Frontiers in Probability and the Statistical Sciences

Dimitris N. Politis

Model-Free Prediction and Regression

A Transformation-Based Approach to Inference

Springer

Frontiers in Probability and the Statistical Sciences

More information about this series at http://www.springer.com/series/11957

Dimitris N. Politis

Model-Free Prediction and Regression

A Transformation-Based Approach to Inference

Springer

Dimitris N. Politis
Department of Mathematics
University of California, San Diego
La Jolla, CA, USA

Frontiers in Probability and the Statistical Sciences
ISBN 978-3-319-21346-0 ISBN 978-3-319-21347-7 (eBook)
DOI 10.1007/978-3-319-21347-7

Library of Congress Control Number: 2015948372

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Für die zwei Violinen und die Viola des
Model-Freien Quartetts

Preface

Prediction has been one of the earliest forms of statistical inference. The emphasis on parametric estimation and testing seems to only have occurred about 100 years ago; see Geisser (1993) for a historical overview. Indeed, parametric models served as a cornerstone for the foundation of Statistical Science in the beginning of the twentieth century by R.A. Fisher, K. Pearson, J. Neyman, E.S. Pearson, W.S. Gosset (also known as "Student"), etc.; their seminal developments resulted into a complete theory of statistics that could be practically implemented using the technology of the time, i.e., pen and paper (and slide-rule!).

While some models are inescapable, e.g., modeling a polling dataset as a sequence of independent Bernoulli random variables, others appear contrived, often invoked for the sole reason to make the mathematics work. As a prime example, the ubiquitous—and typically unjustified—assumption of Gaussian data permeates statistics textbooks to the day. Model criticism and diagnostics were developed as a practical way out; see Box (1976) for an account of the model-building process by one of the pioneers of applied statistics.

With the advent of widely accessible powerful computing in the late 1970s, computer-intensive methods such as resampling and cross-validation created a revolution in modern statistics. Using computers, statisticians became able to analyze big datasets for the first time, paving the way towards the "big data" era of the twenty-first century. But perhaps more important was the realization that the way we do the analysis could/should be changed as well, as practitioners were gradually freed from the limitations of parametric models. For instance, the great success of Efron's (1979) bootstrap was in providing a complete theory for statistical inference under a nonparametric setting much like Maximum Likelihood Estimation had done half a century earlier under the restrictive parametric setup.

Nevertheless, there is a further step one may take, i.e., going beyond even nonparametric models, and this is the subject of the monograph at hand. To explain this, let us momentarily focus on regression, i.e., data that are pairs: $(Y_1, X_1), (Y_2, X_2), \ldots, (Y_n, X_n)$, where Y_i is the measured response associated with a regressor value of X_i. There are several ways to model such a dataset; three main ones are listed below.

They all pertain to the standard, homoscedastic additive model:

$$Y_i = \mu(X_i) + \varepsilon_i \tag{1}$$

where the random variables ε_i are assumed to be independent, identically distributed (i.i.d.) from a distribution $F(\cdot)$ with mean zero.

- **Parametric model:** Both $\mu(\cdot)$ and $F(\cdot)$ belong to parametric families of functions, e.g., $\mu(x) = \beta_0 + \beta_1 x$ and $F(\cdot)$ is $N(0, \sigma^2)$.
- **Semiparametric model:** $\mu(\cdot)$ belongs to a parametric family, whereas $F(\cdot)$ does not; instead, it may be assumed that $F(\cdot)$ belongs to a smoothness class, etc.
- **Nonparametric model:** Neither $\mu(\cdot)$ nor $F(\cdot)$ can be assumed to belong to parametric families of functions.

Despite the nonparametric aspect of it, even the last option constitutes a model, and is thus rather restrictive. To see why, note that Eq. (1) with i.i.d. errors is not satisfied in many cases of interest even after allowing for heteroscedasticity of the errors. For example, consider the model $Y_i = G(X_i, \varepsilon_i)$, where the ε_i are i.i.d., and $G(\cdot,\cdot)$ is a nonlinear/non-additive function of two variables. It is for this reason, i.e., to render the data amenable to an additive model such as (1), that a multitude of transformations in regression have been proposed and studied over the years, e.g., Box-Cox, ACE, AVAS, etc.; see Linton et al. (1997) for a review.

Nevertheless, it is possible to shun Eq. (1) altogether and still conduct inference about a quantity of interest such as the conditional expectation function $E(Y|X = x)$. In contrast to nonparametric model (1), the following model-free assumption can be made:

- **Model-free regression:**
  - **Random design.** The pairs $(Y_1, X_1), (Y_2, X_2), \ldots, (Y_n, X_n)$ are i.i.d.
  - **Deterministic design.** The variables $X_1, \ldots, X_n$ are deterministic, and the random variables $Y_1, \ldots, Y_n$ are independent with common conditional distribution, i.e., $P\{Y_j \leq y | X_j = x\} = D_x(y)$ not depending on j.

Inference for features, i.e., functionals, of the common conditional distribution $D_x(\cdot)$ is still possible under some regularity conditions, e.g., smoothness. Arguably, the most important such feature is the conditional mean $E(Y|X = x)$ that can be denoted $\mu(x)$. While $\mu(x)$ is crucial in the model (1) as the function explaining Y on the basis of $X = x$, it has a key function in model-free prediction as well: $\mu(x_{\mathrm{f}})$ is the mean squared error (MSE) optimal predictor of a future response Y_{f} associated with a regressor value x_{f}.

As will be shown in the sequel, it is possible to accomplish the goal of point and interval prediction of Y_{f} under the above model-free setup; this is achieved via the **Model-free Prediction Principle** described in Part I of the book. In so doing, the solution to interesting estimation problems is obtained as a by-product, e.g., inference on features of $D_x(\cdot)$; the prime example again is $\mu(x)$. Hence, a Model-free approach to frequentist statistical inference is possible, including prediction and confidence intervals.

In nonparametric statistics, it is common to try to develop some asymptotic theory for new methods developed. In addition to offering justification for the accuracy of these methods, asymptotics often provide insights on practical implementation, e.g., on the optimal choice of smoothing bandwidth, etc. All of the methods discussed/employed in the proposed Model-free approach to inference will be based on estimators that have favorable large-sample properties—such as consistency—under regularity conditions. Furthermore, asymptotic information on bandwidth rates, MSE decay rates, etc. will be given whenever available in the form of Facts or Claims together with suggestions on their proof and/or references. However, formal theorems and proofs were deemed beyond the scope of this monograph in order to better focus on the methodology, as well as keep the book's length (and time of completion) under control. Perhaps more importantly, note that it is still unclear how to properly judge the quality of prediction intervals in an asymptotic setting; some preliminary ideas on this issue are given in Sects. 3.6.2 and 7.2.3, and the Rejoinder of Politis (2013).

Interestingly, the emphasis on prediction seems to be coming back full-circle in the twenty-first century with the recent boom in machine learning and data mining; see, e.g., the highly influential book on statistical learning by Hastie et al. (2009), and the recent monograph on predictive modeling by Kuhn and Johnson (2013). The Model-free prediction methods presented here are of a very different nature but share some similarities, e.g., in employing cross-validation and sample re-use for fine-tuning and optimization, and may thus complement well the popular model-based approaches to prediction and classification. Furthermore, ideas from statistical learning and model selection could eventually be incorporated in the Model-free framework as well, e.g., selecting a subset of regressors; this is the subject of ongoing work. Notably, the methods presented in this monograph are very computer-intensive; relevant R functions and software are given at: `http://www.math.ucsd.edu/~politis/DPsoftware.html`.

I would like to thank my colleagues in the Departments of Mathematics and Economics of UCSD for their support, and my Ph.D. students for bearing with some of the material. I have benefited immensely from suggestions and discussions with colleagues from all over the world; a very partial list includes: Ian Abramson, Ery Arias-Castro, Brendan Beare, Patrice Bertail, Ricardo Cao, Anirban DasGupta, Richard Davis, Brad Efron, Peter Hall, Xuming He, Nancy Heckman, Göran Kauermann, Claudia Klüppelberg, Piotr Kokoszka, Jens-Peter Kreiss, Michele La Rocca, Jacek Leskow, Tim McMurry, George Michailidis, Stathis Paparoditis, Mohsen Pourahmadi, Jeff Racine, Joe Romano, Dimitrios Thomakos, Florin Vaida, Slava Vasiliev, Philippe Vieu, and Michael Wolf. Further acknowledgements are given at the end of several chapters.

In closing, I would like to thank the Division of Mathematical Sciences of the National Science Foundation for their continuing support with multiple grants, the most recent ones being DMS-10-07513 and DMS 13-08319, and the John Simon Guggenheim Memorial Foundation for a 2011–2012 fellowship that helped me get started on this monograph. I would also like to thank Marc Strauss and Hannah

Bracken of Springer for a wonderful collaboration, and Somnath Datta and the Editorial Board of the Frontiers Series for hosting this project.

The impetus for putting together this monograph was to show how very different statistical problems can be approached afresh in a Model-free setting. Due to time and space limitations, I could only explicitly address a handful of areas of practical implementation, e.g., regression, autoregression, Markov processes, etc. It is my sincere hope that the monograph will incite the interest of readers to take another look at their favorite problem—either theoretical or applied—in this new light; the insights gained may be well worth it.

San Diego, CA, USA Dimitris N. Politis
Spring 2015

Contents

Acronyms

acf	Autocorrelation function
AIC	Akaike Information Criterion
AR	Auto regressive
ARCH	Auto regressive conditional heteroscedasticity
ARMA	Auto regressive with moving average residuals
cdf	Cumulative distribution function
CVR	Coverage (of interval)
DFT	Discrete Fourier Transform
FSO	Full-Sample Optimal
GARCH	Generalized ARCH
i.i.d.	Independent, identically distributed
i.i.d.–ness	The property of a dataset being i.i.d.
i.i.d. (μ, σ^2)	i.i.d. with mean μ and variance σ^2
LEN	Length (of interval)
LMF	Limit Model Free
LS	Least squares
MA	Moving average
MAD	Mean absolute deviation
MB	Model-based
MF	Model-free
MF^2	Model-free model-fitting
MLE	Maximum likelihood estimation
MSE	Mean squared error
$N(\mu, \sigma^2)$	Normal with mean μ and variance σ^2
NoVaS	Normalizing and variance stabilizing transformation
PMF	Predictive model-free
st.err.	Standard error

Part I
The Model-Free Prediction Principle

Chapter 1
Prediction: Some Heuristic Notions

1.1 To Explain or to Predict?

Statistics is the scientific discipline that enables us to draw inferences about the real world on the basis of observed data. Statistical inference comes in two general flavors:

A. **Explaining/modeling the world**. Here the role of the statistician is much like the role of a natural scientist trying to find how the observed/observable quantity Y depends on another observed/observable quantity X. Natural scientists may have additional information at their disposal, e.g., physical laws of conservation, etc., but the statistician must typically answer this question solely on the basis of the data at hand. Nonetheless, insights provided by the science behind the data can help the statistician formulate a better model.
In a question asked this way, Y is called the *response* variable, and X is called a *regressor* or *predictor* variable. The data are often pairs: $(Y_1,X_1),(Y_2,X_2),\ldots,(Y_n,X_n)$, where Y_i is the measured response associated with a regressor value given by X_i. Figure 1.1a shows an example of a *scatterplot* associated with such a dataset.
The goal is to find a so-called *regression* function, say $\mu(\cdot)$, such that $Y \approx \mu(X)$. The relation $Y \approx \mu(X)$ is written as an approximation because either the association between X and Y is not exact and/or the observation of X and Y is corrupted by measurement error. The inexactness of the association and the possible measurement errors are combined in the discrepancy defined as $\varepsilon = Y - \mu(X)$; the last equation can then be re-written as:

$$Y = \mu(X) + \varepsilon. \tag{1.1}$$

In the above, $\mu(X)$ is the part of Y that is "explainable" by X, and ε is an unexplainable, error term.

D.N. Politis, *Model-Free Prediction and Regression*, Frontiers in Probability and the Statistical Sciences, DOI 10.1007/978-3-319-21347-7_1

Is Eq. (1.1) a *model*? Not yet. It becomes a model if/when it is complemented by an assumed structure for the stochastic nature of the error term. A typical such assumption is that $\varepsilon_1, \ldots, \varepsilon_n$ are independent, identically distributed (i.i.d.) random variables with mean zero, where $\varepsilon_i = Y_i - \mu(X_i)$. For example, without the mean zero assumption on the errors, *any* function could equally serve as the $\mu(\cdot)$ appearing in Eq. (1.1).

In addition to assumptions on the error term ε, typical model assumptions specify the allowed "type" for function $\mu(\cdot)$. This is done by specifying a family of functions, say $\mathscr{F}$, and insisting that $\mu(\cdot)$ must belong to $\mathscr{F}$. If $\mathscr{F}$ is finite-dimensional, then it is called a *parametric* family. A popular two-dimensional example corresponds to

$$\mathscr{F} = \{\text{all } \mu(\cdot) \text{ such that } \mu(x) = \beta_0 + \beta_1 x \text{ for two real numbers } \beta_0, \beta_1\} \quad (1.2)$$

which is the usual straight-line regression with slope β_1 and intercept β_0. If $\mathscr{F}$ is not finite-dimensional, then it is called a *nonparametric* (sometimes also called infinite-parametric) family. For instance, $\mathscr{F}$ could be the family of all functions that are (say) twice continuously differentiable over their support.

Under such model assumptions, the task of the statistician is to use the available data $\{(Y_i, X_i),\ i = 1, \ldots, n\}$ in order to (a) optimally estimate the function $\mu(\cdot)$, and (b) to quantify the statistical/stochastic accuracy of the estimator.

Part (a) can be accomplished after formulating an appropriate optimality criterion; the oldest and most popular such criterion is *Least Squares (LS)*. The LS estimator of $\mu(\cdot)$ is the function, say $\hat{\mu}(\cdot)$, that minimizes the sum of squared errors $\sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n (Y_i - \mu(X_i))^2$ among all $\mu(\cdot) \in \mathscr{F}$. If $\mathscr{F}$ happens to be the two-parameter family of straight-line regression functions, then it is sufficient to obtain LS estimates, say $\hat{\beta}_0$ and $\hat{\beta}_1$, of the intercept and slope β_0 and β_1, respectively. Under a correctly specified model, the LS estimators $\hat{\beta}_0$ and $\hat{\beta}_1$ have minimum variance among all unbiased estimators that are linear functions of the data.

To address part (b) before the 1980s statisticians often resorted to further restrictive model assumptions such as an (exact) Gaussian distribution for the errors ε_i. Fortunately, the bootstrap and other computer-intensive methods have rendered such unrealistic/unverifiable assumptions obsolete; see, e.g., Efron (1979), Efron and Tibshirani (1993), Politis (1998), or Politis et al. (1999).

To fix ideas, consider a toy example involving $n = 20$ patients taken from a group of people with borderline high blood pressure, i.e., (systolic) blood pressure of about 140. A drug for lowering blood pressure may be under consideration, and the question is to empirically see how (systolic) blood pressure Y corresponds to dosage X where the latter is measured as units of the drug taken daily. The Y responses were as follows: $(145, 148, 133, 137)$ for $X = 0$; $(140, 132, 137, 128)$ for $X = 0.25$; $(123, 131, 118, 125)$ for $X = 0.5$; $(115, 118, 120, 126)$ for $X = 1$; and $(108, 115, 111, 112)$ for $X = 2$. Figure 1.1b shows the scatterplot of the 20 data pairs (Y_i, X_i), one for each patient, having superimposed both the LS straight-line regression function (with estimated intercept and slope equal to 136.5 and -13.7, respectively), as well as an estimated nonparametric regression function based on a smoothing spline, i.e., piecewise cubic function, under the assumption that the true function $\mu(\cdot)$ is smooth, e.g., twice continuously differentiable.

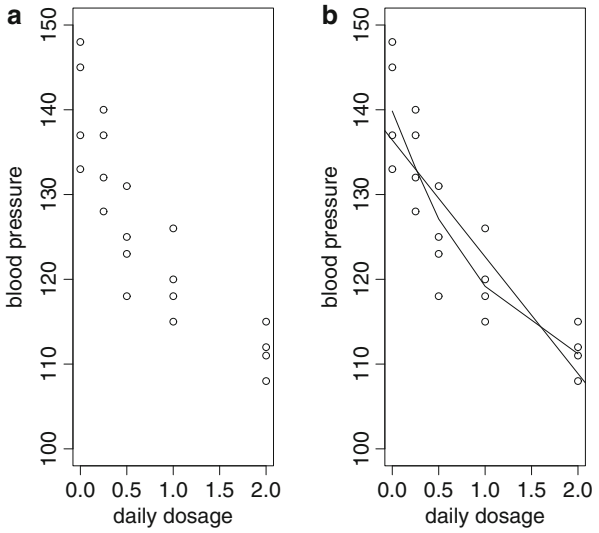

Fig. 1.1 (**a**) Systolic blood pressure vs. daily dosage of a drug. (**b**) Same data with superimposed straight-line regression as well as nonparametric regression based on a smoothing spline

B. **Predicting a future state of the world**. In contrast to trying to model how Y depends on X, one can instead focus on the pragmatic question: what can I say about a future response value $Y_{\rm f}$ that will be associated with the regressor $X_{\rm f}$ taking on a potential value equal to $x_{\rm f}$? For example, with respect to the scatterplot of Fig. 1.1, can one predict the response, i.e., blood pressure, of a patient from the same group who is prescribed a daily dose equal to $x_{\rm f} = 1.5$ units of the drug? The goal now is to find a function, say $\nu(\cdot)$, such that $\nu(X_{\rm f})$ is a good (in some sense) predictor of $Y_{\rm f}$ on the basis of $X_{\rm f}$. The statistician's task now is to (a) identify the optimal (with respect to some criterion) prediction function $\nu(\cdot)$ among a family of allowed functions, say $\mathscr{F}$, and (b) to quantify the statistical/stochastic accuracy of the predictor $\nu(X_{\rm f})$. A popular optimality criterion is minimum Mean Squared Error (MSE) of prediction which can be defined either conditionally or unconditionally. For concreteness, in what follows we will focus on the conditional MSE of prediction given by:

$$\mathrm{MSE}_{x_{\rm f}}^{\rm pred} = E\left((Y_{\rm f} - \nu(X_{\rm f}))^2 | X_{\rm f} = x_{\rm f}\right). \tag{1.3}$$

If the modeling step A has been already accomplished, then the prediction problem appears to immediately be solvable by simply taking $\nu(\cdot) = \mu(\cdot)$. This has been the *modus operandi* in the last 100 or so years: first construct/fit a model, and then use the fitted model for prediction. Nevertheless, the **Model-Free Prediction Principle** to be expounded upon in Chap. 2 shows how prediction can be accomplished in a direct way, without the intermediate step of model-fitting. In so doing, Model-Free Prediction restores the emphasis on observable quantities, i.e., current and future

data, as opposed to unobservable model parameters and estimates thereof. In this sense, the Model-Free prediction principle is in concordance with Bruno de Finetti's statistical philosophy; see, e.g., Dawid (2004) and the references therein. Interestingly, being able to predict the response Y_f associated with the regressor X_f taking on *any* possible value (say x_f) seems to inadvertently also achieve the aforementioned main goal of modeling/explaining the world, i.e., trying to find how Y depends on X.

1.2 Model-Based Prediction

In addition to qualitative/philosophical issues, e.g., disproportionate focus on unobservable parameters, the traditional approach of prediction following (and based on) a preliminary step of model-fitting faces several major difficulties:

(i) **All models are wrong**. This extreme statement was made more than once by one of the pioneers of statistical model-building, George Box—see Box (1976, p. 792). He later revised this to his more famous statement *"essentially, all models are wrong, but some are useful"* which, incidentally, is now the domain of a webpage; see Box and Draper (1987, p. 424). The underlying philosophical notion is that it may be *hybris* to expect that we can capture the *exact* workings and complications of the real world in a simple mathematical equation; indeed, this appears to be true with the exception of a handful of elegant laws of nineteenth century physics.
Leaving philosophy aside, statistical model-building involves a back-and-forth interaction of the statistician with the data. Box (1976) describes this process in detail which roughly goes as follows:

- The practitioner adopts a tentative model as being "true."
- Under the assumption of the "true" model, optimal estimation/fitting is carried out.
- Finally, diagnostics on the "correctness" of the model are carried out which may point to weaknesses and necessary modifications to the assumed model.

The implication is that model-building is always in a state of flux. The final model one settles on is not necessarily the *true* one; it is just a model for which no apparent problems manifest.
For example, consider the straight-line regression model of Fig. 1.1b. With a negative slope (estimated or true), it is apparent that for a high enough value for the dosage, the model will predict a *negative* value for the blood pressure response. This is highly problematic because blood pressure is a non-negative quantity; hence, the straight-line model must be wrong, or at least its range of applicability must be limited to a narrow region of X–values. To fix this problem, a statistician may employ a logarithmic transformation of the

response,[1] e.g., regress $\log Y$ on X, or venture into a Generalized Linear Model; either option would entail a modification of the original model which may itself lead to a further modification, and so on and so forth.

(ii) **Optimal methods for model-fitting are often not robust**. Typical textbook treatments of statistical problems have the following structure: (a) assume a particular model as being true, and (b) identify the optimal statistical procedures under the assumed model. For the most part, published research in classical mathematical statistics has been patterned in a similar way.
However, from the previous discussion it should be apparent that one cannot expect that any model is exactly true. Hence, it was recognized in the 1970s that a good statistical procedure for practical use should not break down under a small deviation from model assumptions. These ideas led to the creation of the sub-field of *Robust Statistics* that deals with estimators that are robust against model misspecification as well as possible outliers; see the early papers by Hampel (1973) and Huber (1973) and the recent monograph by Maronna et al. (2006) and the references therein.
Furthermore, the computational implementation of a good statistical procedure must be devoid of problems otherwise the procedure will be of little use. To give an example of a problematic situation, consider the popular ARCH/GARCH models for financial time series of Engle (1982) and Bollerslev (1986) that are expounded upon in Part IV of the book. The most popular of these models in practice is the GARCH(1,1) that involves four unknown parameters (including a parameter that quantifies the degree of heavy tails for the errors). To achieve optimality in estimation, such parametric models are commonly fitted by Maximum Likelihood Estimation (MLE), i.e., the likelihood function is maximized over the four free parameters. Unfortunately, for the ARCH/GARCH family there is no closed-form solution for the maximizer; thus, the MLEs are found by numerical optimization which is not entirely stable unless the sample size is very large—see Sect. 10.7 for an illustration.

(iii) **Can one perform optimal prediction using the wrong model?** In short, yes. To elaborate, consider the model of Eq. (1.1) with the ε_i assumed i.i.d. with mean zero and variance σ^2. Optimal model-fitting is tantamount to optimal estimation of the function $\mu(\cdot)$ that is assumed to belong to family $\mathscr{F}$. To simplify the problem, let us focus on optimal estimation of $\mu(x_f)$, where x_f is some particular point of interest. We can then define an optimal estimator $\hat{\mu}(x_f)$ as the minimizer of the MSE of estimation defined as

$$\mathrm{MSE}^{\mathrm{est}}_{x_f} = E\left(\hat{\mu}(x_f) - \mu(x_f)\right)^2 = \mathrm{Bias}^2\left(\hat{\mu}(x_f)\right) + \mathrm{Var}\left(\hat{\mu}(x_f)\right)$$

where $\mathrm{Bias}\left(\hat{\mu}(x_f)\right) = E\hat{\mu}(x_f) - \mu(x_f)$ and $\mathrm{Var}\left(\hat{\mu}(x_f)\right) = E\left[\hat{\mu}(x_f) - E\hat{\mu}(x_f)\right]^2$.

[1] Interestingly, for the dataset of Fig. 1.1, straight-line regression of $\log Y$ on X gives a fitted curve that is almost identical to the straight-line regression of Y on X so long as X is in the interval [0,2]; the difference between the two models becomes pronounced only for large X.

Interestingly, under an *additive* model such as Eq. (1.1) with i.i.d. errors having variance σ^2, the MSE of estimation is closely related to the (conditional) MSE of prediction defined in Eq. (1.3), namely

$$\mathrm{MSE}_{x_\mathrm{f}}^{\mathrm{pred}} = \sigma^2 + \mathrm{MSE}_{x_\mathrm{f}}^{\mathrm{est}}$$

provided we use the estimated regression function $\hat{\mu}(\cdot)$ for prediction, i.e., we set $\hat{v}(\cdot) = \hat{\mu}(\cdot)$. Thus, it appears that MSE–optimality in fitting/estimating the model (1.1) is tantamount to MSE–optimality for prediction.

Perhaps surprisingly, *both goals*—estimating $\mu(x_\mathrm{f})$ and predicting Y_f—can sometimes be better achieved using the wrong model. To see why, consider the dataset of Fig. 1.1a; there appears to be a small curvature in the regression function that is captured by the smoothing spline in Fig. 1.1b. The key observation is that if the parameter associated with the curvature is in truth very small, and if the variance associated with estimating it is large, one might be better off omitting the curvature term *even though* it may well belong to the true model; this would entail admitting some bias in order to get significant reduction in variance, resulting into smaller $\mathrm{MSE}_{x_\mathrm{f}}^{\mathrm{pred}}$ and $\mathrm{MSE}_{x_\mathrm{f}}^{\mathrm{est}}$ as compared to using the true model!

As it turns out, there are often quantifiable benefits in using simplified, e.g., *under-fitted*, models for prediction as opposed to using the "true" model even in the unrealistic case when the latter is known. This possibility was pointed out more than 30 years ago by Hocking (1976); see Wu et al. (2007) for an expounded treatment including precisely formulated conditions on when underfitting a regression is optimal for the purpose of prediction.

For instance, consider again the dataset of Fig. 1.1; in such a practical situation, the true model is not known and the practitioner must play the "what if" game: *what if* $\mu(\cdot)$ *were a quadratic, and I fitted a straight-line regression?* In this case, one must also fit a quadratic, and compare the two models. In the traditional approach, the practitioner would include a quadratic term only if the associated estimated coefficient was found to be statistically significant. For the blood pressure dataset, the quadratic coefficient is indeed statistically significant at the 95 % level—but not at the 99 % one.

Figure 1.2a shows a plot of the quadratic which indeed fits the given data quite well; but is the quadratic better for prediction? To appreciate the dangers of over-parameterization, Fig. 1.2b gives a plot of the quadratic predictor using an extended range for the regressor. The implication of adopting the quadratic model is the conclusion that the drug works in reducing blood pressure for dosages up to 2 units daily but that it would actually *increase* blood pressure when the dosage is increased any further; this is counter-intuitive, and certainly not justified based on the data at hand. Notably, this quadratic passes the aforementioned checks given by Eqs. (30)—(32) of Wu et al. (2007), suggesting that a straight-line regression is not recommended here; this shows the caveats in trying to use these conditions in practical work.

Of course, what does work in practice is to abandon the preconception of a "true" model, and try to find instead the most parsimonious model that fits the data well, i.e., perform *model selection*; see, e.g., the comprehensive treatment in Hastie et al. (2009). Interestingly, modern methods for model selection are increasingly based on the concept of cross-validation which, in its essence, is associated with the quality of *prediction* as opposed to model-fitting.

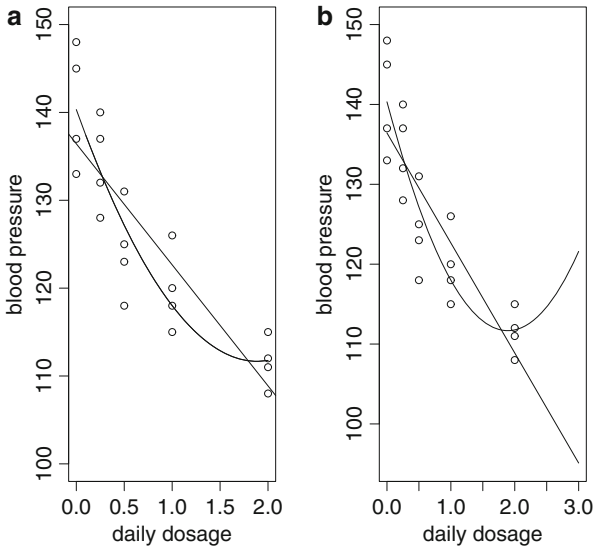

Fig. 1.2 (**a**) Systolic blood pressure vs. daily dosage with superimposed straight-line regression and the LS quadratic fit. (**b**) Same as (**a**) but considering possible dosages up to 3 units of the drug

1.3 Model-Free Prediction

Section 1.2 outlined some perils associated with prediction based on a preliminary step of model-fitting. By contrast, Model-free prediction by-passes the modeling "middle-man" and goes straight to solving the problem at hand, namely prediction of an observable quantity of interest. The essence of the **Model-Free Prediction Principle** that is elaborated upon in Chap. 2 can be heuristically described as follows.

