e., over a neighborhood of the point of interest. One needs, however, to ensure that there are enough points nearby to perform the leave-one-out experiment. Hence, our concrete recommendation goes as follows.

- Choose a positive integer m which can be fixed number or it can be a small fraction of the sample size at hand.
- Then, identify m among the regression points $(Y_1, x_1), \ldots, (Y_n, x_n)$ with the property that their respective x_i's are the m closest neighbors of the point x under consideration.
- Denote this set of m points by $(Y_{g(1)}, x_{g(1)}), \ldots, (Y_{g(m)}, x_{g(m)})$ where the function $g(\cdot)$ gives the index numbers of the selected points.
- For $k = 1, \ldots, m$, compute $\hat{Y}_{g(k)}$ which is the L_2–optimal predictor of $Y_{g(k)}$ using leave-one-out data. In other words, $\hat{Y}_{g(k)}$ is the mean, i.e., center of location, of one of the aforementioned distribution estimators based on the delete–one dataset, i.e. pretending that $Y_{g(k)}$ is unavailable.
- Thus, for a range of values of bandwidth b, we can calculate the following:

$$Err = \sum_{k=1}^{m} (\hat{Y}_{g(k)} - Y_{g(k)})^2. \tag{2.12}$$

- Finally, the optimal bandwidth is given by the value of b that minimizes Err over the range of bandwidths considered.

Coming back to the problem of selecting h_0, define $h = b/n$ and recall that in an analogous regression problem the optimal rates $h_0 \sim n^{-2/5}$ and $h \sim n^{-1/5}$ were suggested in connection with the nonnegative kernel K; see (Li & Racine, 2007). As in (Politis, 2013), this leads to the practical recommendation of letting $h_0 = h^2$. We will adopt the same rule-of-thumb here as well, namely let $h_0 = b^2/n^2$ where b has been chosen previously via local cross-validation. Note that the initial choice of h_0 (before performing the cross-validation to determine the optimal bandwidth b) can be set by a plug-in rule as available in standard statistical software such as R.

2.3 Numerical work: simulations and real data

The performance of the three distribution estimators $\bar{F}_x(y)$, $\bar{F}_x^{LLH}(y)$ or $\bar{F}_x^{LLM}(y)$ described above are empirically compared using both simulated and real-life datasets according to the following metrics.

1. Divergence between the local distribution $\bar{F}_x(\cdot)$ estimated by all three methods and the corresponding local (empirical) distribution calculated from the actual data; this is determined using the mean value of the Kolmogorov-Smirnov (KS) test statistic (Serfling, 2009). The measurement is performed on simulated datasets where multiple realizations of data at both boundary and internal points are available. Therefore

the empirical distribution at any point can be calculated and compared versus the estimated values. Our notation is **KS-LC, KS-LLH and KS-LLM** for the distribution estimators $\bar{F}_x(y)$, $\bar{F}_x^{LLH}(y)$ or $\bar{F}_x^{LLM}(y)$ respectively.

2. Comparison of estimated quantiles of $F_x(\cdot)$ at specified points using all three methods versus the corresponding empirical values calculated using simulated datasets.

3. Point prediction performance as indicated by bias and Mean Squared Error (MSE) on simulated and real-life datasets using all three methods. The MSE values of point prediction are denoted as **MSE-LC, MSE-LLH and MSE-LLM** for the distribution estimators $\bar{F}_x(y)$, $\bar{F}_x^{LLH}(y)$ or $\bar{F}_x^{LLM}(y)$ respectively; the corresponding bias values are denoted **Bias-LC, Bias-LLH and Bias-LLM**. For comparison purposes the point-prediction performance is also measured using the local linear conditional moment estimator as given by equations 3.17 and 2.4. In this case bias and MSE are indicated as **Bias-LL** and **MSE-LL** respectively.

4. Performance comparison of all 4 monotonicity correction schemes described in Algorithms A1, A2, A3 and A4 for the estimator $\bar{F}_x^{LLM}(y)$. The comparison is done using the mean values of the KS statistic between the local distribution $\bar{F}_x^{LLM}(y)$ using all 4 methods and the corresponding (local) empirical distribution obtained from the data. Comparisons are also performed by measuring the MSE and bias of point prediction on the simulated dataset with i.i.d. errors at a given boundary point.

It should be noted that all performance comparisons involving the KS test statistic and point prediction for the estimator $\bar{F}_x^{LLM}(y)$ are done using Algorithm A1 owing to the shorter runtimes in this case as compared to A2, A3 and A4. Comparisons of all of these 4 algorithms for $\bar{F}_x^{LLM}(y)$ are presented in Tables 2.5 and 2.10 for the dataset with i.i.d. errors.

On simulated datasets the performance metrics for all three distribution estimators are calculated both at boundary and internal points to illustrate how performance varies between $\bar{F}_x(y)$, $\bar{F}_x^{LLH}(y)$ and $\bar{F}_x^{LLM}(y)$ in the two cases. Our simulated datasets contain 500 realizations each with 1001 data points. In such cases using Model-Free cross-validation outlined in Section 2.2.5 is computationally expensive. Therefore for these datasets we perform our comparisons using a range of bandwidth values from $3.7, 7.4, \ldots, 51.8$ in steps of 3.7. For the real-life dataset cross-validation is used to determine the most optimal bandwidth for all predictors.

2.3.1 Simulation: Additive model with i.i.d Gaussian errors

Data Y_i for $i = 1, \ldots, 1001$ were simulated as per model (2.1) by setting $\mu(x_i) = \sin(2\pi x_i)$, $\sigma(x_i) = \tau$ where $x_i = \frac{i}{n}$and the errors ε_i as i.i.d. $N(0,1)$. Sample size n was set to 1001. A total of 500 such realizations were generated for this study.

Results for the mean-value of the Kolmogorov-Smirnov test statistic between the LC, LLH and LLM estimated distributions and empirical distribution calculated using available values of the simulated data are given in Tables 2.1, 2.2, 2.3 and 2.4 for boundary point $n = 1001$ and internal point $n = 200$ for values of $\tau = 0.1$ and 0.5 over a range of bandwidths, i.e., b taking values $3.7, 7.4, \ldots, 51.8$ in steps of 3.7.

Point prediction performance values are provided for the same cases in Tables 2.6, 2.7, 2.8 and 2.9.

Estimates of the α–quantile at specific values of α are calculated using all three distribution estimators and compared with corresponding quantiles calculated from the available data. Plots for selected quantile values ($\alpha = 0.1$ and $\alpha = 0.9$) are shown in Figures 2.5, 2.6, 2.7 and 2.8 for both 1 and 2-sided cases ($\tau = 0.5$). Note that the 'true' quantile lines showed in the plots are values calculated from the available data at $n = 1001$ and $n = 200$ over 500 realizations for the case of boundary and internal points respectively. The bandwidths used for estimating the quantiles for LC, LLH and LLM are based on bandwidth values where the best performance for these estimators was obtained using the Kolmogorov-Smirnov test (refer Tables 2.3 and 2.4).

Note that the point $n = 1001$ is excluded from the data used for LC, LLH and LLM estimation at the boundary point. Similarly the point $n = 200$ is excluded for the case of estimation at the internal point.

It can be seen from Tables 2.1 and 2.3 that in the case of boundary point estimation among the estimators based on $\bar{F}_x(y)$, $\bar{F}_x^{LLH}(y)$ and $\bar{F}_x^{LLM}(y)$ the lowest value of the Kolmogorov-Smirnov test statistic is obtained using the LLM estimator $\bar{F}_x^{LLM}(y)$. In addition again for the boundary case the lowest values of MSE are obtained using the LLM estimator as can be seen from Tables 2.6 and 2.8 which is consistent with the trend seen from minimum values of the KS test-statistic. Minimum values of KS-statistic and MSE are highlighted in bold in all respective tables. In addition it can be seen from the plots of the estimated quantiles at $\alpha = 0.1$ and $\alpha = 0.9$ in the boundary case that the center of the estimated quantile distribution for LLM is aligned more closely to the 'true' quantile value calculated from the simulated data as shown by the dotted line (Figures 2.5 and 2.6). Based on these results it can be concluded that for boundary value estimation the estimator based on $\bar{F}_x^{LLM}(y)$ has superior performance as compared to both $\bar{F}_x(y)$ and $\bar{F}_x^{LLH}(y)$.

In the case of $\tau = 0.1$ at the boundary point $n = 1001$ the LC estimator outperforms the LLH and LLM estimators only at very low bandwidths. As can be seen from Table 2.6 for the case of $\tau = 0.1$ with increasing bandwidth the 2 local linear estimators LLH and LLM perform significantly better than the LC owing to lower bias. In this case over a large range of bandwidths from $11.1, 14.8, \ldots, 51.8$ LLH and LLM outperform LC. In the case of $\tau = 0.5$ at the boundary point $n = 1001$ as can be seen from Table 2.8 even though the bias of the local linear estimators are lower at higher bandwidths compared to that of the LC estimator the latter outperforms these for a larger range of bandwidths as compared to the case of $\tau = 0.1$ owing to larger estimation variance for the local linear estimators. In this case the LLM estimator outperforms the LC and LLH estimators only for the highest bandwidths, namely $48.1, 51.8$. However as stated earlier the overall best performance is obtained from the LLM estimator.

For the case of estimation at internal points no appreciable differences in performance are noticeable between the 3 estimators using both the mean values of the Kolmogorov-Smirnov test statistic (Tables 2.2 and 2.4) and also using mean-square error of point prediction (Tables 2.7 and 2.9). Similar trends are noticeable in the quantile plots where the estimated quantiles using LC, LLH and LLM nearly overlap for the internal case (Figures 2.7 and 2.8).

It can also be seen from Tables 2.6, 2.7, 2.8 and 2.9 that across the range of bandwidths considered there is negligible loss in best point prediction performance of LLM versus that of LL.

We also perform a comparison of Algorithms A1, A2, A3 and A4 with this dataset using both the Kolmogorov-Smirnov test statistic and point prediction. The overall best performance among all bandwidths of all algorithms is nearly the same as can be seen from the minimum KS-statistic values for each of A1,A2,A3 and A4 highlighted in bold in Table 2.5. It is only at higher bandwidths that compared to A1 algorithms A2, A3 and A4 perform better and for the highest bandwidth of 51.8 algorithm A2 has the lowest KS-statistic value among all 4 considered. In case of point prediction as shown in Table 2.10 the overall best

Table 2.1: Mean values of KS test statistic over i.i.d. errors at boundary point ($n = 1001, \tau = 0.1$)

Bandwidth	KS-LC	KS-LLH	KS-LLM
3.7	**0.23508**	0.252884	0.275132
7.4	0.241992	0.233996	0.23606
11.1	0.2767	**0.232064**	0.218948
14.8	0.31528	0.240476	0.20744
18.5	0.349924	0.2554	**0.2009**
22.2	0.38438	0.273648	0.204404
25.9	0.418316	0.288032	0.21502
29.6	0.448772	0.307672	0.231588
33.3	0.474796	0.326224	0.253472
37.0	0.502768	0.342884	0.275936
40.7	0.5264	0.360888	0.2993
44.4	0.54664	0.37786	0.320348
48.1	0.56692	0.393392	0.34248
51.8	0.58646	0.407108	0.359404

performance among all bandwidths as highlighted in bold is almost equal among all methods. However a trend similar to that of the Kolmogorov-Smirnov statistic can be observed at higher bandwidths where the MSE of point prediction is better for algorithms A2, A3 and A4 as compared to A1. In this case the best performance at the highest bandwidth of 51.8 is observed from Algorithms A2 and A4.

Table 2.2: Mean values of KS test statistic over i.i.d. errors at internal point ($n = 200, \tau = 0.1$)

Bandwidth	KS-LC	KS-LLH	KS-LLM
3.7	0.212296	0.213792	0.213712
7.4	0.201892	0.203264	0.203704
11.1	0.197736	0.198904	0.197828
14.8	0.19782	0.197296	**0.196772**
18.5	**0.19606**	**0.1949**	0.19684
22.2	0.200164	0.198304	0.198556
25.9	0.202644	0.201472	0.202208
29.6	0.206016	0.20534	0.207628
33.3	0.21412	0.212608	0.21422
37.0	0.220084	0.221096	0.2204
40.7	0.23078	0.23064	0.231744
44.4	0.240556	0.238724	0.240032
48.1	0.250116	0.250692	0.250972
51.8	0.260864	0.260696	0.259292

Table 2.3: Mean values of KS test statistic over i.i.d. errors at boundary point ($n = 1001, \tau = 0.5$)

Bandwidth	KS-LC	KS-LLH	KS-LLM
3.7	0.207104	0.303696	0.352912
7.4	0.148964	0.210324	0.250856
11.1	0.125284	0.171268	0.2058
14.8	0.112412	0.15016	0.182176
18.5	**0.107232**	0.136612	0.16702
22.2	0.107764	0.127176	0.154944
25.9	0.111144	0.121408	0.145624
29.6	0.119836	0.115008	0.136968
33.3	0.126996	0.110716	0.128792
37.0	0.137376	0.108468	0.121452
40.7	0.14676	**0.105504**	0.1165
44.4	0.157364	0.107432	0.111452
48.1	0.165528	0.108692	0.107532
51.8	0.175852	0.110228	**0.103772**

Table 2.4: Mean values of KS test statistic over i.i.d. errors at internal point ($n = 200, \tau = 0.5$)

Bandwidth	KS-LC	KS-LLH	KS-LLM
3.7	0.152968	0.15334	0.152252
7.4	0.119528	0.117216	0.118916
11.1	0.103412	0.104188	0.104388
14.8	0.097028	0.097544	0.097348
18.5	0.0897	0.089944	0.090576
22.2	0.0868	0.086116	0.087244
25.9	0.083068	0.083164	0.084304
29.6	0.082208	0.081544	0.081452
33.3	0.080592	0.081848	0.081572
37.0	**0.07958**	**0.08006**	**0.078328**
40.7	0.080208	0.080568	0.079604
44.4	0.08194	0.08094	0.082332
48.1	0.082628	0.08288	0.082256
51.8	0.084188	0.08518	0.086076

Table 2.5: Mean values of KS test statistic for different monotonicity correction schemes over i.i.d. errors ($n = 1001, \tau = 0.1$)

Bandwidth	KS-A1	KS-A2	KS-A3	KS-A4
3.7	0.275132	0.275149	0.274824	0.276483
7.4	0.23606	0.235748	0.237324	0.237577
11.1	0.218948	0.218420	0.216216	0.217999
14.8	0.20744	0.206292	0.20604	0.205644
18.5	**0.2009**	0.196003	**0.197392**	0.196902
22.2	0.204404	**0.191834**	0.19986	**0.193544**
25.9	0.21502	0.195243	0.208152	0.196858
29.6	0.231588	0.199259	0.225812	0.204090
33.3	0.253472	0.211065	0.247672	0.217721
37.0	0.275936	0.228099	0.269192	0.237243
40.7	0.2993	0.242803	0.29148	0.255088
44.4	0.320348	0.263343	0.313924	0.275137
48.1	0.34248	0.281416	0.334704	0.297277
51.8	0.359404	0.299183	0.35426	0.315886

Table 2.6: Point Prediction for Boundary Value over i.i.d. errors ($n = 1001, \tau = 0.1$)