Consider the regression setup with a vector of observed responses $\underline{Y}_n = (Y_1, \ldots, Y_n)'$ that are associated with the vector of regressors $\underline{X}_n = (X_1, \ldots, X_n)'$. Also consider the enlarged vectors $\underline{Y}_{n+1} = (Y_1, \ldots, Y_n, Y_{n+1})'$ and $\underline{X}_{n+1} = (X_1, \ldots, X_n, X_{n+1})'$ where (Y_{n+1}, X_{n+1}) is an alternative notation for (Y_f, X_f); recall that Y_f is yet unobserved, and X_f will be set equal to the value x_f of interest. If the Y_is were i.i.d. (and

not depending on their associated X value), then prediction would be trivial: the MSE–optimal predictor of Y_{n+1} is simply given by the mean, i.e., expected value, of the i.i.d. Y_is completely disregarding the value of X_{n+1}.

In a nutshell, the Model-Free Prediction Principle amounts to using the structure of the problem in order to **find an invertible transformation H_m that can map the non-i.i.d. vector $\underline{Y}_m$ to a vector $\underline{\varepsilon}_m = (\varepsilon_1, \ldots, \varepsilon_m)'$ that has i.i.d. components;** here m could be taken equal to either n or $n+1$ as needed. Letting H_m^{-1} denote the inverse transformation, we have $\underline{\varepsilon}_m = H_m(\underline{Y}_m)$ and $\underline{Y}_m = H_m^{-1}(\underline{\varepsilon}_m)$, i.e.,

$$\underline{Y}_m \overset{H_m}{\longmapsto} \underline{\varepsilon}_m \quad \text{and} \quad \underline{\varepsilon}_m \overset{H_m^{-1}}{\longmapsto} \underline{Y}_m. \tag{1.4}$$

If the practitioner is successful in implementing the Model-Free procedure, i.e., in identifying the transformation H_m to be used, then the prediction problem is reduced to the trivial one of predicting i.i.d. variables. To see why, note that Eq. (1.4) with $m = n+1$ yields $\underline{Y}_{n+1} = H_{n+1}^{-1}(\underline{\varepsilon}_{n+1}) = H_{n+1}^{-1}(\underline{\varepsilon}_n, \varepsilon_{n+1})$. But $\underline{\varepsilon}_n$ can be treated as known (and constant) given the data $\underline{Y}_n$; just use Eq. (1.4) with $m = n$. Since the unobserved Y_{n+1} is just the $(n+1)^{th}$ coordinate of vector $\underline{Y}_{n+1}$, we have just expressed Y_{n+1} as a function of the unobserved ε_{n+1}. Note that predicting a function, say $g(\cdot)$, of an i.i.d. sequence $\varepsilon_1, \ldots, \varepsilon_n, \varepsilon_{n+1}$ is straightforward because $g(\varepsilon_1), \ldots, g(\varepsilon_n), g(\varepsilon_{n+1})$ is simply another sequence of i.i.d. random variables. Hence, the practitioner can use this simple structure to develop point predictors and prediction intervals for the future response Y_{n+1}.

Under regularity conditions, such a transformation H_m always exists although it is not unique. The challenge to the skills and expertise of the statistician is to be able to devise a workable such transformation for the problem of interest. The monograph at hand is devoted to outlining how to put the Model-free prediction principle to good use in some key problems in statistics, including regression and time series analysis, and to hopefully pave the road for further such applications.

As previously mentioned, the Model-Free Prediction Principle places full emphasis on observable quantities, i.e., current and future data, as opposed to unobservable model parameters and estimates thereof. However, being able to predict the response Y_f associated with the regressor X_f taking on *any* possible value seems to inadvertently also achieve the main goal of modeling, i.e., trying to find how Y depends on X. As a consequence, a practitioner can use Model-Free Prediction ideas in order to obtain point estimates and confidence intervals for relevant parameters if so desired; this may help justify the oxymoron *"Model-free model-fitting"* in the title of the paper by Politis (2013). In other words, as prediction can be treated as a by-product of model-fitting, some key estimation problems can be solved as a by-product of being able to perform prediction.

In anticipation of the detailed discussion on the setup of regression in Part II of the book, it should be mentioned that devising transformations in regression has always been thought to be a crucial issue that received attention by statistics pioneers such as F. Anscombe, M.S. Bartlett, R.A. Fisher, etc.; see the excellent exposition of DasGupta (2008, Chap. 4) and the references therein, as well as Draper and Smith (1998, Chap. 13), Atkinson (1985), and Carroll and Ruppert (1988).

Regarding nonparametric regression in particular, the power family of Box and Cox (1964) has been routinely used in practice, as well as more elaborate, computer-intensive transformation techniques. Of the latter, we single out the ACE algorithm of Breiman and Friedman (1985), and the AVAS transformation of Tibshirani (1988). Both ACE and AVAS are very useful for transforming the data in a way that the usual additive nonparametric regression model is applicable with AVAS also achieving variance stabilization. Notably, as will be apparent in Chap. 4, the Model-free approach to nonparametric regression is *insensitive* to whether such preprocessing by Box/Cox, ACE or AVAS has taken place. Consequently, the Model-free practitioner is relieved from the need to find an optimal transformation; thus, Model-free Model-fitting in regression is a totally *automatic* technique.

To recapitulate, a practitioner can use the Model-Free Prediction Principle to obtain point and interval predictors in regression and other situations with complex datasets, e.g., time series and random fields. In addition, the Model-Free approach can be used to construct point and interval estimators for some parameters of interest as well, e.g., the regression function $\mu(\cdot)$; see, e.g., Chap. 5 and Sect. 8.8 for examples. Since a hypothesis test can be performed by inverting a confidence interval, the Model-Free Prediction Principle is seen to give a complete approach to statistical inference that offers an alternative to the classical treatment.

Acknowledgements

The title of Sect. 1.1 was inspired by the interesting paper of Shmueli (2010).

Chapter 2
The Model-Free Prediction Principle

2.1 Introduction

In the classical setting of an i.i.d. (independent and identically distributed) sample, the problem of prediction is not very interesting. Consequently, practitioners have mostly focused on estimation and hypothesis testing in this case. However, when the i.i.d. assumption no longer holds, the prediction problem is both important and intriguing. Typical examples where the i.i.d. assumption breaks down include regression and time series analysis.

Two key models are given below.

- **Regression**

$$Y_t = \mu(\underline{x}_t) + \sigma(\underline{x}_t)\,\varepsilon_t \quad \text{for } t = 1,\ldots,n. \tag{2.1}$$

- **Time series**

$$Y_t = \mu(Y_{t-1},\cdots,Y_{t-p};\underline{x}_t) + \sigma(Y_{t-1},\cdots,Y_{t-p};\underline{x}_t)\,\varepsilon_t \quad \text{for } t = 1,\ldots,n. \tag{2.2}$$

Here, $Y_1,\ldots,Y_n$ are the data, ε_t are the errors which are assumed[1] i.i.d. $(0,1)$, and $\underline{x}_t$ is a fixed-length vector of explanatory (predictor) variables associated with the observation Y_t. The functions $\mu(\cdot)$ and $\sigma(\cdot)$ are unknown but assumed to belong to a class of functions that is either finite-dimensional (parametric family) or not; the latter case is the usual nonparametric setup in which case the functions $\mu(\cdot)$ and $\sigma(\cdot)$ are typically assumed to belong to a smoothness class.

Given one of these two models, the optimal *model-based* predictors of a future Y-value can be constructed. Nevertheless, the prediction problem can, in principle, be carried out in a fully model-free setting, offering—at the very least—robustness against model misspecification. Perhaps more importantly, the Model-free viewpoint gives a different way to approach standard problems, leading to new insights.

[1] The notation i.i.d. (μ,σ^2) is shorthand for i.i.d. with mean μ and variance σ^2.

D.N. Politis, *Model-Free Prediction and Regression*, Frontiers in Probability and the Statistical Sciences, DOI 10.1007/978-3-319-21347-7_2

In this chapter, we identify the underlying principles and elements of Model-free prediction that apply equally to cases where the breakdown of the i.i.d. assumption is either due to non-identical distributions, i.e., the regression example (2.1), and/or due to dependence in the data as in example (2.2). In the following sections, these general principles for Model-free prediction are theoretically formulated; their essence is based on the notion of transforming a given setup into one that is easier to work with, e.g., i.i.d. or Gaussian. Being i.i.d., the transformed data are amenable to resampling; indeed, the Model-free prediction principle can be combined with the bootstrap to yield frequentist predictive distributions in a very general framework.

2.2 Model-Free Approach to Prediction

2.2.1 Motivation: The i.i.d. Case

As already mentioned, the prediction problem is most interesting in cases where the i.i.d. assumption breaks down. However, we briefly outline the i.i.d. case in order to motivate the more general results.

Consider real-valued data $Y_1, \ldots, Y_n$ i.i.d. from the (unknown) distribution F_Y. The goal is prediction of a future value Y_{n+1} based on the data. It is apparent that F_Y *is* the predictive distribution, and its quantiles could be used to form predictive intervals. Furthermore, different measures of center of location of the distribution F_Y can be used as (point) predictors of Y_{n+1}. In particular, the mean and median of F_Y are of interest since they represent optimal predictors under an L_2 and L_1 loss criterion, respectively.

Of course, F_Y is unknown but can be estimated by the empirical distribution of the data $Y_1, \ldots, Y_n$, denoted by $\hat{F}_Y$, that has favorable consistency properties. Hence, the L_2 and L_1 optimal predictors can be well approximated by the mean and median of $\hat{F}_Y$, respectively. Furthermore, simple Model-free predictive intervals could also be based on quantiles of $\hat{F}_Y$. Such intervals will have asymptotically correct coverage level but are expected to exhibit under-coverage in finite samples as the variability of the estimated quantiles is ignored; see Sect. 2.4 for an elaboration.

2.2.2 The Model-Free Prediction Principle

In general, the data $\underline{Y}_n = (Y_1, \ldots, Y_n)'$ may not be i.i.d. so the predictive distribution of Y_{n+1} given the data may depend on $\underline{Y}_n$ and on $\mathbf{X}_{n+1}$ which is an array of observable, explanatory (predictor) variables. The notation $\mathbf{X}_n$ here is cumulative, i.e., $\mathbf{X}_n$ is the collection of all predictor variables associated with the data $\{\underline{Y}_t$ for $t = 1, \ldots, n\}$; in the regression example of Eq. (2.1), the array $\mathbf{X}_n$ would be formed by concatenating together all the (fixed-length) predictor vectors $\underline{x}_t, t = 1, \ldots, n$. The predictors can be deterministic or random; in the latter case, prediction (and regression) will be carried out *conditionally* on $\mathbf{X}_{n+1}$.

Let Y_t take values in the linear space $\mathbf{B}$ which often will be $\mathbf{R}^d$ for some integer d. The goal is to predict $g(Y_{n+1})$ based on $\underline{Y}_n$ and $\mathbf{X}_{n+1}$ *without* invoking any particular model; here g is some real-valued (measurable) function on $\mathbf{B}$. The key to successful Model-free prediction is the following **Model-free Prediction Principle** that was first presented in the extended abstract of Politis (2007b), and expounded upon in Politis (2013). Intuitively, the basic idea is to transform the non-i.i.d. setup to an i.i.d. dataset for which prediction is easy—even trivial— and then transform back.

Model-Free Prediction Principle

(a) *For any integer $m \geq$ some m_0, suppose that a transformation H_m is found that maps the data $\underline{Y}_m = (Y_1, \ldots, Y_m)'$ onto the vector $\underline{\varepsilon}_m^{(m)} = (\varepsilon_1^{(m)}, \ldots, \varepsilon_m^{(m)})'$ where the $\{\varepsilon_i^{(m)}, i = 1, \ldots, m\}$ are* i.i.d. *with distribution F_m satisfying*

$$F_m \stackrel{\mathscr{L}}{\Longrightarrow} F \text{ as } m \to \infty; \tag{2.3}$$

in the above, F is some limit distribution, and $\stackrel{\mathscr{L}}{\Longrightarrow}$ denotes convergence in law. The transformation H_m may depend on the structure of the problem as well as the explanatory variables $\mathbf{X}_m$.

(b) *Suppose that the transformation H_m is invertible for all $m \geq m_0$ (possibly modulo some initial conditions denoted by IC), and—in particular—that one can solve for Y_m in terms of $\underline{Y}_{m-1}, \mathbf{X}_m$, and $\varepsilon_m^{(m)}$ alone, i.e., that*

$$Y_m = g_m(\underline{Y}_{m-1}, \mathbf{X}_m, \varepsilon_m^{(m)}) \tag{2.4}$$

and

$$\underline{Y}_{m-1} = f_m(\underline{Y}_{m-2}, \mathbf{X}_{m-1};\ \varepsilon_1^{(m)}, \ldots, \varepsilon_{m-1}^{(m)};\ IC) \tag{2.5}$$

for some functions g_m and f_m and for all $m \geq m_0$.

(c) *From the above, a* predictive equation *for the unobserved Y_{n+1} can be formed:*

$$Y_{n+1} = g_{n+1}(\underline{Y}_n, \mathbf{X}_{n+1}, \varepsilon_{n+1}^{(n+1)}). \tag{2.6}$$

Hence, the L_2-optimal Model-free predictor of $g(Y_{n+1})$ given the data $\underline{Y}_n$ and the predictors $\mathbf{X}_{n+1}$ is obtained by the (conditional) expectation

$$\int G_{n+1}(\underline{Y}_n, \mathbf{X}_{n+1}, \varepsilon) dF_{n+1}(\varepsilon) \tag{2.7}$$

where $G_{n+1} = g \circ g_{n+1}$ and $\circ$ denotes composition of functions.

(d) *The* predictive distribution *of $g(Y_{n+1})$ is given by the (conditional) distribution of the random variable $G_{n+1}(\underline{Y}_n, \mathbf{X}_{n+1}, \varepsilon_{n+1})$ where ε_{n+1} is drawn from distribution F_{n+1} and is independent of $\underline{Y}_n$. The median of this predictive distribution yields the L_1-optimal Model-free predictor of $g(Y_{n+1})$ given $\underline{Y}_n$ and $\mathbf{X}_{n+1}$.*

Parts (c) and (d) above outline a general approach to the problem of prediction of (a function of) Y_{n+1} given a dataset of size n. As will be apparent in the sequel, the application of Model-free prediction hinges on the aforementioned transformation H_m and its inverse for $m = n$ and $m = n + 1$; see, e.g., the predictive equation (2.6).

Remark 2.2.1 The predictive distribution in part (d) above is understood to be *conditional* on the value of $\underline{Y}_n$ (and the value of $\mathbf{X}_{n+1}$ when the latter is random); the same is true for the expectation in part (c). Note also the tacit understanding that the "future" ε_{n+1} is independent to the conditioning variable $\underline{Y}_n$; this assumption is directly implied by Eq. (2.5) which itself—under some assumptions on the function g_m—could be obtained by iterating (back-solving) Eq. (2.4). The presence of initial conditions such as *IC* in Eq. (2.5) is familiar in time series problems of autoregressive nature where *IC* would typically represent values $Y_0, Y_{-1}, \ldots, Y_{-p}$ for a finite value p; the effect of the initial conditions is negligible for large n. In regression problems, the presence of initial conditions would not be required if the regression errors can be assumed to be i.i.d. as in Eq. (2.1).

Remark 2.2.2 Equation (2.4) with $\varepsilon_i^{(m)}$ being i.i.d. from distribution F_m looks like a model equation but it is more general than a typical model. For one thing, the functions g_m and F_m may change with m, and so does $\varepsilon_i^{(m)}$ which, in essence, is a triangular array of i.i.d. random variables. Furthermore, no assumptions are made *a priori* on the form of g_m. However, the process of starting without a model, and—by this transformation technique—arriving at a model-like equation deserves the name *Model-Free Model-Fitting* (MF^2 for short).

Remark 2.2.3 The predictive distribution in part (d) above is the *true* distribution in this setup, but it is unusable as such since it depends on many potentially unknown quantities. For example, the distribution F_{n+1} will typically be unknown but it can be consistently estimated by $\hat{F}_n$, the empirical distribution of $\varepsilon_1^{(n)}, \ldots, \varepsilon_n^{(n)}$, which can then be plugged-in to compute estimates of the aforementioned (conditional) mean, median, and predictive distribution. Similarly, if the form of function g_{n+1} is unknown, a consistent estimator $\hat{g}_{n+1}$ that is based on the data $\underline{Y}_n$ (and the regressors $\mathbf{X}_n$) could be plugged-in. The resulting empirical estimates of the (conditional) mean and median would typically be quite accurate but such a "plug-in" empirical estimate of the predictive distribution will be too narrow, i.e., possessing a smaller variance and/or inter-quartile range than ideal. The correct predictive distribution should incorporate the variability of estimated quantities such as $\hat{F}_n$ and $\hat{g}_{n+1}$. The only general *frequentist* way to nonparametrically capture such a widening of the predictive distribution via *resampling*; see Sect. 2.4 for more details.

Remark 2.2.4 Although the distribution F_{n+1} will typically be unknown, sometimes the large-sample distribution F of Eq. (2.3) will be known (perhaps up to estimating a finite-dimensional parameter). It is then possible to use F in place of F_{n+1} in connection with the Model-Free Prediction Principle, as well as the resampling step alluded to in Remark 2.2.3 and elaborated upon in Sect. 2.4. Using the

large-sample distribution F in this context can be denoted a **Limit Model-Free** method that combines knowledge/estimation of an asymptotic distribution with the Model-Free Prediction notions.

2.3 Tools for Identifying a Transformation Towards i.i.d.–Ness

2.3.1 Model-Free Prediction as an Optimization Problem

The task of finding a set of candidate transformations H_n for any given particular setup is challenging, and demands expertise and ingenuity. Sometimes, however, the problem itself suggests the appropriate transformation; this is the case with Model-free regression analyzed in Chap. 4, and Model-free inference for Markov processes studied in Chap. 8.

In other situations, a whole class of candidate transformations may be identified (and denoted by $\mathscr{H}$). In this case, the procedure is easy to formalize: **find the transformation $H_n \in \mathscr{H}$ that minimizes the (pseudo)distance $d(\mathscr{L}(H_n(\underline{Y}_n)), \mathscr{F}_{iid,n})$ over all $H_n \in \mathscr{H}$;** here $\mathscr{L}(H_n(\underline{Y}_n))$ is the probability law of $H_n(\underline{Y}_n)$, and $\mathscr{F}_{iid,n}$ is the space of all distributions associated with an n-dimensional random vector $\underline{Y}_n$ whose **B**–valued coordinates are i.i.d., i.e., the space of all distributions of the type $G \times G \times \cdots \times G$ where G is some distribution on space **B**. There are many choices for the (pseudo)distance $d(\cdot, \cdot)$; see, e.g., Hong and White (2005). As an example, the NoVaS transformation for financial returns developed in Chap. 10 is the outcome of optimization over a class $\mathscr{H}$ that is a parametric family of transformations.

Framed as above, the application of the prediction principle may appear similar in spirit to the Minimum Distance Method (MDM) of Wolfowitz (1957). Nevertheless, their objectives are quite different since MDM is typically employed for parameter estimation and testing whereas in the prediction paradigm parameters are of secondary importance. A typical MDM searches for the parameter $\hat{\theta}$ that minimizes the distance $d(\hat{F}_n, \mathscr{F}_\theta)$, i.e., the distance of the empirical distribution $\hat{F}_n$ to a parametric family $\mathscr{F}_\theta$. In this sense, it is apparent that MDM sets an ambitious target (the parametric family $\mathscr{F}_\theta$) but there is no necessity of actually "hitting" this target. By contrast, the prediction principle sets the minimal target of independence but its successful application requires that this minimal target is achieved (at least approximately/asymptotically).

2.3.2 Transformation into Gaussianity as a Stepping Stone

If a model such as (2.1) and (2.2) is plausible, then the model itself suggests the form of the transformation H_n, and the residuals from model-fitting would serve as the "transformed" values $\varepsilon_t^{(n)}$. Of course, the goodness of the model should now

be assessed in terms of achieved "i.i.d.–ness" of these residuals, i.e., by how close the residuals are to being i.i.d. It is relatively straightforward—via the usual graphical methods—to check that the residuals have identical distributions but checking their independence is trickier; see, e.g., Hong (1999). However, if the residuals happened to be (jointly) Gaussian, then checking their independence is easy since it is equivalent to checking for correlation, e.g., portmanteau test, Ljung-Box, etc.

This observation motivates the following variation of the Model-free prediction principle that is particularly useful in the case of dependent data; here, and for the remainder of Sect. 2.3, we will assume that $\mathbf{B} = \mathbf{R}$, i.e., that the responses Y_t are real valued.

(a′) For any $m(> m_0)$, find an invertible transformation H_m on $\mathbf{R}^m$ that maps the data $\underline{Y}_m = (Y_1, \ldots, Y_m)'$ into a Gaussian vector $\underline{W}_m^{(m)} = (W_1^{(m)}, \ldots, W_m^{(m)})'$ having covariance matrix V_m.

(b′) Use a linear transformation to map $\underline{W}_m^{(m)}$ into the i.i.d. Gaussian vector $\underline{\varepsilon}_m^{(m)} = (\varepsilon_1^{(m)}, \ldots, \varepsilon_m^{(m)})'$, and then continue with parts (c) and (d) of the Model-free prediction principle.

In applications, the linear transformation in step (b′) above may be estimated by fitting a linear model and/or by direct estimation of the covariance matrix V_n from the transformed data $W_1^{(n)}, \ldots, W_n^{(n)}$ using some extra assumption on its structure, e.g., a Toeplitz structure in stationary time series as in McMurry and Politis (2010), or an appropriate shrinkage/regularization technique as in Bickel and Li (2006); then, the estimate $\hat{V}_n$ must be extrapolated to give an estimate of V_{n+1}; see Chap. 6 for details.

Normalization as a prediction "stepping stone" can be formalized in much the same way as before. To elaborate, once $\mathscr{H}$, the set of candidate transformations is identified, the procedure is to: *choose the transformation $H_n \in \mathscr{H}$ that minimizes the (pseudo)distance $d(\mathscr{L}(H_n(\underline{Y}_n)), \Phi_n)$ over all $H_n \in \mathscr{H}$* where now Φ_n is the space of all n-dimensional Gaussian distributions. Many choices for the (pseudo)distance d are again available, including usual goodness-of-fit favorites such as the Kolmogorov-Smirnov distance or χ^2 test; a pseudo-distance based on the Shapiro-Wilk statistic is also a valid alternative here. Interestingly, in the setting of financial data, i.e., a heteroskedastic time series setup like example (2.2) with $\mu \equiv 0$ and heavy tails, Politis (2003a, 2007a) was able to achieve normalization using a kurtosis-based distance measure; see Chap. 10 for details.

Remark 2.3.1 Now that H_n is essentially a *normalizing* transformation, a collection of graphical and exploratory data analysis (EDA) tools are also available. Some of these tools include: (a) Q-Q plots of the $W_1^{(n)}, \ldots, W_n^{(n)}$ data to test for Gaussianity; (b) Q-Q plots of linear combinations of $W_1^{(n)}, \ldots, W_n^{(n)}$ to test for *joint* Gaussianity; and (c) autocorrelation plots of $\varepsilon_1^{(n)}, \ldots, \varepsilon_n^{(n)}$ to test for independence—since in the (jointly) Gaussian case, independence is tantamount to zero correlation.

2.3.3 Existence of a Transformation Towards i.i.d.–Ness

Under regularity conditions, a transformation such as H_m of part (a) of the Model-Free Prediction Principle always exists but is not necessarily unique. For example, if the variables $(Y_1,\ldots,Y_m)$ have an absolutely continuous joint distribution and no explanatory variables $\mathbf{X}_m$ are available, then the Rosenblatt (1952) transformation can map them onto a set of i.i.d. random variables with F_m being Uniform (0,1). Nevertheless, estimating the Rosenblatt transformation from data may be infeasible except in special cases. On the other hand, a practitioner may exploit a given structure for the data at hand, e.g., a regression structure, in order to construct a different, case-specific transformation that may be practically estimable from the data.

To elaborate, recall that the Rosenblatt transformation maps an arbitrary random vector $\underline{Y}_m = (Y_1,\ldots,Y_m)'$ having absolutely continuous joint distribution onto a random vector $\underline{U}_m = (U_1,\ldots,U_m)'$ whose entries are i.i.d. Uniform(0,1). This is done via the probability integral transform based on conditional distributions. For $k > 1$ define the conditional distributions $D_k(y_k|y_{k-1},\ldots,y_1) = P\{Y_k \le y_k|Y_{k-1} = y_{k-1},\ldots,Y_1 = y_1\}$, and let $D_1(y_1) = P\{Y_1 \le y_1\}$. Then the Rosenblatt transformation amounts to letting

$$U_1 = D_1(Y_1), U_2 = D_2(Y_2|Y_1), U_3 = D_3(Y_3|Y_2,Y_1),$$
$$\ldots, \text{ and } U_m = D_m(Y_m|Y_{m-1},\ldots,Y_2,Y_1). \tag{2.8}$$

The problem is that the conditional distributions D_k for $k \ge 1$ are typically unknown and must be estimated (in a continuous fashion) from the $\underline{Y}_n$ data at hand. It is apparent that unless there is some additional structure, e.g., Markovian as in Chap. 8, this estimation task may be unreliable or even infeasible for large k. As an extreme example, note that to estimate D_n we would have only one point (in n-dimensional space) to work with. Hence, without additional assumptions, the estimate of D_n would be a point mass which is a completely unreliable estimate, and of little use in terms of constructing a probability integral transform due to its discontinuity.

Remark 2.3.2 (Blocking a time series) Suppose the data $Y_1,\ldots,Y_n$ constitute a stretch of a univariate time series $\{Y_t \in \mathbf{Z}\}$. A time-honored time series technique is **blocking** the data—also known as "vectorizing" the time series; see, e.g., Bartlett (1946), Künsch (1989), and Politis and Romano (1992). 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 B_t for $t = 1,\ldots,q$, and let $D_t^{(m)}$ denote the distribution function of vector B_t which is nothing else than the m–dimensional joint marginal of $Y_t,\ldots,Y_{t+m-1}$. Using the knowledge (or a consistent estimate) of $D_t^{(m)}$, one can then use the Rosenblatt transformation (2.8) to map the vector B_t to a vector U_t having components $U_t^{(1)},\ldots,U_t^{(m)}$ that are i.i.d. Uniform (0,1). Hence, we can create Gaussian data by a further transformation, i.e., letting $Z_t^{(j)} = \Phi^{-1}(U_t^{(j)})$ for $j = 1,\ldots,m$, and $t = 1,\ldots,q$; here Φ is the cumulative distribution of a standard normal. The new vector time series $Z_t = (Z_t^{(1)},\ldots,Z_t^{(m)})'$ is

multivariate Gaussian, and standard autocorrelation-based methods can be used to handle it as discussed in Sect. 2.3.2. For example, after estimating the autocorrelation structure of $\{Z_t\}$, a further "whitening" transformation can bring us to the setup of an independent Gaussian time series where vector Z_t is independent to vector Z_s for $t \neq s$ but also the elements of vector Z_t are independent from each other. The composition of these three successive transformations gives the transformation to i.i.d.–ness as required in premise (a) of the Model-free prediction principle; see Chap. 9 for an application.

2.3.4 A Simple Check of the Model-Free Prediction Principle

Section 2.3.3 showed the existence of a transformation towards i.i.d.–ness (under regularity conditions), i.e., the viability of finding a transformation H_m satisfying premise (a) of the Model-Free Prediction Principle with $m = n$. However, for any practical implementation we would need to also satisfy premise (b) of the Model-Free Prediction Principle, and in particular Eq. (2.4) with $m = n+1$. In other words, we need to be able to express the yet unobserved Y_{n+1} as a function of the previous data and the unobserved $\varepsilon_{n+1}^{(n+1)}$. This might not be possible with a given transformation H_m (say) but may be made possible using a different transformation towards i.i.d.–ness; see Sect. 6.6.1 for an example.

The difficulty is that, in general, the variables $\varepsilon_1^{(m)}, \ldots, \varepsilon_m^{(m)})'$ constitute the mth row of a triangular array of i.i.d. data. Interestingly, if it so happens that $\varepsilon_j^{(m)} = \varepsilon_j$ does not depend on m, i.e., the variables $\varepsilon_1^{(m)}, \ldots, \varepsilon_m^{(m)})'$ constitute an i.i.d. sequence instead of a triangular array, then premise (b) of the Model-Free Prediction Principle comes for free. To see why, recall that in this case we would have $\underline{Y}_m \overset{H_m}{\longmapsto} \underline{\varepsilon}_m$ and $\underline{\varepsilon}_m \overset{H_m^{-1}}{\longmapsto} \underline{Y}_m$, where $\underline{\varepsilon}_m = (\varepsilon_1, \ldots, \varepsilon_m)'$. Letting $m = n+1$ yields $\underline{Y}_{n+1} = H_{n+1}^{-1}(\underline{\varepsilon}_{n+1}) = H_{n+1}^{-1}(\underline{\varepsilon}_n, \varepsilon_{n+1})$. But $\underline{\varepsilon}_n$ can be treated as known given the data $\underline{Y}_n$ since $\underline{\varepsilon}_n = H_n(\underline{Y}_n)$. Hence, the function g_{n+1} needed in Eq. (2.4) can be simply constructed by extracting the last coordinate of the vector $H_{n+1}^{-1}(H_n(\underline{Y}_n), \varepsilon_{n+1})$.

Fact 2.3.1 *If, for any $m(> m_0)$, the invertible transformation H_m maps the vector $\underline{Y}_m$ to vector $\underline{\varepsilon}_m = (\varepsilon_1, \ldots, \varepsilon_m)'$ where $\varepsilon_1, \ldots, \varepsilon_m$ are the first m elements of an i.i.d. sequence $\varepsilon_1, \varepsilon_2, \ldots$, then H_m also satisfies premise (b) of the Model-Free Prediction Principle.*

Interestingly, the Rosenblatt transformation (2.8) manages to map $\underline{Y}_m$ to $\varepsilon_1, \ldots, \varepsilon_m$ that are the first m elements of an i.i.d. sequence. Hence, existence of a transformation that satisfies both premises (a) and (b) of the Model-Free Prediction Principle is ensured when the data vector $\underline{Y}_n$ has an absolutely continuous joint distribution.

2.3.5 Model-Free Model-Fitting in Practice

As mentioned in Sect. 2.3.2, the task of identifying the transformation H_n for a given particular setup is expected to be challenging since it is analogous to the difficult task of identifying a good model for the data at hand, i.e., model-building. Thus, faced with a new dataset, the Model-free practitioner could/should take advantage of all the model-building know-how associated with the particular problem. The resulting "best" model can then serve as the starting point in concocting the desired transformation.

As with model-building, the candidate transformation may well depend on some unknown parameter, say θ, that may be finite-dimensional or infinite-dimensional—the latter corresponding to a "nonparametric" situation. If θ is high-dimensional, the optimization procedure outlined in Sect. 2.3.1 may be problematic. In this case, there are several potential strategies for choosing an optimal value for the parameter θ based on the data. The simplest strategy is the following.

(A) Continue with the model-building analogy, and use standard estimation techniques such as Maximum Likelihood (ML) or Least Squares (LS) when θ is finite-dimensional, or standard nonparametric/smoothing techniques when θ is infinite-dimensional.

If step (A) is not successful in rendering the transformed data i.i.d., then the strategy may be modified as follows.

(B) The parameter θ may be divided in two parts, i.e., $\theta=(\theta_1, \theta_2)$ where θ_1 is of finite (and hopefully small) dimension. Firstly, (θ_1, θ_2) are fitted using standard methods as in strategy (A). Then, using the fitted value for θ_2, a new search for θ_1 is initiated choosing the θ_1 value that renders the transformed data closest to being i.i.d.; i.e., the procedure follows the optimization procedure outlined in Sect. 2.3.1 only as far as θ_1 is concerned.

A final option involves a different kind of optimization focusing directly on the objective at hand, i.e., prediction; this appears similar to optimization techniques used in machine learning due to the focus of the latter in predictive modeling.

(C) The parameter θ is divided in three parts, i.e., $\theta = (\theta_1, \theta_2, \theta_3)$ where the first two parameters are estimable as in step (B) above; the last parameter θ_3 is a low-dimensional (hopefully univariate) parameter reserved for "fine-tuning." To proceed, construct a discrete grid $\{\theta_3^{(j)}$ for $j = 1,\ldots,J\}$ that spans the (finite) range of θ_3. For example, if θ_3 is univariate taking values in an interval $[a,b]$, then we may let $\theta_3^{(j)} = a + (b-a)j/J$ for some large enough J. For each value of $j = 1,\ldots,J$, assume θ_3 equals $\theta_3^{(j)}$ and perform step (B) above to come up with optimal values $\theta_1^{(j)}$ and $\theta_2^{(j)}$ for θ_1 and θ_2, respectively, yielding $\theta^{(j)} = (\theta_1^{(j)}, \theta_2^{(j)}, \theta_3^{(j)})$. Finally, rank the parameter choices $\{\theta^{(j)}$ for $j = 1,\ldots,J\}$ in terms of their predictive ability in premises (c) and (d) of the Model-free prediction principle (according to an L_2 or L_1 loss criterion), and choose the parameter value $\theta^{(j)}$ that ranks best; see Sect. 10.5.3 for a worked-out example.

Fortunately, in many examples the form of the desired transformation H_n is self-evident, and optimization is not needed; this is the case in the regression example whether an additive model is true (as in Chap. 3) or not (as in Chap. 4).

Remark 2.3.3 It has been noted that the Model-free (MF) approach relinquishes the notion of a model only to replace it with that of a transformation; indeed, the Model-free approach could equally be termed a **Transformation-based approach to inference**. To further elucidate the similarities and differences between the Model-free and the Model-based (MB) approaches, consider a setup where an additive model with respect to i.i.d. errors is indeed available, e.g., assume $Y_t = \mu(x_t) + \varepsilon_t$ with ε_t being i.i.d. $(0, \sigma^2)$, and $\mu(\cdot)$ an unknown function; this is the setup that will be analyzed more generally in Part II of the book. It is apparent that in order to concoct a transformation towards i.i.d.–ness, the Model-free practitioner would do well by estimating the mean $\mu(x_t)$ and subtracting it from the Y_t data. Hence, when a model with respect to i.i.d. errors is available, the Model-free practitioner may base his/her transformation on the model and thus appear to proceed in a similar way as in the model-based approach. Still, the Model-free/Transformation-based approach may offer new insights; see Chap. 3. Interestingly, the Model-Free principle appears to be more primitive than Least Squares, i.e., implying Least Squares (or even L_1 regression) under certain conditions—see Sect. 3.7.3. So using (say) a Least Squares estimator of $\mu(\cdot)$ is very much in line with the Model-free Prediction Principle. Of course, when a model is *not* available, the Model-free approach has little competition—see, e.g., Chap. 4.

2.4 Model-Free Predictive Distributions

2.4.1 Prediction Intervals and Asymptotic Validity

Statistical inference is not considered complete if it is not accompanied by a measure of its inherent accuracy. With point estimators, the accuracy is measured either by a standard error or a confidence interval. With (point) predictors, the accuracy is measured either by the predictor error variance or by the *prediction interval*. Focusing on the latter, given the data $\underline{Y}_n$ and the regressor value $\mathbf{X}_{n+1}$, the goal is to construct a prediction interval that will contain the future value $g(Y_{n+1})$ with a prespecified coverage probability. Hence the prediction interval's coverage probability should be interpreted as *conditional probability* given $\underline{Y}_n$ and $\mathbf{X}_{n+1}$ (when the latter is random).

Definition 2.4.1 *Let L_n, U_n be functions of $\underline{Y}_n$ and $\mathbf{X}_{n+1}$. The interval $[L_n, U_n]$ will be called a* $(1-\alpha)100\%$ asymptotically valid *prediction interval for $g(Y_{n+1})$ given $\underline{Y}_n$ and $\mathbf{X}_{n+1}$ if*

$$P(L_n \leq g(Y_{n+1}) \leq U_n) \to 1 - \alpha \text{ as } n \to \infty \tag{2.9}$$

for all values of $\underline{Y}_n$ and $\mathbf{X}_{n+1}$ in a set of probability one.

The probability P in (2.9) should thus be interpreted as *conditional* probability given $\underline{Y}_n$ and $\mathbf{X}_{n+1}$ although it is not explicitly denoted. Hence, property (2.9) indicates the large-sample *conditional* validity of the prediction interval $[L_n, U_n]$. All prediction intervals developed in the book will be constructed in such a way to satisfy the asymptotic validity property (2.9) under regularity conditions. Different notions of validity for prediction intervals are reviewed in Lei et al. (2013).

Asymptotic validity is a fundamental property but it does not tell the whole story. Prediction intervals are particularly useful if they can also capture the uncertainty involved in model estimation although the latter is asymptotically negligible; see Sect. 3.6 for an elaboration. Similarly, in the Model-free approach, the practitioner is estimating the requisite transformation towards i.i.d.–ness from the data at hand; this entails some variability as it is analogous to estimating/fitting a model. Ideally, this variability should be captured/incorporated in the construction of prediction intervals, and resampling gives a unique way to do just that.

2.4.2 Predictive Roots and Model-Free Bootstrap

As mentioned in Remark 2.2.3, plugging-in estimates of $\hat{F}_n$ and/or $\hat{g}_{n+1}$ in the theoretical predictive distribution defined in part (d) of the Model-Free Prediction Principle may yield an estimated predictive distribution that is too narrow, and prediction intervals that are too short resulting in *undercoverage*. The only general frequentist way to practically correct for that is via resampling. Having created the i.i.d. variables $\varepsilon_1^{(n)}, \ldots, \varepsilon_n^{(n)}$, the Model-free Prediction Principle appears ideally amenable to the i.i.d. bootstrap of Efron (1979) as will be shown in what follows.

For simplicity—and concreteness—we assume henceforth that the effect of the initial conditions IC is negligible as is, e.g., in the regression example (2.1). We will focus on constructing bootstrap prediction integrals of the **'root'** type in analogy to the well-known confidence interval construction; cf. Hall (1992), Efron and Tibshirani (1993), Davison and Hinkley (1997), or Shao and Tu (1995).

To start with, let us denote by $\Pi(g, g_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, F_{n+1})$ the optimal (with respect to either an L_1 or L_2 loss function) *point predictor* of $g(Y_{n+1})$ as obtained by the Model-free Prediction Principle. For example, in the L_2–optimal case,

$$\Pi(g, g_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, F_{n+1}) = \int G_{n+1}(\underline{Y}_n, \mathbf{X}_{n+1}, \varepsilon) dF_{n+1}(\varepsilon)$$

where $G_{n+1} = g \circ g_{n+1}$ by Eq. (2.7); note that the above integral is typically approximated by Monte Carlo. Similarly, in the L_1–optimal case, we can approximate $\Pi(g, g_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, F_{n+1})$ by Monte Carlo as the median of the set

$$\{G_{n+1}(\underline{Y}_n, \mathbf{X}_{n+1}, \varepsilon_j^*) \text{ for } j = 1, \ldots, M\}$$

where $\varepsilon_1^*, \ldots, \varepsilon_M^*$ are generated as i.i.d. from F_{n+1}, and M is some large integer.

To use predictor $\Pi(g, g_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, F_{n+1})$ in practice, it is necessary to estimate the functions g_{n+1} and F_{n+1}. The latter can be estimated by $\hat{F}_n$, i.e., the empirical distribution of $\varepsilon_1^{(n)}, \ldots, \varepsilon_n^{(n)}$ as in Remark 2.2.3. Furthermore, estimating the transformation H_n in the Model-free Prediction Principle by (say) $\hat{H}_n$, yields an estimator $\hat{g}_{n+1}$ of the prediction function g_{n+1}. Plugging-in the estimates $\hat{g}_{n+1}$ and $\hat{F}_n$ in the theoretical expression $\Pi(g, g_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, F_{n+1})$ gives $\Pi(g, \hat{g}_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n)$ which is our data-based approximation to the optimal point predictor of $g(Y_{n+1})$.

Hence, under an L_2–loss, the data-based optimal predictor $\Pi(g, \hat{g}_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n)$ is given by

$$\int g\left(\hat{g}_{n+1}(\underline{Y}_n, \mathbf{X}_{n+1}, \varepsilon)\right) d\hat{F}_n(\varepsilon) = \frac{1}{n}\sum_{j=1}^{n} g\left(\hat{g}_{n+1}(\underline{Y}_n, \mathbf{X}_{n+1}, \varepsilon_j^{(n)})\right),$$

whereas, under an L_1–loss, it is given by the median of the set

$$\{g\left(\hat{g}_{n+1}(\underline{Y}_n, \mathbf{X}_{n+1}, \varepsilon_j^{(n)})\right) \text{ for } j = 1, \ldots, n\}.$$

Definition 2.4.2 *The* **predictive root** *is the error in prediction, i.e.,*

$$g(Y_{n+1}) - \Pi(g, \hat{g}_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n). \tag{2.10}$$

Furthermore, given bootstrap data $\underline{Y}_n^* = (Y_1^*, \ldots, Y_n^*)'$ *and* Y_{n+1}^*, *the* **bootstrap predictive root** *is the error in prediction in the bootstrap world, i.e.,*

$$g(Y_{n+1}^*) - \Pi(g, \hat{g}_{n+1}^*, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n) \tag{2.11}$$

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

To elaborate, under an L_2–loss, $\Pi(g, \hat{g}_{n+1}^*, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n)$ is given by

$$\int g\left(\hat{g}_{n+1}^*(\underline{Y}_n, \mathbf{X}_{n+1}, \varepsilon)\right) d\hat{F}_n(\varepsilon) = \frac{1}{n}\sum_{j=1}^{n} g\left(\hat{g}_{n+1}^*(\underline{Y}_n, \mathbf{X}_{n+1}, \varepsilon_j^{(n)})\right),$$

whereas, under an L_1–loss, it is given by the median of the set

$$\{g\left(\hat{g}_{n+1}^*(\underline{Y}_n, \mathbf{X}_{n+1}, \varepsilon_j^{(n)})\right) \text{ for } j = 1, \ldots, n\}.$$

Remark 2.4.1 Note that Eq. (2.11) depends on the bootstrap data $\underline{Y}_n^*$ *only* through the estimated function $\hat{g}_{n+1}^*$. One might be tempted to define[2] the bootstrap predictive root as

$$g(Y_{n+1}^*) - \Pi(g, \hat{g}_{n+1}^*, \underline{Y}_n^*, \mathbf{X}_{n+1}, \hat{F}_n). \tag{2.12}$$

[2] Indeed, Politis (2013) employed definition (2.12) in a regression context; note, however, that when the responses $Y_1, Y_2, \ldots$ are independent, definitions (2.11) and (2.12) coincide since in this case the theoretical predictor $\Pi(g, g_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, F_{n+1})$ does not depend on its third argument, namely $\underline{Y}_n$, at all.

The reason for defining the bootstrap predictive root via Eq. (2.11) is to give validity to bootstrap prediction intervals *conditionally* on the data $\underline{Y}_n$; see the discussion after Definition 2.4.1.

Remark 2.4.2 It is also plausible to instead define the bootstrap predictive root as

$$g(Y^*_{n+1}) - \Pi(g, \hat{g}^*_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}^*_n) \tag{2.13}$$

where $\hat{F}^*_n$ is a bootstrap version of $\hat{F}_n$, i.e., $\hat{F}^*_n$ is the empirical distribution of random variables $\varepsilon_1, \ldots, \varepsilon_n$ generated as i.i.d. from $\hat{F}_n$. Nevertheless, in all the examples treated in this book the major source of variability is due the estimation of the function g_{n+1}; thus, definitions (2.11) and (2.13) are practically equivalent, and the former is simpler. In fact, as will be apparent in Sect. 2.4.3, plugging-in the asymptotic limit F instead of either $\hat{F}_n$ or $\hat{F}^*_n$ also gives a viable option.

Algorithm 2.4.1 MODEL-FREE (MF) BOOTSTRAP FOR PREDICTIVE DISTRIBUTION AND PREDICTION INTERVALS FOR $g(Y_{n+1})$

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., let $(\varepsilon_1^{(n)}, \ldots, \varepsilon_n^{(n)})' = \hat{H}_n(\underline{Y}_n)$. By construction, the variables $\varepsilon_1^{(n)}, \ldots, \varepsilon_n^{(n)}$ are approximately i.i.d.; let $\hat{F}_n$ denote their empirical distribution.*
   a. *Sample randomly (with replacement) the variables $\varepsilon_1^{(n)}, \ldots, \varepsilon_n^{(n)}$ to create the bootstrap pseudo-data $\varepsilon_1^*, \ldots, \varepsilon_n^*$.*
   b. *Use the inverse transformation $\hat{H}_n^{-1}$ to create pseudo-data in the Y domain, i.e., let $\underline{Y}^*_n = (Y_1^*, \ldots, Y_n^*)' = \hat{H}_n^{-1}(\varepsilon_1^*, \ldots, \varepsilon_n^*)$.*
   c. *Calculate a bootstrap pseudo-response Y^*_{n+1} as the point $\hat{g}_{n+1}(\underline{Y}_n, \mathbf{X}_{n+1}, \varepsilon)$ where ε is drawn randomly from the set $(\varepsilon_1^{(n)}, \ldots, \varepsilon_n^{(n)})$.*
   d. *Based on the pseudo-data $\underline{Y}^*_n$, re-estimate the transformation H_n and the corresponding function g_{n+1} by $\hat{H}^*_n$ and $\hat{g}^*_{n+1}$ respectively.*
   e. *Calculate a bootstrap root replicate using Eq. (2.11).*
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 $g(Y_{n+1})$ is given by*

$$[\Pi + q(\alpha/2),\ \Pi + q(1-\alpha/2)] \tag{2.14}$$

   where Π is short-hand for $\Pi(g, \hat{g}_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n)$.
5. *Finally, our Model-free estimate of the predictive distribution of $g(Y_{n+1})$ is the empirical distribution of bootstrap roots obtained in step 3 shifted to the right by the number Π; this is equivalent to the empirical distribution of the B bootstrap root replicates when the quantity Π is added to each. [Recall that the predictive*

distribution of $g(Y_{n+1})$ is—by definition—conditional on $\underline{Y}_n$ and $\mathbf{X}_{n+1}$; hence, the quantity $\Pi = \Pi(g, \hat{g}_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n)$ is a constant given $\underline{Y}_n$ and $\mathbf{X}_{n+1}$.]

Algorithm 2.4.1 is analogous to the so-called residual bootstrap schemes in model-based situations—cf. Efron (1979). The key difference is that, in the Model-free setting, the i.i.d. variables $\varepsilon_1^{(n)}, \ldots, \varepsilon_n^{(n)}$ are not residuals but the outcome of the data-transformation.

Remark 2.4.3 Note that in Step 2(c), the bootstrap pseudo-response Y_{n+1}^* is constructed based on the original data $\underline{Y}_n$—as opposed to the bootstrap data $\underline{Y}_n^*$—for the same reasons discussed in Remark 2.4.1, i.e., to ensure conditional validity of the bootstrap prediction intervals.

Using an estimate of the prediction error variance, bootstrap prediction intervals can also be constructed based on *studentized* predictive roots. However, in contrast to what happens in confidence intervals, studentization does not ensure second order accuracy of prediction intervals; see, e.g., Shao and Tu (1995, Chap. 7.3) and Remark 3.6.3. For completeness, we give some relevant details below.

Let $\hat{V}_n^2$ be an estimate of $\mathrm{Var}\left(g(Y_{n+1}) - \Pi | \underline{Y}_n, \mathbf{X}_{n+1}\right)$ where Π is short-hand for $\Pi(g, \hat{g}_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n)$ and $\hat{V}_n$ is short-hand for $V_n(g, \hat{g}_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n)$; so, its bootstrap version will be $\hat{V}_n^* = V_n(g, \hat{g}_{n+1}^*, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n)$ by analogy to Remark 2.4.1.

Definition 2.4.3 *The* **studentized predictive root** *is defined as*

$$\left(g(Y_{n+1}) - \Pi(g, \hat{g}_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n)\right) / \hat{V}_n, \tag{2.15}$$

and the **bootstrap studentized predictive root** *is defined as*

$$\left(g(Y_{n+1}^*) - \Pi(g, \hat{g}_{n+1}^*, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n)\right) / \hat{V}_n^*. \tag{2.16}$$

Algorithm 2.4.2 MODEL-FREE BOOTSTRAP BASED ON STUDENTIZED ROOTS

1–3. Same as Steps 1–3 of Algorithm 2.4.1 with one exception; step 2(e) should read: Calculate a bootstrap root replicate using Eq. (2.16).

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

$$[\Pi + q(\alpha/2)\hat{V}_n,\ \Pi + q(1-\alpha/2)\hat{V}_n]. \tag{2.17}$$

2.4.3 Limit Model-Free Resampling Algorithm

As mentioned in Remark 2.2.4, sometimes the limit distribution F appearing Eq. (2.3) may be known (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 outlined in the sequel.

The LMF data-based approximation to the optimal (with respect to either L_1 or L_2 loss) *point predictor* of $g(Y_{n+1})$ will be denoted by $\Pi(g, \hat{g}_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, F)$. To elaborate, the L_2–optimal LMF point predictor of $g(Y_{n+1})$ would be given by

$$\int g\left(\hat{g}_{n+1}(\underline{Y}_n, \mathbf{X}_{n+1}, \varepsilon)\right) dF(\varepsilon)$$

where the integral can be approximated by numerical integration or Monte Carlo methods. Similarly, the L_1–optimal LMF predictor can be approximated via Monte Carlo by the median of the set $\{g\left(\hat{g}_{n+1}(\underline{Y}_n, \mathbf{X}_{n+1}, \varepsilon_j^*)\right),\ \text{for } j = 1, \ldots, N\}$ where $\varepsilon_1^*, \ldots, \varepsilon_N^*$ are drawn i.i.d. from F, and N is some large integer.

Let $V_n^2(g, \hat{g}_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, F)$ be an estimate of $\text{Var}\left(g(Y_{n+1}) - \Pi | \underline{Y}_n, \mathbf{X}_{n+1}\right)$, where $\Pi = \Pi(g, \hat{g}_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, F)$. Then, our LMF predictive root is denoted by

$$\left(g(Y_{n+1}) - \Pi(g, \hat{g}_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, F)\right) / \hat{V}_n \tag{2.18}$$

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

$$\left(g(Y_{n+1}^*) - \Pi(g, \hat{g}_{n+1}^*, \underline{Y}_n, \mathbf{X}_{n+1}, F)\right) / \hat{V}_n^*. \tag{2.19}$$

The above covers both possibilities, studentized as well as unstudentized roots. For studentized roots, let $\hat{V}_n = V_n(g, \hat{g}_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, F)$ and $\hat{V}_n^* = V_n(g, \hat{g}_{n+1}^*, \underline{Y}_n, \mathbf{X}_{n+1}, F)$; for unstudentized roots, just let $\hat{V}_n = 1 = \hat{V}_n^*$.

Algorithm 2.4.3 LIMIT MODEL-FREE (LMF) BOOTSTRAP FOR PREDICTIVE DISTRIBUTION AND PREDICTION INTERVALS FOR $g(Y_{n+1})$

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 $\varepsilon_1^*, \ldots, \varepsilon_n^*$ 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 $\underline{Y}_n^* = (Y_1^*, \ldots, Y_n^*)' = \hat{H}_n^{-1}(\varepsilon_1^*, \ldots, \varepsilon_n^*)$.
   c. Calculate a bootstrap pseudo-response Y_{n+1}^* as the point $\hat{g}_{n+1}(\underline{Y}_n, \mathbf{X}_{n+1}, \varepsilon)$ where ε is a random draw from distribution F.
   d. Based on the pseudo-data $\underline{Y}_n^*$, re-estimate the transformation H_n and the corresponding function g_{n+1} by $\hat{H}_n^*$ and $\hat{g}_{n+1}^*$ respectively.
   e. Calculate a bootstrap root replicate using Eq. (2.19).
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 $g(Y_{n+1})$ is given by*

$$\left[\Pi + q(\alpha/2)\hat{V}_n,\ \Pi + q(1-\alpha/2)\hat{V}_n\right] \tag{2.20}$$

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

5. *Finally, our Model-free estimate of the predictive distribution of $g(Y_{n+1})$ is the empirical distribution of bootstrap roots obtained in step 3 shifted to the right by the number Π; this is equivalent to the empirical distribution of the B bootstrap root replicates when the quantity Π is added to each.*

Remark 2.4.4 Note that the forward step of the Model-free transformation, i.e., the transformation

$$\underline{Y}_n \stackrel{\hat{H}_n}{\longmapsto} \underline{\varepsilon}_n = (\varepsilon_1^{(n)}, \ldots, \varepsilon_n^{(n)})'$$

is not needed in the Limit Model-Free algorithm; thus step 2 of the algorithm is considerably simplified. In addition, note that the Limit Model-Free algorithm only requires the explicit construction of the inverse transformation $\hat{H}_m^{-1}$ with $m = n+1$; the form of the forward transformation $\hat{H}_m$ is not used at all—see Chaps. 4 and 9 for examples.

2.4.4 Prediction of Discrete Variables

The variable to be predicted, i.e., $g(Y_{n+1})$, has been assumed real-valued. However, it can be the case that $g(Y_{n+1})$ is not a continuous random variable. If the (conditional) distribution of $g(Y_{n+1})$ is mixed, i.e., it has both a continuous and a discrete part, then the Model-free bootstrap algorithms are still applicable if/when the transformation H_n and its inverse can be identified and estimated. In some cases it may be easier to estimate/identify just H_n^{-1} in which case the LMF Algorithm 2.4.3 would be more useful—see the discussion at the end of Sect. 4.3.1. However, if the (conditional) distribution of $g(Y_{n+1})$ is purely discrete, many items have to be modified.

So in this subsection, assume that $g(Y_{n+1})$ takes values in a discrete, i.e., countable, set $S \subset \mathbf{R}$. In this case, the L_1 and L_2-optimal predictors of $g(Y_{n+1})$ make little sense. More appropriate is a 0–1 loss function under which the optimal predictor of $g(Y_{n+1})$ is the **mode** of the predictive distribution of the random variable $G_{n+1}(\underline{Y}_n, \mathbf{X}_{n+1}, \varepsilon_{n+1})$ defined in premise (d) of the Model-free Prediction Principle; as before ε_{n+1} is drawn from distribution F_{n+1} and is independent of the conditioning variable $\underline{Y}_n$. In addition, having a prediction interval around such a discrete predictor is not appropriate unless the set S is of *lattice* form; even then, the problem of non-achievable α–levels ensues. Fortunately, the predictive distribution of $G_{n+1}(\underline{Y}_n, \mathbf{X}_{n+1}, \varepsilon_{n+1})$ is still very informative, and can be presented in graphical form *in lieu* of prediction intervals.

Let $\Pi(g, \hat{g}_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, F_{n+1})$ denote the abovementioned optimal data-based predictor under 0–1 loss; here, F_{n+1} would be estimated by either $\hat{F}_n$ or F under the Model-free Algorithm 2.4.1 or the Limit Model-free Algorithm 2.4.3, respectively. Hence, premise (d) of the Model-free Prediction Principle can still be used to obtain the predictive distribution of $\Pi(g, \hat{g}_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, F_{n+1})$. However, as previously mentioned, plugging-in unknown features in this predictive distribution—although often justifiable asymptotically—fails to take into account the added variance due to the estimation error of these plug-in values.

As already claimed, except in the case of the set S being lattice, taking differences makes little sense; therefore, we cannot rely on resampling predictive roots such as $g(Y_{n+1}) - \Pi(g, \hat{g}_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n)$ in order to capture the estimator variability. If roots are not to be used, then we could approximate the (conditional) distribution of $g(Y_{n+1})$ by the bootstrap distribution of $g(Y^*_{n+1})$ (without centering them) where Y^*_{n+1} is generated by either Algorithm 2.4.1 or 2.4.3. But doing so, is tantamount to a Monte Carlo implementation of premise (d) of the Model-free Prediction Principle; in other words, the variability of the estimated features of the optimal predictor is not captured. An *ad hoc* way to try to capture (some of) this variability is given by the following algorithm in which the generation of Y^*_{n+1} itself relies on a bootstrap sample, therefore the notation Y^{**}_{n+1}.

Algorithm 2.4.4 MF AND LMF BOOTSTRAP FOR PREDICTIVE DISTRIBUTION OF DISCRETE-VALUED $g(Y_{n+1})$

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., $(\varepsilon_1^{(n)}, \ldots, \varepsilon_n^{(n)})' = \hat{H}_n(\underline{Y}_n)$. Denote by $\hat{F}_n$ the empirical distribution of the (approximately) i.i.d. variables $\varepsilon_1^{(n)}, \ldots, \varepsilon_n^{(n)}$. [Note: Step 2 is not needed in the LMF case.]*
3. *a. Generate bootstrap pseudo-data $\varepsilon_1^*, \ldots, \varepsilon_n^*$ as i.i.d. from $\hat{F}_n$ (case MF) or as i.i.d. from F (case LMF).*
   b. Use the inverse transformation $\hat{H}_n^{-1}$ to create pseudo-data in the Y domain, i.e., let $\underline{Y}^*_n = (Y_1^*, \ldots, Y_n^*)' = \hat{H}_n^{-1}(\varepsilon_1^*, \ldots, \varepsilon_n^*)$.
   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-response $Y^{**}_{n+1} = \hat{g}^*_{n+1}(\underline{Y}_n, \mathbf{X}_{n+1}, \varepsilon)$ where ε is generated either from $\hat{F}_n$ (case MF) or from F (case LMF).
   e. Compute the bootstrap pseudo-value $g(Y^{**}_{n+1})$.
4. *Steps (a)—(e) in the above should be repeated a large number of times (say B times), and the B bootstrap replicates of the pseudo-values $g(Y^{**}_{n+1})$ are collected in the form of an empirical distribution; this is our Model-free estimate of the predictive distribution of $g(Y_{n+1})$ whose* mode *can serve as an alternative point predictor for $g(Y_{n+1})$.*

Algorithm 2.4.4 is suggested here only as a plan B since plan A—the bootstrap approximation of predictive roots—is not available in the case of discrete variables. Interestingly, Alonso et al. (2002) and Pascual et al. (2004) have used a version of Algorithm 2.4.4 for prediction interval construction in the setting of linear time series with continuous random variables.