Ban	Bias-LC	MSE-LC	Bias-LLH	MSE-LLH	Bias-LLM	MSE-LLM	Bias-LL	MSE-LL
3.7	-0.01887676	0.01265856	-0.0087034	0.01453471	0.0004694887	0.01667712	0.00279478	0.01713243
7.4	-0.03782673	**0.01261435**	-0.01818502	0.0126929	0.0005444976	0.01323652	0.003247646	0.01340418
11.1	-0.05753609	0.01418224	-0.02725602	**0.01232877**	-0.001022256	0.01200918	0.0039133	0.01219628
14.8	-0.07724901	0.01672728	-0.03718728	0.01259729	-0.005397138	0.01148354	0.00354838	0.01167496
18.5	-0.09692561	0.0200906	-0.04758345	0.01327841	-0.01222596	**0.01130622**	0.002834568	0.01139095
22.2	-0.116533	0.02423279	-0.05831195	0.01431087	-0.02106315	0.01142789	0.002008806	0.01120327
25.9	-0.1359991	0.02911512	-0.06918129	0.0156254	-0.03138586	0.01185914	0.001102312	0.01106821
29.6	-0.1555938	0.03480583	-0.08021998	0.01722284	-0.04274234	0.01263368	8.912064e-05	0.01096947
33.3	-0.1752324	0.04128715	-0.09144259	0.01910772	-0.05473059	0.01375585	-0.001070282	0.01089842
37.0	-0.1947342	0.04848954	-0.1027918	0.02127558	-0.0670785	0.01521865	-0.002416635	0.01084951
40.7	-0.2145001	0.05656322	-0.1142845	0.02374615	-0.07967838	0.01704094	-0.003988081	0.01081946
44.4	-0.2343967	0.06548142	-0.1259372	0.02651703	-0.09236019	0.01919461	-0.005818943	**0.01080699**
48.1	-0.2543523	0.07522469	-0.1377167	0.02960364	-0.1050934	0.02168698	-0.007939144	0.01081259
51.8	-0.2740635	0.08563245	-0.1496325	0.03301117	-0.1178388	0.02451228	-0.01037417	0.01083832

Table 2.7: Point Prediction for Internal Value over i.i.d. errors ($n = 200, \tau = 0.1$)

Ban	Bias-LC	MSE-LC	Bias-LLH	MSE-LLH	Bias-LLM	MSE-LLM	Bias-LL	MSE-LL
3.7	0.005693694	0.01026982	0.005815108	0.01027252	0.005811741	0.01027231	0.005672309	0.01027341
7.4	0.004548762	0.009868812	0.004644668	0.009871222	0.004640743	0.009871005	0.004547984	0.009883257
11.1	0.003077572	0.009736622	0.003193559	0.009739295	0.003189924	0.009738919	0.003108078	0.009754927
14.8	0.001168265	0.009684642	0.001329604	0.009685997	0.001325573	0.009685696	0.001205735	0.009703492
18.5	-0.001163283	**0.009671566**	-0.0009392514	**0.009670138**	-0.0009440976	**0.009670008**	-0.001162398	**0.009689214**
22.2	-0.003874557	0.009682447	-0.00359328	0.009680945	-0.003598744	0.009680969	-0.003997042	0.009703
25.9	-0.006944759	0.009723612	-0.006615935	0.009717111	-0.006621406	0.009717225	-0.007307346	0.009745675
29.6	-0.01035534	0.009789969	-0.009987875	0.009781065	-0.009992804	0.009781194	-0.01109961	0.009822695
33.3	-0.01407319	0.009888265	-0.01368629	0.009877023	-0.01369037	0.009877157	-0.01537421	0.009942768
37.0	-0.01808254	0.01002258	-0.01768867	0.01001026	-0.01769184	0.01001041	-0.02012788	0.01011708
40.7	-0.02234318	0.01020278	-0.02197526	0.01018668	-0.02197765	0.01018686	-0.02535515	0.01035866
44.4	-0.02686568	0.01042781	-0.02652964	0.01041258	-0.02653147	0.0104128	-0.03104801	0.0106819
48.1	-0.03163397	0.01071166	-0.03133849	0.01069454	-0.03133999	0.01069479	-0.03719388	0.01110199
51.8	-0.03662567	0.01105637	-0.03639079	0.01103926	-0.03639212	0.01103955	-0.04377252	0.0116341

Table 2.8: Point Prediction for Boundary Value over i.i.d. errors ($n = 1001, \tau = 0.5$)

Ban	Bias-LC	MSE-LC	Bias-LLH	MSE-LLH	Bias-LLM	MSE-LLM	Bias-LL	MSE-LL
3.7	0.04888178	0.301925	0.07073083	0.3540897	0.07920865	0.384878	0.0808868	0.4035579
7.4	0.02525561	0.2802656	0.05074344	0.3037839	0.06735271	0.3233827	0.07335949	0.3276234
11.1	0.00374298	0.2731737	0.038811	0.2892723	0.06222332	0.3013529	0.07053577	0.3027942
14.8	-0.01695169	0.270537	0.02715055	0.2805281	0.05849931	0.2922475	0.06822172	0.293552
18.5	-0.03718522	**0.2696291**	0.01614152	0.2761872	0.05612087	0.2867147	0.06656515	0.2892179
22.2	-0.05753523	0.2699832	0.005048478	0.2739688	0.05384922	0.2829767	0.0649322	0.2860192
25.9	-0.07760465	0.271603	-0.005574987	0.2723361	0.0513094	0.2798923	0.06320544	0.2830793
29.6	-0.09765073	0.2742877	-0.01633413	0.271128	0.04834131	0.2770397	0.06143642	0.2803242
33.3	-0.1176859	0.2780296	-0.02722356	0.2704552	0.04514186	0.2748099	0.05960562	0.2778554
37.0	-0.1373472	0.2827116	-0.0383895	**0.2701542**	0.04137961	0.2727937	0.05763437	0.2757286
40.7	-0.1572939	0.2883236	-0.04971082	0.2703248	0.03701344	0.2709994	0.05544761	0.2739321
44.4	-0.1769863	0.294608	-0.0611495	0.2709176	0.03212707	0.2695289	0.0530012	0.2724221
48.1	-0.1965911	0.3018083	-0.07255455	0.2717088	0.02680826	0.2683285	0.05027668	0.2711495
51.8	-0.2158054	0.3097015	-0.08401642	0.2728317	0.02098977	**0.2673724**	0.04726651	**0.2700701**

Table 2.9: Point Prediction for Internal Value over i.i.d. errors ($n = 200, \tau = 0.5$)

Ban	Bias-LC	MSE-LC	Bias-LLH	MSE-LLH	Bias-LLM	MSE-LLM	Bias-LL	MSE-LL
3.7	0.009184716	0.2520511	0.01220923	0.2521932	0.01220434	0.2521952	0.007409901	0.2516582
7.4	0.01372525	0.2431836	0.01526585	0.2435718	0.01526117	0.2435718	0.01263826	0.2426903
11.1	0.0148307	0.2398743	0.01582708	0.2401292	0.01582349	0.2401341	0.01395436	0.2395701
14.8	0.0135934	0.2381523	0.01432564	0.2382689	0.01432314	0.2382728	0.01288775	0.2379284
18.5	0.01125721	0.236852	0.011912	0.2369737	0.01190766	0.2369759	0.01078182	0.2367428
22.2	0.008293956	0.2359636	0.008883749	0.2359976	0.008879099	0.2360007	0.007971824	0.2358225
25.9	0.004809638	0.2352631	0.005346559	0.2352719	0.005342764	0.235277	0.004580992	0.235121
29.6	0.0009735356	0.2347759	0.001361408	0.2347516	0.001357999	0.2347585	0.0006901448	0.2346118
33.3	-0.003467449	0.234453	-0.003042608	0.2344041	-0.003046705	0.2344117	-0.00365963	0.2342717
37.0	-0.008232451	0.2342181	-0.007859816	0.2342051	-0.007864671	0.2342125	-0.008456445	0.2340811
40.7	-0.01347908	**0.2341583**	-0.01309377	**0.2341384**	-0.01309954	**0.234145**	-0.01370081	**0.2340256**
44.4	-0.01912791	0.2342317	-0.01874779	0.2341951	-0.01875384	0.2342009	-0.01939379	0.2340969
48.1	-0.02516629	0.2344631	-0.0248178	0.2343727	-0.02482374	0.234378	-0.02553028	0.2342927
51.8	-0.0316367	0.2347946	-0.0312908	0.2346738	-0.03129606	0.2346788	-0.03209508	0.2346152

Table 2.10: Point prediction for different monotonicity correction schemes over i.i.d. errors ($n = 1001, \tau = 0.1$)

Ban	Bias-LLM-A1	MSE-LLM-A1	Bias-LLM-A2	MSE-LLM-A2	Bias-LLM-A3	MSE-LLM-A3	Bias-LLM-A4	MSE-LLM-A4
3.7	0.0004694887	0.01667712	0.000841778	0.01677949	0.001150729	0.0167038	0.001141823	0.0167835
7.4	0.0005444976	0.01323652	0.0009324214	0.01327941	0.001407732	0.0132403	0.001360984	0.01327423
11.1	-0.001022256	0.01200918	-0.0003834476	0.01205771	0.0002215077	0.01200811	0.0003438006	0.01205052
14.8	-0.005397138	0.01148354	-0.004193309	0.01154498	-0.003389566	0.01146531	-0.002862872	0.0115196
18.5	-0.01222596	**0.01130622**	-0.009974438	**0.01133137**	-0.0087616	0.01123066	-0.007608046	0.01126663
22.2	-0.02106315	0.01142789	-0.01703069	0.0113444	-0.01551549	**0.01122922**	-0.01342514	**0.01121847**
25.9	-0.03138586	0.01185914	-0.02501421	0.01156919	-0.02386251	0.01144391	-0.02043524	0.01135003
29.6	-0.04274234	0.01263368	-0.03364703	0.01198791	-0.0338179	0.01194188	-0.02872835	0.01167511
33.3	-0.05473059	0.01375585	-0.04277015	0.01263349	-0.04477437	0.01275655	-0.03799226	0.01223478
37.0	-0.0670785	0.01521865	-0.05219742	0.01347599	-0.05636865	0.0138868	-0.04802831	0.01305605
40.7	-0.07967838	0.01704094	-0.06192437	0.01455663	-0.06848161	0.01537956	-0.05849318	0.01413659
44.4	-0.09236019	0.01919461	-0.07185436	0.01585587	-0.08077665	0.01719065	-0.06922099	0.0154773
48.1	-0.1050934	0.02168698	-0.08190542	0.01738198	-0.09315366	0.01932202	-0.08017627	0.01709562
51.8	-0.1178388	0.02451228	-0.09206765	0.01913345	-0.1055826	0.02178116	-0.09116538	0.0189627

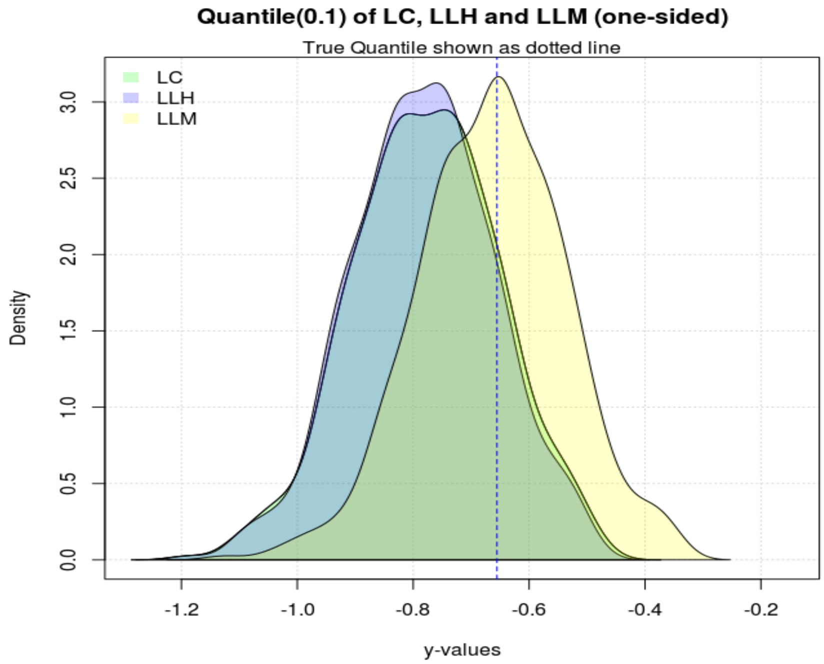

Figure 2.5: Estimated versus true quantile values ($\alpha = 0.1$) for 1-sided estimation, i.i.d. errors ($\tau = 0.5$)

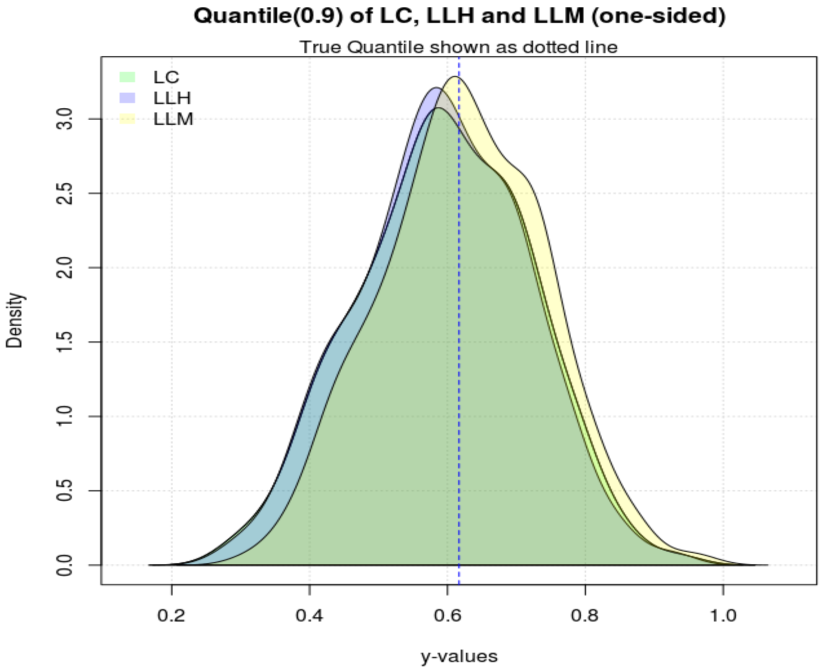

Figure 2.6: Estimated versus true quantile values ($\alpha = 0.9$) for 1-sided estimation, i.i.d. errors ($\tau = 0.5$)

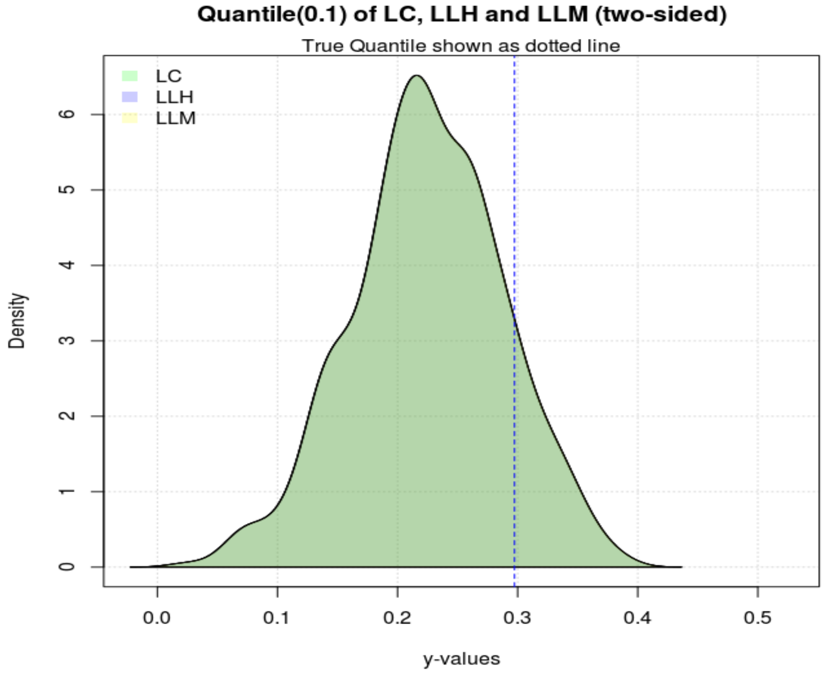

Figure 2.7: Estimated versus true quantile values ($\alpha = 0.1$) for 2-sided estimation, i.i.d. errors ($\tau = 0.5$)

2.3.2 Simulation: Additive model with heteroskedastic errors

Data Y_i for $i = 1, \ldots, 1001$ were simulated as per model (2.1) with $\mu(x_i) = \sin(2\pi x_i)$, $\sigma(x_i) = \tau x_i$ where $x_i = \frac{i}{n}$ and the errors ε_i as i.i.d. $\frac{1}{2}\chi^2_2 - 1$. Sample size n was set to 1001. A total of 500 such realizations were generated for this study.