A further justification for Algorithm 2.4.4 is via its analogy to Breiman's (1996) *bagging*. The difference is that in bagging, the B bootstrap series are used to construct B point predictors that are then averaged and/or otherwise aggregated in order to give a single, more stable point predictor; see Bühlmann and van de Geer

(2011) for details. In contrast, Algorithm 2.4.4 uses the B bootstrap series in order to construct B future pseudo-realizations of Y_{n+1}, and thus quantify the predictive distribution of $g(Y_{n+1})$.

Remark 2.4.5 Algorithm 2.4.4 may find application even beyond the case of discrete data. For instance, a summary measure of location—e.g., the mean/median for continuous variables or the mode for discrete—based on the empirical distribution of the B bootstrap pseudo-values $g(Y_{n+1}^{**})$ can be seen as an alternative point predictor for $g(Y_{n+1})$ that is more stable than the analogous (mean/median/mode) standard predictor of the plug-in type, i.e., $\Pi(g, \hat{g}_{n+1}, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n)$, while being intrinsically different from Breiman's (1996) aggregation of predictors.

Part II
Independent Data: Regression

Chapter 3
Model-Based Prediction in Regression

3.1 Model-Based Regression

In this chapter, we focus on the nonparametric regression model of Eq. (2.1) noting that regression is quintessential in statistical practice. For simplicity, the regressor $\underline{x}_t$ will be assumed univariate, and denoted simply as x_t. The case of a multivariate—even functional—regressor can be handled in an identical fashion although, of course, the *caveat* of the curse of dimensionality must always be born in mind; see, e.g., Appendix 1 in Chap. 4.

Thus, throughout Chap. 3 our data $\{(Y_t, x_t), t = 1, \ldots, n\}$ are assumed to have been generated by the model

$$Y_t = \mu(x_t) + \sigma(x_t)\, \varepsilon_t \text{ for } t = 1, \ldots, n \tag{3.1}$$

with ε_t being i.i.d. (0,1) from the (unknown) distribution F. The default assumption will be that the regressors x_t are deterministic. However, random regressors can easily be accommodated as long as they are independent of the errors ε_t; in this case, inference will be conducted *conditionally* on the realization of regressor values—see Sect. 4.1 for more discussion.

The functions $\mu(\cdot)$ and $\sigma(\cdot)$ are unknown but assumed to possess some degree of smoothness (differentiability, etc.). There are many approaches towards nonparametric estimation of $\mu(\cdot)$ and $\sigma(\cdot)$, e.g., wavelets and orthogonal series, smoothing splines, local polynomials, and kernel smoothers. The reviews by Altman (1992) and Schucany (2004) give concise introductions to popular methods of nonparametric regression with emphasis on kernel smoothers; book-length treatments are given by Härdle (1990), Hart (1997), Fan and Gijbels (1996), and Loader (1999). The above references focus on estimation of the conditional mean (and other moments). Regarding estimation of conditional quantiles, the book by Koenker (2005) is an

D.N. Politis, *Model-Free Prediction and Regression*, Frontiers in Probability and the Statistical Sciences, DOI 10.1007/978-3-319-21347-7_3

excellent reference, and includes a chapter on bootstrapping quantile regression estimators; see also Gangopadhyay and Sen (1990), Hahn (1995), Horowitz (1998), and Li and Racine (2007) to that effect.

For simplicity of presentation, we will focus here on nonparametric regression based on kernel smoothing. Nevertheless, it is important to stress that our predictive inference procedures can equally be implemented with *any* other appropriate regression estimator, be it of parametric or nonparametric form. For example, local linear smoothers are an attractive alternative to kernel smoothing especially when boundary issues are concerned; see Chap. 9 for an elaboration.

A popular—and very intuitive—form of a kernel smoother is the Nadaraya-Watson estimator (Nadaraya 1964; Watson 1964) defined by

$$m_x = \sum_{i=1}^{n} Y_i \tilde{K}\left(\frac{x - x_i}{h}\right) \tag{3.2}$$

where h is the bandwidth, $K(x)$ is a symmetric kernel function with $\int K(x)dx = 1$, and

$$\tilde{K}\left(\frac{x - x_i}{h}\right) = \frac{K\left(\frac{x-x_i}{h}\right)}{\sum_{k=1}^{n} K\left(\frac{x-x_k}{h}\right)}. \tag{3.3}$$

Similarly, the Nadaraya-Watson estimator of $\sigma(x)$ is given by s_x where

$$s_x^2 = M_x - m_x^2 \text{ with } M_x = \sum_{i=1}^{n} Y_i^2 \tilde{K}\left(\frac{x - x_i}{q}\right), \tag{3.4}$$

and q is another bandwidth parameter. Selection of the bandwidth parameters h and q is often done by **cross-validation**. To elaborate, let e_t denote the **fitted** residuals, i.e.,

$$e_t = (Y_t - m_{x_t})/s_{x_t} \text{ for } t = 1, \ldots, n. \tag{3.5}$$

and $\tilde{e}_t$ the **predictive** residuals, i.e.,

$$\tilde{e}_t = \frac{Y_t - m_{x_t}^{(t)}}{s_{x_t}^{(t)}}, \quad t = 1, \ldots, n \tag{3.6}$$

where $m_x^{(t)}$ and $M_x^{(t)}$ denote the estimators m_x and M_x respectively computed from the *delete-*Y_t dataset: $\{(Y_i, x_i),\ i = 1, \ldots, t-1 \text{ and } i = t+1, \ldots, n\}$. As usual, we define $s_{x_t}^{(t)} = \sqrt{M_{x_t}^{(t)} - (m_{x_t}^{(t)})^2}$. In other words, $\tilde{e}_t$ is the (standardized) error in trying to predict Y_t from the aforementioned delete-Y_t dataset.

Cross-validation amounts to picking the bandwidths h and q that minimize PRESS $= \sum_{t=1}^{n} \tilde{e}_t^2$, i.e., the PREdictive Sum of Squared residuals. PRESS is an L_2 measure that is obviously non-robust in case of heavy-tailed errors and/or outliers. For this reason, in this paper, we favor cross-validation based on an L_1 criterion. It is more robust, and is not any more computationally expensive than PRESS—see Appendix 2 for more details. L_1–cross-validation amounts to picking the bandwidths

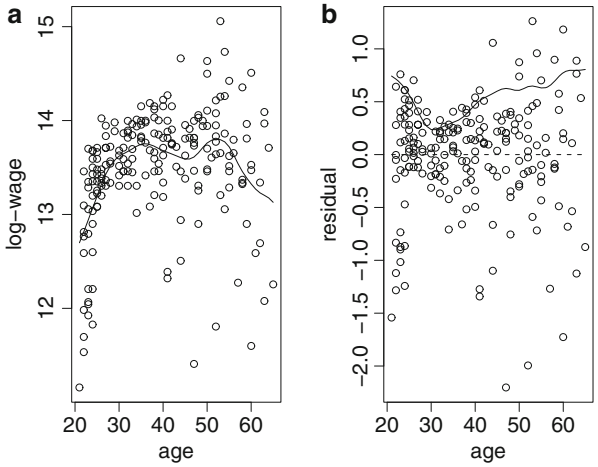

Fig. 3.1 (**a**) Log-wage vs. age data with fitted kernel smoother m_x (*solid line*). (**b**) Plot of the unstudentized residuals $Y - m_x$ with superimposed estimated standard deviation s_x (*solid line*)

that minimize $\sum_{t=1}^{n} |\tilde{e}_t|$; the latter could be denoted PRESAR, i.e., PREdictive Sum of Absolute Residuals. Note that L_1–cross-validation imposes the L_1 penalty on the predictive residuals, and thus should be distinguished from Tibshirani's (1996) Lasso that imposes an L_1 penalty on the regression parameters.

Remark 3.1.1 Rather than doing a two-dimensional search over h and q to minimize PRESS, the simple constraint $q = h$ will be imposed in what follows which has the additional advantage of rendering $M_x \geq m_x^2$ as needed for a nonnegative estimator s_x^2 in Eq. (3.4). Note, that the choice $q = h$ is not necessarily optimal; see, e.g., Wang et al. (2008). Furthermore, note that these are global bandwidths; techniques for picking *local* bandwidths, i.e., a different optimal bandwidth for each x, are widely available but will not be discussed further here in order not to obscure our main focus. Similarly, there are several recent variations on the cross-validation theme such as the one-sided cross-validation of Hart and Yi (1998), and the far casting cross-validation for dependent data of Carmack et al. (2009) that present attractive alternatives. Our discussion will focus on the well-known standard form of cross-validation for concreteness especially since our aim is to show how the Model-Free prediction principle applies in nonparametric regression with any type of kernel smoother, and any type of bandwidth selector.

Remark 3.1.2 If there are large "gaps" in the scatterplot of the data, i.e., if there are large x–regions within the range of $x_1, \ldots, x_n$ where no data are available, then a "local" bandwidth, i.e., a bandwidth that depends on x, or a *k–nearest neighbor* technique may be used; see, e.g., Li and Racine (2007, Chap. 14).

As a running example in Chaps. 3 and 4, we will use the Canadian high-school graduate earnings data from the 1971 Canadian Census; this is a wage vs. age dataset concerning 205 male individuals with common education (13th grade). The data are

available under the name `cps71` within the `np` package of R, and are discussed in Pagan and Ullah (1999). Figure 3.1a presents a scatterplot of the data with the fitted kernel estimator m_x superimposed using a normal kernel. The kernel smoother seems to be problematic at the left boundary. The problem can be alleviated either using a local linear smoother as in Fig. 2 of Schucany (2004), or by employing the reflection technique of Hall and Wehrly (1991); see also the paper by Dai and Sperlich (2010) for a comparison of different boundary correction techniques for kernel smoothers. We will not elaborate further here on these issues since our focus is on the general Model-free Prediction method which can equally be implemented with *any* chosen regression estimator. Finally, Fig. 3.1b shows a scatterplot of the unstudentized residuals $Y - m_x$ with the estimated standard deviation s_x superimposed.

3.2 Model-Based Prediction in Regression

The prediction problem amounts to predicting the future response Y_f associated with a potential design point x_f. Recall that the L_2–optimal (point) predictor of Y_f is $E(Y_f|x_f)$, i.e., the expected value[1] of the response Y_f associated with design point x_f. Under model (3.1), we have that $E(Y_f|x_f) = \mu(x_f)$. However, if the Y_t–data are heavy-tailed, the L_1–optimal predictor might be preferred; this would be given by the *median* response Y_f associated with design point x_f; under model (3.1), this is given by $\mu(x_f) + \sigma(x_f) \cdot median(F)$. If the error distribution F is symmetric, then the L_2– and L_1–optimal predictors coincide.

To obtain practically useful predictors, the unknown quantities $\mu(x), \sigma(x)$, and $median(F)$ must be estimated and plugged in the formulas of optimal predictors. Naturally, $\mu(x_f)$ and $\sigma(x_f)$ are estimated by m_{x_f} and s_{x_f} of Eqs. (3.2) and (3.4). The unknown F can be estimated by $\hat{F}_e$, the empirical distribution of the residuals $e_1, \ldots, e_n$ that are defined in Eq. (3.5). Hence, the practical L_2– and L_1–optimal *model-based* predictors of Y_f are given respectively by $\widehat{Y}_f = m_{x_f}$ and $\widetilde{Y}_f = m_{x_f} + s_{x_f} \cdot median(\hat{F}_e)$.

Suppose, however, that our objective is predicting the future value $g(Y_f)$ associated with design point x_f where $g(\cdot)$ is a function of interest; this possibility is of particular importance due to the fact that data transformations such as Box/Cox, ACE, AVAS, etc., are often applied in order to arrive at a reasonable additive model such as (3.1). For example, the wages in dataset `cps71` have been logarithmically transformed before model (3.1) was fitted in Fig. 3.1a; in this case, $g(x) = \exp(x)$ since naturally we are interested in predicting wage not log-wage.

[1] In general, the L_2–optimal predictor of Y_f would be given by the conditional expectation of Y_f given $Y_1, \ldots, Y_n$ as well as x_f. However, under model (3.1), the Y data are independent, and $E(Y_f|Y_1, \ldots, Y_n, x_f)$ simplifies to just $E(Y_f|x_f)$; the same will be true under the Model-free regression setting of Chap. 4. The study of dependent data will be undertaken in Part III of the book.

In such a case, the model-based L_2–optimal (point) predictor of $g(Y_f)$ is $E(g(Y_f)|x_f)$ which can be estimated by

$$n^{-1}\sum_{i=1}^{n} g\left(m_{x_f} + \sigma_{x_f} e_i\right).$$

Unfortunately, practitioners sometimes use the *naive* plug-in predictor $g(m_{x_f})$ that can be grossly suboptimal since g is typically nonlinear. For instance, if g is convex, as in the exponential example above, Jensen's inequality immediately implies that the naive predictor $g(m_{x_f})$ under-estimates its target, i.e., it is biased downward.

Similarly, the model-based L_1–optimal (point) predictor of $g(Y_f)$ can be approximated by the sample median of the set $\{g\left(m_{x_f} + \sigma_{x_f} e_i\right),\ i = 1,\ldots,n\}$; interestingly, the latter would be equivalent to the naive plug-in predictor $g(\widetilde{Y}_f)$ as long as g is monotone.

3.3 A First Application of the Model-Free Prediction Principle

Consider a dataset like the one depicted in Fig. 3.1. Faced with this type of data, a practitioner may well decide to entertain a model like Eq. (3.1) for his/her statistical analysis. However, even while fitting—and working with—model (3.1), it is highly unlikely that the practitioner will believe that this model is *exactly* true; more often than not, the model will be simply regarded as a convenient approximation.

Thus, in applying strategy (A) of Sect. 2.3.5, the model-free practitioner computes the fitted residuals $e_t = (Y_t - m_{x_t})/s_{x_t}$ that can be interpreted as an effort to center and studentize the $Y_1,\ldots,Y_n$ data. In this sense, they can be viewed as a preliminary transformation of the Y–data towards "i.i.d.–ness" since the residuals $e_1,\ldots,e_n$ have (approximately) same first and second moment while the Y–data do not; see also Remark 2.3.3. Here, and for the remainder of Chap. 3, we will assume that the form of the estimator m_x is *linear* in the Y data; our running example of a kernel smoother obviously satisfies this requirement, and so do other popular methods such as local polynomial fitting.

Recall that throughout Chap. 3 we have assumed that model (3.1) is true. Hence, the model-free practitioner should find (via the usual diagnostics) that to a good approximation the fitted residuals $e_t = (Y_t - m_{x_t})/s_{x_t}$ are close to being i.i.d. However, the model-free practitioner does not see this as model confirmation, and may well try additional choices for centering and/or studentizing the data. Motivated by the studentizing transformation in Politis (2007a), we may consider a more general centering/studentization that may provide a better transformation for the Model-free Prediction Principle. Such a transformation is given by:

$$W_t = \frac{Y_t - \tilde{m}_{x_t}}{\tilde{s}_{x_t}},\ t = 1,\ldots,n. \tag{3.7}$$

where

$$\tilde{m}_{x_t} = cY_t + (1-c)m_{x_t}^{(t)},\ \tilde{M}_{x_t} = cY_t^2 + (1-c)M_{x_t}^{(t)} \text{ and } \tilde{s}_{x_t}^2 = \tilde{M}_{x_t} - \tilde{m}_{x_t}^2 . \quad (3.8)$$

In the above, $m_x^{(t)}$ and $M_x^{(t)}$ denote the estimators m and M respectively computed from the delete-Y_t dataset: $\{(x_i, Y_i),\ i = 1,\ldots,t-1 \text{ and } i = t+1,\ldots,n\}$, and evaluated at the point x. Note that the W_t's, as well as $\tilde{m}_{x_t}, \tilde{M}_{x_t}$, depend on the parameter $c \in [0,1)$ but this dependence is not explicitly denoted. The optimal choice of c will be discussed later. The case $c = 1$ is excluded as it leads to the trivial setting of $W_t = 0$, and an inconsistent $\tilde{m}_{x_t}$ that simply interpolates the data on the scatterplot; similarly problematic would be choosing c close to unity.

Nevertheless, Eq. (3.7) is a general—and thus more flexible—reduction to residuals since it includes the fitted and predictive residuals as special cases. To see this, note that (3.2) implies that the choice $c = K(0)/\sum_{k=1}^{n} K\left(\frac{x_t - x_k}{h}\right)$ corresponds to $\tilde{m}_{x_t} = m_{x_t}$ and $\tilde{M}_{x_t} = M_{x_t}$ in which case Eq. (3.7) reduces to Eq. (3.5), i.e., the fitted residuals. Furthermore, consider the extreme case of $c = 0$; in this case, W_t is tantamount to a predictive residual, i.e., $W_t = \tilde{e}_t$ as defined in Eq. (3.6).

Thus, Eq. (3.7) is a good candidate for our search for a general transformation H_n towards "i.i.d.—ness" as the Model-free Prediction Principle of Chap. 2 requires. With a proper choice of bandwidth (and the constant c), $W_1,\ldots,W_n$ would be—by construction—centered and studentized; hence, the first two moments of the W_t's are (approximately) constant. Since the original data are assumed independent, the W_t's are also approximately independent.

Remark 3.3.1 The independence and constancy of the first two moments of the W_t's generally would fall short of claiming that they are i.i.d.; in this case, however, the claim is true since model (3.1) implies that the nonconstancy of distributions is only attributed to the first two moments. Furthermore, note that the W_t's are not exactly independent because of dependence of m_{x_t} and s_{x_t} to m_{x_k} and s_{x_k}. Nevertheless, under typical conditions, $m_x \xrightarrow{P} E(Y|x)$ and $s_x^2 \xrightarrow{P} \mathrm{Var}(Y|x)$ as $n \to \infty$, i.e., they are both asymptotically nonrandom—hence the asymptotic independence of the W_t's follows.

3.4 Model-Free/Model-Based Prediction

Recall that the prediction problem amounts to predicting the future value Y_f associated with a potential design point x_f. As customary in a prediction problem one starts by investigating the distributional characteristics of the unobserved Y_f centered and studentized. To this effect, note that Eq. (3.7) is also valid for the unobserved Y_f, i.e., the yet unobserved Y_f is related to the yet unobserved W_f by

$$W_f = \frac{Y_f - \tilde{m}_{x_f}^f}{\tilde{s}_{x_f}^f} \quad (3.9)$$

where $\tilde{m}^{\mathrm{f}}$ and $\tilde{s}^{\mathrm{f}}$ are the estimators from Eqs. (3.2) and (3.4) but computed from the *augmented* dataset that includes the full original dataset $\{(x_i, Y_i),\ i = 1, \ldots, n\}$ *plus* the pair $(x_{\mathrm{f}}, Y_{\mathrm{f}})$. As in Eq. (3.8) we have:

$$\tilde{m}^{\mathrm{f}}_{x_{\mathrm{f}}} = cY_{\mathrm{f}} + (1-c)m_{x_{\mathrm{f}}},\ \tilde{M}^{\mathrm{f}}_{x_{\mathrm{f}}} = cY^2_{\mathrm{f}} + (1-c)M_{x_{\mathrm{f}}} \text{ and } \tilde{s}^{\mathrm{f}}_{x_{\mathrm{f}}} = \sqrt{\tilde{M}^{\mathrm{f}}_{x_{\mathrm{f}}} - (\tilde{m}^{\mathrm{f}}_{x_t})^2} \quad (3.10)$$

where $m_{x_{\mathrm{f}}}, M_{x_{\mathrm{f}}}$ are the estimators m, M computed from the original dataset as in Sect. 3.2 and evaluated at the candidate point x_{f}.

Solving Eq. (3.9) for Y_{f} is the key to model-free prediction as it would yield an equation like (2.4). As it turns out, the solution of Eq. (3.9) is given by

$$Y_{\mathrm{f}} = m_{x_{\mathrm{f}}} + s_{x_{\mathrm{f}}} \frac{W_{\mathrm{f}}}{\sqrt{1 - c - cW^2_{\mathrm{f}}}}; \quad (3.11)$$

see Appendix 1 for details. Equation (3.11) is the regression analog of the general Eq. (2.4) of Sect. 2.2.2, and will form the basis for our model-free prediction procedure when the model (3.1) is actually true.

One may now ponder on the optimal choice of c. It is possible to opt to choose c with the goal of normalization of the empirical distribution of the W's in the spirit of the "Gaussian stepping stone" of Sect. 2.3.2. But inasmuch as prediction is concerned, Gaussianity is not required. Since the W_t are (at least approximately) i.i.d., the model-free prediction principle can be invoked, and is equally valid for *any* value of c. It is interesting then to ask how the predictors based on Eq. (3.11) depend on the value of c. Surprisingly (and thankfully), the answer is *not at all* ! After a little algebra it is immediate that

$$\frac{W_t}{\sqrt{1 - c - cW^2_t}} \equiv \tilde{e}_t \text{ for any } c \in [0, 1), \text{ and for all } t = 1, \ldots, n \quad (3.12)$$

where the $\tilde{e}_t$s are the *predictive* residuals defined in Eq. (3.6). In other words, the prediction equation (3.11) does *not* depend on the value of c, and can be simplified to:

$$Y_{\mathrm{f}} = m_{x_{\mathrm{f}}} + s_{x_{\mathrm{f}}} \tilde{e}_{\mathrm{f}}. \quad (3.13)$$

Equation (3.13) will form the basis for our application of the Model-free Prediction Principle under model (3.1).

Remark 3.4.1 Since the model-free philosophy is implemented in a setup where model (3.1) is true, we will denote the resulting predictors by **MF/MB** to indicate both the model-free (MF) *construction*, as well as the predictor's model-based (MB) *realm of validity*. Recall that the Model-Free methodology is a transformation-based approach to inference. Hence, a different notation of the MF/MB setup could be: **transformation-based inference when the model is true.**

To elaborate on the construction of MF/MB predictors, let $\hat{F}_{\tilde{e}}$ denote the empirical distribution of the predictive residuals $\tilde{e}_1, \ldots, \tilde{e}_n$. Then, the L_2– and L_1–optimal

	Model-based	MF/MB
Predictive equation	$Y_f = m_{x_f} + s_{x_f} e_f$	$Y_f = m_{x_f} + s_{x_f} \tilde{e}_f$
L_2–predictor of Y_f	m_{x_f}	$m_{x_f} + s_{x_f} \cdot \text{mean}(\tilde{e}_i)$
L_1–predictor of Y_f	$m_{x_f} + s_{x_f} \cdot \text{median}(e_i)$	$m_{x_f} + s_{x_f} \cdot \text{median}(\tilde{e}_i)$
L_2–predictor of $g(Y_f)$	$n^{-1} \sum_{i=1}^{n} g\left(m_{x_f} + \sigma_{x_f} e_i\right)$	$n^{-1} \sum_{i=1}^{n} g\left(m_{x_f} + \sigma_{x_f} \tilde{e}_i\right)$
L_1–predictor of $g(Y_f)$	$\text{median}\left(g\left(m_{x_f} + \sigma_{x_f} e_i\right)\right)$	$\text{median}\left(g\left(m_{x_f} + \sigma_{x_f} \tilde{e}_i\right)\right)$

Table 3.1 Comparison of the model-based and MF/MB point prediction procedures obtained when model (3.1) is true

model-free predictors of the function $g(Y_f)$ are given, respectively, by the expected value and median of the random variable $g(Y_f)$ where Y_f as given in Eq. (3.13) and $\tilde{e}_f$ is a random variable drawn from distribution $\hat{F}_{\tilde{e}}$.

Focusing on the case $g(x) = x$, it follows that the L_2– and L_1–optimal MF/MB predictors of Y_f are given, respectively, by the expected value and median of the random variable given in Eq. (3.13). Note, however, that the only difference between Eq. (3.13) and the fitted regression equation $Y_t = m_{x_t} + s_{x_t} e_t$ as applied to the case where x_t is the future point x_f is the use of the predictive residuals $\tilde{e}_t$ instead of the regression residuals e_t. The different predictors are summarized in Table 3.1.

3.5 Model-Free/Model-Based Prediction Intervals

The model-based L_2—optimal predictor of Y_f from Table 3.1 uses the model information that the mean of the errors is exactly zero and does not attempt to estimate it. Another way of enforcing this model information is to center the residuals e_i to their mean, and use the centered residuals for prediction. The need to center regression residuals was first pointed out by Freedman (1981) in a linear model setting, and will also be used in the Resampling Algorithm in what follows.

The use of predictive residuals is both natural and intuitive since the objective is prediction. Furthermore, in case $\sigma^2(x)$ can be assumed to be constant,[2] simple algebra shows

$$\tilde{e}_t = e_t / (1 - \delta_{x_t}) \quad \text{where } \delta_{x_t} = K(0) / \sum_{k=1}^{n} K\left(\frac{x_t - x_k}{h}\right). \tag{3.14}$$

[2] If $\sigma^2(x)$ is not assumed constant, then $\tilde{e}_t = e_t C_t / (1 - \delta_{x_t})$ where $C_t = s_{x_t} / s_{x_t}^{(t)}$.

Equation (3.14) suggests that the main difference between the fitted and predictive residuals is their scale; their center should be about the same (and close to zero) since typically

$$\text{mean}(\tilde{e}_i) \approx 0 \ \text{ and } \text{median}(\tilde{e}_i) \approx 0. \tag{3.15}$$

Therefore, the model-based and MF/MB *point* predictors of Y_f are almost indistinguishable; this is, of course, reassuring since, when model (3.1) is true, the model-based procedures are obviously optimal. Nevertheless, due to the different scales of the fitted and predictive residuals, the difference between the two approaches is more pronounced in terms of construction of a predictive *distribution* for Y_f in which case the correct scaling of residuals is of paramount importance; see also the discussion in Sect. 3.7.1.

With regards to the construction of an accurate predictive distribution of Y_f, both approaches (model-based and MF/MB) are formally identical, the only difference being in the use of fitted vs. predictive residuals. The Resampling Algorithm of Sect. 2.4 reads as follows for the case at hand where the predictive function g_{n+1} needed in the Model-free Prediction Principle is essentially determined by $\mu(x)$ and $\sigma(x)$.

Algorithm 3.5.1 RESAMPLING ALGORITHM FOR PREDICTIVE DISTRIBUTION AND PREDICTION INTERVALS FOR $g(Y_f)$

1. *Based on the data $\underline{Y}_n$, construct the estimates m_x and s_x from which the fitted residuals e_i, and predictive residuals $\tilde{e}_i$ are computed for $i = 1, \ldots, n$.*
2. *For the model-based approach, let $r_i = e_i - n^{-1}\sum_j e_j$, for $i = 1, \ldots n$, whereas for the MF/MB approach, let $r_i = \tilde{e}_i$, for $i = 1, \ldots, n$. Let $\hat{F}_n$ denote the empirical distribution of $r_1, \ldots, r_n$. Also let Π be a short-hand for $\Pi(g, m_x, s_x, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n)$, the chosen predictor from Table 3.1; e.g., for the L_2–optimal predictor we have $\Pi = n^{-1}\sum_{i=1}^{n} g\,(m_{x_f} + \sigma_{x_f} r_i)$*
   a. *Sample randomly (with replacement) the variables $r_1, \ldots, r_n$ to create the bootstrap pseudo-data $r_1^*, \ldots, r_n^*$.*
   b. *Create pseudo-data in the Y domain by letting $Y_i^* = m_{x_i} + s_{x_i} r_i^*$, for $i = 1, \ldots n$.*
   c. *Calculate a bootstrap pseudo-response as $Y_f^* = m_{x_f} + s_{x_f} r$ where r is drawn randomly from the set $(r_1, \ldots, r_n)$.*
   d. *Based on the pseudo-data $\{(Y_t^*, x_t), t = 1, \ldots, n\}$, re-estimate the functions $\mu(x)$ and $\sigma(x)$ by the estimators m_x^* and s_x^* using the same methodology,[3] e.g., kernel smoothing with the same kernel, as the original estimators m_x and s_x.*
   e. *Calculate a bootstrap root replicate as $g(Y_f^*) - \Pi(g, m_x^*, s_x^*, \underline{Y}_n, \mathbf{X}_{n+1}, \hat{F}_n)$.*
3. *Steps (a)—(e) in the above are repeated B times, and the B bootstrap root replicates are collected in the form of an empirical distribution with α–quantile denoted by $q(\alpha)$.*

[3] m_x^* and s_x^* can use the same bandwidth as the original estimators m_x and s_x provided these are slightly undersmoothed; otherwise, a two bandwidth trick is recommended—see Remark 3.5.2.

4. *Then, a* $(1-\alpha)100\%$ *equal-tailed predictive interval for* $g(Y_f)$ *is given by:*

$$[\Pi + q(\alpha/2), \Pi + q(1-\alpha/2)]. \tag{3.16}$$

5. *Finally, our estimate of the predictive distribution of* $g(Y_f)$ *is the empirical distribution of bootstrap roots obtained in step 3 shifted to the right by the number* Π.

Remark 3.5.1 As an example, suppose $g(x) = x$ and the L_2–optimal point predictor of Y_f is chosen in which case $\Pi \simeq m_{x_f}$. Then, our $(1-\alpha)100\%$ equal-tailed, predictive interval for Y_f boils down to $[m_{x_f} + q(\alpha/2),\ m_{x_f} + q(1-\alpha/2)]$ where $q(\alpha)$ is the α—quantile of the empirical distribution of the B bootstrap root replicates of type $Y_f^* - m_{x_f}^*$.

Fact 3.5.1 *When* $\sigma^2(x)$ *is constant, Eq. (3.14) implies that* $\delta_{x_t} > 0$, *and thus* $\tilde{e}_t$ *will always be* larger *in absolute value (i.e., inflated) as compared to* e_t. *As a consequence, MF/MB prediction intervals will tend to be wider than their MB counterparts. Nevertheless, this difference disappears asymptotically since* $\delta_{x_t} \to 0$ *under the usual bandwidth condition* $h \to 0$ *but* $hn \to \infty$.

Remark 3.5.2 As in all nonparametric smoothing problems, choosing the bandwidth is often a key issue due to the ever-looming problem of bias; the addition of a bootstrap algorithm as above further complicates things. In the closely related problem of constructing bootstrap confidence bands in nonparametric regression, different authors have used various tricks to account for the bias. For example, Härdle and Bowman (1988) construct a kernel estimate for the second derivative $\mu''(x)$, and use this estimate to explicitly correct for the bias; the estimate of the second derivative is known to be consistent but it is difficult to choose its bandwidth. Härdle and Marron (1991) estimate the (fitted) residuals using the optimal bandwidth but the resampled residuals are then added to an oversmoothed estimate of μ; they then smooth the bootstrapped data using the optimal bandwidth. Neumann and Polzehl (1998) use only one bandwidth but it is of smaller order than the mean square error optimal rate; this *undersmoothing* of curve estimates was first proposed by Hall (1993) and is perhaps the easiest theoretical solution towards confidence band construction although the recommended degree of undersmoothing for practical purposes is not obvious. In a recent paper, McMurry and Politis (2008) show that the use of infinite-order, flat-top smoothing kernels alleviates the bias problem significantly permitting the use of the optimal bandwidth. The above literature pertains to confidence intervals; the construction of prediction intervals is expected to suffer from similar difficulties but not as pronounced. The reason is that the main thrust of prediction interval accuracy is capturing the variability due to the unobserved error; the variability of the estimated features m_x and s_x is of secondary importance—see Sect. 3.6.2.

Remark 3.5.3 An important feature of all bootstrap procedures is that they can handle *joint* prediction intervals, i.e., prediction *regions*, with the same ease as the univariate ones. For example, x_f can represent a collection of p "future" x–points in the above Resampling Algorithm. The only difference is that in Step 2(c) we

would need to draw p pseudo-errors r randomly (with replacement) from the set $(r_1, \ldots, r_n)$, and thus construct p bootstrap pseudo-responses, one for each of the p points in x_f. Then, Step 5 of the Algorithm would give a multivariate (joint) predictive distribution for the response Y at the p points in x_f from which a joint prediction *region* can be extracted. If it is desired that the prediction region is of rectangular form, i.e., joint prediction *intervals* as opposed to a general-shaped region, then these can be based on the distribution of the maximum (and minimum) of the p targeted responses that is obtainable from the multivariate predictive distribution via the continuous mapping theorem; see Wolf and Wunderli (2015) for an elaboration.

For completeness, we now briefly discuss the predictive interval that follows from an assumption of normality of the errors ε_t in the model (3.1). In that case, m_{x_f} is also normal, and independent of the "future" error ε_f. If $\sigma^2(x)$ can be assumed to be at least as smooth as $\mu(x)$, then a normal approximation to the distribution of the root $Y_f - m_{x_f}$ implies an approximate $(1-\alpha)100\%$ equal-tailed, predictive interval for Y_f given by:

$$[m_{x_f} + V_{x_f} \cdot z(\alpha/2),\ m_{x_f} + V_{x_f} \cdot z(1-\alpha/2)] \tag{3.17}$$

where $V_{x_f}^2 = s_{x_f}^2 \left(1 + \sum_{i=1}^n \tilde{K}^2(\frac{x_f - x_i}{h})\right)$ with $\tilde{K}$ defined in Eq. (3.3), and $z(\alpha)$ being the α-quantile of the standard normal. If the "density" (e.g., histogram) of the design points $x_1, \ldots, x_n$ can be thought to approximate a given functional shape (say, $f(\cdot)$) for large n, then the large-sample approximation

$$\sum_{i=1}^{n} \tilde{K}^2(\frac{x_f - x_i}{h}) \sim \frac{\int K^2(x)dx}{nh\, f(x_f)} \tag{3.18}$$

can be used—provided $K(x)$ is such that $\int K(x)dx = 1$; see, e.g., Li and Racine (2007).

Interval (3.17) is problematic in at least two respects: (a) it completely ignores the bias of m_x, so it must be either explicitly bias-corrected, or a suboptimal bandwidth must be used to ensure undersmoothing; and (b) it crucially hinges on *exact*, finite-sample normality of the data as its validity cannot be justified by a central limit approximation. As a result, the usefulness of interval (3.17) is limited.

3.6 Pertinent Prediction Intervals

3.6.1 The i.i.d. Case

As mentioned in Sect. 2.4.1, asymptotic validity is a fundamental property but it does not tell the whole story. Prediction intervals are particularly useful if they can also capture the uncertainty involved in model estimation although the latter is asymptotically negligible.

To give a concrete example, consider the simple case where $Y_1, Y_2, \cdots$ are i.i.d. $N(\mu, \sigma^2)$. Given the data $Y_1, \cdots, Y_n$, we estimate the unknown μ, σ^2 by the sample mean and variance $\hat{\mu}, \hat{\sigma}^2$, respectively. Then, the exact Normal theory $(1-\alpha)100\%$ prediction interval for Y_{n+1} is given by

$$\hat{\mu} \pm t_{n-1}(\alpha/2)\hat{\sigma}\sqrt{1+n^{-1}}. \tag{3.19}$$

One could use the standard normal quantile $z(\alpha/2)$ instead of $t_{n-1}(\alpha/2)$, i.e., construct the prediction interval:

$$\hat{\mu} \pm z(\alpha/2)\hat{\sigma}\sqrt{1+n^{-1}}. \tag{3.20}$$

Since $1+n^{-1} \approx 1$ for large n, an even simpler prediction interval is available:

$$\hat{\mu} \pm z(\alpha/2)\hat{\sigma}. \tag{3.21}$$

Notably, all three above prediction intervals are asymptotically valid in the sense of definition (2.9). Nonetheless, interval (3.21) can be called *naive* since it fails to take into account the variability that results from the error in estimating the theoretical predictor μ by $\hat{\mu}$. The result is that, although asymptotically valid, interval (3.21) will be characterized by *undercoverage* in finite samples; see Geisser (1993) for an in-depth discussion.

By contrast, interval (3.20) does take into account the variability resulting from estimating the theoretical predictor. Therefore, interval (3.20) deserves to be called something stronger than asymptotically valid; we will call it **pertinent** to indicate that it asymptotically captures *all* three elements of the exact interval (3.19), namely:

(i) the quantile $t_{n-1}(\alpha/2)$ associated with the studentized root;
(ii) the error variance σ^2; and
(iii) the variability associated with the estimated parameters, i.e., the factor $\sqrt{1+n^{-1}}$.

In general, an exact interval analogous to (3.19) will not be available because of non-normality of the errors and/or nonlinearity of the optimal predictor. A "pertinent" interval such as (3.20) would be something to strive for. Notably, the bootstrap is an attempt to create prediction intervals that are asymptotically pertinent in that (a) they are able to capture the variability due to the estimated quantities—note that in parametric models with p parameters the correction term inside the square root of (3.19) would be $O(p/n)$ not just $1/n$, and in nonparametric models it would be $O(\frac{1}{hn})$ with $h \to 0$ as $n \to \infty$, i.e., this correction is not so trivial; and (b) they are able to approximate well the necessary quantiles.

Interestingly, while interval (3.19) is based on the distribution of the studentized predictive root, the bootstrap can also work with nonstudentized roots; in this case, the bootstrap would attempt to estimate the product $t_{n-1}(\alpha/2)\ \hat{\sigma}$ as a whole instead of breaking it up in its two constituent pieces. One might be tempted to think that the studentized bootstrap would lead to better approximations, and therefore more accurate prediction intervals but the phenomenon is not as clear-cut as in the case of bootstrap confidence intervals; the Rejoinder of Politis (2013) gives a discussion

to that effect. Finally, note that bootstrap prediction intervals are not restricted to be symmetric around the predictor like (3.19); thus, they may also capture the skewness of the predictive distribution which is valuable in its own right.

3.6.2 Asymptotic Pertinence of Bootstrap Prediction Intervals

To formally define the notion of pertinence, consider model (3.1) under homoscedasticity, i.e.,

$$Y_t = \mu(x_t) + \sigma \cdot \varepsilon_t \text{ for } t = 1, \ldots, n \tag{3.22}$$

with ε_t being i.i.d. (0,1). Recall that the MSE–optimal predictor of Y_{n+1} associated with regressor value $X_{n+1} = x_{n+1}$ is $\mu(x_{n+1})$. Hence we let $\hat{Y}_{n+1} = \hat{\mu}(x_{n+1})$ where $\hat{\mu}(\cdot)$ is some consistent estimator of $\mu(\cdot)$. Assume that $\hat{\mu}(\cdot)$ has rate of convergence a_n, i.e., $a_n(\hat{\mu}(\cdot) - \mu(\cdot))$ has a well-defined, nontrivial asymptotic distribution where $a_n \to \infty$ as $n \to \infty$. Then, the predictive root is given by

$$Y_{n+1} - \hat{Y}_{n+1} = \varepsilon_{n+1} + A_\mu \tag{3.23}$$

where $A_\mu = \mu(x_{n+1}) - \hat{\mu}(x_{n+1}) = O_p(1/a_n)$ represents the estimation error.

Similarly, the bootstrap predictive root can be written as

$$Y^*_{n+1} - \hat{Y}^*_{n+1} = \varepsilon^*_{n+1} + A^*_\mu \tag{3.24}$$

where $A^*_\mu = \hat{\mu}(x_{n+1}) - \hat{\mu}^*(x_{n+1})$. By construction, the model-based bootstrap described in Algorithm 3.5.1 should be capable of capturing both the pure prediction error, i.e., the distribution of ε_{n+1}, as well as the estimation error. We are then led to the following definition.

Definition 3.6.1 ***Asymptotic pertinence of bootstrap prediction intervals under model*** **(3.22).** *Consider a bootstrap prediction interval for Y_{n+1} that is based on approximating the distribution of the predictive root $Y_{n+1} - \hat{Y}_{n+1}$ of Eq.* (3.23) *by the distribution of the bootstrap predictive root $Y^*_{n+1} - \hat{Y}^*_{n+1}$ of Eq.* (3.24)*. The interval will be called* **asymptotically pertinent** *provided the bootstrap satisfies the following three conditions as $n \to \infty$ conditionally on $X_{n+1} = x_{n+1}$.*

- *(i)* $\sup_a |P(\varepsilon_{n+1} \le a) - P^*(\varepsilon^*_{n+1} \le a)| \xrightarrow{P} 0$*, presupposing that the error distribution is continuous.*
- *(ii)* $|P(a_n A_\mu \le a) - P^*(a_n A^*_\mu \le a)| \xrightarrow{P} 0$ *for some sequence $a_n \to \infty$, and for all points a where the assumed nontrivial limit of $P(a_n A_\mu \le a)$ is continuous.*
- *(iii)* ε^*_{n+1} *and* A^*_μ *are independent in the bootstrap world as their analogs are in the real world.*

Let $\hat{V}_n^2$ be an estimate of $\mathrm{Var}(Y_{n+1} - \hat{Y}_{n+1} | X_{n+1} = x_{n+1})$, and let $\hat{V}^*_n$ be its bootstrap counterpart, i.e., an estimate of $\mathrm{Var}^*(Y^*_{n+1} - \hat{Y}^*_{n+1} | X_{n+1} = x_{n+1})$. The bootstrap prediction interval for Y_{n+1} that is based on approximating the distribution of the studentized *predictive root $(Y_{n+1} - \hat{Y}_{n+1})/\hat{V}_n$ by the distribution of the bootstrap*

studentized predictive root $(Y^*_{n+1} - \hat{Y}^*_{n+1})/\hat{V}^*_n$ *will be called asymptotically pertinent if, in addition to (i)—(iii) above, the following also holds:*

(iv) $\hat{V}_n/\hat{V}^*_n \xrightarrow{P} 1$.

Remark 3.6.1 Note that asymptotic pertinence is a stronger property than asymptotic validity. In fact, under model (3.22), just part (i) of Definition 3.6.1 together with the consistency of $\hat{\mu}(\cdot)$ and $\hat{\mu}^*(\cdot)$, i.e., the fact that both A_μ and A^*_μ are $o_p(1)$, are enough to imply asymptotic validity of the bootstrap prediction interval. Also note that part (ii) of Definition 3.6.1 is the condition needed in order to show that the bootstrap can yield asymptotically valid *confidence intervals* for the conditional mean $\mu(\cdot)$. In many cases in the literature, this condition has been already established; we can build upon this for the purpose of constructing pertinent prediction intervals.

Consider again the heteroscedastic model (3.1). Much of the above discussion carries over *verbatim*; for example, the MSE–optimal predictor of Y_{n+1} given $X_{n+1} = x_{n+1}$ is still $\hat{Y}_{n+1} = \hat{\mu}(x_{n+1})$. The only difference is that the predictive root now is expressed as

$$Y_{n+1} - \hat{Y}_{n+1} = \sigma(x_{n+1})\varepsilon_{n+1} + A_\mu, \tag{3.25}$$

and the bootstrap predictive root as

$$Y^*_{n+1} - \hat{Y}^*_{n+1} = \hat{\sigma}(x_{n+1})\varepsilon^*_{n+1} + A^*_\mu \tag{3.26}$$

where $\hat{\sigma}(\cdot)$ is the (consistent) estimator of $\sigma(\cdot)$ that is employed in the bootstrap data generation mechanism. Hence, the following definition is immediate.

Definition 3.6.2 ***Asymptotic pertinence of bootstrap prediction intervals under heteroscedastic model*** **(3.1).** *Consider a bootstrap prediction interval for* Y_{n+1} *that is based on approximating the distribution of the predictive root* $Y_{n+1} - \hat{Y}_{n+1}$ *of Eq.* (3.25) *by the distribution of the bootstrap predictive root* $Y^*_{n+1} - \hat{Y}^*_{n+1}$ *of Eq.* (3.26)*. The interval will be called* asymptotically pertinent *provided the bootstrap satisfies conditions (i)—(iii) of Definition 3.6.1 together with the additional requirement:*

(iv′) $\sigma(x_{n+1}) - \hat{\sigma}(x_{n+1}) \xrightarrow{P} 0$.

Furthermore, the bootstrap prediction interval for Y_{n+1} *that is based on the approximating the distribution of the* studentized *predictive root* $(Y_{n+1} - \hat{Y}_{n+1})/\hat{V}_n$ *by the distribution of the bootstrap studentized predictive root* $(Y^*_{n+1} - \hat{Y}^*_{n+1})/\hat{V}^*_n$ *will be called asymptotically pertinent if, in addition condition (iv) of Definition 3.6.1 also holds.*

Fact 3.6.1 *Under model* (3.1) *and standard regularity conditions, the model-based bootstrap prediction interval* (3.16) *will be asymptotically pertinent provided the bandwidth* h *is chosen in such a way that undersmoothing occurs, i.e., letting* $h = o(n^{-1/5})$ *when the kernel* K *is nonnegative. Otherwise, interval* (3.16) *will be asymptotically valid but not pertinent.*

Remark 3.6.2 Taking into account that $A_\mu = o_p(1)$ as $n \to \infty$, an immediate estimator for the (conditional) variance of the predictive root $Y_{n+1} - \hat{Y}_{n+1}$ under model (3.1) is simply $\hat{V}_n = \hat{\sigma}(x_{n+1})$. Therefore, condition (iv) of Definition 3.6.1 can be rewritten as $\hat{\sigma}(x_{n+1}) - \hat{\sigma}^*(x_{n+1}) \xrightarrow{P} 0$, i.e., it is just a bootstrap version of condition (iv′) of Definition 3.6.2. As a matter of fact, resampling in the heteroscedastic model (3.1) as described in Algorithm 3.5.1 entails using unstudentized predictive roots but it is based on *studentized* residuals. In this case, the unstudentized predictive root method gives prediction intervals that are very close to the intervals that would be obtained from studentized predictive roots so long as the simple estimator $\hat{V}_n = \hat{\sigma}(x_{n+1})$ is used for the latter.

Remark 3.6.3 To continue the discussion of studentized vs. unstudentized prediction intervals for Y_{n+1} under model (3.1), let $\hat{F}_n$ denote the empirical distribution of $r_1, \ldots, r_n$ that were defined in step 2 of Algorithm 3.5.1, and recall that the original errors $\varepsilon_1, \ldots, \varepsilon_n$ are i.i.d. F. Under smoothness assumptions, the typical nonparametric rate of estimating $\mu(\cdot)$ is $a_n = \sqrt{nh}$ where $h \to 0$ but $nh \to \infty$; hence, $A_\mu = O_p(1/\sqrt{nh})$. Interestingly, under regularity conditions, the residual distribution F is estimated at a fast parametric rate, i.e., $\hat{F}_n = F + O_p(1/\sqrt{n})$; see Akritas and VanKeilegom (2001).

Let us compare $\Pi + q(\alpha/2)$, the left end-point of prediction interval (3.16) that is based on unstudentized roots, to $\Pi + Q(\alpha/2)\hat{\sigma}(x_{n+1})$ that would be the left end-point of the corresponding prediction interval based on studentized roots using $\hat{V}_n = \hat{\sigma}(x_{n+1})$. Here, $q(\alpha)$ denotes the α–quantile of $\mathscr{L}^*(Y^*_{n+1} - \hat{Y}^*_{n+1})$ which is the bootstrap probability law of the unstudentized root $Y^*_{n+1} - \hat{Y}^*_{n+1}$. From (3.26) and the above discussion it follows that

$$\mathscr{L}^*(Y^*_{n+1} - \hat{Y}^*_{n+1}) = \hat{\sigma}(x_{n+1})F + O_p(1/\sqrt{nh}).$$

Similarly, $Q(\alpha)$ denotes the α–quantile of $\mathscr{L}^*\left((Y^*_{n+1} - \hat{Y}^*_{n+1})/\hat{\sigma}^*(x_{n+1})\right)$ which is the bootstrap probability law of the studentized root $(Y^*_{n+1} - \hat{Y}^*_{n+1})/\hat{\sigma}^*(x_{n+1})$. It is now apparent that

$$\mathscr{L}^*\left((Y^*_{n+1} - \hat{Y}^*_{n+1})/\hat{\sigma}^*(x_{n+1})\right) = F + O_p(1/\sqrt{nh})$$

Therefore, the two end-points, $\Pi + q(\alpha/2)$ and $\Pi + Q(\alpha/2)\hat{\sigma}(x_{n+1})$, have the same asymptotic accuracy dictated by the $O_p(1/\sqrt{nh})$ term.

3.7 Application to Linear Regression

3.7.1 Better Prediction Intervals in Linear Regression

The literature on predictive intervals in regression is not large; see, e.g., Caroll and Ruppert (1991), Patel (1989), Schmoyer (1992), and the references therein.

Furthermore, the literature on predictive distributions seems virtually non-existent outside the Bayesian framework. What is striking is that even the problem of undercoverage of prediction intervals in *linear* regression reported by Stine (1985) remained open three decades later; see, e.g., Olive (2007).

Thus, in this section we focus on the usual linear regression model:

$$Y_i = \underline{x}_i'\underline{\beta} + Z_i, \text{ for } i = 1, \ldots, n, \tag{3.27}$$

with $Z_t \sim$ i.i.d. $(0,\sigma^2)$. Equivalently, $\underline{Y}_n = X\underline{\beta} + \underline{Z}_n$ where $\underline{Y}_n = (Y_1, \ldots, Y_n)'$ and $\underline{Z}_n = (Z_1, \ldots, Z_n)'$ are $n \times 1$ random vectors, $\underline{\beta}$ is a $p \times 1$ deterministic parameter vector, and X is an $n \times p$ deterministic design matrix of full rank with ith row given by vector $\underline{x}_i'$.

Let $\hat{\underline{\beta}}$ be an estimator of $\underline{\beta}$ that is linear in the data $\underline{Y}_n$ so that the MF/MB methodology of Sect. 3.4, and in particular Eq. (3.13), applies; an obvious possibility is the Least Squares (LS) estimator. Also let $\hat{\underline{\beta}}^{(i)}$ be the same estimator based on the delete-Y_i dataset. The predictive and fitted residuals ($\tilde{z}_i$ and z_i, respectively) corresponding to data point Y_i are defined in the usual manner, i.e., $\tilde{z}_i = Y_i - \underline{x}_i'\hat{\underline{\beta}}^{(i)}$ and $z_i = Y_i - \underline{x}_i'\hat{\underline{\beta}}$. Analogously to Eq. (3.14), here too the predictive residuals are always larger in absolute value (i.e., "inflated") as compared to the fitted residuals. To see this, recall that

$$\tilde{z}_i = \frac{z_i}{1 - h_i}, \text{ for } i = 1, \ldots, n, \tag{3.28}$$

where $h_i = \underline{x}_i'(X'X)^{-1}\underline{x}_i$ is the ith diagonal element of the "hat" matrix $X(X'X)^{-1}X'$; see, e.g., Seber and Lee (2003, Theorem 10.1). Assuming that the regression has an intercept term, Eq. (10.12) of Seber and Lee (2003) further implies $1/n \leq h_i \leq 1$ from which it follows that $|\tilde{z}_i| \geq |z_i|$ for all i.

Noting that the fitted residuals have variance depending on h_i, Stine (1985) suggested resampling the *studentized* residuals $\hat{z}_i = z_i/\sqrt{1 - h_i}$ in his construction of bootstrap prediction intervals. The studentized residuals $\hat{z}_i$ are also "inflated" as compared to the fitted residuals z_i, so Stine's (1985) suggestion was an effort to reduce the undercoverage of bootstrap prediction intervals that was first pointed out by Efron (1983). However, Stine's proposal does not seem to fully correct the problem; for example, Olive (2007) recommends the use of an *ad hoc* further inflation of the residuals arguing that "since residuals underestimate the errors, finite sample correction factors are needed."

Nevertheless, it is apparent from the above discussion that $|\tilde{z}_i| \geq |\hat{z}_i|$. Hence, using the predictive residuals is not only intuitive and natural as motivated by the model-free prediction principle, but it also goes further towards the goal of increasing coverage without cumbersome (and arbitrary) correction factors.[4] To obtain predictive intervals for Y_f, the Resampling Algorithm of Sect. 3.5 now applies *verbatim* with the understanding that in the linear regression setting $m_x \equiv \underline{x}'\hat{\underline{\beta}}$.

[4] Efron (1983) proposed an iterated bootstrap method in order to correct the downward bias of the bootstrap estimate of variance of prediction error; notably, his method involved the use of predictive residuals albeit at the 2nd bootstrap tier—see also Efron and Tibshirani (1993, Chap. 17.7).

As the following subsection confirms, the MF/MB method based on predictive residuals seems to correct the undercoverage of bootstrap prediction intervals. Finally, note that the methodology of Sect. 3.5 can equally address the *heteroscedastic* case when $\mathrm{Var}(Z_i) = \sigma^2(\underline{x}_i)$, and an estimate of $\sigma^2(\underline{x}_i)$ is available via parametric or nonparametric methods.

3.7.2 Simulation: Prediction Intervals in Linear Regression

We now conduct a small simulation in the linear regression setup with $p = 2$, i.e., $\underline{x}_i = (1, x_i)'$, and $Y_i = \beta_0 + \beta_1 x_i + Z_i$, for $i = 1, \ldots, n$. For the simulation, the values $\beta_0 = -1$ and $\beta_1 = 1$ were used, and $Z_t \sim$ i.i.d. (0,1) from distribution Normal or Laplace, i.e., two-sided exponential. The design points $x_1, \ldots, x_n$ for $n = 50$ were generated from a standard normal distribution, and the prediction carried out at the point $x_f = 1$. The simulation focused on constructing 90 % prediction intervals, and was based on 900 repetitions of each experiment. Both LS regression and L_1 regression were considered for estimating β_0 and β_1.

Table 3.2 reports the empirical coverage levels (CVR), and (average) lower and upper limits of the different prediction intervals in the linear regression case. The standard error of the CVR entries is 0.01; the provided standard error (st.err.) applies equally to either the lower or upper limit of the interval. For the first five rows of Table 3.2, β_0 and β_1 were estimated by Least Squares which is optimal in the Normal case; in the last two rows of Table 3.2, β_0 and β_1 are estimated via L_1 regression which is optimal in the Laplace case. Note that the ideal point predictor of Y at $x_f = 1$ is zero; so the prediction intervals are expected to be centered around zero. Indeed, all (average) intervals of Table 3.2 are approximately symmetric around zero.

Linear regression is, of course, a model-based setup; so both interval constructions MB (=model-based) and MF/MB (=model-free/model-based) of Sect. 3.5 are applicable; they were both considered here in addition to three competing intervals: Stine's (1985) interval that is analogous to the MB construction except that Stine used the studentized residuals; the usual NORMAL theory interval, namely $m_{x_f} \pm t_{n-2}(\alpha/2) S \sqrt{1 + h_f}$; and Olive's (2007) "semi-parametric" interval:

$$\left(m_{x_f} + a_n e(\alpha/2) \sqrt{1 + h_f},\ m_{x_f} + a_n e(1 - \alpha/2) \sqrt{1 + h_f} \right).$$

In the above, m_{x_f} is the usual point predictor given by $\hat{\beta}_0 + \hat{\beta}_1 x_f$, $h_f = \underline{x}_f'(X'X)^{-1}\underline{x}_f$ is the "leverage" at point x_f, and $S^2 = (n-2)^{-1} \sum_{i=1}^n e_i^2$. In Olive's interval, $e(\alpha)$ is the α–quantile of the empirical distribution of the residuals $\{e_1, \ldots, e_n\}$ while $a_n = (1 + \frac{15}{n}) \sqrt{\frac{n}{n-2}}$ is an *ad hoc* "correction" factor employed to increase coverage.

Distribution:	Normal		Laplace	
Case $x_f = 1$	CVR	INTERVAL (st.err.)	CVR	INTERVAL (st.err.)
MF/MB	0.890	$[-1.686, 1.682]$ (.011)	0.901	$[-1.685, 1.691]$ (.016)
MB	0.871	$[-1.631, 1.609]$ (.011)	0.886	$[-1.611, 1.619]$ (.015)
MB Stine	0.881	$[-1.656, 1.641]$ (.011)	0.892	$[-1.640, 1.663]$ (.015)
MB Olive	0.941	$[-2.111, 2.097]$ (.017)	0.930	$[-2.072, 2.089]$ (.025)
NORMAL	0.901	$[-1.723, 1.711]$ (.009)	0.910	$[-1.699, 1.716]$ (.011)
MF/MB L_1	0.896	$[-1.715, 1.709]$ (.012)	0.908	$[-1.699, 1.705]$ (.016)
MB L_1	0.871	$[-1.647, 1.632]$ (.012)	0.896	$[-1.619, 1.636]$ (.015)

Table 3.2 Empirical coverage levels (CVR), and (average) lower and upper bounds of different prediction intervals with nominal coverage of 0.90 in linear regression; the standard error (st.err.) applies equally to either the lower or upper limit

The findings of Table 3.2 are quite interesting:

- The NORMAL theory interval (based on t–quantiles) has exact coverage with Normal data—as expected—but slightly over-covers in the Laplace case. It is also the interval with smallest length variability.
- Olive's interval shows striking *over*-coverage which is an indication that the a_n correction factor is too extreme. Also surprising is the large variability in the length of Olive's interval that is 50 % larger than that of our bootstrap methods.
- Looking at rows 1—3, the expected monotonicity in terms of increasing coverage is observed; i.e., CVR(MB) $<$ CVR(MB Stine) $<$ CVR(MF/MB).
- The MF/MB intervals have (almost) uniformly better coverage than their MB analogs indicating that using the predictive residuals is indeed a good solution to the widely reported undercoverage of MB and Stine's intervals.

3.7.3 Model-Free vs. Least Squares: A Reconciliation

As claimed throughout the book, the Model-Free approach can form the basis for a complete statistical inference that includes point estimators and predictors in addition to confidence and prediction intervals without assuming an additive model such as (2.1). Interestingly, however, when an additive regression model is known to hold true, little is lost by adhering to the Model-Free approach, i.e., trying to find a transformation towards "i.i.d.-ness."