Results for the mean-value of the Kolmogorov-Smirnov test statistic between the LC, LLH and LLM estimated distributions and empirical distribution calculated using available values of the simulated data are given in Tables 2.11, 2.12, 2.13 and 2.14 for boundary point $n = 1001$ and internal point $n = 200$ for values of $\tau = 0.1$ and 0.5 over a range of bandwidths, i.e., b taking values $3.7, 7.4, \ldots, 51.8$ in steps of 3.7.

Point prediction performance values are provided for the same cases in Tables 2.15, 2.16, 2.17 and 2.18.

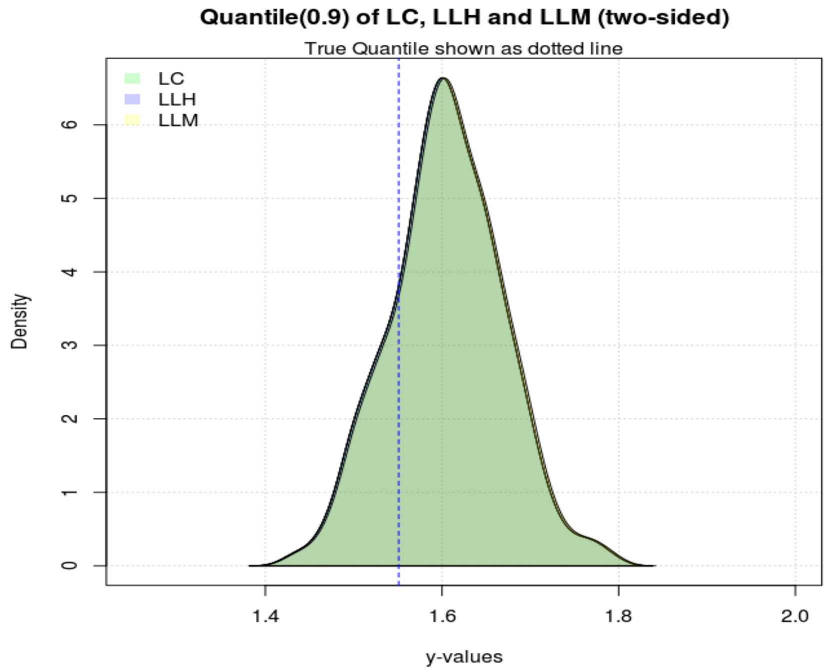

Figure 2.8: Estimated versus true quantile values ($\alpha = 0.9$) for 2-sided estimation, i.i.d. errors ($\tau = 0.5$)

Note that the point $n = 1001$ is excluded from the data used for LC, LLH and LLM estimation at the boundary point. Similarly the point $n = 200$ is excluded for the case of estimation at the internal point.

It can be seen from Tables 2.11 and 2.13 that in the case of boundary point estimation among the estimators based on $\bar{F}_x(y)$, $\bar{F}_x^{LLH}(y)$ and $\bar{F}_x^{LLM}(y)$ the lowest value of the Kolmogorov-Smirnov test statistic is obtained using the LLM estimator $\bar{F}_x^{LLM}(y)$. In addition again for the boundary case the lowest values of MSE are obtained using the LLM estimator as can be seen from Tables 2.15 and 2.17 which is consistent with the trend seen from minimum values of the KS test-statistic. Minimum values of KS-statistic and MSE are highlighted in bold in all respective tables. Based on these results it can be concluded that for boundary value estimation the estimator based on $\bar{F}_x^{LLM}(y)$ has superior performance as compared to both $\bar{F}_x(y)$ and $\bar{F}_x^{LLH}(y)$.

In the case of $\tau = 0.1$ at the boundary point $n = 1001$ the LC estimator outperforms the LLH, LLM and LL estimators only at very low bandwidths. As can be seen from Table 2.15 for the case of $\tau = 0.1$ with increasing bandwidth the 2 local linear estimators LLH and LLM perform significantly better than LC owing to lower bias. In this case over a large range of bandwidths from $11.1,\ldots,51.8$ LLH and LLM outperform LC. In the case of $\tau = 0.5$ at the boundary point $n = 1001$ as can be seen from Table 2.17, even though the bias of the local linear estimators are lower at higher bandwidths compared to that of the LC estimator the latter outperforms these for a larger range of bandwidths as compared to the case of $\tau = 0.1$ owing to larger estimation variance for the local linear estimators. In this case the LLM and LL estimators outperform the LC and LLH estimators only for the higher bandwidths $25.9,\ldots,51.8$. However as stated earlier the overall best performance is obtained from the LLM estimator.

For the case of estimation at internal points no appreciable differences in performance are noticeable between the 3 estimators using both the mean values of the Kolmogorov-Smirnov test statistic (Tables 2.12 and 2.14) and also using mean-square error of point prediction (Tables 2.16 and 2.18).

It can also be seen from Tables 2.15, 2.16, 2.17 and 2.18 that—across the range of bandwidths considered—there is negligible loss in best point prediction performance of LLM versus that of LL. This finding is unexpected since it has been widely believed that the LL method gives optimal point estimators and/or predictors. It appears that the monotonicity correction does not hurt the resulting point estimators/predictors which is encouraging.

Table 2.11: Mean values of KS test statistic over heteroskedastic errors at boundary point ($n = 1001, \tau = 0.1$)

Bandwidth	KS-LC	KS-LLH	KS-LLM
3.7	**0.361228**	0.3619	0.368288
7.4	0.39358	**0.3606**	0.336436
11.1	0.43216	0.371372	0.326076
14.8	0.470316	0.388952	**0.325116**
18.5	0.506436	0.408316	0.335152
22.2	0.53998	0.42548	0.350864
25.9	0.572256	0.44356	0.371324
29.6	0.599836	0.462808	0.393896
33.3	0.6269	0.47816	0.415468
37.0	0.651132	0.499376	0.44184
40.7	0.670604	0.516304	0.462756
44.4	0.69	0.529796	0.485004
48.1	0.706968	0.545344	0.505352
51.8	0.72394	0.562432	0.5257

2.3.3 Real-life example: Wage dataset

The `Wage` dataset from the `ISLR` package (James, Witten, Hastie, & Tibshirani, 2013) was selected as a real-life example to demonstrate the differences in estimated local densities estimated using the LC, LLH and LLM methods. The full dataset has 3000 points and has been constructed from the Current Population Survey (CPS) data for year 2011. Point Prediction is used as the criterion for demonstrating performance differences between the three distribution estimators. This dataset is an example of regression data distributed non-uniformly and hence the local linear estimator (LL) based on equations 3.17 and 2.4 is expected to give the best performance in such cases. However our study involves using point-prediction using the three distribution estimators $\bar{F}_x(y)$, $\bar{F}_x^{LLH}(y)$ or $\bar{F}_x^{LLM}(y)$. Among these 3 estimators LLM gives the best point prediction performance and we show that using this estimator causes negligible loss in performance compared to using LL.

Table 2.12: Mean values of KS test statistic over heteroskedastic errors at internal point ($n = 200, \tau = 0.1$)

Bandwidth	KS-LC	KS-LLH	KS-LLM
3.7	**0.459776**	**0.461528**	0.461176
7.4	0.461872	0.462716	**0.4603**
11.1	0.46576	0.467308	0.464956
14.8	0.468904	0.471824	0.470172
18.5	0.47436	0.475916	0.474864
22.2	0.482716	0.482476	0.47912
25.9	0.488952	0.488444	0.486656
29.6	0.495916	0.495736	0.495056
33.3	0.503672	0.503052	0.502708
37.0	0.5105	0.513116	0.51026
40.7	0.519052	0.518104	0.518928
44.4	0.528456	0.528444	0.527104
48.1	0.537336	0.536916	0.535632
51.8	0.545264	0.545496	0.543776

Table 2.13: Mean values of KS test statistic over heteroskedastic errors at boundary point ($n = 1001, \tau = 0.5$)

Bandwidth	KS-LC	KS-LLH	KS-LLM
3.7	0.208708	0.28022	0.323664
7.4	**0.176304**	0.210876	0.241228
11.1	0.178416	0.189656	0.206996
14.8	0.189136	0.17842	0.186628
18.5	0.204484	**0.175508**	0.173096
22.2	0.220652	0.177144	0.163916
25.9	0.240692	0.181092	0.158476
29.6	0.25784	0.186648	0.15736
33.3	0.277888	0.191396	**0.156008**
37.0	0.295264	0.20092	0.159028
40.7	0.312968	0.20922	0.163296
44.4	0.330008	0.216464	0.167872
48.1	0.345432	0.22344	0.17522
51.8	0.36082	0.234392	0.181376

Table 2.14: Mean values of KS test statistic over heteroskedastic errors at internal point ($n = 200, \tau = 0.5$)

Bandwidth	KS-LC	KS-LLH	KS-LLM
3.7	0.3289	0.329088	0.329112
7.4	**0.327172**	**0.326072**	**0.3268**
11.1	0.327236	0.32788	0.3275
14.8	0.331784	0.3309	0.33186
18.5	0.337856	0.337888	0.337692
22.2	0.343504	0.344328	0.343368
25.9	0.350048	0.351444	0.349592
29.6	0.3588	0.359188	0.358944
33.3	0.36826	0.368708	0.368008
37.0	0.378308	0.376472	0.377692
40.7	0.386636	0.3864	0.388256
44.4	0.39642	0.395744	0.39754
48.1	0.4055	0.408072	0.40714
51.8	0.418516	0.4171	0.41794

Table 2.15: Point Prediction for Boundary Value over heteroskedastic errors ($n = 1001, \tau = 0.1$)

Ban	Bias-LC	MSE-LC	Bias-LLH	MSE-LLH	Bias-LLM	MSE-LLM	Bias-LL	MSE-LL
3.7	-0.01646515	**0.0110308**	-0.008415339	0.01301928	0.003362834	0.01521503	0.002122231	0.01532911
7.4	-0.03418985	0.01113183	-0.01803682	0.01111592	0.001465109	0.01164382	0.003045892	0.01196587
11.1	-0.05291763	0.01251871	-0.02791687	**0.0110065**	-0.001493594	0.01066538	0.003162759	0.01102039
14.8	-0.07217657	0.01484132	-0.03844108	0.01144334	-0.007355843	0.01025364	0.003252626	0.01051494
18.5	-0.09186859	0.0180368	-0.0493472	0.01222871	-0.01604004	**0.01020275**	0.003291589	0.01020678
22.2	-0.1116673	0.02205503	-0.06052473	0.01337097	-0.0266163	0.0105145	0.003183081	0.01002049
25.9	-0.1312554	0.02681084	-0.07204081	0.01484635	-0.03845618	0.01120131	0.002843088	0.009915576
29.6	-0.1512692	0.03246252	-0.08373385	0.01662921	-0.05099656	0.01226805	0.002239256	0.009858149
33.3	-0.1714417	0.03896746	-0.09557852	0.01872622	-0.06394962	0.01372077	0.00136753	0.009824624
37.0	-0.1916003	0.04627765	-0.1075785	0.02114855	-0.07708492	0.01554708	0.0002256174	0.009802568
40.7	-0.2119687	0.05448537	-0.1197012	0.02389638	-0.09028337	0.01774215	-0.001196002	0.009787441
44.4	-0.2326798	0.06368262	-0.1320067	0.02699023	-0.1035047	0.0202921	-0.002912961	**0.009779257**
48.1	-0.2535364	0.07381161	-0.1444581	0.03043434	-0.1167033	0.02319127	-0.004943721	0.009780505
51.8	-0.2740579	0.08462823	-0.1570383	0.03422973	-0.1299138	0.02644559	-0.007307095	0.009795173

Table 2.16: Point Prediction for Internal Value over heteroskedastic errors ($n = 200, \tau = 0.1$)

Ban	Bias-LC	MSE-LC	Bias-LLH	MSE-LLH	Bias-LLM	MSE-LLM	Bias-LL	MSE-LL
3.7	-0.00078446	0.0004397085	-0.001282314	0.0004403816	-0.001281847	0.0004403461	-0.001460641	0.0004417506
7.4	-0.001122367	**0.0004306633**	-0.001431476	**0.0004311207**	-0.001431238	**0.0004311334**	-0.001977922	**0.0004335427**
11.1	-0.002288569	0.0004309951	-0.002426097	0.0004313394	-0.002424798	0.0004313337	-0.003182182	0.0004360195
14.8	-0.00405804	0.0004390668	-0.004017686	0.0004388123	-0.004015654	0.0004387818	-0.004960882	0.0004476175
18.5	-0.006300199	0.0004597561	-0.006090049	0.0004573097	-0.006086971	0.0004572689	-0.007269184	0.0004732545
22.2	-0.008952297	0.0004986956	-0.00857106	0.0004917471	-0.008566653	0.0004916759	-0.01008903	0.0005201175
25.9	-0.01195461	0.0005599192	-0.0114063	0.0005469568	-0.01140055	0.0005468245	-0.01341156	0.0005966537
29.6	-0.01524307	0.000648231	-0.01455151	0.0006275842	-0.01454456	0.0006273696	-0.01723004	0.000712577
33.3	-0.0188042	0.0007686332	-0.0179713	0.0007381116	-0.01796344	0.0007378019	-0.02153766	0.0008788525
37.0	-0.02260511	0.0009254938	-0.02163909	0.0008829351	-0.02163065	0.000882528	-0.02632699	0.00110763
40.7	-0.02662906	0.001123084	-0.02553604	0.001066478	-0.02552742	0.001065985	-0.03158974	0.001412141
44.4	-0.03085926	0.001365925	-0.02964955	0.001293297	-0.02964117	0.00129274	-0.03731584	0.001806523
48.1	-0.03531386	0.001660546	-0.03397167	0.001568158	-0.03396393	0.00156757	-0.0434914	0.002305438
51.8	-0.03995794	0.002010171	-0.0384976	0.001896071	-0.03849081	0.00189549	-0.05009551	0.002923419

Table 2.17: Point Prediction for Boundary Value over heteroskedastic errors ($n = 1001, \tau = 0.5$)

Ban	Bias-LC	MSE-LC	Bias-LLH	MSE-LLH	Bias-LLM	MSE-LLM	Bias-LL	MSE-LL
3.7	-0.01641585	0.273216	-0.01371259	0.3278422	0.01500573	0.3662851	0.01063269	0.3832281
7.4	-0.02085331	0.2520507	-0.0253276	0.274159	0.002731055	0.2896534	0.01538866	0.2991516
11.1	-0.02981426	0.2462187	-0.03060796	0.2589025	0.003270715	0.2685365	0.0163369	0.2755266
14.8	-0.04068759	**0.2442488**	-0.03742699	0.2526514	0.002433103	0.2586642	0.01748551	0.2629147
18.5	-0.05443176	0.2442541	-0.04586018	0.2488821	0.0005299281	0.2526573	0.01882287	0.2552529
22.2	-0.06977487	0.245767	-0.05475483	0.2474683	-0.001728694	0.248843	0.0199724	0.2506579
25.9	-0.08589639	0.2481108	-0.06470145	**0.2471712**	-0.005360827	0.2463975	0.02061821	0.2481124
29.6	-0.1036357	0.25121	-0.07550857	0.2474051	-0.01066184	0.2448518	0.02070028	0.2467569
33.3	-0.1221155	0.2551902	-0.08684923	0.2482818	-0.01739231	0.2440367	0.02029725	0.2459808
37.0	-0.1410488	0.2599296	-0.09877418	0.2499431	-0.02522804	**0.2438554**	0.01949336	0.2454429
40.7	-0.1599352	0.2653016	-0.1111362	0.252298	-0.03400529	0.2440469	0.0183332	0.2449864
44.4	-0.1798873	0.2718873	-0.1241105	0.2552008	-0.04372748	0.2444396	0.01682687	0.2445524
48.1	-0.2001088	0.2793124	-0.1376482	0.2586551	-0.05435831	0.2450597	0.01496614	0.2441256
51.8	-0.2196351	0.2872558	-0.1514669	0.2625969	-0.06555938	0.2460242	0.01273652	**0.2437067**

Table 2.18: Point Prediction for Internal Value over heteroskedastic errors ($n = 200, \tau = 0.5$)