To see why, let us assume Eq. (3.27) with an $n \times p$ design matrix X that has a column of 1's as its first column; then we can write:

$$Y_j = \underline{x}_j' \underline{\beta} + Z_j = \beta_0 + \underline{x}_{j,p-1}' \underline{\beta}_{p-1} + Z_j \text{ for } j = 1, \ldots, n, \tag{3.29}$$

where $Z_t \sim$ i.i.d. $(0, \sigma^2)$, $\underline{\beta}_{p-1} = (\beta_1, \ldots, \beta_{p-1}$, and the jth row of X is denoted by the (row) vector $\underline{x}_j = (1, \underline{x}_{j,p-1})$.

From the point of view of Model-free prediction, the essence of this model is that the variables $\varepsilon_j \equiv Y_j - \underline{x}_{j,p-1}' \underline{\beta}_{p-1}$ are i.i.d. albeit with (possibly) nonzero mean β_0. Thus, a candidate transformation to "i.i.d.–ness" may be constructed by letting $r_j = Y_j - \underline{x}_{j,p-1}' \hat{\underline{\beta}}_{p-1}$, where $\hat{\underline{\beta}}_{p-1}$ is a candidate vector. The Model-Free prediction principle now mandates choosing $\hat{\underline{\beta}}_{p-1}$ with the objective of having the r_js become as close to i.i.d. as possible. However, under the stated regression model, the r_js would be i.i.d. if only their first moment was properly adjusted.

To elaborate, a homoscedastic regression model such as (3.29) implies that all central moments of order two or higher are constant; the only non-i.i.d. feature of the data is in the first moment. So, in this case, the Model-Free prediction principle suggests choosing $\hat{\underline{\beta}}_{p-1}$ in such a way as to make $r_1, \ldots, r_n$ have (approximately) the *same* first moment. Noting that the first moment—if it is common—would be naturally approximated by the empirical value $\hat{r} = n^{-1} \sum_{i=1}^{n} r_i$, we can use a *subsampling* construction to make this happen; see, e.g., Politis et al. (1999).

To fix ideas, assume for simplicity that $p = 2$, and that the univariate design points $x_1, \ldots, x_n$ found in the 2nd column of X are sorted in ascending order. Then compute the overlapping block means

$$\bar{r}_{k,b} = b^{-1} \sum_{j=k}^{k+b-1} r_j \text{ for } k = 1, \ldots, q \tag{3.30}$$

where b is the block size, and $q = n - b + 1$ is the number of available blocks.

Note that $\bar{r}_{k,b}$ is an estimate of the first moment of the r_is found in the kth block. In order to achieve the target requirement that all $r_1, \ldots, r_n$ have first moment that is the same (and thus approximately equal to $\hat{r}$), the Model-free practitioner may

$$\text{choose } \hat{\beta}_1 \text{ that minimizes } LS(b) = \sum_{k=1}^{q} (\bar{r}_{k,b} - \hat{r})^2 \text{ or } L1(b) = \sum_{k=1}^{q} |\bar{r}_{k,b} - \hat{r}| \tag{3.31}$$

according to whether an L_2 or L_1 loss criterion is preferred. Instead of $\hat{r}$, we could equally use the mean of means, i.e., $\bar{\bar{r}} = q^{-1} \sum_{k=1}^{q} \bar{r}_{k,b}$ as the centering value in Eq. (3.31). If $b = 1$, then $\hat{r} = \bar{\bar{r}}$ whereas if $b > 1$, then $\hat{r} = \bar{\bar{r}} + O_P(b/n)$; thus, the difference is negligible provided b is small as compared to n.

Recall that in the typical application of subsampling for variance or distribution estimation, it is suggested to take the block size b to be large (but still of smaller order than n); this is for the purpose of making the subsample statistics $\bar{r}_{k,b}$ have asymptotically the same distribution as the statistic $\hat{r}$ computed from the

full sample. Nevertheless, it is not crucial in our current setting that each of the $\bar{r}_{k,b}$ have asymptotically the same distribution as $\hat{r}$. What is important is that all the $\bar{r}_{k,b}$ (for $k = 1, \ldots, q$) have approximately the *same* distribution whatever that may be. Therefore, it is not necessary in Eq. (3.31) to use a large value for b. Even the value $b = 1$ is acceptable, in which case we have:

$$\frac{d}{d\hat{\beta}_1} LS(1) = 0 \Rightarrow \hat{\beta}_1 = \frac{\sum_{i=1}^{n}(Y_i - \bar{Y})(x_i - \bar{x})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \text{ where } \bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i \text{ and } \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

In other words, the Model-free fitting procedure (3.31) with L_2 loss and $b = 1$ is re-assuringly *identical* to the usual Least Squares estimator!

Note that the r_is serve as proxies for the unobservable ε_is which have expected value β_0 under model (3.29). Hence, β_0 is naturally estimated by the sample mean of the r_is, i.e.,

$$\hat{\beta}_0 = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{\beta}_1 x_i) = \bar{Y} - \hat{\beta}_1 \bar{x}$$

which is again the Least Squares estimator.

Minimizing $LS(b)$ with $b > 1$ gives a more robust way of doing Least Squares in which the effect of potential outliers is diminished by the local averaging of b neighboring values, Similarly, minimizing $L1(1)$ is equivalent to L_1 regression, whereas minimizing $L1(b)$ with $b > 1$ gives additional robustness.

Finally, let us revisit the general p case of model (3.29). When $p > 2$, the vector regressors $\{\underline{x}_{i,p-1} \text{ for } i = 1, \ldots, n\}$ cannot be sorted in ascending order. One could instead use a local-averaging or nearest-neighbor technique to compute the subsample means. But no such trick is needed in the most interesting case of $b = 1$ since then the quantities $LS(1)$ and $L1(1)$ are unequivocally defined as

$$LS(1) = \sum_{k=1}^{n}(r_k - \hat{r})^2 \text{ and } L1(1) = \sum_{k=1}^{n}|r_k - \hat{r}|. \tag{3.32}$$

It is now easy to see that the Model-free practitioner that chooses the β's in order to minimize $LS(1)$ or $L1(1)$ is effectively doing Least Squares or L_1 regression, respectively.

Hence, when an additive linear regression model is available, there is no discrepancy between the Model-free point of view and traditional model fitting. Nevertheless, the Model-free approach can still lend some insights such as the aforementioned use of predictive residuals in connection with the (model-based) residual bootstrap.

Appendix 1: The Solution of Eq. (3.9)

Squaring Eq. (3.9) and using (3.10) we obtain the double solution:

$$Y_{\rm f} = \frac{m_{x_{\rm f}}(1-c)(1-c-cW_{\rm f}^2) \pm |W_{\rm f}|\sqrt{(1-c)^2 m_{x_{\rm f}}^2(-1+c+cW_{\rm f}^2)+(1-c)M_{x_{\rm f}}D_{\rm f}}}{D_{\rm f}} \tag{3.33}$$

where $s_{x_{\rm f}}^2 = M_{x_{\rm f}} - m_{x_{\rm f}}^2$, and $D_{\rm f} = (1-c)^2 + (c^2-c)W_{\rm f}^2$. A little algebra shows that the denominator $D_{\rm f}$ is strictly positive and the argument of the square root in Eq. (3.33) is nonnegative provided the bound (3.34) below holds[5]:

$$|W_t| < \sqrt{\frac{1-c}{c}} \quad \text{for all } t. \tag{3.34}$$

To see that (3.34) is indeed true, note that Eq. (3.7) implies

$$\begin{aligned} \frac{1}{W_t^2} &= \frac{\tilde{s}_{x_t}^2}{(Y_t - \tilde{m}_{x_t})^2} = \frac{\tilde{M}_{x_t} - \tilde{m}_{x_t}^2}{(Y_t - \tilde{m}_{x_t})^2} \\ &= \frac{cY_t^2 + (1-c)M_{x_t}^{(t)} - (cY_t + (1-c)m_{x_t}^{(t)})^2}{(1-c)^2(Y_t - m_{x_t}^{(t)})^2} \\ &= \frac{c-c^2}{(1-c)^2} + \frac{(1-c)\left(M_{x_t}^{(t)} - (m_{x_t}^{(t)})^2\right)}{(1-c)^2(Y_t - m_{x_t}^{(t)})^2} \geq \frac{c-c^2}{(1-c)^2} \end{aligned}$$

since $M_{x_t}^{(t)} - (m_{x_t}^{(t)})^2 \geq 0$ having assumed that the bandwidths h and q are the same. From the above, it follows that $|W_t| \leq \sqrt{(1-c)/c}$ as desired, with *strict* inequality provided $M_{x_t}^{(t)} > (m_{x_t}^{(t)})^2$.

Now as previously noted, c is in general a small number. For example, if $c = K(0)/\sum_{k=1}^n K\left(\frac{x_t - x_k}{h}\right)$, then c tends to zero as $h \to 0$ in which case Eq. (3.33) becomes

$$Y_{\rm f} \simeq m_{x_{\rm f}} \pm |W_{\rm f}| s_{x_{\rm f}}. \tag{3.35}$$

Comparing Eq. (3.35) to Eq. (3.9), it follows that the solution $Y_{\rm f} \simeq m_{x_{\rm f}} + W_{\rm f} s_{x_{\rm f}}$ is the *uniquely* correct one for Eq. (3.35). By the same token (and due to the continuity in the variable c), the double solution (3.33) reduces to the *unique* solution of Eq. (3.9)

$$Y_{\rm f} = \frac{m_{x_{\rm f}}(1-c)(1-c-cW_{\rm f}^2) + W_{\rm f}\sqrt{(1-c)^2 m_{x_{\rm f}}^2(-1+c+cW_{\rm f}^2)+(1-c)M_{x_{\rm f}}D_{\rm f}}}{D_{\rm f}} \tag{3.36}$$

that simplifies to Eq. (3.11) as claimed.$\diamond$

[5] If $c = 0$, the bound (3.34) is trivial: $|W_t| < \infty$.

Appendix 2: L_1 vs. L_2 Cross-Validation

Early proponents of cross-validation include Allen (1971, 1974), Geisser (1971, 1975), and Stone (1974). Minimizing the PREdictive Sum of Squared residuals (PRESS) has been shown to be generally consistent for the optimal bandwidth—although characterized by slow rates of convergence; see, e.g., Härdle and Marron (1991), and Härdle, Hall, and Marron (1988).

To further discuss the cross-validation procedure, we will focus here on the nonparametric model (3.1) with the objective of prediction of Y_f under the two criteria L_1 and L_2; see Table 3.1 for a summary. Since the L_2–optimal predictor is the one minimizing the Mean Squared Error (MSE) of prediction, the minimization of PRESS makes perfect sense in order to further reduce this MSE. However, the L_1–optimal predictor is the one minimizing the Mean Absolute Error (MAE) of prediction; to fine-tune it, it may be preferable to use an L_1–cross-validation criterion, i.e., to minimize the PREdictive Sum of Absolute Residuals abbreviated as PRESAR $= \sum_{t=1}^{n} |\tilde{e}_t|$ where $\tilde{e}_t$ are the predictive residuals of Eq. (3.6).

In addition, L_1–cross-validation may be advisable on robustness considerations. Note that the random variable ε_t^2 (of which $\tilde{e}_t^2$ is a proxy) has a distribution with potentially heavy tails. For example, if $\varepsilon_t \sim N(0,1)$, then the density of ε_t^2 at point u has tails of type: $|u|^{-1/2}\exp(-|u|)$, i.e., tails of exponential thickness. If ε_t is itself a (two-sided) exponential, then the matters are much worse: the density of ε_t^2 at point u has tails of type: $|u|^{-1/2}\exp(-\sqrt{|u|})$. Now recall that $n^{-1}\times\text{PRESS} = n^{-1}\sum_{t=1}^{n}\tilde{e}_t^2$ is an empirical version of $E\varepsilon_t^2$. Although this expectation is finite in the two cases discussed above, the heavy tails of ε_t^2 make a sample average like $n^{-1}\times$PRESS somewhat unstable in practice. In other words, the presence of a large value generated by the heavy tails (or by potential outliers) can throw off PRESS together with the resulting bandwidths estimated by cross-validation. For this reason, L_1–cross-validation may be preferable, and is not any more computationally expensive than the usual L_2–cross-validation.[6]

To see the difference between L_1 and L_2 cross-validation in practice, a small simulation was conducted. For the simulation, data were generated from model (3.1) with the choices $\mu(x) = \sin(x)$, $\sigma(x) = 1/10$, $\varepsilon_t \sim$ i.i.d. $(0,\tau^2)$ with distribution normal or two-sided exponential (Laplace), and different values for τ; reducing the error standard deviation τ has a similar effect as increasing sample size. For each of the error distributions, 999 datasets each of size $n = 100$ were created; the design points $x_1,\ldots,x_n$ were drawn each time from a uniform distribution on $(0,2\pi)$.

The MSE of estimator m_x is denoted by MSE_x and was empirically evaluated at 25 different x–points taken equi-spaced on a grid of the interval $(0,2\pi)$; those points were: $0.24, 0.48, \cdots, 5.79, 6.03$. Figure 3.2 shows a plot of the estimated MSE_x as a function of x in the case $\tau = 4$ using either L_1 or L_2 cross-validation. The peaking of the MSE at the boundaries is a well-known problem associated with kernel smoothers; it can be alleviated using the reflection technique of Hall and Wehrly

[6] In the rare case of non-unique minima in PRESAR cross-validation, the dilemma may be resolved by picking the result closest to one given by PRESS.

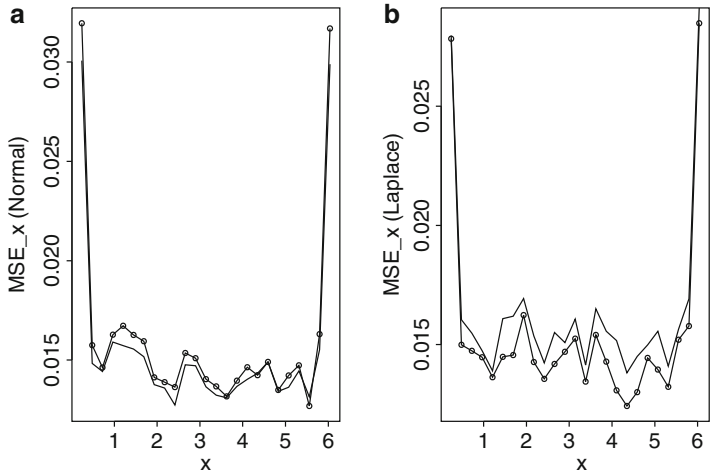

Fig. 3.2 Plot of estimated MSE_x as a function of x in the case $\tau = 4$ using either L_1 (*dashed circles*) or L_2 cross-validation (*solid lines*). (**a**) Normal data; (**b**) Laplace data

$\tau =$	1	2	4
Normal	1.010	1.026	1.034
Laplace	0.970	0.959	0.941
Contam.	0.987	0.934	0.887

Table 3.3 Entries are estimated ratios $\text{IMSE}(L_1)/\text{IMSE}(L_2)$ where L_1 and L_2 indicate the type of cross-validation used, and τ^2 is the error variance; the standard error of each entry is approximately 0.01 as found by subsampling

(1991) which, in effect, makes the kernel estimator approximately equivalent to local linear fitting when the data are evenly distributed on the x-scale—see, e.g., Fan and Gijbels (1996) or Hastie and Loader (1993).

The performance of PRESS appears slightly better in the Normal case—see Fig. 3.2a, while PRESAR has a definite (and seemingly uniform) advantage in the Laplace case—see Fig. 3.2b. This is hardly surprising since minimization of $\sum_{t=1}^{n} \varepsilon_t^2$ (resp. $\sum_{t=1}^{n} |\varepsilon_t|$) is tantamount to Maximum Likelihood in the Normal (resp. Laplace) case. However, note that PRESAR's target is minimization of the Mean Absolute Error (MAE) of estimator m_x and *not* its MSE; the fact that PRESAR yields MSE's that are smaller than that from PRESS (whose target is MSE minimization) is quite noteworthy.

Estimating MSE_x on a grid of points gives a natural estimate of the Integrated MSE of m_x denoted by $\text{IMSE} = \int_0^{2\pi} \text{MSE}_x \, dx$. Table 3.3 compares the IMSE of m_x using either L_1 or L_2 cross-validation for the bandwidth. The implication is that the

two methods are very similar in the Gaussian case (with PRESS being slightly better). However, as expected, L_1 cross-validation has a definite advantage in the heavy-tailed case, and this is particularly true when the error variance is large (and/or the sample size is small).

The simulation was repeated in a situation involving outliers; here the errors were $\varepsilon_t \sim$ i.i.d. $N(0, \tau^2)$ with a 5 % contamination of $N(0, (10\tau)^2)$. Not surprisingly, PRESAR displays *robustness* to outliers and clearly outperforms PRESS in this case as indicated by the last row of Table 3.3. As a consequence of the above discussion, it seems that PRESAR may be preferable to PRESS overall since (a) it is optimal for the L_1 predictor, and (b) it works very well *even* for the L_2 predictor and MSE minimization—outperforming PRESS cross-validation in the non-normal examples.

Chapter 4
Model-Free Prediction in Regression

4.1 Introduction

In Chap. 3, the data $\{(Y_t, x_t), t = 1, \ldots, n\}$ were assumed to have been generated by model (3.1). As already mentioned, the regressor x_j is often thought of as deterministic, and $\mu(x_j)$ has the interpretation of the expected value of the response Y_j associated with regressor x_j. If the regressors are random, i.e., if $x_1, \ldots, x_n$ are the realizations of the random variables $X_1, \ldots, X_n$, we can still write an analog of model (3.1), i.e.,

$$Y_t = \mu(X_t) + \sigma(X_t)\, \varepsilon_t \text{ for } t = 1, \ldots, n \tag{4.1}$$

where the ε_t are i.i.d. (0,1). Coupled with the "exogeneity" assumption that the ε_t are independent of $X_1, \ldots, X_n$, all results of Chap. 3 go through *verbatim* with the understanding that inference is conducted conditionally on event $S_n = \{X_j = x_j \text{ for } j = 1, \ldots, n\}$. A weaker assumption is to simply assume that the pairs

$$(Y_j, X_j) \text{ for } j = 1, \ldots, n \text{ are i.i.d.} \tag{4.2}$$

where the joint distribution of (Y_j, X_j) is not restricted to belong to a parametric family. Define $\mu(x_j) = E(Y_j | X_j = x_j)$, and recall that $E(Y|X)$ is the Hilbert space projection of Y on the subspace of all (measurable) functions of X. It then follows that the discrepancy $Z_j = Y_j - \mu(X_j)$ is uncorrelated with all (measurable) functions of X_j. One can then write down the equation

$$Y_t = \mu(X_t) + Z_t \text{ for } t = 1, \ldots, n$$

but the above could not plausibly be considered as a regression model with i.i.d. errors. For instance, the second moment of Z_j could very well depend on X_j; this would make the heteroscedastic random regressor model (4.1) more plausible by defining $\varepsilon_t = Z_t / \sigma(X_t)$ and $\sigma^2(x_t) = \text{Var}(Y_t | X_t = x_t)$ since, by construction, the ε_t would have conditional—and therefore also unconditional—mean zero and variance one. However, there is no reason that the third (or higher moment) of Z_j will not

D.N. Politis, *Model-Free Prediction and Regression*, Frontiers in Probability and the Statistical Sciences, DOI 10.1007/978-3-319-21347-7_4

depend on X_j, in which case the ε_t will *not* be i.i.d. (0,1) as model (4.1) requires. The above goes to show that Eq. (4.2) is a vague structural assumption, and does not constitute a nonparametric model *per se*. Throughout this chapter, we will work with a weaker version of (4.2) that is elaborated upon in the next section.

Remark 4.1.1 (On resampling pairs) In Chap. 3, the residual-based bootstrap method was employed in order to construct prediction intervals. As mentioned in Sect. 2.4, the Model-free resampling algorithm also bears some similarity with the residual bootstrap. However, the random regressor set of assumption (4.2) motivates the use of the well-known *pairs bootstrap* which is nothing else than the i.i.d. bootstrap of Efron (1979) applied to the i.i.d. pairs (Y_j, X_j); see Bose and Chatterjee (2002) for a review and comparison of several different resampling methods for (linear) regression. By contrast to the residual bootstrap, the pairs bootstrap can be performed without appeal to a particular model as it is based on the model-free assumption (4.2); this makes it very useful for evaluating parameter uncertainty incorporating the uncertainty due to model selection—see Efron (2014) including the discussion pieces. However, the pairs bootstrap cannot create a pseudo-value for the future response Y_f associated with a chosen regressor value x_f; hence, it is not immediately useful as a method that yields prediction intervals.

4.2 Constructing the Transformation Towards i.i.d.–Ness

We now approach the nonparametric regression setup when a model such as Eq. (3.1) cannot be considered to hold true. As discussed in the last section, it may be the case that the skewness and/or kurtosis of Y_t depends on x_t; therefore, centering and studentization alone cannot yield "i.i.d.–ness." For example, kernel estimates of skewness and kurtosis from dataset `cps71`—although slightly undersmoothed—clearly point to the nonconstancy of these two functions; see Fig. 4.1. Throughout this chapter, we will work with a weaker version of (4.2) that is described as follows:

Model-free Regression setup. *The dataset is* $\{(Y_t, x_t),\ t = 1, \ldots, n\}$ *where the regressors* $x_1, \ldots, x_n$ *are either deterministic, or represent a realization of the random variables* $X_1, \ldots, X_n$. *In the latter case, it will be assumed that* Y_j *is independent of* $\{X_k$ *for* $k \neq j\}$, *and inference will be conducted conditionally on event* $S_n = \{X_j = x_j$ *for* $j = 1, \ldots, n\}$. *Conditionally on* S_m, *for any* $m \geq 1$, *the responses* $Y_1, \ldots, Y_m$ *will be assumed independent although not identically distributed. Also assume that the conditional distribution* $P\{Y_j \leq y | X_j = x\}$ *does not depend on* j.

As before, the objective is inference regarding the future response Y_f associated with predictor X_f taking the value x_f.

Remark 4.2.1 In the case of deterministic design, the above Model-free Regression setup implies that the Y_ts are independent although, of course, not identically distributed. In the case of random design, the Model-free Regression setup implies Eq. (4.2) if one additionally assumes that $X_1, \ldots, X_n$ are i.i.d.; the latter is a convenient assumption that may be adopted if/when needed.

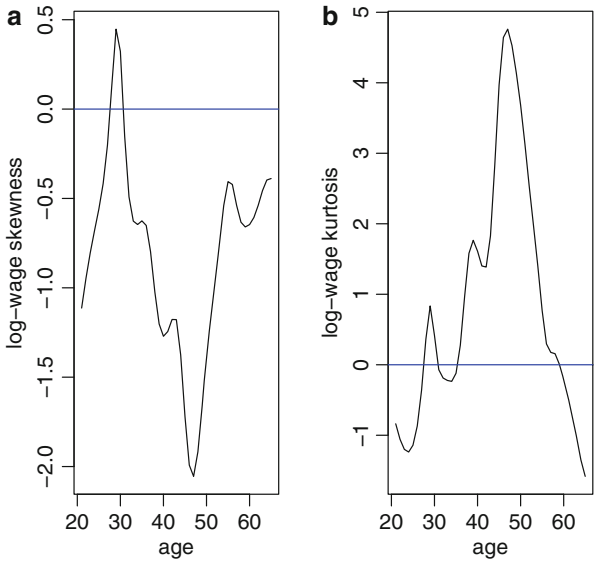

Fig. 4.1 (**a**) Skewness of log-wage vs. age. (**b**) Kurtosis of log-wage vs. age [Both are kernel-based estimates from dataset `cps71`; kurtosis is given relative to the normal value of 3]

For nonparametric estimation, some smoothness assumptions are typically needed as well. We will generally work under the simple assumption that the common conditional distribution $D_x(y) = P\{Y_j \le y | X_j = x\}$ is *continuous* in both x and y. Assuming that $D_x(y)$ is continuous in y implies that $Y_1, \ldots, Y_n$ are continuous random variables; otherwise standard methods like Generalized Linear Models (GLM) can be invoked, e.g., logistic regression, Poisson regression, etc.—see, e.g., McCullagh and Nelder (1983). Nevertheless, one of the variations of the methodology, namely the Limit Model-Free method, remains valid when the responses have a discrete (or even mixed) distribution thus presenting an alternative to a GLM with "link" that depends smoothly on x; see Remark 4.4.2. Since the collection of functions $D_x(\cdot)$ is assumed to depend in a smooth way on x, we can estimate $D_x(y)$ by a "local" empirical distribution such as

$$N_{x,h}^{-1} \sum_{t:|x_t - x| < h/2} \mathbf{1}\{Y_t \le y\} \tag{4.3}$$

where $\mathbf{1}\{Y_t \le y\}$ denotes the indicator of event $\{Y_t \le y\}$, and $N_{x,h}$ is the number of summands, i.e., $N_{x,h} = \#\{t : |x_t - x| < h/2\}$. More generally, we can estimate $D_x(y)$ by

$$\hat{D}_x(y) = \sum_{i=1}^{n} \mathbf{1}\{Y_i \le y\} \tilde{K}\left(\frac{x - x_i}{h}\right) \tag{4.4}$$

where $\tilde{K}\left(\frac{x-x_i}{h}\right) = K\left(\frac{x-x_i}{h}\right) / \sum_{k=1}^{n} K\left(\frac{x-x_k}{h}\right)$ as before; for any fixed y, this is just a Nadaraya-Watson smoother of the variables $\mathbf{1}\{Y_t \le y\}$, $t = 1, \ldots, n$. Note that

Eq. (4.3) is just $\hat{D}_x(y)$ with K chosen as the rectangular kernel, i.e., $K(x) = \mathbf{1}\{|x| \leq 1/2\}$; in general, we can use any *nonnegative* integrable kernel $K(\cdot)$ in (4.4). The nonnegativity of the kernel $K(\cdot)$ is important in order to ensure that $\hat{D}_x(\cdot)$ is a *bona fide* distribution function for any x.

Remark 4.2.2 For $\hat{D}_x$ to be an accurate estimator of D_x, the value x must be such that it has an appreciable number of h-close neighbors among the original predictors $x_1, \ldots, x_n$, i.e., that the number $N_{x,h}$ is not too small. For example, if $N_{x,h} \leq 1$ the estimation of D_x is not just inaccurate—it is simply infeasible.

Estimator $\hat{D}_x(y)$ enjoys many good properties including asymptotic consistency under regularity conditions. For example,

$$\text{Var}\,(\hat{D}_x(y)) = O(\frac{1}{hn}) \text{ and } \text{Bias}\,(\hat{D}_x(y)) = O(h^2) \tag{4.5}$$

with $h \to 0$ but such that $hn \to \infty$; see Theorem 6.1 of Li and Racine (2007). Nevertheless, $\hat{D}_x(y)$ is discontinuous as a function of y, and therefore unacceptable for our immediate purposes. In Politis (2010) a piecewise linear—and strictly increasing—version of $\hat{D}_x(y)$ was proposed; here, we will take a slightly different approach. Observe that the discontinuity of $\hat{D}_x(y)$ as a function of y stems from the discontinuity of the indicator functions $\mathbf{1}\{Y_t \leq y\}$. We may therefore replace $\mathbf{1}\{Y_t \leq y\}$ by the smooth function $\Lambda\left(\frac{y-Y_t}{h_0}\right)$ in Eq. (4.4) leading to the estimator

$$\bar{D}_x(y) = \sum_{i=1}^{n} \Lambda\left(\frac{y - Y_t}{h_0}\right) \tilde{K}\left(\frac{x - x_i}{h}\right) \tag{4.6}$$

that is also studied in Li and Racine (2007, Chap. 6). In the above, h_0 is a positive bandwidth parameter and $\Lambda(y)$ is a smooth distribution function that is strictly increasing, rendering the estimator $\bar{D}_x(y)$ continuous and strictly increasing in y. For example, we may define $\Lambda(y) = \int_{-\infty}^{y} \lambda(s)ds$, where $\lambda(s)$ is a symmetric density function that is continuous and nonnegative over its support. In this case, it is apparent that $\bar{D}_x(y)$ will not only be continuous—it will actually be differentiable with respect to y. Thus, a different interpretation of estimator $\bar{D}_x(y)$ is that it is the indefinite integral of a local estimate of the *density* associated with distribution $D_x(y)$, i.e., an estimate of the derivative of $D_x(y)$ with respect to y (provided that exists).

Remark 4.2.3 A local linear (or polynomial) smoother of the indicator variables $\mathbf{1}\{Y_t \leq y\}$ or the smooth variables $\Lambda\left(\frac{y-Y_t}{h_0}\right)$ could conceivably be used in place of the local constant estimators (4.4) and (4.6); this may be preferable in view of better handling of edge effects and non-equally spaced x–points. There is a difficulty here as these local linear (or polynomial) smoothers are not guaranteed to yield an estimated conditional distribution that is a proper distribution function, i.e., nondecreasing in y with a left limit of 0 and right limit of 1; see Li and Racine (2007). 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). In addition, Hansen (2004) has proposed a quick-and-easy adjustment to the local linear

estimator that yields a proper distribution function while maintaining its favorable asymptotic properties. The local linear versions of $\hat{D}_x(y)$ and $\bar{D}_x(y)$ using Hansen's (2004) adjustment are given by:

$$\hat{D}_x^{LL}(y) = \frac{\sum_{i=1}^n w_i^{\diamond} \mathbf{1}(Y_i \leq y)}{\sum_{i=1}^n w_i^{\diamond}} \text{ and } \bar{D}_x^{LL}(y) = \frac{\sum_{i=1}^n w_i^{\diamond} \Lambda\left(\frac{y-Y_i}{h_0}\right)}{\sum_{i=1}^n w_i^{\diamond}}. \tag{4.7}$$

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

$$w_i^{\diamond} = \begin{cases} 0 & \text{when } \hat{\beta}(x-X_i) > 1 \\ w_i(1-\hat{\beta}(x-X_i)) & \text{when } \hat{\beta}(x-X_i) \leq 1 \end{cases} \tag{4.8}$$

where

$$w_i = \frac{1}{h} K\left(\frac{x-X_i}{h}\right) \text{ and } \hat{\beta} = \frac{\sum_{i=1}^n w_i(x-X_i)}{\sum_{i=1}^n w_i(x-X_i)^2}. \tag{4.9}$$

See Chap. 9 for an application of the above to a time series prediction problem.

Fact 4.2.1 *Under regularity conditions that include a well-behaved "density" $f(\cdot)$ (e.g., large-sample histogram) of the design points $x_1, \ldots, x_n$ and the assumption that, for all x, $D_x(y)$ is twice continuously differentiable as a function of y, it follows that $\bar{D}_x(y)$ satisfies an equation similar to Eq. (4.5), namely:*

$$\text{Var}\left(\bar{D}_x(y)\right) = O\left(\frac{1}{hn}\right) \text{ and } \text{Bias}\left(\bar{D}_x(y)\right) = O(h^2 + h_0^2) \tag{4.10}$$

assuming that $h_0 = o(h)$, $h \to 0$, $hn \to \infty$, and $\sqrt{hn}(h^3 + h_0^3) = o(1)$; see Theorem 6.2 of Li and Racine (2007). Furthermore, the two estimators $\bar{D}_x(y)$ and $\hat{D}_x(y)$ are asymptotically equivalent, *i.e., for any fixed x, $\sqrt{hn}\left(\bar{D}_x(y) - \hat{D}_x(y)\right) = o_p(1)$.*

Interestingly, although the two estimators $\bar{D}_x(y)$ and $\hat{D}_x(y)$ have Mean Squared Errors (MSE) that are of the same asymptotic order, smoothing may give a finite-sample advantage when the true $D_x(y)$ is smooth (at least twice continuously differentiable) as a function of y. Comparing Eq. (6.2) and (6.4) of Li and Racine (2007), it follows that:

$$\text{MSE}\left[\hat{D}_y(x)\right] - \text{MSE}\left[\bar{D}_y(x)\right] = c_{y,x}\frac{h_0}{nh} + o(\max\{h^4, \frac{1}{nh}\}) \tag{4.11}$$

where $c_{y,x} = C\frac{\partial}{\partial y} D_x(y)/f(x)$ for some constant $C \geq 0$.

Remark 4.2.4 (On choice of bandwidths) In order to minimize the asymptotic MSE of $\bar{D}_x(y)$, the optimal bandwidth specifications are $h \sim c_h n^{-1/5}$ and $h_0 \sim c_0 n^{-2/5}$ for some positive constants c_h, c_0. This suggests the following bandwidth choice rule-of-thumb which works reasonably well in practice: pick h via cross-validation, and then let $h_0 = h^2$.

Remark 4.2.5 (Quantile estimation) Recall that $D_x(y)$ is assumed continuous as a function of y. Suppose also that for some $\alpha \in [0,1]$ of interest, $D_x(y)$ is strictly increasing at $y = D_x^{-1}(\alpha)$. Then, the asymptotic consistency of $\bar{D}_x(y)$ that follows from Eq. (4.10) implies that the inverse $\bar{D}_x^{-1}(\alpha)$ will be consistent for $D_x^{-1}(\alpha)$; see, e.g., Lemma 1.2.1 of Politis et al. (1999). Similarly, Eq. (4.5) implies that $\hat{D}_x^{-1}(\alpha)$ is also consistent for $D_x^{-1}(\alpha)$ where $\hat{D}_x^{-1}(\alpha)$ now denotes the quantile inverse.

Recall that the Y_ts are non-i.i.d. only because they do not have identical distributions. Since they are continuous random variables, the *probability integral transform* is the key idea to transform them towards "i.i.d.–ness." To see why, note that if we let

$$\eta_i = D_{x_i}(Y_i) \quad \text{for } i = 1, \ldots, n$$

our transformation objective would be exactly achieved since $\eta_1, \ldots, \eta_n$ would be i.i.d. Uniform (0,1). Of course, $D_x(\cdot)$ is not known but we have the consistent estimator $\bar{D}_x(\cdot)$ as its proxy. Therefore, our practical transformation to be used in connection with the Model-Free Prediction Principle amounts to defining

$$u_i = \bar{D}_{x_i}(Y_i) \quad \text{for } i = 1, \ldots, n. \tag{4.12}$$

Claim 4.2.1 *Under the regularity conditions implicit in Fact 4.2.1, including the requirement that $D_x(y)$ is (absolutely) continuous in y for all x, the variables $u_1, \ldots, u_n$ are approximately i.i.d. Uniform (0,1).*

The word "approximately" in the above should be interpreted as "asymptotically" for large n; note that, technically, $u_1, \ldots, u_n$ represent the nth row of a triangular array although this is not explicitly denoted. A way to prove Claim 4.2.1 is to use the uniform consistency of $\bar{D}_x(\cdot)$, i.e., that $\sup_y |\bar{D}_x(y) - D_x(y)| \xrightarrow{P} 0$ under regularity conditions, and thus show that $u_i - \eta_i \xrightarrow{P} 0$ for each fixed i and x_i.

Remark 4.2.6 If a parametric specification for $D_x(y)$ happens to be available, i.e., if $P\{Y_t \leq y | x_t = x\}$ has known form up to a finite-dimensional parameter θ_x that may depend on x, then obviously our probability integral transform of Y_t would be based on the parametric distribution with parameter θ_x estimated from a local neighborhood of the associated regressor x_t.

The probability integral transform has been previously used by Ruppert and Cline (1994) as an intermediate step towards building better density estimators; however, our application is quite different as the following sections make clear.

4.3 Model-Free Optimal Predictors

4.3.1 Model-Free and Limit Model-Free Optimal Predictors

Since a transformation of the data towards i.i.d.–ness is available from Eq. (4.12), we can now formulate optimal predictors in the Model-free paradigm. As a first step, we formulate the inverse transformation needed in premise (b) of the Model-Free Prediction Principle. To do this, consider the inverse transformation $\bar{D}_{x_f}^{-1}$ which is well-defined since $\bar{D}_{x_f}(\cdot)$ is strictly increasing by construction. Note that, for any $i = 1, \ldots, n$, $\bar{D}_{x_f}^{-1}(u_i)$ is a *bona fide* potential response Y_f associated with predictor x_f since $\bar{D}_{x_f}^{-1}(u_i)$ has (approximately) the same distribution as Y_f. These n valid potential responses given by $\{\bar{D}_{x_f}^{-1}(u_i) \text{ for } i = 1, \ldots, n\}$ can be gathered together to give us an approximate empirical distribution for Y_f from which our predictors will be derived. Thus, analogously with the discussion associated with the entries of Table 3.1 from Chap. 3, it follows that the L_2–optimal predictor of $g(Y_f)$ will be the expected value of $g(Y_f)$ that is approximated by

$$\Pi_{x_f} = n^{-1} \sum_{i=1}^{n} g\left(\bar{D}_{x_f}^{-1}(u_i)\right). \tag{4.13}$$

Similarly, the L_1–optimal predictor of $g(Y_f)$ will be approximated by the sample median of the set $\{g\left(\bar{D}_{x_f}^{-1}(u_i)\right),\ i = 1, \ldots, n\}$. The model-free predictors are given in the middle column of Table 4.1 that can be compared to Table 3.1 of the previous chapter. Note that any of the two optimal model-free predictors (mean or median) can be used to give the equivalent of a model *fit*. To fix ideas, suppose we focus on the L_2–optimal case and that $g(x) = x$. Calculating the value of the optimal predictor of Eq. (4.13) for many different x_f values, e.g., taken on a grid, the equivalent of a nonparametric smoother of a regression function is constructed, and can be plotted over the (Y, x) scatterplot. In this sense, *Model-Free Model-Fitting* is achieved as discussed in Remark 2.2.2.

Remark 4.3.1 Following the discussion of Remark 4.2.3, recall that for $\bar{D}_{x_f}^{-1}$ to be an accurate estimator of $D_{x_f}^{-1}$, the value x_f must be such that it has an appreciable number of h-close neighbors among the original predictors $x_1, \ldots, x_n$ as discussed in Remark 4.2.2. As an extreme example, note that prediction of Y_f when x_f is outside the range of the original predictors $x_1, \ldots, x_n$, i.e., extrapolation, is *not* feasible in the model-free paradigm. It is also apparent that the Model-free predictors of Table 4.1 are still computable in the case where the x–variable is discrete-valued provided, of course, that $N_{x,h}$ the number of data points in the local neighborhood of each of these discrete values is large enough to permit accurate estimation of $D_{x_f}(\cdot)$ locally. What allows the method to work here—and also to still work in terms of predictive intervals to be developed shortly—is that x_f will by necessity be one of these discrete values as well.

	Model-free (MF)	LMF
L_2–predictor of Y_f	$n^{-1}\sum_{i=1}^{n}\bar{D}_{x_f}^{-1}(u_i)$	$\int_0^1 \hat{D}_{x_f}^{-1}(u)du$
L_1–predictor of Y_f	median$\{\bar{D}_{x_f}^{-1}(u_i),\ i=1,\ldots,n\}$	$\hat{D}_{x_f}^{-1}(1/2)$
L_2–predictor of $g(Y_f)$	$n^{-1}\sum_{i=1}^{n} g\left(\bar{D}_{x_f}^{-1}(u_i)\right)$	$\int_0^1 g\left(\hat{D}_{x_f}^{-1}(u)\right)du$
L_1–predictor of $g(Y_f)$	median$\{g\left(\bar{D}_{x_f}^{-1}(u_i),\ i=1,\ldots,n\right)\}$	$g\left(\hat{D}_{x_f}^{-1}(1/2)\right)$

Table 4.1 Middle column: Model-free (MF) optimal point predictors, where $u_i = \bar{D}_{x_i}(Y_i)$. Last column: Limit Model-Free (LMF) optimal point predictors; in the bottom LMF entry, it was assumed that $g(\cdot)$ is monotone [Recall that if $g(\cdot)$ is a monotone function, and X a random variable with median M, the median of $g(X)$ equals $g(M)$]

Finally, recall that under the Limit Model-Free (LMF) paradigm of Sect. 2.4.3, F is the limit distribution of the i.i.d. variables $u_1,\ldots,u_n$. In our case, F is known to be the Uniform (0,1) distribution which can be used explicitly in the construction of the optimal predictors that are given in the last column of Table 4.1. Also recall that under the LMF methodology it is not required to generate the i.i.d. variables $u_1,\ldots,u_n$. Hence, it is not necessary to estimate $D_x(y)$ as a continuous function (in y), and the smoothing step involved in constructing $\bar{D}_x(y)$ is not needed.[1] Thus, for the LMF point predictors, as well as the LMF prediction intervals developed in Sect. 4.4, all that is needed is a consistent estimator of $D_x^{-1}(\cdot)$ which can be immediately provided by the (quantile) inverse $\hat{D}_x^{-1}(\cdot)$; see Remark 4.2.5. Note that the Limit Model-free L_2– and L_1–optimal predictors are very intuitive, corresponding to the mean and median of the (estimated) conditional distribution of Y_f.

4.3.2 Asymptotic Equivalence of Point Predictors

To fix ideas, let us continue to focus on the simple case where $g(x) = x$, recall that the L_2–optimal predictor of Y_f associated with design point x_f is simply the conditional expectation $E(Y_f|x_f)$. The latter is well approximated by our kernel estimator m_{x_f} (or a local polynomial) even *without* the validity of model (3.1), therefore also qualifying to be called a model-free *point* predictor. Table 4.1 gave two alternative approximations to the theoretical conditional expectation $E(Y_f|x_f)$. How different are these expressions? To start with, note that the Nadaraya-Watson estimator m_{x_f} can be expressed alternatively as

[1] Nevertheless, if $D_x(y)$ is smooth in y, $\bar{D}_x(y)$ may be more accurate than $\hat{D}_x(y)$; see Eq. (4.11). So, if a practitioner has taken the trouble to construct $\bar{D}_x(y)$, they may well decide to use it in place of $\hat{D}_x(y)$ in all entries of Table 4.1.

$$m_{x_f} = \sum_{i=1}^{n} Y_i \tilde{K}\left(\frac{x_f - x_i}{h}\right) = \int y\, \hat{D}_{x_f}(dy) = \int_0^1 \hat{D}_{x_f}^{-1}(u)du; \tag{4.14}$$

where the last equality is due to the identity $\int y\, F(dy) = \int_0^1 F^{-1}(u)du$ that holds true for any distribution F. In other words, the LMF L_2–optimal predictor of Y_f from Table 4.1 is *identical* to the Nadaraya-Watson estimator. From the case $g(x) = x$ of Eq. (4.13) we have

$$\Pi_{x_f} = n^{-1} \sum_{i=1}^{n} \bar{D}_{x_f}^{-1}(u_i);$$

also denote

$$\hat{\Pi}_{x_f} = n^{-1} \sum_{i=1}^{n} \hat{D}_{x_f}^{-1}(u_i). \tag{4.15}$$

Under the regularity conditions implicit in Fact 4.2.1, it is not hard to show that Π_{x_f} and $\hat{\Pi}_{x_f}$ are asymptotically equivalent, i.e., that for any x_f, $\sqrt{nh}\,(\Pi_{x_f} - \hat{\Pi}_{x_f}) = o_p(1)$. But perhaps more important is a relationship of Π_{x_f} and $\hat{\Pi}_{x_f}$ to the Nadaraya-Watson smoother m_{x_f}. To motivate it, note that for large n, a law of large numbers gives the approximation

$$m_{x_f} = \int_0^1 \hat{D}_{x_f}^{-1}(u)du \simeq n^{-1} \sum_{i=1}^{n} \hat{D}_{x_f}^{-1}(u_i) = \hat{\Pi}_{x_f}. \tag{4.16}$$

Claim 4.3.1 *Under the regularity conditions implicit in Fact 4.2.1, including the assumption that $D_x(y)$ is continuous in x, and differentiable in y with derivative that is everywhere positive on its support, $\hat{\Pi}_{x_f}$ and m_{x_f} are* asymptotically equivalent, *i.e., $\sqrt{nh}\,(\hat{\Pi}_{x_f} - m_{x_f}) = o_p(1)$ for any x_f that is not a boundary point.*

One way to prove the above is to show that the average appearing in (4.15) is close to a Riemann sum approximation to the integral $\int_0^1 \hat{D}_{x_f}^{-1}(u)du$ from Eq. (4.16) based on a grid of n points. The law of the iterated logarithm for order statistics of uniform spacings can be useful here; see Devroye (1981) and the references therein.

Remark 4.3.2 The above line of arguments indicates that there is a variety of estimators that are asymptotically equivalent to m_{x_f} in the sense of Claim 4.3.1. For example, the Riemann sum $M^{-1}\sum_{k=1}^{M} \hat{D}_{x_f}^{-1}(k/M)$ is such an approximation as long as $M \geq n$. A stochastic approximation can also be concocted as $M^{-1}\sum_{i=1}^{M} \hat{D}_{x_f}^{-1}(U_i)$ where $U_1, \ldots, U_M$ are i.i.d. generated from a Uniform (0,1) distribution and $M \geq n$.

Remark 4.3.3 Reverting momentarily to the L_1–optimal predictors, note that the Model-free L_1–predictor of Y_f can be expressed as:

$$\text{median}\,\{\bar{D}_{x_f}^{-1}(u_i)\} = \bar{D}_{x_f}^{-1}(\text{median}\{u_i\}) \simeq \bar{D}_{x_f}^{-1}(1/2)$$

since the u_is are approximately Uniform (0,1). But $\bar{D}_{x_f}^{-1}(1/2) \simeq \hat{D}_{x_f}^{-1}(1/2)$; hence, the Model-free L_1–optimal point predictor is practically equivalent to the Limit Model-free L_1–optimal point predictor.

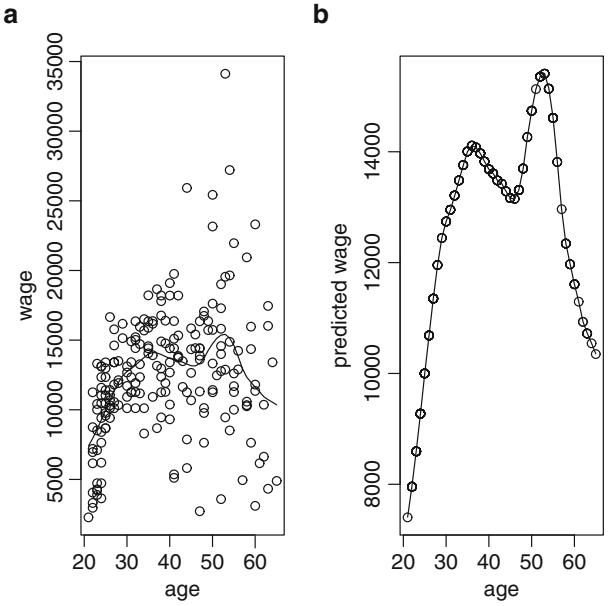

Fig. 4.2 (**a**) Wage vs. age scatterplot. (**b**) *Circles* indicate the salary predictor from Eq. (4.13) calculated from log-wage data with $g(x) = e^x$. In both figures, the superimposed *solid line* represents the MF L_2–optimal salary predictor calculated from the *raw* data (without the log-transformation)

Remark 4.3.4 From Claim 4.3.1, it is apparent that Model-free and Limit Model-free point predictors are asymptotically equivalent to the standard predictors based on the mean and median of the (estimated) conditional distribution of Y_f in the L_2 and L_1 cases, respectively. However, the advantages of the Model-free philosophy are twofold: (i) it allows us to go *beyond* the point predictions and obtain valid predictive distributions and intervals for Y_f as will be described in Sect. 4.4—this is simply not possible on the basis of the kernel estimator m_{x_f} without resort to a model like (3.1); and (ii) it is a totally automatic method that **relieves the practitioner from the need to find an optimal transformation for additivity and variance stabilization.** This is a significant practical advantage because of the multitude of such proposed transformations, e.g., the Box/Cox power family, ACE, AVAS, etc.; see Linton et al. (2008) and the references therein. For example, Fig. 4.2a depicts the `cps71` dataset using the raw salary data, i.e., without the logarithmic transformation employed in Fig. 3.1a; superimposed is the MF L_2–optimal predictor of salary that uses transformation (4.12) on the raw data. As Fig. 4.2b shows, the latter is virtually identical to the MF L_2–optimal predictor obtained from the logarithmically transformed data and then using an exponential as the function $g(x)$ for predictor (4.13). Furthermore, Fig. 4.3a shows the Q-Q plot of the transformed variables u_i based on the logarithmically transformed data whereas Fig. 4.3b is its analogue based on the raw data; in both cases, the uniformity seems to be largely achieved, and the practitioner can equally choose either domain to work with.

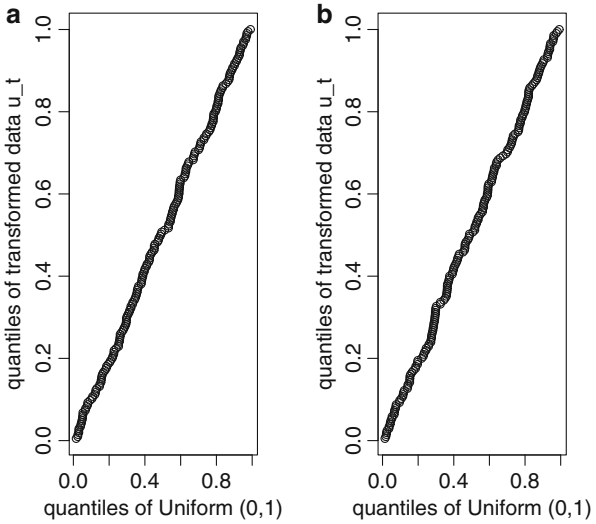

Fig. 4.3 Q-Q plots of the transformed variables u_i vs. the quantiles of Uniform (0,1) for the `cps71` data: (**a**) The u_i's are obtained from the log-wage vs. age dataset of Fig. 3.1 using bandwidth 5.5. (**b**) The u_i's are obtained from the raw (untransformed) dataset of Fig. 4.2 using bandwidth 7.3

4.3.3 Cross-Validation for Model-Free Prediction

As seen in the last subsection, estimating the conditional distribution $D_x(\cdot)$ by $\bar{D}_x(\cdot)$ or $\hat{D}_x(\cdot)$ is a crucial part of the Model-free procedures; the accuracy of this estimation depends on the choice of bandwidth h. Recall that cross-validation is a predictive criterion since it aims at minimizing the sum of squares (or absolute values) of *predictive* residuals. Nevertheless, we can still construct predictive residuals in model-free prediction, and thus cross-validation is possible in the model-free framework as well. To fix ideas, suppose we focus on the L_2–optimal predictor of Eq. (4.13), and let $\Pi_t^{(t)}$ denote the predictor of Y_t as computed from the delete-Y_t dataset: $\{(Y_i, x_i)$ for $i = 1, \ldots, t-1$ and $i = t+1, \ldots, n\}$, i.e., pretending the (Y_t, x_t) data pair is unavailable; this involves estimating $D_x(\cdot)$ by $\bar{D}_x^{(t)}(\cdot)$ computed from the delete-Y_t dataset, and having only $n-1$ values of u_i in connection with Eqs. (4.12) and (4.13). Finally, define the MF *predictive residuals*:

$$\tilde{e}_t = g(Y_t) - \Pi_t^{(t)} \quad \text{for } t = 1, \ldots, n. \tag{4.17}$$

Choosing the best bandwidth h to use in our model-free predictor (4.13) can then be based on minimizing PRESS=$\sum_{t=1}^n \tilde{e}_t^2$ or PRESAR=$\sum_{t=1}^n |\tilde{e}_t|$ as before. If $\hat{D}_x$ and $\bar{D}_x$ are based on k–nearest neighbor estimation as in Remark 4.2.3, then minimizing PRESS or PRESAR would yield the cross-validated choice of k to be used. Note that cross-validation using the MF predictive residuals of Eq. (4.17) can be quite computationally expensive. In view of Claim 4.3.1 arguing that the Model-free L_2–optimal

predictor (4.13) is asymptotically equivalent to a kernel smoother of the $(g(Y),x)$ scatterplot, it follows that cross-validation on the latter should give a quick approximate solution to the bandwidth choice for the Model-free predictors of Sect. 4.3.1 as well.

4.4 Model-Free Bootstrap

The empirical distribution of $g(Y_{\rm f})$ that was mentioned in the construction of predictor (4.13) cannot be regarded as a predictive distribution because it does not capture the variability of estimator $\bar{D}_x$; resampling gives us a way out of this difficulty once again. Generally, the predictive distribution and prediction intervals for $g(Y_{\rm f})$ can be obtained by the resampling algorithm of Sect. 2.4 that is re-cast below in the model-free regression framework. Let $g(Y_{\rm f})-\Pi_{x_{\rm f}}$ be the prediction root with $\Pi_{x_{\rm f}}$ denoting either the L_2- or L_1-optimal predictor from Table 4.1, i.e., $\Pi_{x_{\rm f}}=n^{-1}\sum_{i=1}^{n} g\left(\bar{D}_{x_{\rm f}}^{-1}(u_i)\right)$ or $\Pi_{x_{\rm f}}=\text{median}\left\{g\left(\bar{D}_{x_{\rm f}}^{-1}(u_i)\right)\right\}$. Then, our Model-free (MF) bootstrap algorithm for regression goes as follows.

Algorithm 4.4.1 RESAMPLING ALGORITHM FOR MF PREDICTIVE DISTRIBUTION AND PREDICTION INTERVALS FOR $g(Y_{\rm f})$

1. *Based on the Y–data, estimate the conditional distribution $D_x(\cdot)$ by $\bar{D}_x(\cdot)$, and use Eq. (4.12) to obtain the transformed data $u_1,\ldots,u_n$ that are approximately i.i.d.; let $\hat{F}_n$ denoted the empirical distribution of $u_1,\ldots,u_n$.*
   a. *Sample randomly (with replacement) the transformed data $u_1,\ldots,u_n$ to create bootstrap pseudo-data $u_1^*,\ldots,u_n^*$ whose empirical distribution is denoted $\hat{F}_n^*$.*
   b. *Use the inverse transformation $\bar{D}_x^{-1}$ to create pseudo-data in the Y domain, i.e., let $\underline{Y}_n^*=(Y_1^*,\ldots,Y_n^*)$ where $Y_t^*=\bar{D}_{x_t}^{-1}(u_t^*)$.*
   c. *Generate a bootstrap pseudo-response $Y_{\rm f}^*$ by letting $Y_{\rm f}^*=\bar{D}_{x_{\rm f}}^{-1}(u)$ where u is drawn randomly from the set $(u_1,\ldots,u_n)$.*
   d. *Based on the pseudo-data $\underline{Y}_n^*$, re-estimate the conditional distribution $D_x(\cdot)$; denote the bootstrap estimator by $\bar{D}_x^*(\cdot)$.*
   e. *Calculate a replicate of the bootstrap root $g(Y_{\rm f}^*)-\Pi_{x_{\rm f}}^*$ where we define $\Pi_{x_{\rm f}}^*=n^{-1}\sum_{i=1}^{n} g\left(\bar{D}_{x_{\rm f}}^{*^{-1}}(u_i^*)\right)$ or $\Pi_{x_{\rm f}}^*=\text{median}\left\{g\left(\bar{D}_{x_{\rm f}}^{*^{-1}}(u_i^*)\right)\right\}$ according to whether L_2- or L_1-optimal prediction has been used for the original $\Pi_{x_{\rm f}}$.*
2. *Steps (a)—(e) in the above are repeated B times, and the B bootstrap root replicates are collected in the form of an empirical distribution with $\alpha-$quantile denoted by $q(\alpha)$.*
3. *Then, the model-free $(1-\alpha)100\%$ equal-tailed, prediction interval for $g(Y_{\rm f})$ is*

$$[\Pi_{x_{\rm f}}+q(\alpha/2),\Pi_{x_{\rm f}}+q(1-\alpha/2)] \tag{4.18}$$

and our estimate of the predictive distribution of $g(Y_{\rm f})$ is the empirical distribution of bootstrap roots obtained in step 2 shifted to the right by the number $\Pi_{x_{\rm f}}$.

Remark 4.4.1 (On edge effects) Smoothing techniques are often plagued by edge effects, and this is especially true for kernel smoothers; Figs. 3.1a and 4.2a show the bias problems near the left boundary. Thus, to implement the MF Resampling Algorithm for prediction intervals given in this section—but also to construct the MF point predictors of Table 4.1—it is practically advisable to only include the u_is obtained from x_is that are away from either boundary by more than a bandwidth. From these u_is, a full-size resample $(u_1^*, \ldots, u_n^*)$ can be generated that, in turn, gives rise to a full-size pseudo-sample $(Y_1^*, \ldots, Y_n^*)$ which allows us to compute the bootstrap estimator $\bar{D}_x^*(\cdot)$. Similarly, only the Y^*s that are away from the boundaries by more than a bandwidth will be used in the construction of $\Pi_{x_f}^*$ in Step 1(e) above.

Algorithm 4.4.1 is essentially the Model-free Bootstrap Algorithm 2.4.1 as applied to the nonparametric regression setup. We could devise an analog of the Limit Model-Free Bootstrap Algorithm 2.4.3 by just modifying Step 1(a) of Algorithm 4.4.1, i.e., having $u_1^*, \ldots, u_n^*$ drawn as i.i.d. from F (instead of $\hat{F}_n$); here, of course, the limit distribution F is Uniform (0,1). Substituting F instead of $\hat{F}_n$ makes little impact in practice because even with reasonably big sample sizes, $\hat{F}_n$ is already very close to Uniform (0,1); see, e.g., Fig. 4.3. Note, however, that defining $u_1^*, \ldots, u_n^*$ to be i.i.d. Uniform (0,1) does avoid the edge effects issues mentioned in Remark 4.4.1. Hence, we will reserve the term Limit Model-Free (LMF) bootstrap in regression for the following algorithm that takes it a step further: since we do not need to rely (or even construct) the "uniformized" data $u_1, \ldots, u_n$, the whole algorithm can be based on the step function estimator $\hat{D}_x(\cdot)$ instead of the smoothed $\bar{D}_x(\cdot)$. As $\hat{D}_x(\cdot)$ is a step function, $\hat{D}_x^{-1}(\beta)$ will be interpreted as a *quantile inverse*, i.e., $\hat{D}_x^{-1}(\beta) = \inf\{y \text{ such that } \hat{D}_x(y) \geq \beta\}$. As before, let Π_{x_f} denote either the L_2– or L_1–optimal predictor from Table 4.1, i.e., $\Pi_{x_f} = n^{-1} \sum_{i=1}^n g\left(\bar{D}_{x_f}^{-1}(u_i)\right)$ or $\Pi_{x_f} = \text{median}\,\{g\left(\bar{D}_{x_f}^{-1}(u_i)\right)\}$.