Ban	Bias-LC	MSE-LC	Bias-LLH	MSE-LLH	Bias-LLM	MSE-LLM	Bias-LL	MSE-LL
3.7	-0.005989017	0.01091506	-0.009105295	0.01090718	-0.009100397	0.01090687	-0.006151798	0.01102828
7.4	-0.004317512	0.01067232	-0.006852094	0.01066366	-0.006845515	0.01066378	-0.005549238	0.01077156
11.1	-0.004591794	0.01059617	-0.006678386	0.0105944	-0.006665835	0.01059435	-0.006333703	0.01068745
14.8	-0.005937551	0.01054613	-0.007674744	0.01055486	-0.007656429	0.01055456	-0.00795147	0.01063841
18.5	-0.007974463	**0.01051967**	-0.009442124	**0.01053534**	-0.009416436	**0.01053501**	-0.01019012	**0.01061418**
22.2	-0.01058953	0.01053145	-0.01180495	0.01054554	-0.01176999	0.01054509	-0.01297439	0.01062656
25.9	-0.01373489	0.01057533	-0.01467215	0.01059193	-0.01462675	0.01059104	-0.01627809	0.01068457
29.6	-0.01725266	0.01066128	-0.01798338	0.01067947	-0.01792693	0.01067781	-0.02008891	0.01079614
33.3	-0.02118215	0.0107964	-0.02169107	0.01081295	-0.02162338	0.01081019	-0.0243973	0.01096978
37.0	-0.02546816	0.01098609	-0.02575577	0.01099723	-0.02567729	0.01099311	-0.0291937	0.01121525
40.7	-0.03007643	0.01123397	-0.0301445	0.01123745	-0.03005627	0.01123178	-0.03446792	0.01154379
44.4	-0.03496193	0.01154587	-0.03483024	0.01153901	-0.03473374	0.01153169	-0.04020819	0.01196797
48.1	-0.04015664	0.01193249	-0.03979071	0.01190758	-0.03968792	0.01189862	-0.04639914	0.0125013
51.8	-0.04561132	0.01240015	-0.04500812	0.01234911	-0.04490124	0.01233864	-0.05301874	0.01315745

Table 2.19: Point Prediction for ISLR Wage Dataset

Method	Bias	MSE
LC	-0.01800635	0.08682027
LLH	-0.0149775	0.08441505
LLM	-0.0001609517	0.08201118
LL	0.001865924	0.0825116

From the plot of the dataset in Figure 2.9 with superimposed smoother (obtained using `loess` fitting from the R package `lattice`) it can be noted that the regression function is sloping upwards at the left boundary whereas it flattens out at the right boundary. Hence, at the right boundary, local constant methods suffice and should be practically equivalent to local linear methods. The left boundary is more interesting, and this is where our numerical work will focus. To carry this out, we created a second version of the data where logwage is tabulated versus decreasing age and performed point prediction over the last 231 values of this backward dataset, i.e., the first 231 values of the original. Since this is a regression dataset with non-uniformly distributed design points we determine bandwidths for LC, LLH and LLM using the 2-sided predictive cross-validation procedure outlined in Section 2.2.5. We predict the value of logwage at i and compare it with the known value at that point where $i = 2770, \ldots, 3000$ to determine the MSE of point prediction.

Point prediction results for all three methods over data points $2770, \ldots, 3000$ (logwage versus decreasing age) are given in Table 2.19. It can be seen from this table that LLM has the best point prediction performance and this closely matches that of LL. As in the case of simulated data, this is an unexpected and encouraging result indicating that the LLM distribution may be an all-around favorable estimator both in terms of its quantiles as well as its center of location used for point estimation and prediction purposes.

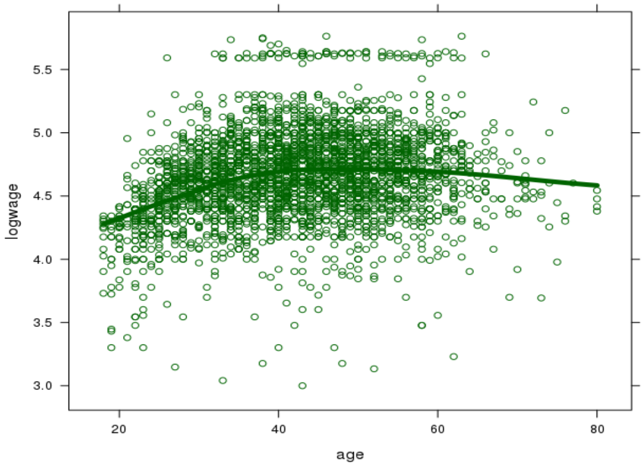

Figure 2.9: Plot of logwage versus age from Wage dataset (ISLR package)

2.4 Conclusions

Improved estimation of conditional distributions at boundary points is possible via local linear smoothing and other methods that, however, do not guarantee that the resulting estimator is a proper distribution function. In this Chapter we propose a simple monotonicity correction procedure that is immediately applicable, easy to implement, and performs well with simulated and real data.

To elaborate, it has been shown using boundary points on simulated datasets that the LLM distribution estimator outperforms that of LLH and LC as seen by the values of the Kolmogorov-Smirnov test statistic, accuracy of estimated quantiles, and also by its performance in point prediction—the latter finding being entirely unexpected. In contrast, for internal points on these datasets there seem to be no significant differences between the 3 estimators using these performance metrics.

In addition, among all three methods over a wide range of selected bandwidths the overall best performance is obtained using Monotone Local Linear Estimation. As can be seen from the point prediction tables, the predictor based on $\bar{F}_x^{LLM}(y)$ has lower bias compared to $\bar{F}_x(y)$ and $\bar{F}_x^{LLH}(y)$; this is consistent with the discussion in Section 2.2, i.e. that $\bar{F}_x^{LLM}(y)$ has improved performance because of reduced bias in extrapolation for the boundary case. No such differences in bias are noticed for the case of internal points.

As in the case of simulated data, in the real data example as well the point prediction performance of LLM closely matches in performance to that of LL which implies that the LLM distribution estimator can be used for all practical applications, including point prediction.

Acknowldgements: Chapter 2 is based on the paper ”Nonparametric estimation of the conditional distribution at regression boundary points”, by S. Das and D. N. Politis and has been submitted for publication. The dissertation author was the primary investigator and author of this paper.

Chapter 3

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

3.1 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.,

$$Y_t = \mu(t) + W_t \tag{3.1}$$

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

$$Y_t = \mu(t) + \sigma(t)W_t. \tag{3.2}$$

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 (3.2), it is also assumed that $\text{Var}(W_t) = 1$. 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 3.1.1 (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

$$\mu(t) = \mu_{[0,1]}(a_t) \;\text{ and }\; \sigma(t) = \sigma_{[0,1]}(a_t) \tag{3.3}$$

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 3.2.2 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 (3.1) and (3.2) 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 (3.2), $EY_t = \mu(t)$ and $\mathrm{Var}\,Y_t = \sigma^2(t)$; hence,

$$W_t = \frac{Y_t - \mu(t)}{\sigma(t)} \tag{3.4}$$

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 $\mathrm{Corr}(Y_t, Y_{t+1})$ changes smoothly (and slowly) with t. Notably, models (3.1) and (3.2) 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 (3.1) and (3.2). 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 $H_n : \underline{Y}_n \mapsto \underline{\varepsilon}_n$ where $\underline{\varepsilon}_n = (\varepsilon_1, \ldots, \varepsilon_n)'$ is a random vector with i.i.d. components; the details are given in Section 3.3. 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 3.2 and Model-free of Section 3.3, are novel, and they are empirically compared to each other in Section 3.5 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 3.6 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 3.1 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 3.6 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 3.4 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.

3.2 Model-based inference

Throughout Section 3.2, we will assume model (3.2)—that includes model (3.1) as a special case—together with a nonparametric assumption on smoothness of $\mu(\cdot)$ and $\sigma(\cdot)$

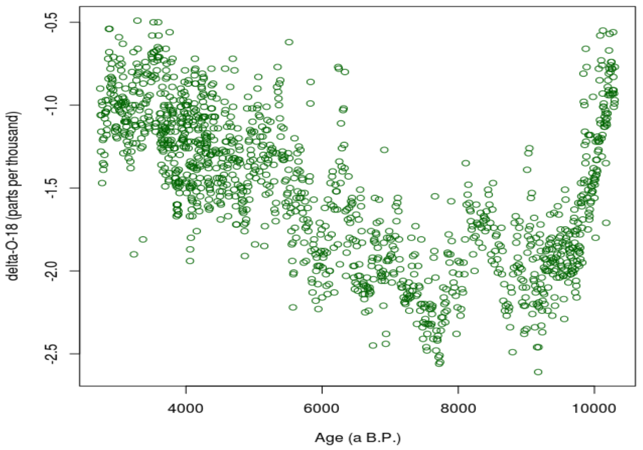

Figure 3.1: Oxygen Isotope Record from stalagmite Q5 from southern Oman (1345 samples)

as described in Remark 3.1.1.

3.2.1 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 (3.2), we have

$$E(Y_{n+1}|\underline{Y}_n) = \mu(n+1) + \sigma(n+1)E(W_{n+1}|\underline{Y}_n). \tag{3.5}$$

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.,

$$E(W_{n+1}|\underline{Y}_n) \simeq E(W_{n+1}|Y_s, s \leq n) = E(W_{n+1}|W_s, s \leq n) \simeq E(W_{n+1}|\underline{W}_n) \qquad (3.6)$$

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

$$\bar{E}(W_{n+1}|\underline{W}_n) = \phi_1(n)W_n + \phi_2(n)W_{n-1} + \ldots + \phi_n(n)W_1, \qquad (3.7)$$

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:

$$W_t = \phi_1 W_{t-1} + \phi_2 W_{t-2} + \cdots + \phi_p W_{t-p} + V_t \qquad (3.8)$$

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

$$\bar{E}(W_{n+1}|\underline{W}_n) = \phi_1 W_n + \phi_2 W_{n-1} + \cdots + \phi_p W_{n-p+1}. \tag{3.9}$$

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 (3.7) coincides with the above AR–type predictor if the matrix Γ_n is the one implied by the fitted AR(p) model (3.8). We will use the AR–type predictor in the sequel because it additionally affords us the possibility of resampling based on model (3.8).

3.2.2 Trend estimation and practical prediction

To construct the L_2–optimal predictor (3.5), 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 3.1.1 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 3.2.1 (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 (3.5), 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. (3.7) 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. (3.4) 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

$$b \to \infty \text{ as } n \to \infty \text{ but } b/n \to 0, \tag{3.10}$$

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

$$\hat{\mu}(t) = \sum_{i=1}^{t} Y_i \, \hat{K}\left(\frac{t-t_i}{b}\right) \text{ and } \hat{M}(t) = \sum_{i=1}^{t} Y_i^2 \, \hat{K}(\frac{t-t_i}{b}) \tag{3.11}$$

where

$$\hat{\sigma}(t) = \sqrt{\hat{M}_t - \hat{\mu}(t)^2} \text{ and } \hat{K}\left(\frac{t-t_i}{b}\right) = \frac{K(\frac{t-t_i}{b})}{\sum_{k=1}^{t} K(\frac{t-t_k}{b})}. \tag{3.12}$$

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

$$\hat{W}_t = \frac{Y_t - \hat{\mu}(t)}{\hat{\sigma}(t)} \text{ for } t = b+1, \ldots, n. \tag{3.13}$$

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

$$\tilde{\mu}(t) = \sum_{i=1}^{t-1} Y_i \, \tilde{K}\left(\frac{t-t_i}{b}\right) \text{ and } \tilde{M}(t) = \sum_{i=1}^{t-1} Y_i^2 \, \tilde{K}(\frac{t-t_i}{b}) \tag{3.14}$$

where

$$\tilde{\sigma}(t) = \sqrt{\tilde{M}_t - \tilde{\mu}(t)^2} \text{ and } \tilde{K}\left(\frac{t-t_i}{b}\right) = \frac{K(\frac{t-t_i}{b})}{\sum_{k=1}^{t-1} K(\frac{t-t_k}{b})}. \tag{3.15}$$

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

$$\tilde{W}_t = \frac{Y_t - \tilde{\mu}(t)}{\tilde{\sigma}(t)} \text{ for } t = b+1, \ldots, n. \tag{3.16}$$

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

$$\hat{\mu}(t) = \frac{\sum_{j=1}^{t} w_j Y_j}{\sum_{j=1}^{t} w_j + n^{-2}} \text{ and } \hat{M}(t) = \frac{\sum_{j=1}^{t} w_j Y_j^2}{\sum_{j=1}^{t} w_j + n^{-2}} \tag{3.17}$$

where

$$w_j = K(\frac{t-t_j}{b})\left[s_{t,2} - (t-t_j)s_{t,1}\right], \tag{3.18}$$

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. (3.17) is just to ensure the denominator is not zero; see Fan (1993). Eq. (3.12) then yields $\hat{\sigma}(t)$, and eq. (3.13) yields $\hat{W}_t$.

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

$$\tilde{\mu}(t) = \frac{\sum_{j=1}^{t-1} w_j Y_j}{\sum_{j=1}^{t-1} w_j + n^{-2}} \text{ and } \tilde{M}(t) = \frac{\sum_{j=1}^{t-1} w_j Y_j^2}{\sum_{j=1}^{t-1} w_j + n^{-2}} \tag{3.19}$$

where

$$w_j = K(\frac{t-t_j}{b})\left[s_{t-1,2} - (t-t_j)s_{t-1,1}\right]. \tag{3.20}$$

Eq. (3.15) then yields $\tilde{\sigma}(t)$, and eq. (3.16) 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 (3.5). 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 3.2.1.

Remark 3.2.2 (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 3.2.3 (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.

3.2.3 Model-based prediction intervals

To go from point prediction to prediction intervals, some form of resampling is required. Since model (3.2) 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 (3.2). 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. (3.8) as if they were i.i.d., performing an i.i.d. bootstrap on them, and then generating $\{W_t^*\}$ via the recursion (3.8) 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 3.2.1, 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. (3.8) 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

$$\Pi = \check{\mu}(n+1) + \check{\sigma}(n+1)\left[\hat{\phi}_1\check{W}_n + \cdots + \hat{\phi}_p\check{W}_{n-p+1}\right] \tag{3.21}$$

where $\hat{\phi}_1, \ldots, \hat{\phi}_p$ are the Yule-Walker estimators of $\phi_1, \ldots, \phi_p$ appearing in eq. (3.8).

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 3.2.1 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* (3.2).

2. *Fit the AR(p) model* (3.8) *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*

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

   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*

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

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

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

(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* $\hat{\phi}_1^*, \ldots, \hat{\phi}_p^*$.

(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*

[1] The bootstrap estimators $\check{\mu}^*(\cdot)$ and $\check{\sigma}^*(\cdot)$ are based on bandwidth b' determined by Algorithm A.0.3 given in Appendix A. This may be different from the bandwidth b found using model-based cross-validation.

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*

$$[\Pi + q(\alpha/2),\ \Pi + q(1-\alpha/2)]. \tag{3.22}$$

It is easy to see that prediction interval (3.22) 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 (3.10). 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 (3.22) must be gauged by finite-sample simulations. Results of these simulations are shown in Section 3.5.

3.3 Model-free inference

Model (3.2) 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 $H_n : \underline{Y}_n \mapsto \underline{\varepsilon}_n$ where $\underline{\varepsilon}_n = (\varepsilon_1, \ldots, \varepsilon_n)'$ 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 (3.2) 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 3.3.1 (Quantifying smoothness–model-free case) As in Remark 3.1.1, 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

$$D_t^{(m)}(\underline{s}) = D_{a_t}^{[0,1]}(\underline{s}) \text{ for } t = 1, \ldots, n. \tag{3.23}$$

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 3.1.1, 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,

$$Y_t = \mathbf{f}_t(W_t, W_{t-1}, \ldots, W_{t-m+1}) \tag{3.24}$$

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. (3.24) 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. (3.24) can be used to describe a rather general class of locally stationary processes. Note that model (3.2) is a special case of Eq. (3.24) 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. (3.2), we will focus in the sequel on the case $m = 1$. Section 3.3.10 discusses how to handle the case $m > 1$.

3.3.1 Constructing the theoretical transformation

Hereafter, adopt the setup of Eq. (3.24) with $m = 1$, and let

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

denote the 1st marginal distribution of time series $\{Y_t\}$. Throughout Section 3.3, 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 3.3.8.

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

$$U_t = D_t(Y_t) \text{ for } t = 1, \ldots, n; \tag{3.25}$$

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

$$Z_t = \Phi^{-1}(U_t) \text{ for } t = 1, \ldots, n; \tag{3.26}$$

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:

$$\underline{\varepsilon}_n = C_n^{-1} \underline{Z}_n. \tag{3.27}$$

[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.

It then follows that the entries of $\underline{\varepsilon}_n = (\varepsilon_1, \ldots, \varepsilon_n)'$ are uncorrelated standard normal. Assuming that the random variables $Z_1, \ldots, Z_n$ were *jointly* normal, this can be strenghtened to claim that $\varepsilon_1, \ldots, \varepsilon_n$ are i.i.d. $N(0,1)$; see Section 3.3.10 for further discussion. Consequently, the transformation of the dataset $\underline{Y}_n = (Y_1, \ldots, Y_n)'$ to the vector $\underline{\varepsilon}_n$ 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. (3.25), (3.26) and (3.27), are invertible; hence, the composite transformation $H_n : \underline{Y}_n \mapsto \underline{\varepsilon}_n$ is invertible as well.

3.3.2 Kernel estimation of the ‘uniformizing’ transformation

We first focus on estimating the ‘uniformizing’ part of the transformation, i.e., eq. (3.25). 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:

$$\hat{D}_t(y) = \sum_{i=1}^{T} \mathbf{1}\{Y_{t_i} \leq y\} \tilde{K}(\frac{t-t_i}{b}) \tag{3.28}$$

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 (3.28) is *one-sided* for the same reasons discussed in Remark 3.2.1. Since $\hat{D}_t(y)$ is a step function in y, a smooth estimator can be defined as:

$$\bar{D}_t(y) = \sum_{i=1}^{T} \Lambda(\frac{y-Y_{t_i}}{h_0}) \tilde{K}(\frac{t-t_i}{b}) \tag{3.29}$$

where h_0 is a secondary bandwidth. Furthermore, as in Section 3.2.2, 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 3.3.5.

3.3.3 Local linear estimation of the 'uniformizing' transformation

Note that the kernel estimator $\hat{D}_t(y)$ defined in eq. (3.28) is just the Nadaraya-Watson smoother, i.e., local average, of the variables $u_1,\ldots,u_n$ where $u_i = \mathbf{1}\{Y_i \le y\}$. Similarly, $\bar{D}_t(y)$ defined in eq. (3.29) 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 $\mathbf{1}\{Y_i \le y\}$ 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 et al., 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:

$$\hat{D}_t^{LLH}(y) = \frac{\sum_{i=1}^{T} w_i^{\diamond} \mathbf{1}(Y_i \le y)}{\sum_{i=1}^{T} w_i^{\diamond}} \text{ and } \bar{D}_t^{LLH}(y) = \frac{\sum_{i=1}^{T} w_i^{\diamond} \Lambda(\frac{y-Y_i}{h_0})}{\sum_{i=1}^{T} w_i^{\diamond}}. \tag{3.30}$$

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

$$w_i^{\diamond} = \begin{cases} 0 & \text{when } \hat{\beta}(t-t_i) > 1 \\ w_i(1-\hat{\beta}(t-t_i)) & \text{when } \hat{\beta}(t-t_i) \le 1 \end{cases} \tag{3.31}$$

where

$$w_i = \frac{1}{b} K(\frac{t-t_i}{b}) \text{ and } \hat{\beta} = \frac{\sum_{i=1}^{T} w_i(t-t_i)}{\sum_{i=1}^{T} w_i(t-t_i)^2}. \tag{3.32}$$

As with eq. (3.28)and (3.29), 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 $\bar{D}_t^{LLH}(y)$.

3.3.4 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 4.4.1 given below.

Algorithm 3.3.1 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*

$$\bar{d}_t^{LLM}(y) = \frac{\bar{d}_t^{LL+}(y)}{\int_{-\infty}^{\infty} \bar{d}_t^{LL+}(s)ds}. \tag{3.33}$$

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. (3.28)and (3.29), 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 4.4.1 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.

3.3.5 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 3.3.2 *BANDWIDTH DETERMINATION FOR MODEL-FREE INFERENCE*

1. *Perform the uniformizing transform described in* (3.25) *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 3.2.2. For this*

purpose let $k < n$, and suppose only subseries $Y_1, \dots, Y_k$ has been observed. Denote $\hat{Y}_{k+1}$ the best predictor of Y_{k+1} based on the data $Y_1, \dots, Y_k$ 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.

3.3.6 Estimation of the whitening transformation

To implement the whitening transformation (3.27), it is necessary to estimate Γ_n, i.e., the $n \times n$ covariance matrix of the random vector $\underline{Z}_n = (Z_1, \dots, Z_n)'$ where the Z_t are the normal random variables defined in eq. (3.26).

As discussed in the analogous model-based problem in Section 3.2.1, 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 $H_n : \underline{Y}_n \mapsto \underline{\varepsilon}_n$. 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 $\hat{\Gamma}^{AR}_{n+1}$ be the $(n+1)\times(n+1)$ covariance matrix associated with the fitted AR(p) model.

B$'$. Denote by $\hat{\gamma}^{\star}_{|i-j|}$ the i,j element of $\hat{\Gamma}^{\star}_{n}$ for $i,j=1,\ldots,n$. Then, define $\hat{\Gamma}^{\star}_{n+1}$ to be the symmetric, banded Toeplitz $(n+1)\times(n+1)$ matrix with ij element given by $\hat{\gamma}^{\star}_{|i-j|}$ when $|i-j|<n$. Recall that $\hat{\Gamma}^{\star}_{n}$ 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 $\hat{\Gamma}^{\star}_{n+1}$ 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 $\underline{\varepsilon}_{n+1}=(\varepsilon_1,\ldots,\varepsilon_n,\varepsilon_{n+1})'$ 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 $H^{-1}_{n+1}:\underline{\varepsilon}_{n+1}\mapsto\underline{Y}_{n+1}$. Recall that $\underline{\varepsilon}_n$ 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 $\underline{Y}_n=H_n^{-1}(\varepsilon_n)$. Hence, we just need to show how to create the unobserved Y_{n+1} from $\underline{\varepsilon}_{n+1}$; this is done in the following three steps.

Algorithm 3.3.3 *GENERATION OF UNOBSERVED DATAPOINT FROM FUTURE INNOVATIONS*

i. Let

$$\underline{Z}_{n+1}=C_{n+1}\underline{\varepsilon}_{n+1} \tag{3.34}$$

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

$$Z_{n+1}=\underline{c}_{n+1}\underline{\varepsilon}_{n+1} \tag{3.35}$$

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*

$$U_{n+1} = \Phi(Z_{n+1}). \tag{3.36}$$

iii. *Finally, define*

$$Y_{n+1} = D_{n+1}^{-1}(U_{n+1}); \tag{3.37}$$

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) $\underline{\varepsilon}_{n+1}$, the above completes the construction of $\underline{Y}_{n+1}$ based on $\underline{\varepsilon}_{n+1}$, i.e., the mapping $H_{n+1}^{-1} : \underline{\varepsilon}_{n+1} \mapsto \underline{Y}_{n+1}$.

3.3.7 Model-free predictors and prediction intervals

In the previous sections, it was shown how the construct the transformation $H_n : \underline{Y}_n \mapsto \underline{\varepsilon}_n$ and its inverse $H_{n+1}^{-1} : \underline{\varepsilon}_{n+1} \mapsto \underline{Y}_{n+1}$, where the random variables $\varepsilon_1, \varepsilon_2, \ldots,$ are i.i.d. Note that by combining eq. (3.35), (3.36) and (3.37) we can write the formula:

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

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

$$Y_{n+1} = g_{n+1}(\varepsilon_{n+1}) \text{ where } g_{n+1}(x) = D_{n+1}^{-1}\left(\Phi\left(\sum_{i=1}^{n} c_i\varepsilon_i + c_{n+1}x\right)\right). \tag{3.38}$$

Eq. (3.38) 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 3.3.4 MODEL-FREE (MF) PREDICTORS AND PREDICTION INTERVALS FOR Y_{n+1}

1. *Construct $U_1,\ldots,U_n$ by eq. (3.25) 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. (3.26), and use the methods of Section 3.3.6 to estimate Γ_n by either $\hat{\Gamma}_n^{AR}$ or $\hat{\Gamma}_n^{\star}$.*

3. *Construct $\varepsilon_1,\ldots,\varepsilon_n$ by eq. (3.27), and let $\hat{F}_n$ denote their empirical distribution.*

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

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

   where the function g_{n+1} is defined in the predictive equation (3.38) *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 $\{g_{n+1}(\varepsilon_i)$ 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.0.1 based on either the L_2– or L_1–optimal point predictor.*

Algorithm 3.3.4 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 3.3.5 PREDICTIVE MODEL-FREE (PMF) PREDICTORS AND PREDICTION INTERVALS FOR Y_{n+1}
The algorithm is identical to Algorithm 3.3.5 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 3.3.2 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 $\varepsilon_1, \ldots, \varepsilon_n$ 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 $\varepsilon_1, \ldots, \varepsilon_n$ 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 3.3.5 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 $H_{n+1}^{-1} : \underline{\varepsilon}_{n+1} \mapsto \underline{Y}_{n+1}$; the forward transformation $H_n : \underline{Y}_n \mapsto \underline{\varepsilon}_n$ 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 $D^{-1}(\beta) = \inf\{y$ such that $D(y) \geq \beta\}$.

Algorithm 3.3.6 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*

$$\hat{Y}_{n+1} = \int g_{n+1}(x) d\Phi(x) \qquad (3.39)$$

where the function g_{n+1} is defined in the predictive equation (3.38) *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* (3.39) *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 Boootstrap of Algorithm A.0.2 based on either the L_2– or L_1–optimal point predictor.*

Remark 3.3.3 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 3.3.4. 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. (3.36) and (3.37) 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

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

$$\simeq D_{n+1}^{-1}\left(\Phi\left(Median\left(Z_{n+1}|\mathcal{F}_1^n(Z)\right)\right)\right) = D_{n+1}^{-1}\left(\Phi\left(E\left(Z_{n+1}|\mathcal{F}_1^n(Z)\right)\right)\right), \qquad (3.40)$$

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. (3.7), 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. (3.40), and also employing consistent estimates of Γ_n and $\gamma(n)$ completes the calculation. As discussed in Section 3.3.6, 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 3.3.4 (Robustness of LMF approach) The LMF approach focuses completely on the predictive equation (3.38) 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

3.3.6. 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 3.3.6 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.

3.3.8 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 3.3.8 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 3.3.6 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 3.3.7 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 mode of this distribution is the LMF optimal predictor of Y_{n+1} under 0-1 loss.*

3.3.9 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 3.3.6—as well as the discrete data Algorithm 3.3.7—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 3.3.4 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 3.3.5, the optimal rate is $h_0 \sim n^{-2/5}$ when the estimand $D_t(y)$ is sufficiently smooth in y.

3.3.10 Local stationarity in a higher-dimensional marginal

The success of the theoretical transformation of Section 3.3.1 in transforming the data vector $\underline{Y}_n$ to the vector of i.i.d. components $\underline{\varepsilon_n}$ 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 $\text{Corr}(Y_t, Y_{t+m})$ 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 3.3.1.

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

[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 $V_m = D_m(Y_m|Y_{m-1}, \ldots, Y_2, Y_1)$.

whitening as required by the Model-Free Principle. Thus, when the time series $\{Y_t\}$ is locally stationary in its *m*th 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 $\text{Cov}(Z_t, Z_{t+k})$ 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 *m*th 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.

3.4 Diagnostics for Model-Free Inference

The steps outlined in Section 3.3.1 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.

3.4.1 QQ-plots after uniformization

The success of the uniformization step outlined in Section 3.3.1 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.

3.4.2 Shapiro-Wilk test for joint normality

The random vector $\underline{Z}_n = (Z_1, \ldots, Z_n)'$ from Section 3.3.6 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 3.2 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.

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

Provided that the inputs are jointly normal the whitening transformation described in Section 3.3.6 produces i.i.d. standard normal variables. The covariance matrix used in this 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 $N[0,1]$.

3.4.4 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 3.3 where it can be noticed from the ACF plot that the Model-Free transformations were successful in

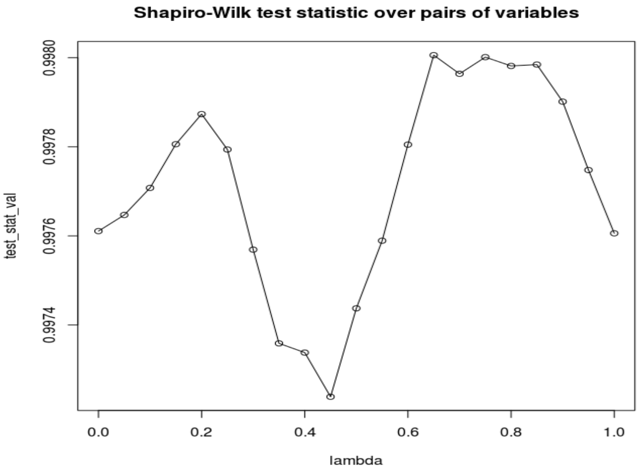

Figure 3.2: Values of Shapiro-Wilk test statistic for joint normality test. Note that corresponding p-values range from 0.09 to 0.29.

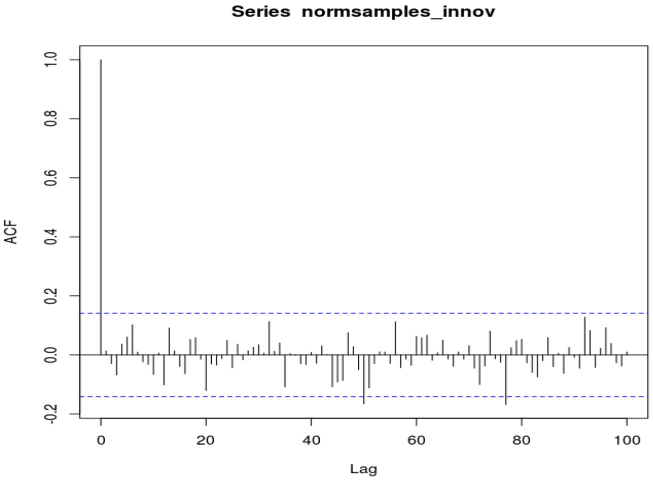

Figure 3.3: Autocorrelation plot showing decorrelation/independence of data after whitening transformation

producing decorrelated and therefore i.i.d. (normal) data.

3.5 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 and MB-LL are described in Section 3.2. 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 3.3. The covariance estimators using the flat-top tapered kernel and fitting an AR(p) model are discussed in Section 3.3.6. 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%.

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

Data Y_i for $t = 1,\ldots,1000$ were simulated as per model (3.1) with trend as in eq. (3.3), 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 $\tau = 0.16$. 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 3.2.2 and 3.3.5 respectively.

Results for point prediction including bias and mean square error (MSE) over all MB and MF methods are shown in Table 3.1 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 3.2 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 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).

3.5.2 Simulation: Additive model with nonlinearly generated errors

Data Y_i for $t = 1,\dots,1000$ were simulated from model (3.1) with trend as in eq. (3.3), 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:

Table 3.1: 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 3.2: 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

$$W_t = \begin{cases} 1+\alpha W_{t-1}+e_t & if\ W_{t-1} \le r \\ -1+\beta W_{t-1}+\gamma e_t & if\ W_{t-1} > r \end{cases} \tag{3.41}$$

where the errors e_t are assumed i.i.d. $N(0,\tau^2)$. Eq. (3.41) 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 3.4. The process of eq. (3.41) 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 3.2.2 and 3.3.5 respectively.

Results for point prediction including bias and mean square error (MSE) over all MB and MF methods are shown in Table 3.3 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 3.4 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

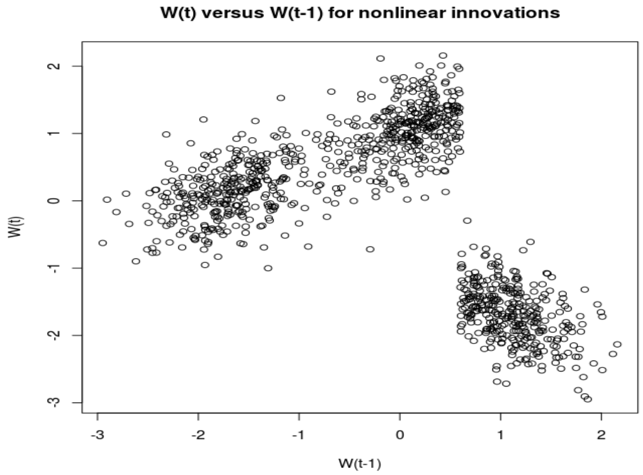

Figure 3.4: Nonlinear time series scatterplot of W_t versus W_{t-1}.

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 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).

Table 3.3: 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 3.4: 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

3.6 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 3.1 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 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 3.5 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 3.5 that depicts the age versus sample number. Hence we will assume even time spacing

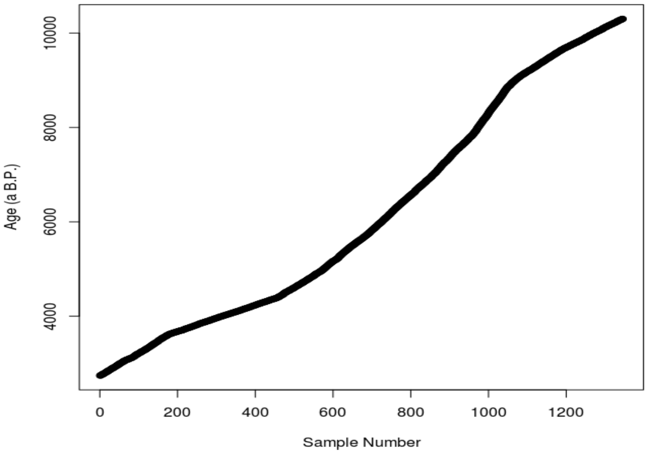

Figure 3.5: Age (a B.P.) of delta-O-18 versus sample number

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 3.6. 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 3.2.2 and 3.3.5 respectively; the last 62 points are used to calculate point prediction and prediction intervals. It can be noticed from Figures 3.1 and 3.6 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 3.3.2.

Results for point prediction including bias and mean square error (MSE) over all MB and MF methods are shown in Table 3.5 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 3.6 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 3.5. 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.

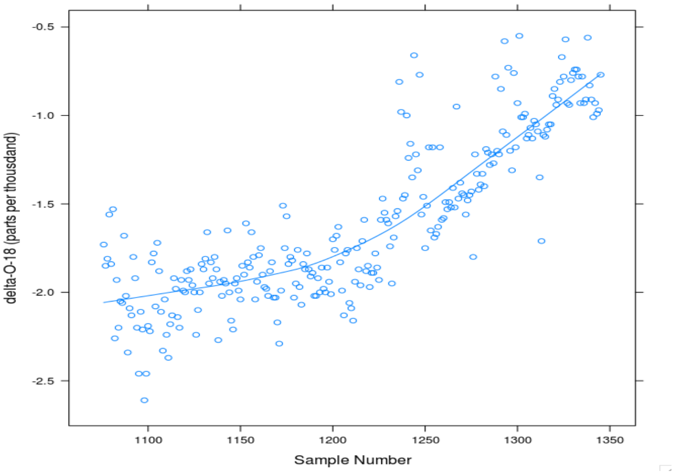

Figure 3.6: Speleothem data segment used for cross-validation and prediction

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 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.

Table 3.5: 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 3.6: 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

Acknowldgements: Chapter 3 is based on the paper ”Predictive inference for locally stationary time series with an application to climate data”, by S. Das and D. N. Politis and has been submitted for publication. The dissertation author was the primary investigator and author of this paper.

Chapter 4

Predictive inference for locally stationary random fields

4.1 Introduction

Consider a real-valued random field dataset $\{Y_{\underline{t}}, \underline{t} \in Z^2\}$ defined over a 2-D index-set D e.g. pixel values over an image or satellite data observed on an ocean surface. It may be unrealistic to assume that the stochastic structure of such a random field $Y_{\underline{t}}$ has stayed invariant over the entire region of definition D hence, we can not assume that $\{Y_{\underline{t}}\}$ 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 a previously unobserved data point $Y_{\underline{t}_k}$, i.e., constructing a point predictor for $Y_{\underline{t}_k}$. The usual approach for dealing with nonstationary series is to assume that the data can be decomposed as the sum of three components:

$$\mu(\underline{t}) + S_{\underline{t}} + W_{\underline{t}}$$

where $\mu(\underline{t})$ is a deterministic trend function, $S_{\underline{t}}$ is a seasonal (periodic) series, and $\{W_{\underline{t}}\}$ is (strictly) stationary with mean zero; this is the 'classical' decomposition of a time series to trend, seasonal and stationary components which can also be used for decomposition of random field data. 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.,

$$Y_{\underline{t}} = \mu(\underline{t}) + W_{\underline{t}} \tag{4.1}$$

which can be generalized to accommodate a coordinate-changing variance as well, i.e.,

$$Y_{\underline{t}} = \mu(\underline{t}) + \sigma(\underline{t}) W_{\underline{t}}. \tag{4.2}$$

In both above models, the series $\{W_{\underline{t}}\}$ is assumed to be (strictly) stationary, weakly dependent, e.g. strong mixing, and satisfying $EW_{\underline{t}} = 0$; in model (4.2), it is also assumed that $\mathrm{Var}(W_{\underline{t}}) = 1$. 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 $\mu(\underline{t})$ and $\sigma(\underline{t})$ change smoothly (and slowly) with $\underline{t}$.

As far as capturing the first two moments of $Y_{\underline{t}}$, models (4.1) and (4.2) are considered general and flexible—especially when $\mu(\cdot)$ and $\sigma(\cdot)$ are not parametrically specified—and have been studied extensively in the case of time series; see e.g. (Zhou & Wu, 2009), (Zhou & Wu, 2010). However, it may be that the skewness and/or kurtosis of $Y_{\underline{t}}$ changes with $\underline{t}$, in which case centering and studentization alone can not render the problem stationary. To see why, note that under model (4.2), $EY_{\underline{t}} = \mu(\underline{t})$ and $\text{Var}\, Y_{\underline{t}} = \sigma^2(\underline{t})$; hence,

$$W_{\underline{t}} = \frac{Y_{\underline{t}} - \mu(\underline{t})}{\sigma(\underline{t})} \tag{4.3}$$

cannot be (strictly) stationary unless the skewness and kurtosis of $Y_{\underline{t}}$ 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 $\text{Corr}(Y_{\underline{t}_j}, Y_{\underline{t}_k})$ changes smoothly (and slowly) with $\underline{t}$. Notably, models (4.1) and (4.2) 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 locally stationary random fields that does not rely on simple additive models such as (4.1) and (4.2). Fortunately, the Model-free Prediction Principle of (Politis, 2013), (Politis, 2015) suggests a way to accomplish Model-free inference in the general setting of random fields that are only locally stationary. The key towards Model-free inference is to be able to construct an invertible transformation $H_n : \underline{Y}_n \mapsto \underline{\varepsilon}_n$ where $\underline{\varepsilon}_n = (\varepsilon_1, \ldots, \varepsilon_n)'$ is a random vector with i.i.d. components. Section 4.3 describes the methodology for Model-based point prediction and Section 4.4 outlines the steps necessary for Model-free point prediction in case of locally stationary random fields. The 2 approaches are empirically compared to each other in Section 4.5 using finite sample experiments.

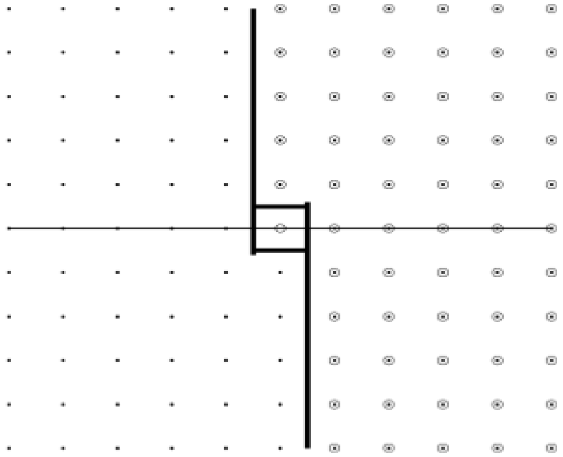

Figure 4.1: Non Symmetric Half-Plane

4.2 Causality of Random Fields

Given the random field observations $Y_{\underline{t}_1}, \ldots, Y_{\underline{t}_n}$ our goal is predictive inference for the ”next” unknown datapoint $Y_{\underline{t}_{n+1}}$. In this context a definition of causality is necessary to specify $\underline{t}_{n+1}$ where predictive inference will be performed. For this purpose we consider the region of support (ROS) of random fields discussed in this Chapter to be defined over a non symmetric half-plane (NSHP) denoted as H_∞. Figure 4.1 shows an NSHP centered at (0, 0). The NSHP can also be defined for any other point $\underline{t}$ as follows:

$$NSHP(\underline{t}) = \underline{t} + \underline{s} \quad \forall \underline{s} \in NSHP \tag{4.4}$$

Such non symmetric half-planes have been used previously for specifying causal 2-D AR models (Choi & Politis, 2007). In such cases a causal 2-D AR model with ROS

$H_p \subset H_\infty$ can be defined as:

$$Y_{t_1,t_2} = \sum_{(j,k)\in H_p} \beta_{j,k} Y_{t_1-j,t_2-k} + v_{t_1,t_2} \tag{4.5}$$

Here v_{t_1,t_2} is a 2-D white noise process with variance $\sigma^2 > 0$. Based on (Dudgeon & Mersereau, 1984) a 2-D AR process with ROS S is causal if there exists a subset C of Z^2 satisfying the following conditions:

- The set C consists of 2 rays emanating from the origin and the points between the rays
- The angle between the 2 rays is strictly less than 180 degrees
- $S \subset C$

In this case since $H_p \subset H_\infty$ satisfies these conditions the 2-D AR process satisfying (4.5) is causal. Therefore this framework can be used to describe a causal random field defined over the NSHP and used for construction of one-step ahead point predictors. This is described below.

Consider random field data $\{Y_{\underline{t}},\ \underline{t} \in E\}$ where E can be any finite subset of Z^2 for e.g. $E_{\underline{n}} = \{\underline{t} \in Z^2 \text{ with } 0 < t_1 < n_1 \ \& \ 0 < t_2 < n_2,\ \underline{n} = (n_1, n_2)\}$. Our goal is predictive inference at $\underline{t} = (t_1, t_2)$. This "future" value Y_{t_1,t_2} is determined using data defined over the region:

$$E_{\underline{t},\underline{n}} = NSHP(\underline{t}) \cap E_{\underline{n}}$$

Both model-based and model-free causal inference for Y_{t_1,t_2} are performed using the data specified over this region $E_{\underline{t},\underline{n}}$. We consider predictive inference at $Y_{\underline{t}} = Y_{t_1,t_2}$ given the data ($Y_{\underline{s}} \mid \underline{s} \prec \underline{t}$ & $\underline{s} \in E_{\underline{t},\underline{n}}$) where the symbol $\prec$ denotes lexicographical ordering on the region of support of the random field as described in (Choi & Politis, 2007). In the subsequent discussion this lexicographically ordered ”past” data $Y_{\underline{s}}$ will be denoted as $Y_{\underline{t}_1}, \ldots, Y_{\underline{t}_k}, \ldots, Y_{\underline{t}_n}$ and prediction will be performed at $Y_{\underline{t}} = Y_{\underline{t}_{n+1}}$.

4.3 Model-based inference

Throughout Section 4.3, we will assume model (4.2)—that includes model (4.1) as a special case—together with assuming that $\mu(\cdot)$ and $\sigma(\cdot)$ change smoothly (and slowly) with $\underline{t}$.

4.3.1 Theoretical optimal point prediction

It is well-known that the L_2–optimal predictor of $Y_{\underline{t}_{n+1}}$ given the data $\underline{Y}_{\underline{t}_n} = (Y_{\underline{t}_1}, \ldots, Y_{\underline{t}_n})'$ is the conditional expectation $E(Y_{\underline{t}_{n+1}}|\underline{Y}_{\underline{t}_n})$ where $\underline{Y}_{\underline{t}_n}$ indicates the data $Y_{\underline{t}_1}, \ldots, Y_{\underline{t}_n}$. Furthermore, under model (4.2), we have

$$E(Y_{\underline{t}_{n+1}}|\underline{Y}_{\underline{t}_n}) = \mu(\underline{t}_{n+1}) + \sigma(\underline{t}_{n+1})E(W_{\underline{t}_{n+1}}|\underline{Y}_{\underline{t}_n}). \tag{4.6}$$

Define $\mathcal{F}_{\underline{t}_j}^{\underline{t}_J}(Y)$ to be the *information set* $\{Y_{\underline{t}_j}, Y_{\underline{t}_{j+1}}, \ldots, Y_{\underline{t}_J}\}$, also known as σ–field, and note that the information sets $\mathcal{F}_{-\infty}^{\underline{t}}(Y)$ and $\mathcal{F}_{-\infty}^{\underline{t}}(W)$ are identical for any $\underline{t}$, i.e., knowledge of $\{Y_{\underline{s}}$ for $\underline{s} \prec \underline{t}\}$ is equivalent to knowledge of $\{W_{\underline{s}}$ for $\underline{s} \prec \underline{t}\}$; here, $\mu(\cdot)$ and $\sigma(\cdot)$ are assumed known and the symbol $\prec$ denotes lexicographical ordering on the region of support of the random field as described in (Choi & Politis, 2007). Hence, for large n, and due to the assumption that $W_{\underline{t}}$ is weakly dependent (and therefore the same must be true for $Y_{\underline{t}}$ as well), the following large-sample approximation is useful, i.e.,

$$E(W_{\underline{t}_{n+1}}|\underline{Y}_{\underline{t}_n}) \simeq E(W_{\underline{t}_{n+1}}|Y_{\underline{t}_s}, \underline{s} \preceq \underline{n}) = E(W_{\underline{t}_{n+1}}|W_{\underline{t}_s}, \underline{s} \preceq \underline{n}) \simeq E(W_{\underline{t}_{n+1}}|\underline{W}_{\underline{t}_n}) \tag{4.7}$$

where $\underline{W}_{\underline{t}_n} = (W_{t_1}, \ldots, W_{t_n})'$.

All that is needed now is to construct an approximation for $E(W_{\underline{t}_{n+1}}|\underline{W}_{\underline{t}_n})$. We construct this L_2–optimal linear predictor of $W_{\underline{t}_{n+1}}$ by fitting a (causal) AR(p,q) model to the data $W_{\underline{t}_1}, \ldots, W_{\underline{t}_n}$ with p,q chosen by minimizing AIC, BIC or a related criterion as described in (Choi & Politis, 2007); this would entail fitting the model:

$$W_{t_{n_1},t_{n_2}} = \sum_{(j,k)\in H_p} \beta_{j,k} W_{t_{n_1}-j,t_{n_2}-k} + v_{t_{n_1},t_{n_2}} \tag{4.8}$$

where $\{v_{t_{n_1},t_{n_2}}\}$ is a stationary white noise, i.e., an uncorrelated sequence, with mean zero and variance $\tau^2 > 0$ and (t_{n_1}, t_{n_2}) denote the components of $\underline{t}_{n+1}$. The implication then is that

$$\bar{E}(W_{\underline{t}_{n+1}}|\underline{W}_{\underline{t}_n}) = \sum_{(j,k)\in H_p} \beta_{j,k} W_{t_{n_1}-j,t_{n_2}-k} \tag{4.9}$$

4.3.2 Trend estimation and practical prediction

To construct the L_2–optimal predictor (4.6), 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). Since the goal is predictive inference on $Y_{\underline{t}_{n+1}}$, Nadaraya-Watson (NW or local constant) and/or local linear fitting must be performed in a *one-sided way* 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; see Fan and Gijbels (1996), Fan and Yao (2003), or Li and Racine (2007).

Furthermore to compute $\bar{E}(W_{\underline{t}_{n+1}}|\underline{W}_{\underline{t}_n})$ in eq. (4.9) we need access to the stationary data $W_{\underline{t}_1},\ldots,W_{\underline{t}_n}$. The $W_{\underline{t}}$'s are not directly observed, but—much like residuals in a regression—they can be reconstructed by eq. (4.3) with estimates of $\mu(\underline{t})$ and $\sigma(\underline{t})$ plugged-in. What is important is that **the way $W_{\underline{t}}$ is reconstructed/estimated by (say) $\hat{W}_{\underline{t}}$ must remain the same for all** $\underline{t}$, otherwise the reconstructed data $\hat{W}_{\underline{t}_1},\ldots,\hat{W}_{\underline{t}_n}$ can not be considered stationary. Since $W_{\underline{t}}$ can only be estimated in a one-sided way for $\underline{t}$ close to $\underline{t}_n$, the same one-sided way must also be implemented for $\underline{t}$ in the middle of the dataset even though in that case two-sided estimation is possible.

We will assume throughout that $K(\cdot)$ in the NW or local linear case is a nonnegative, symmetric 2-D Gaussian kernel function for which the diagonal values are set to a bandwidth value b and the off-diagonal terms are set to 0. Random field data is denoted as $Y_{\underline{t}_1},\ldots,Y_{\underline{t}_k},\ldots Y_{\underline{t}_n}$.

NW fitting: Let $\underline{t}_k \in [\underline{t}_{b+1},\underline{t}_n]$, and define

$$\hat{\mu}(\underline{t}_k) = \sum_{i=1}^{k} Y_{\underline{t}_i}\,\hat{K}\left(\frac{\underline{t}_k-\underline{t}_i}{b}\right) \text{ and } \hat{M}(\underline{t}_k) = \sum_{i=1}^{k} Y_{\underline{t}_i}^2\,\hat{K}(\frac{\underline{t}_k-\underline{t}_i}{b}) \tag{4.10}$$

where

$$\hat{\sigma}(\underline{t}_k) = \sqrt{\hat{M}_{\underline{t}_k}-\hat{\mu}(\underline{t}_k)^2} \text{ and } \hat{K}\left(\frac{\underline{t}_k-\underline{t}_i}{b}\right) = \frac{K(\frac{\underline{t}_k-\underline{t}_i}{b})}{\sum_{j=1}^{k} K(\frac{\underline{t}_k-\underline{t}_j}{b})}. \tag{4.11}$$

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

$$\hat{W}_{\underline{t}_k} = \frac{Y_{\underline{t}_k}-\hat{\mu}(\underline{t}_k)}{\hat{\sigma}(\underline{t}_k)} \text{ for } \underline{t}_k = \underline{t}_{b+1},\ldots,\underline{t}_n. \tag{4.12}$$

Similarly, the one-sided local linear (LL) fitting estimators of $\mu(\underline{t}_k)$ and $\sigma(\underline{t}_k)$ can be defined as below.

LL–Regular fitting: Let $\underline{t}_k \in [\underline{t}_{b+1}, \underline{t}_n]$, and define

$$\hat{\mu}(\underline{t}_k) = \frac{\sum_{j=1}^{k} w_j Y_{\underline{t}_j}}{\sum_{j=1}^{k} w_j + n^{-2}} \text{ and } \hat{M}(\underline{t}_k) = \frac{\sum_{j=1}^{k} w_j Y_{\underline{t}_j}^2}{\sum_{j=1}^{t} w_j + n^{-2}} \tag{4.13}$$

where

$$\underline{a} = (a_1, a_2) = (\underline{t}_j - \underline{t}_k) \tag{4.14}$$

$$s_{t1,1} = \sum_{j=1}^{k} K\left(\frac{\underline{t}_j - \underline{t}_k}{b}\right) a_1 \tag{4.15}$$

$$s_{t2,1} = \sum_{j=1}^{k} K\left(\frac{\underline{t}_j - \underline{t}_k}{b}\right) a_2 \tag{4.16}$$

$$s_{t1,2} = \sum_{j=1}^{k} K\left(\frac{\underline{t}_j - \underline{t}_k}{b}\right) a_1^2 \tag{4.17}$$

$$s_{t2,2} = \sum_{j=1}^{k} K\left(\frac{\underline{t}_j - \underline{t}_k}{b}\right) a_2^2 \tag{4.18}$$

$$s_{t1,t2} = \sum_{j=1}^{k} K\left(\frac{\underline{t}_j - \underline{t}_k}{b}\right) a_1 a_2 \tag{4.19}$$

$$w_j = K(\frac{\underline{t}_j - \underline{t}_k}{b}) \left[s_{t1,2}s_{t2,2} - s_{t1,t2}^2 - a_1(s_{t1,1}s_{t2,2} - s_{t2,1}s_{t1,t2}) + a_2(s_{t1,1}s_{t1,t2} - s_{t1,2}s_{t2,1})\right], \tag{4.20}$$

The term n^{-2} in eq. (4.13) is just to ensure the denominator is not zero; see Fan (1993). Eq. (4.11) then yields $\hat{\sigma}(\underline{t}_k)$, and eq. (4.12) yields $\hat{W}_{\underline{t}_k}$.

Using one of the above methods (NW vs. LL) gives estimates of the quantities needed to compute the L_2–optimal predictor (4.6). In order to approximate $E(W_{\underline{t}_{n+1}}|\underline{Y}_{\underline{t}_n})$, one would treat the proxies $\hat{W}_{\underline{t}_k}$ or $\tilde{W}_{\underline{t}_k}$ as if they were the true $W_{\underline{t}_k}$, and proceed as outlined in Section 4.3.1.

Remark 4.3.1 (Random Field 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 prediction setting, i.e., always one-step-ahead. To elaborate, let $k < n$, and suppose only subseries $Y_{\underline{t}_1}, \ldots, Y_{\underline{t}_k}$ has been observed. Denote $\hat{Y}_{\underline{t}_{k+1}}$ the best predictor of $Y_{\underline{t}_{k+1}}$ based on the data $Y_{\underline{t}_1}, \ldots, Y_{\underline{t}_k}$ constructed according to the above methodology and some choice of b. However, since $Y_{\underline{t}_{k+1}}$ is known, the quality of the predictor can be assessed. So, for each value of b over a reasonable range, we can form either $PRESS(b) = \sum_{k=k_o}^{n-1}(\hat{Y}_{\underline{t}_{k+1}} - Y_{\underline{t}_{k+1}})^2$ or $PRESAR(b) = \sum_{k=k_o}^{n-1}|\hat{Y}_{\underline{t}_{k+1}} - Y_{\underline{t}_{k+1}}|$; 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 $\hat{Y}_{\underline{t}_{k+1}} \simeq \hat{\mu}(\underline{t}_{k+1})$ and base $PRESS(b)$ or $PRESAR(b)$ on this approximation.

4.4 Model-free inference

Model (4.2) is a flexible way to account for a coordinate-changing mean and variance of $Y_{\underline{t}}$. However, nothing precludes that the random field $\{Y_{\underline{t}} \text{ for } \underline{t} \in \mathbf{Z}^{\mathbf{2}}\}$ 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 $H_n : \underline{Y}_{\underline{t}_n} \mapsto \underline{\varepsilon}_n$ where $\underline{\varepsilon}_n = (\varepsilon_1, \ldots, \varepsilon_n)'$ 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 $\mathcal{L}(Y_{\underline{t}_k}, Y_{\underline{t}_{k-1}}, \ldots, Y_{\underline{t}_{k-m+1}})$ the mth marginal of the random field $Y_{\underline{t}}$, i.e. the joint probability law of the vector $(Y_{\underline{t}_k}, Y_{\underline{t}_{k-1}}, \ldots, Y_{\underline{t}_{k-m+1}})'$. Although we abandon model (4.2) in what follows, we still want to employ nonparametric smoothing for estimation; thus, we must assume that $\mathcal{L}(Y_{\underline{t}_k}, Y_{\underline{t}_{k-1}}, \ldots, Y_{\underline{t}_{k-m+1}})$ changes smoothly (and slowly) with $\underline{t}_k$. For concreteness and easy comparison with the model-based case of Eq. (4.2), we will focus in the sequel on the case $m = 1$.

4.4.1 Constructing the theoretical transformation

Let $D_{\underline{t}}(y) = P\{Y_{\underline{t}} \leq y\}$ denote the 1st marginal distribution of random field $\{Y_{\underline{t}}\}$. Throughout Section 4.4, the default assumption will be that $D_{\underline{t}}(y)$ is (absolutely) continuous in y for all $\underline{t}$.

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

$$U_{\underline{t}} = D_{\underline{t}}(Y_{\underline{t}}) \quad \text{for } \underline{t} = \underline{t}_1, \ldots, \underline{t}_n; \tag{4.21}$$

the assumed continuity of $D_{\underline{t}}(y)$ in y implies that $U_{\underline{t}_1}, \ldots, U_{\underline{t}_n}$ are random variables having distribution Uniform $(0,1)$. However, $U_{\underline{t}_1}, \ldots, U_{\underline{t}_n}$ 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

$$Z_{\underline{t}} = \Phi^{-1}(U_{\underline{t}}) \text{ for } \underline{t} = \underline{t}_1, \ldots, \underline{t}_n; \tag{4.22}$$

it then follows that $Z_{\underline{t}_1}, \ldots, Z_{\underline{t}_n}$ are standard normal—albeit correlated—random variables.

Let Γ_n denote the $n \times n$ covariance matrix of the random vector $\underline{Z}_{\underline{t}_n} = (Z_{\underline{t}_1}, \ldots, Z_{\underline{t}_n})'$. Under standard assumptions, e.g. that the spectral density of the series $\{Z_{\underline{t}_n}\}$ is continuous and bounded away from zero, 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:

$$\underline{\varepsilon}_n = C_n^{-1} \underline{Z}_{\underline{t}_n}. \tag{4.23}$$

It then follows that the entries of $\underline{\varepsilon}_n = (\varepsilon_1, \ldots, \varepsilon_n)'$ are uncorrelated standard normal. Assuming that the random variables $Z_{\underline{t}_1}, \ldots, Z_{\underline{t}_n}$ were *jointly* normal, this can be strenghtened to claim that $\varepsilon_1, \ldots, \varepsilon_n$ are i.i.d. $N(0,1)$. Consequently, the transformation of the dataset $\underline{Y}_n = (Y_1, \ldots, Y_n)'$ to the vector $\underline{\varepsilon}_n$ 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. (4.21), (4.22) and (4.23), are invertible; hence, the composite transformation $H_n : \underline{Y}_n \mapsto \underline{\varepsilon}_n$ is invertible as well.

4.4.2 Kernel estimation of the 'uniformizing' transformation

We first focus on estimating the 'uniformizing' part of the transformation, i.e., eq. (4.21). Recall that the Model-free setup implies that the function $D_{\underline{t}}(\cdot)$ changes smoothly (and slowly) with $\underline{t}$; hence, local constant and/or local linear fitting can be used to estimate it. Consider random field data denoted as $Y_{\underline{t}_1}, \ldots, Y_{\underline{t}_k}, \ldots Y_{\underline{t}_n}$. Using local constant, i.e., kernel estimation, a consistent estimator of the marginal distribution $D_{\underline{t}_k}(y)$ is given by:

$$\hat{D}_{\underline{t}_k}(y) = \sum_{i=1}^{k} \mathbf{1}\{Y_{\underline{t}_i} \leq y\} \tilde{K}(\frac{\underline{t}_k - \underline{t}_i}{b}) \tag{4.24}$$

where $\tilde{K}(\frac{\underline{t}_k - \underline{t}_i}{b}) = K(\frac{\underline{t}_k - \underline{t}_i}{b}) / \sum_{j=1}^{T} K(\frac{\underline{t}_k - \underline{t}_j}{b})$. Similar to the model-based case we will assume throughout that $K(\cdot)$ is a nonnegative, symmetric 2-D Gaussian kernel function for which the diagonal values are set to the bandwidth b and the off-diagonal terms are set to 0. Note that the kernel estimator (4.24) is *one-sided* for the same reasons discussed in Section 4.3.2. Since $\hat{D}_{\underline{t}_k}(y)$ is a step function in y, a smooth estimator can be defined as:

$$\bar{D}_{\underline{t}_k}(y) = \sum_{i=1}^{k} \Lambda(\frac{y - Y_{\underline{t}_i}}{h_0}) \tilde{K}(\frac{\underline{t}_k - \underline{t}_i}{b}) \tag{4.25}$$

where h_0 is a secondary bandwidth. Cross-validation can be used to determine the bandwidths h_0 and b ; details are described in Section 4.4.5.

4.4.3 Local linear estimation of the 'uniformizing' transformation

Note that the kernel estimator $\hat{D}_{\underline{t}_k}(y)$ defined in eq. (4.24) is just the Nadaraya-Watson smoother, i.e., local average, of the variables $u_1, \ldots, u_n$ where $u_i = \mathbf{1}\{Y_{\underline{t}_i} \leq y\}$. Similarly, $\bar{D}_{\underline{t}_k}(y)$ defined in eq. (4.25) is just the Nadaraya-Watson smoother of the variables

$v_1, \ldots, v_n$ where $v_i = \Lambda(\frac{y-Y_{\underline{t}_i}}{h_0})$. 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 in 1-sided estimation on the boundary of the random field.

Let $\hat{D}^{LL}_{\underline{t}_k}(y)$ and $\bar{D}^{LL}_{\underline{t}_k}(y)$ denote the local linear estimators of $D_{\underline{t}_k}(y)$ based on either the indicator variables $\mathbf{1}\{Y_{\underline{t}_i} \leq y\}$ or the smoothed variables $\Lambda(\frac{y-Y_{\underline{t}_i}}{h_0})$ respectively. Keeping y fixed, $\hat{D}^{LL}_{\underline{t}_k}(y)$ and $\bar{D}^{LL}_{\underline{t}_k}(y)$ exhibit good behavior for estimation at the boundary, e.g. smaller bias than either $\hat{D}_{\underline{t}_k}(y)$ and $\bar{D}_{\underline{t}_k}(y)$ 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.

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 $\hat{D}_{\underline{t}_k}(y)$ and $\bar{D}_{\underline{t}_k}(y)$ adjusted via Hansen's (2004) proposal are given as follows:

$$\hat{D}^{LLH}_{\underline{t}_k}(y) = \frac{\sum_{i=1}^k w_i^\diamond \mathbf{1}(Y_{\underline{t}_i} \leq y)}{\sum_{i=1}^k w_i^\diamond} \quad \text{and} \quad \bar{D}^{LLH}_{\underline{t}_k}(y) = \frac{\sum_{i=1}^k w_i^\diamond \Lambda(\frac{y-Y_{\underline{t}_i}}{h_0})}{\sum_{i=1}^k w_i^\diamond}. \tag{4.26}$$

The weights $w_i^\diamond$ are derived from weights w_i described in equation (4.20) where:

$$w_i^\diamond = \begin{cases} 0 & \text{when } w_i < 0 \\ w_i & \text{when } w_i \geq 0 \end{cases} \tag{4.27}$$

4.4.4 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 random fields), 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 $\bar{D}^{LL}_{\underline{t}_k}(y)$ to construct a monotonic version denoted by $\bar{D}^{LLM}_{\underline{t}_k}(y)$.

The Monotone Local Linear Distribution Estimator $\bar{D}^{LLM}_{\underline{t}_k}(y)$ can be constructed by Algorithm 4.4.1 given below.

Algorithm 4.4.1 Monotone Local Linear Distribution Estimation

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

$$\bar{d}^{LL}_{\underline{t}_k}(y) = \frac{\frac{1}{h_0}\sum_{j=1}^{n} w_j \lambda(\frac{y-Y_{\underline{t}_j}}{h_0})}{\sum_{j=1}^{n} w_j}$$

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

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

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

$$\bar{d}^{LLM}_{\underline{t}_k}(y) = \frac{\bar{d}^{LL+}_{\underline{t}_k}(y)}{\int_{-\infty}^{\infty} \bar{d}^{LL+}_{\underline{t}_k}(s)ds}. \tag{4.28}$$

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

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.

4.4.5 Cross-validation Bandwidth Choice for Model-Free Inference

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

Algorithm 4.4.2 *BANDWIDTH DETERMINATION FOR MODEL-FREE INFERENCE*

1. *Perform the uniformizing transform described in* (4.21) *over the given random field dataset* $Y_{\underline{t}_1}, \ldots, Y_{\underline{t}_k}, \ldots, Y_{\underline{t}_n}$ *using either of the estimators* $\bar{D}_{\underline{t}_k}(y)$, $\bar{D}^{LLH}_{\underline{t}_k}(y)$ *or* $\bar{D}^{LLM}_{\underline{t}_k}(y)$ *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* $\bar{D}_{\underline{t}_k}(y)$, $\bar{D}^{LLH}_{\underline{t}_k}(y)$ *or* $\bar{D}^{LLM}_{\underline{t}_k}(y)$.

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 4.3.2. For*

this purpose let $j < n$, and suppose only subseries $Y_{\underline{t}_1}, \ldots, Y_{\underline{t}_j}$ has been observed. Denote $\hat{Y}_{\underline{t}_{j+1}}$ the best predictor of $Y_{\underline{t}_{j+1}}$ based on the data $Y_1, \ldots, Y_{\underline{t}_k}$ constructed using $\bar{D}_{\underline{t}_k}(y)$, $\bar{D}^{LLH}_{\underline{t}_k}(y)$ or $\bar{D}^{LLM}_{\underline{t}_k}(y)$ and a value of b selected among the p values obtained above. Since $Y_{\underline{t}_{j+1}}$ is known, the quality of the predictor can be assessed. So, for each value of b we can form either $PRESS(b) = \sum_{j=j_o}^{n-1} (\hat{Y}_{\underline{t}_{j+1}} - Y_{\underline{t}_{j+1}})^2$ or $PRESAR(b) = \sum_{j=j_o}^{n-1} |\hat{Y}_{\underline{t}_{j+1}} - Y_{\underline{t}_{j+1}}|$; here j_o should be big enough so that estimation is accurate, e.g., j_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.

4.4.6 Estimation of the whitening transformation

To implement the whitening transformation (4.23), it is necessary to estimate Γ_n, i.e., the $n \times n$ covariance matrix of the random vector $\underline{Z}_{\underline{t}_n} = (Z_{\underline{t}_1}, \ldots, Z_{\underline{t}_n})'$ where the $Z_{\underline{t}}$ are the normal random variables defined in eq. (4.22). Let $\{Z_{\underline{t}} = Z_{\underline{t}_1, \underline{t}_2} | t_1 = 1, \ldots, T_1, t_2 = 1, \ldots, T_2, n = T_1 T_2\}$. The problem involves positive definite estimation of Γ_n based on the sample $Z_{\underline{t}_1}, \ldots, Z_{\underline{t}_n}$. One method to implement this involves the following steps:

- Calculate the sample autocovariance function as follows:
$\breve{\gamma}(i,j) = \breve{\gamma}(-i,-j) = \frac{1}{T_1T_2}\sum_{t_1=1}^{T_1-i}\sum_{t_2=1}^{T_2-j} Z_{t_1+i,t_2+j}Z_{t_1,t_2},\ i,j = 0,1,2,\ldots$
$\breve{\gamma}(i,-j) = \breve{\gamma}(-i,j) = \frac{1}{T_1T_2}\sum_{t_1=1}^{T_1-i}\sum_{t_2=1}^{T_2-j} Z_{t_1+i,t_2-j}Z_{t_1,t_2},\ i,j = 0,1,2,\ldots$

- Let $\hat{\Gamma}_n = \left[\hat{\gamma}_{i,j}\right]_{i,j=1}^{n}$ be the banded, tapered covariance matrix where $\hat{\gamma}(i,j) = \lambda(i,j)\breve{\gamma}(i,j)$ and $\lambda(i,j)$ is the 2-D flat-top taper defined in (Politis & Romano, 1996). The final estimator of Γ_n can be $\hat{\Gamma}_n^{\star}$ which is a a positive definite version of $\hat{\Gamma}_n$ that is banded and Toeplitz.

Consider the 'augmented' vectors $\underline{Y}_{\underline{t}_{n+1}} = (Y_{\underline{t}_1}, \ldots, Y_{\underline{t}_n}, Y_{\underline{t}_{n+1}})'$, $\underline{Z}_{\underline{t}_{n+1}} = (Z_{\underline{t}_1}, \ldots, Z_{\underline{t}_n}, Z_{\underline{t}_{n+1}})'$ and $\underline{\varepsilon}_{n+1} = (\varepsilon_1, \ldots, \varepsilon_n, \varepsilon_{n+1})'$ where the values $Y_{\underline{t}_{n+1}}, Z_{\underline{t}_{n+1}}$ and ε_{n+1} are yet unobserved. As described before estimating the 'uniformizing' transformation $D_{\underline{t}}(\cdot)$ and the whitening transformation based on Γ_n allows us to estimate the transformation $H_n : \underline{Y}_{\underline{t}_n} \mapsto \underline{\varepsilon}_n$. We now show how to obtain the inverse transformation $H_{n+1}^{-1} : \underline{\varepsilon}_{n+1} \mapsto \underline{Y}_{\underline{t}_{n+1}}$. Recall that $\underline{\varepsilon}_n$ and $\underline{Y}_{\underline{t}_n}$ are related in a one-to-one way via transformation H_n, so the values $Y_{\underline{t}_1}, \ldots, Y_{\underline{t}_n}$ are obtainable by $\underline{Y}_{\underline{t}_n} = H_n^{-1}(\underline{\varepsilon}_n)$. Hence, we just need to show how to create the unobserved $Y_{\underline{t}_{n+1}}$ from $\underline{\varepsilon}_{n+1}$; this is done in the following three steps.

Algorithm 4.4.3 *GENERATION OF UNOBSERVED DATAPOINT FROM FUTURE INNOVATIONS*

i. Let

$$\underline{Z}_{\underline{t}_{n+1}} = C_{n+1}\underline{\varepsilon}_{n+1} \tag{4.29}$$

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

$$Z_{\underline{t}_{n+1}} = \underline{c}_{n+1}\underline{\varepsilon}_{n+1} \tag{4.30}$$

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*

$$U_{\underline{t}_{n+1}} = \Phi(Z_{\underline{t}_{n+1}}). \tag{4.31}$$

iii. *Finally, define*

$$Y_{\underline{t}_{n+1}} = D_{n+1}^{-1}(U_{\underline{t}_{n+1}}); \tag{4.32}$$

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

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

4.4.7 Model-free point prediction

In the previous sections, it was shown how the construct the transformation $H_n : \underline{Y}_{\underline{t}_n} \mapsto \underline{\varepsilon}_n$ and its inverse $H_{n+1}^{-1} : \underline{\varepsilon}_{n+1} \mapsto \underline{Y}_{\underline{t}_{n+1}}$, where the random variables $\varepsilon_1, \varepsilon_2, \ldots,$ are i.i.d. Note that by combining eq. (4.30), (4.31) and (4.32) we can write the formula:

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

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

$$Y_{\underline{t}_{n+1}} = g_{n+1}(\varepsilon_{n+1}) \;\text{ where }\; g_{n+1}(x) = D^{-1}_{\underline{t}_{n+1}}\left(\Phi\left(\sum_{i=1}^{n} c_i\varepsilon_i + c_{n+1}x\right)\right). \tag{4.33}$$

Eq. (4.33) is the predictive equation required in the Model-free Prediction Principle; conditionally on $\underline{Y}_{\underline{t}_n}$, it can be used like a model equation in computing the L_2– and L_1–optimal point predictors of $Y_{\underline{t}_{n+1}}$. Based on this the algorithm for constructing Model-free point predictors is described below.

Algorithm 4.4.4 MODEL-FREE (MF) POINT PREDICTION FOR $Y_{\underline{t}_{n+1}}$

1. *Construct $U_{\underline{t}_1},\ldots,U_{\underline{t}_n}$ by eq. (4.21) with $D_{\underline{t}_k}(\cdot)$ estimated by either $\bar{D}_{\underline{t}_k}(\cdot)$, $\bar{D}^{LLH}_{\underline{t}_k}(\cdot)$ or $\bar{D}^{LLM}_{\underline{t}_k}(\cdot)$; for all the 3 types of estimators, use the respective formulas with $T = k$.*

2. *Construct $Z_{\underline{t}_1},\ldots,Z_{\underline{t}_n}$ by eq. (4.22), and use the method of Section 4.4.6 to estimate Γ_n by $\hat{\Gamma}^{\star}_n$.*

3. *Construct $\varepsilon_1,\ldots,\varepsilon_n$ by eq. (4.23), and let $\hat{F}_n$ denote their empirical distribution.*

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

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

   where the function g_{n+1} is defined in the predictive equation (4.33) *with $D_{\underline{t}_{n+1}}(\cdot)$ being again estimated by either $\bar{D}_{\underline{t}_{n+1}}(\cdot)$, $\bar{D}^{LLH}_{\underline{t}_{n+1}}(\cdot)$ or $\bar{D}^{LLM}_{\underline{t}_{n+1}}(\cdot)$*

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

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

The performance of the Model-free and Model-based predictors described above are empirically compared using simulated data based on point prediction. The Model-based local constant and local linear methods are denoted as MB-LC and MB-LL respectively. Model-based predictors MB-LC and MB-LL are described in Section 4.3. 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 predictors are described in Section 4.4. Point prediction performance as indicated by Mean Squared Error (MSE) are used to compare the estimators.

4.5.1 Simulation: Additive model with stationary 2-D AR errors

Let a random field be generated using the 2-D AR process as below:

$$y(t_1,t_2) = 0.25y_{t_1-1,t_2-1} + 0.2y_{t_1-1,t_2+1} - 0.05y_{t_1-2,t_2} + v(t_1,t_2) \tag{4.34}$$

Let this field be generated over the region defined by $0 \leq t_1 \leq n_1$ & $0 \leq t_2 \leq n_2$ where $n_1 = 101, n_2 = 101$ Let $t_1 = 50, t_2 = 50$ where point prediction is performed, Here $v(t_1,t_2)$ are i.i.d. $N(0,\tau^2)$ where $\tau = 0.1$. The data Y_i is generated using the additive model in eq. (4.1) with trend specified as $\mu(\underline{t}) = \mu(t_1,t_2) = \sin(4\pi\frac{t_2-1}{n_2-1})$ where $0 \leq t_1 \leq n_1$ & $0 \leq t_2 \leq n_2$.

Results for point prediction using mean square error (MSE) over all MB and MF methods are shown for a range of bandwidths b in Table 4.1. A total of 96 realizations of the dataset were used for measuring point prediction performance. From this table it can be

Table 4.1: Point Prediction performance for 2-D AR dataset

Bandwidth	MB-LC	MB-LL	MF-LC	MF-LLH	MF-LLM
1	4.456e-01	3.900e-02	3.721e-02	3.800e-02	3.778e-02
10	3.870e-02	3.777e-02	3.847e-02	3.646e-02	3.458e-02
15	3.839e-02	3.783e-02	3.900e-02	3.682e-02	3.497e-02
20	3.851e-02	3.795e-02	3.946e-02	3.721e-02	3.540e-02

seen that among the model-based methods MB-LL outperforms MB-LC. However when the performance of all estimators are considered the MF-LLM estimator has the overall best MSE performance across all considered bandwidths.

Appendix A

Basic Model-free Bootstrap and Double Bootstrap Algorithms

This section describes in detail algorithms A.0.1 and A.0.2 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.0.3 and A.0.4 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.,

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

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.,

$$Y^*_{n+1} - \Pi(\hat{g}^*_{n+1}, \underline{Y}_n, \hat{F}_n) \tag{A.2}$$

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

Remark A.0.1 Note that eq. (A.2) 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.0.1 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. (A.2).*

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*

$$[\Pi + q(\alpha/2),\ \Pi + q(1-\alpha/2)] \tag{A.3}$$

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

$$Y_{n+1} - \Pi(\hat{g}_{n+1}, \underline{Y}_n, F) \tag{A.4}$$

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

$$Y_{n+1}^* - \Pi(\hat{g}_{n+1}^*, \underline{Y}_n, F). \tag{A.5}$$

Algorithm A.0.2 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. (A.5).

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*

$$[\Pi + q(\alpha/2),\ \Pi + q(1-\alpha/2)] \tag{A.6}$$

   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 3.2.2 and 3.3.5 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.0.3 and A.0.4 below enable the determination of this adjusted bandwidth b'.

Algorithm A.0.3 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* (3.2).

2. *Fit the AR(p) model* (3.8) *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*

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

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*

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

This is the first bootstrap loop.

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

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

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* (3.2).

6. *Fit the AR(p) model* (3.8) *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*

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

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*

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

This is the second bootstrap loop.

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

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

(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*

$$[\Pi^{*} + q(\alpha/2),\ \Pi^{*} + q(1-\alpha/2)]. \tag{A.7}$$

Here Π^{*} *is given by:*

$$\Pi^{*} = \check{\mu}^{*}(n+1) + \check{\sigma}^{*}(n+1)\left[\hat{\phi}_1^{*} W_n^{*} + \cdots + \hat{\phi}_p^{*} W_{n-p+1}^{*}\right] \tag{A.8}$$

where $\hat{\phi}_1^{*}, \ldots, \hat{\phi}_p^{*}$ *are the Yule-Walker estimators of* $\phi_1, \ldots, \phi_p$ *appearing in eq.* (3.8).

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 3.2.1.*

Algorithm A.0.4 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*

$$Y_{n+1}^{**} - \Pi(\hat{g}_{n+1}^{**}, \underline{Y}_n^*, \hat{F}_n^*). \tag{A.9}$$

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*

$$[\Pi^* + q(\alpha/2),\ \Pi^* + q(1-\alpha/2)] \tag{A.10}$$

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.0.1 and A.0.2.*

Appendix B

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 Chapter 3. 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):

$$x(i) = x_{fit}(i) + \varepsilon(i) \tag{B.1}$$

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

$$x_{fit}(t) = \begin{cases} x1, & for\, t \leq t1, \\ x1 + (t - t1)(x2 - x1)/(t2 - t1), & for\, t1 \leq t \leq t2, \\ x2, & for\, t \geq t2 \end{cases} \tag{B.2}$$

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:

$$SSQW(t1, x1, t2, x2) = \sum_{i=1}^{n} \frac{[x(i) - x_{fit}(i)]^2}{\sigma(i)^2} \tag{B.3}$$

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.0.5 *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 (B.2) and (B.3) 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.*

References

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

Brunk, H. D. (1955). Maximum likelihood estimates of monotone parameters. The Annals of Mathematical Statistics, 607–616.

Choi, B., & Politis, D. N. (2007). Modeling 2-d ar processes with various regions of support. IEEE transactions on signal processing, 55(5), 1696–1707.

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.

Dette, H., Neumeyer, N., Pilz, K. F., et al. (2006). A simple nonparametric estimator of a strictly monotone regression function. Bernoulli, 12(3), 469–490.

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.

Dudgeon, D. E., & Mersereau, R. M. (1984). Multidimensional digital signal processing prentice-hall signal processing series. Prentice-Hall, Englewood Cliffs, NJ.

Efron, B., & Tibshirani, R. J. (1993). An introduction to the bootstrap. Monographs on *statistics and applied probability*, *57*.

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., & Huang, L.-S. (2001). Nonparametric kernel regression subject to monotonicity constraints. *Annals of Statistics*, 624–647.

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.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). Islr: Data for an introduction to statistical learning with applications in r [Computer software manual]. Retrieved from `http://CRAN.R-project.org/package=ISLR` (R package version 1.0)

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.

Koenker, R. (2005). *Quantile regression* (No. 38). Cambridge University Press, Cambridge.

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.

Politis, D. N., & Romano, J. P. (1996). On flat-top kernel spectral density estimators for homogeneous random fields. *Journal of Statistical Planning and Inference*, *51*(1), 41–53.

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.

Schucany, W. R. (2004). Kernel smoothers: an overview of curve estimators for the first graduate course in nonparametric statistics. Statistical Science, 663–675.

Serfling, R. J. (2009). Approximation theorems of mathematical statistics (Vol. 162). John Wiley & Sons.

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

Wand, M. P., & Jones, M. C. (1994). Kernel smoothing. CRC Press, Boca Raton.

Yu, K., & Jones, M. (1998). Local linear quantile regression. Journal of the American statistical Association, 93(441), 228–237.

Yu, K., Lu, Z., & Stander, J. (2003). Quantile regression: applications and current research areas. Journal of the Royal Statistical Society: Series D (The Statistician), 52(3), 331–350.

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.