Algorithm 4.4.2 LMF Resampling Algorithm for predictive distribution and prediction intervals for $g(Y_f)$

1. *Based on the Y–data, estimate the conditional distribution $D_x(\cdot)$ by $\hat{D}_x(\cdot)$.*
   a. *Generate bootstrap pseudo-data $u_1^*, \ldots, u_n^*$ i.i.d. Uniform $(0,1)$, and denote $\hat{F}_n^*$ the empirical distribution of $u_1^*, \ldots, u_n^*$.*
   b. *Use the quantile inverse $\hat{D}_x^{-1}$ to create pseudo-data in the Y domain, i.e., let $\underline{Y}_n^* = (Y_1^*, \ldots, Y_n^*)$ where $Y_t^* = \hat{D}_{x_t}^{-1}(u_t^*)$.*
   c. *Generate a bootstrap pseudo-response Y_f^* by letting $Y_f^* = \hat{D}_{x_f}^{-1}(u)$ where u is drawn randomly from a Uniform $(0,1)$ distribution.*
   d. *Based on the pseudo-data $\underline{Y}_n^*$, re-estimate the conditional distribution $D_x(\cdot)$ by the step-function estimator denoted by $\hat{D}_x^*(\cdot)$.*
   e. *Calculate a replicate of the bootstrap root $g(Y_f^*) - \Pi_{x_f}^*$ where we define $\Pi_{x_f}^* = n^{-1} \sum_{i=1}^n g\left(\hat{D}_{x_f}^{*^{-1}}(u_i^*)\right)$ or $\Pi_{x_f}^* = \text{median}\,\{g\left(\hat{D}_{x_f}^{*^{-1}}(u_i^*)\right)\}$ according to whether L_2– or L_1–optimal prediction has been used for the original Π_{x_f}.*

2. *Steps (a)—(e) in the above are repeated B times, and the B bootstrap root replicates are collected in the form of an empirical distribution with α–quantile denoted by $q(\alpha)$.*
3. *The Limit Model-Free $(1-\alpha)100\%$ equal-tailed, prediction interval for $g(Y_f)$ is*

$$[\Pi_{x_f} + q(\alpha/2), \Pi_{x_f} + q(1-\alpha/2)] \tag{4.19}$$

and our estimate of the predictive distribution of $g(Y_f)$ is the empirical distribution of bootstrap roots obtained in step 2 shifted to the right by the number Π_{x_f}.

Remark 4.4.2 (On discrete responses) Until now, it has been assumed that $D_x(y)$ is continuous both in x and y. Continuity in x is a *sine qua non* in terms of estimating $D_x(\cdot)$ by a local (in x) window technique. However, it is interesting to see that the above LMF algorithm remains valid *verbatim* when $D_x(y)$ is the distribution of a discrete random variable, and even when $D_x(y)$ is the distribution of a mixed discrete–continuous random variable. The reason is that the step function estimator $\hat{D}_x(y)$ remains consistent in this case as it does not rely on continuity of $D_x(y)$ in y. However, when $g(Y_f)$ takes values in a countable set, the caveats of Sect. 2.4.4 apply; in this case, optimality of the point predictor should be gauged with respect to 0–1 loss, and the whole predictive distribution of $g(Y_f)$ should be estimated/presented instead of just prediction intervals.

As we have seen, the LMF method has two advantages namely: (a) it is not affected by edge effect corruption of $u_1, \ldots, u_n$, and (b) it can accommodate discrete (or mixed) responses. However, it is the basic MF method that affords us a generalization that is analogous to using the predictive residuals in model-based regression in Sect. 3.5; this is the subject of the following section.

4.5 Predictive Model-Free Bootstrap

The success of the MF/MB method of Sect. 3.5 was based on the fact that the distribution of the prediction error can be approximated better by the (empirical) distribution of the predictive residuals as compared to the (empirical) distribution of the fitted residuals. Using the latter—as in the Model-Based (MB) method—typically results in variance underestimation and under-coverage of prediction intervals. Since Model-Free (MF) predictive residuals are computable from Eq. (4.17), one might be tempted to try to use them in order to mimic the MF/MB construction. Unfortunately, the MF predictive residuals of Eq. (4.17) are *not* i.i.d. in the model-free context of the present chapter; hence, i.i.d. bootstrap on them is not recommended. In what follows, we will try to identify analogs of the i.i.d. predictive residuals in this model-free setting. Recall that the accuracy of our bootstrap prediction intervals hinges on the accuracy of the approximation of the prediction root $g(Y_f) - \Pi_{x_f}$ by its bootstrap analog, namely $g(Y_f^*) - \Pi_{x_f}^*$. However, Π_{x_f} is based on

a sample of size n, and Y_f is *not* part of the sample. Using predictive residuals is a trick that helps the bootstrap root mimic this situation by making Y_f^* into a genuinely "out-of-the-sample" point; the reason is that *every* data point is treated as an "out-of-the-sample" point as far as the computation of predictive residuals is concerned. We can still achieve this effect within the basic MF paradigm using an analogous trick; to see how, let $\bar{D}_{x_t}^{(t)}$ denote the estimator $\bar{D}_{x_t}$ as computed from the delete-Y_t dataset: $\{(Y_i, x_i),\ i = 1, \ldots, t-1 \text{ and } i = t+1, \ldots, n\}$. Now let

$$u_t^{(t)} = \bar{D}_{x_t}^{(t)}(Y_t) \quad \text{for } t = 1, \ldots, n; \tag{4.20}$$

the $u_t^{(t)}$ variables will serve as the analogs of the predictive residuals $\tilde{e}_t$ of Sect. 3.5. Although the latter are approximately i.i.d. *only* when model (3.1) holds true, the $u_t^{(t)}$s are approximately i.i.d. in general under the weak assumptions of smoothness of $D_x(y)$. The Predictive Model-Free (PMF) bootstrap algorithm goes as follows.

Algorithm 4.5.1 PMF RESAMPLING ALGORITHM FOR PREDICTIVE DISTRIBUTION AND PREDICTION INTERVALS FOR $g(Y_f)$

- *The PMF Resampling Algorithm is identical to Algorithm 4.4.1 with one exception: replace the variables $u_1, \ldots, u_n$ by $u_1^{(1)}, \ldots, u_n^{(n)}$ throughout the construction.*

In addition, the PMF optimal point predictors are identical to the MF predictors given in the middle column of Table 4.1 with the same exception: replace the variables $u_1, \ldots, u_n$ by $u_1^{(1)}, \ldots, u_n^{(n)}$. However, as was the case in the model-based case of Sect. 3.5, there is little advantage in using the PMF point predictors. It is the finite-sample variability of $u_1^{(1)}, \ldots, u_n^{(n)}$ that is generally bigger compared to that of $u_1, \ldots, u_n$ that results into prediction intervals with better coverage.

4.6 Model-Free Diagnostics

The three Model-Free prediction schemes in regression (MF, LMF, and PMF) have been developed under minimal assumptions, e.g., continuity of $D_x(y)$ in both x and y—although the latter can be dropped in the LMF scheme—and availability of enough data so that "local" estimation can take place. With regards to the latter, traditional conditions for asymptotic validity would include the usual requirement that $h \to 0$ as $n \to \infty$ but also ensuring $N_{x,h} \to \infty$ for all x over an interval of interest; see Remark 4.2.2. For good finite-sample results, however, we would like $D_x(\cdot)$ to remain largely unchanged over an x–interval of length $2h$, where h is the chosen bandwidth in the practical application. Hence continuity is not enough in practice; what is needed is that $D_x(\cdot)$ is changing *smoothly* with x. As an illustration, consider the following problematic model $Y_t = \beta_0 + \beta_1 x_t + c_t \varepsilon_t$ where $x_t = t$ for $t = 1, \ldots, n$, $c_t = \mathbf{1}\{t \geq n/2\}$, and $\varepsilon_t \sim$ i.i.d. $N(0, \sigma^2)$. The change-point that is

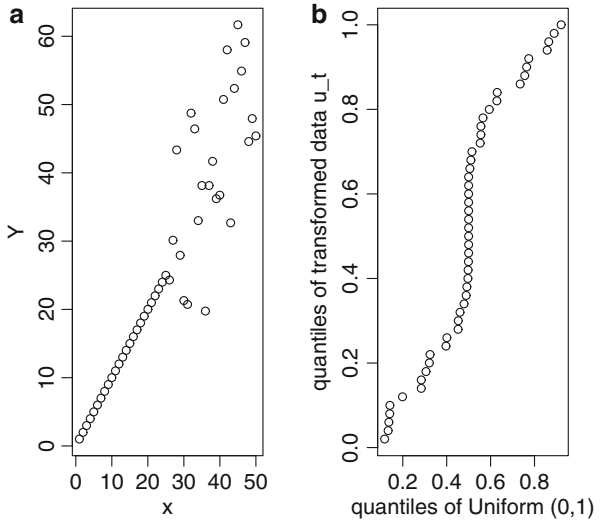

Fig. 4.4 (**a**) Scatterplot of model $Y = 2x + \mathbf{1}\{x \geq 25\} \cdot \varepsilon_x$ for $x = 1, \ldots, 50$ with $\varepsilon_x \sim$ i.i.d. $N(0, 100)$. (**b**) Q-Q plot of the transformed variables u_t vs. the quantiles of Uniform (0,1)

present in the error variance obviously negates our practical requirement that $D_x(\cdot)$ changes slowly with x; see Fig. 4.4a for an illustration. Letting $u_t = \bar{D}_t(Y_t)$, it is easy to see that $u_t \simeq 1/2$ for all $t < n/2$, but $u_t \sim$ i.i.d. Uniform (0,1) for $t \geq n/2$. This mixed quality of the transformed variables u_t causes the basic Model-free prediction method to break down. Interestingly, the LMF method is robust in such a setting as it does not rely on the variables $u_1, \ldots, u_n$. The LMF method would be problematic here only in the neighborhood of the change-point, i.e., for x within h of the middle value. Fortunately, the problem can be diagnosed by an exploratory investigation of the transformed variables u_i much like the usual diagnostics on residuals in regression. It is obvious that non-uniformity of the u_is is a red flag, and can be easily diagnosed by a histogram and/or Q-Q plot. In particular, if the distribution of the u_is appears to contain a point mass at 1/2 or elsewhere, then a problem is identified; for example, the Q-Q plot of Fig. 4.4b clearly indicates the presence of a point mass on 1/2.

4.7 Simulations

4.7.1 When a Nonparametric Regression Model Is True

The building block for the simulation in this subsection is model (3.1) with $\mu(x) = \sin(x)$, $\sigma(x) = 1/2$, and errors ε_t that are i.i.d. N(0,1) or Laplace (two-sided exponential) rescaled to unit variance. Knowledge that the variance $\sigma(x)$ is constant

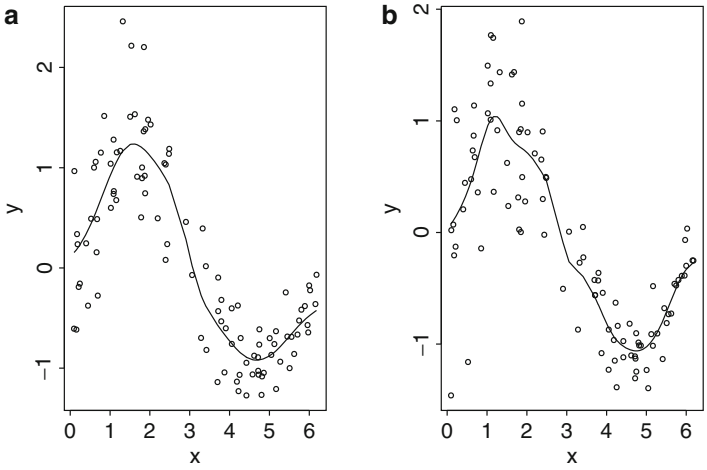

Fig. 4.5 Typical scatterplots with superimposed kernel smoothers; (**a**) Normal data; (**b**) Laplace data

was not used in the estimation procedures, i.e., $\sigma(x)$ was estimated from the data. For each distribution, 500 datasets each of size $n = 100$ were created with the design points $x_1, \ldots, x_n$ being equi-spaced on $(0, 2\pi)$, and Nadaraya-Watson estimates of $\mu(x) = E(Y|x)$ and $\sigma^2(x) = \text{Var}(Y|x)$ were computed using a normal kernel in R. Two typical datasets are depicted in Fig. 4.5. Prediction intervals with nominal level 90 % were constructed using the two methods presented in Chap. 3: Model-Based (MB) and Model-Free/Model-Based (MF/MB); the three methods presented in this chapter: Model-Free (MF), Limit Model-Free (LMF) and Predictive Model-Free (PMF), and the NORMAL approximation interval (3.17). For all methods (except the LMF and the NORMAL) the correction of Remark 4.4.1 was employed. The required bandwidths were computed by L_1 (PRESAR) cross-validation. For simplicity—and to guarantee that $M_x \geq m_x^2$—equal bandwidths were used for both m_x and M_x, i.e., the constraint $h = q$ was imposed. Before evaluating the performance of the resulting prediction intervals, it is of interest to check whether the u_i defined in (4.12) are indeed "uniformized" as their usage in the MF and PMF procedures requires. From each of the 500 replications, the set of $u_1, \ldots, u_n$ was constructed, and compared to the Uniform (0,1) via a Kolmogorov-Smirnov (K-S) test. Only 1 out of the 500 cases resulted in a rejection of the Uniform (0,1) null hypothesis at level 0.05. This could be regarded as good news for the "uniformize" procedure of Eq. (4.12) but it also underscores an interesting issue: the variability of the K-S distances is smaller than that expected from i.i.d. Uniform (0,1) samples, and that is why the number of rejections is smaller than expected. The reason for this reduced variability could be attributed to the fact that the $u_1, \ldots, u_n$ are not exactly independent in our finite-sample setup; instead, they exhibit lag-1 and lag-2 autocorrelations of the order of -0.07 which is not statistically significant but nevertheless present. The negative—albeit small—autocorrelation may result into a reduced

probability of clustering of the $u_1, \ldots, u_n$ data, and therefore explain the reduced variability of the K-S statistics. Note, however, that this is a finite-sample effect; with a larger n, the bandwidth h decreases, and so does the correlation present in the $u_1, \ldots, u_n$ data. In any case, this correlation is destroyed in the bootstrap reshuffling that is implemented in the MF and PMF procedures. For each type of prediction interval constructed, the corresponding empirical coverage level (CVR) and average length (LEN) were recorded together with the (empirical) standard error associated with each average length. The standard error of the reported coverage levels over the 500 replications is 0.013; notably, these coverage levels represent *overall* (i.e., unconditional) probabilities in the terminology of Cox (1975) and Beran (1990). As previously mentioned, in the practical construction of bootstrap predictive intervals one would employ a large number of bootstrap simulations, e.g., $B = 999$; we did so here, effectively re-running the simulations of Politis (2013) that were based on $B = 249$ due to limitations on computing time. Each method, NORMAL, MB, etc., is represented by 3 lines of entries in Tables 4.2 and 4.3. The first line of entries gives the empirical coverage levels (CVR) of prediction intervals calculated at several x_f points spanning the interval $(0, 2\pi)$; nominal coverage was **0.90**. The second line of entries gives the average length (LEN) of the corresponding interval; and the third line gives the standard error associated with interval length. Tables 4.2 and 4.3 summarize our findings, and contain a number of important features.

- As mentioned before, the standard error of the reported CVRs is 0.013. In addition, note that—by construction—this simulation problem has some symmetry that helps us further appreciate the variability of the CVRs. For example, the expected CVRs should be the same for $x_f = 0.3\pi$ and 1.7π in all methods; so for the NORMAL case of Table 4.2, the CVR would be better estimated by the average of 0.886 and 0.866, i.e., closer to 0.876. Similarly, the CVR of PMF for $x_f = 0.15\pi$ in Table 4.3 can be better estimated by $(0.918 + 0.878)/2 = 0.898$.
- The NORMAL intervals are characterized by under-coverage even when the true distribution is Normal. This under-coverage is a bit more pronounced when $x_f = \pi/2$ or $3\pi/2$ due to the high bias of the kernel estimator at the points of a "peak" or "valley" that the normal interval (3.17) "sweeps under the carpet."
- The length of the NORMAL intervals is quite less variable than those based on bootstrap; this should come as no surprise since the extra randomization implicit in any bootstrap procedure is expected to inflate the overall variances. [Note that the standard deviation of the length can be estimated by st. err. $\times\sqrt{500}$.]
- The MF/MB intervals are *always* more accurate (in terms of coverage) than their MB analogs in Tables 4.2 and 4.3. This was not unexpected since (i) the regression model (3.1) holds true here; (ii) bootstrap model-based intervals are expected to under-cover; and (iii) by Fact 3.5.1, MF/MB intervals are expected to be wider, and therefore partially correct this under-coverage. The increase in coverage of the MF/MB intervals comes at the cost of increased variability of interval length.

- The three Model-free methods have generally comparable coverages and variabilities; in addition, they are all comparable to the MF/MB intervals. Hence, it appears there is little to lose in conducting model-free inference even when the model is true.
- All bootstrap methods (model-free and model-based) result in overcoverage when $x_f \approx \pi$; this could be explained by the phenomenon of *"bias leakage"* that will be discussed in more detail below. The only bootstrap method that, in principle, should be immune to "bias leakage" is LMF; this is confirmed in Table 4.3 while the unusually large value of 0.926 of Table 4.2 could be attributed to the randomness of the simulation (it is two standard errors away from 0.90).

The case $x_f \approx \pi$ deserves special discussion. In principle, this should be an easy case since kernel smoothers have approximately zero bias there. Nevertheless, smoothers will have appreciable bias at *all* other points where the curvature is nonzero, and in particular, at the peak/valley points $x_f = \pi/2$ and $x_f = 3\pi/2$. This bias is passed on to the residuals (fitted, predictive, or even the u_i variables of MF and PMF) in the following way: residuals obtained near the point $x_f = \pi/2$ will tend to be larger (their distribution being skewed right), while residuals near the point $x_f = 3\pi/2$ will

$x_f/\pi =$	0.15	0.3	0.5	0.75	1	1.25	1.5	1.7	1.85
	0.878	0.886	0.854	0.886	0.878	0.860	0.876	0.866	0.870
NORMAL	1.6147	1.6119	1.6117	1.6116	1.6117	1.6116	1.6117	1.6119	1.6146
	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006
	0.852	0.864	0.818	0.854	0.878	0.866	0.802	0.808	0.818
MB	1.6021	1.5326	1.4547	1.5855	1.7120	1.5955	1.4530	1.5223	1.5666
	0.013	0.013	0.012	0.014	0.015	0.013	0.012	0.012	0.013
	0.904	0.894	0.890	0.900	0.928	0.910	0.870	0.888	0.896
MF/MB	1.8918	1.8097	1.7248	1.8602	2.006	1.8669	1.7170	1.7930	1.8482
	0.017	0.016	0.017	0.016	0.016	0.015	0.016	0.015	0.016
	0.916	0.872	0.860	0.898	0.926	0.910	0.888	0.914	0.890
LMF	1.8581	1.7730	1.6877	1.8286	1.9685	1.8334	1.6921	1.7681	1.8213
	0.016	0.015	0.014	0.016	0.017	0.015	0.015	0.015	0.015
	0.910	0.888	0.902	0.892	0.906	0.922	0.874	0.896	0.894
MF	1.8394	1.7531	1.6784	1.8117	1.9423	1.8139	1.6808	1.7500	1.8085
	0.016	0.015	0.014	0.016	0.017	0.016	0.015	0.015	0.015
	0.900	0.884	0.880	0.906	0.912	0.912	0.884	0.890	0.902
PMF	1.8734	1.7814	1.7013	1.8394	1.9705	1.8462	1.7076	1.7759	1.8339
	0.016	0.014	0.014	0.015	0.016	0.015	0.014	0.014	0.015

Table 4.2 Simulation results for additive model with i.i.d. Normal errors

$x_f/\pi =$	0.15	0.3	0.5	0.75	1	1.25	1.5	1.7	1.85
NORMAL	0.886	0.892	0.872	0.896	0.896	0.878	0.894	0.904	0.890
	1.6296	1.6268	1.6266	1.6265	1.6266	1.6266	1.6266	1.6268	1.6296
	0.008	0.008	0.008	0.008	0.008	0.008	0.008	0.008	0.008
MB	0.872	0.836	0.856	0.868	0.890	0.860	0.846	0.880	0.868
	1.5881	1.5743	1.5114	1.6276	1.7526	1.6255	1.4487	1.5426	1.5804
	0.017	0.017	0.018	0.017	0.017	0.017	0.016	0.016	0.016
MF/MB	0.914	0.904	0.906	0.898	0.938	0.898	0.892	0.874	0.912
	1.8663	1.8602	1.7735	1.9157	2.044	1.9043	1.7049	1.8130	1.8575
	0.021	0.022	0.022	0.020	0.020	0.020	0.020	0.020	0.019
LMF	0.902	0.868	0.904	0.912	0.910	0.912	0.870	0.888	0.886
	1.8418	1.8470	1.8034	1.8777	1.9907	1.8978	1.7110	1.8025	1.8361
	0.022	0.022	0.025	0.022	0.021	0.022	0.021	0.021	0.021
MF	0.898	0.884	0.886	0.914	0.938	0.904	0.874	0.860	0.866
	1.8134	1.8307	1.7847	1.8632	1.9704	1.8756	1.7054	1.7932	1.8282
	0.022	0.022	0.025	0.023	0.021	0.023	0.022	0.021	0.022
PMF	0.918	0.910	0.868	0.880	0.946	0.928	0.882	0.842	0.878
	1.8504	1.8633	1.8090	1.8954	1.9953	1.8995	1.7236	1.8144	1.8341
	0.022	0.022	0.024	0.022	0.021	0.022	0.021	0.021	0.020

Table 4.3 Simulation results for additive model with i.i.d. Laplace errors

tend to be smaller (more negative, i.e., skewed left). By the bootstrap reshuffling of residuals, the skewness disappears but an artificial inflation of the residual distribution ensues; this contamination of the residual pool may adversely influence the prediction interval coverage. This is the phenomenon previously referred to as *"bias leakage"* that is expected to result in *over*-coverage of bootstrap prediction (or confidence) intervals at points where the regression function has small curvature. "Bias leakage" would be alleviated with a larger sample size and/or using higher-order smoothing kernels or other low bias approximation methods, e.g., wavelets. It could also be alleviated using bandwidth tricks such as *undersmoothing*—see the detailed discussion in Remark 3.5.2.

4.7.2 When a Nonparametric Regression Model Is Not True

In this subsection, we investigate the performance of the different prediction intervals in a setup where model (3.1) is not true. For easy comparison with Sect. 4.7.1, we will keep the same (conditional) mean and variance, i.e., we will generate

$x_f/\pi =$	0.15	0.3	0.5	0.75	1	1.25	1.5	1.7	1.85
	0.906	0.890	0.890	0.884	0.908	0.900	0.870	0.890	0.872
NORMAL	1.5937	1.5911	1.5908	1.5908	1.5908	1.5908	1.5908	1.5911	1.5937
	0.009	0.009	0.009	0.009	0.009	0.009	0.009	0.009	0.009
	0.846	0.878	0.860	0.882	0.894	0.862	0.804	0.814	0.826
MB	1.4846	1.4530	1.3485	1.5421	1.6795	1.5329	1.4012	1.4745	1.5315
	0.021	0.019	0.018	0.019	0.019	0.017	0.015	0.014	0.014
	0.928	0.946	0.886	0.964	0.932	0.912	0.846	0.862	0.880
MF/MB	1.8161	1.7776	1.6409	1.8833	2.051	1.8695	1.7162	1.8017	1.8609
	0.031	0.025	0.023	0.026	0.024	0.022	0.021	0.019	0.019
	0.916	0.934	0.908	0.928	0.918	0.898	0.846	0.884	0.900
LMF	1.7555	1.7460	1.5870	1.8489	1.9798	1.7985	1.6652	1.7407	1.8094
	0.027	0.025	0.023	0.024	0.024	0.020	0.019	0.017	0.017
	0.908	0.932	0.882	0.910	0.906	0.910	0.860	0.876	0.876
MF	1.7344	1.7265	1.5561	1.8300	1.9345	1.7707	1.6355	1.7181	1.7963
	0.027	0.025	0.023	0.025	0.023	0.020	0.019	0.017	0.018
	0.926	0.936	0.932	0.922	0.932	0.872	0.872	0.902	0.902
PMF	1.7748	1.7636	1.5991	1.8550	1.9898	1.8083	1.6737	1.6737	1.8246
	0.026	0.024	0.022	0.023	0.023	0.019	0.019	0.016	0.017

Table 4.4 Simulation results using regression model with non-identically distributed errors

independent Y data such that $E(Y|x) = \sin(x)$, $\mathrm{Var}(Y|x) = 1/4$, and design points $x_1, \ldots, x_{100}$ equi-spaced on $(0, 2\pi)$ as before. However, the error structure $\varepsilon_x = (Y - E(Y|x))/\sqrt{\mathrm{Var}(Y|x)}$ may have skewness and/or kurtosis that depends on x, thereby violating the i.i.d. assumption. For our simulation we considered the simple construction:

$$\varepsilon_x = \frac{c_x Z + (1 - c_x)W}{\sqrt{c_x^2 + (1 - c_x)^2}} \tag{4.21}$$

where $c_x = x/(2\pi)$ for $x \in [0, 2\pi]$, and $Z \sim N(0, 1)$ independent of W that has mean zero and variance one but has an exponential shape, i.e., it is distributed as $\frac{1}{2}\chi_2^2 - 1$, to capture a changing skewness. Table 4.4 presents our findings; it is qualitatively similar to Tables 4.2 and 4.3 although the problem at hand is more complicated because of the skewness or kurtosis changing with x. In particular:

- Note the coverage of the NORMAL intervals decreases monotonically as x_f increases, yielding correct coverage in the region where skewness exists, and under-coverage in the region with (close to) normal errors; this is counter-intuitive but explained by the fact that the NORMAL interval undercovers in

the case of normal data; see Table 4.2. The boost in coverage in the case of a skewed distribution is a fluke.
- The MF/MB intervals correct (and sometimes over-correct) the under-coverage of MB intervals.
- The three model-free methods perform well throughout, with the PMF being the most conservative.

All in all, the Model-free Prediction Principle suggests ways to do inference in nonparametric regression in the presence or absence of an additive regression model. Interestingly, the Model-free methodology seems competitive to the model-based ones when an additive model is true, suggesting that there is little loss in adopting the Model-free approach throughout.

Acknowledgements

Chapters 3 and 4 are based on the paper: D.N. Politis (2013). "Model-free Model-fitting and Predictive Distributions" (with Discussion), *Test*, vol. 22, no. 2, pp. 183–250. Figures and Tables are reproduced with kind permission of Springer Science+Business Media. Sincere thanks are due to Srinjoy Das and Liang Wang for correcting, updating, and parallelizing the software, and re-running all simulations on the UCSD supercomputer cluster. The author is grateful to the Editors of *Test*, Ricardo Cao and Domingo Morales, for hosting the above Discussion paper, and the discussants: Juan Cuesta-Albertos, Manuel Febrero-Bande, Ingrid Van Keilegom, Stefan Sperlich, and Carlos Velasco for their inspiring comments. In particular, the favorable comments of Juan Cuesta-Albertos for one of the methods mentioned in Remark 4.5 of Politis (2013) led to the present definition of the Limit Model-Free variation of the methodology. Also note that the terminology MF^2 and MF/MF^2 of Politis (2013) has been replaced by the less cumbersome MF and PMF, respectively.

Appendix 1: High-Dimensional and/or Functional Regressors

So far in Chap. 4, it has been assumed for simplicity that the regressors are univariate; we now relax this assumption and show how the Model-free ideas are immediately applicable bearing in mind, of course, the curse of dimensionality. Throughout this Appendix we consider regression data $(Y_1, x_1), \ldots, (Y_n, x_n)$ where Y_k is the *univariate* response associated with a regressor value x_k that takes values in a linear vector space $\mathbf{E}$ equipped with a semi-metric $d(\cdot, \cdot)$. The space $\mathbf{E}$ can be high-dimensional or even infinite-dimensional, e.g., a function space; see Chap. 5 of Ferraty and Vieu (2006) for details. We will assume that the data adhere to the Model-free regression setup defined in Sect. 4.2. As before, we can estimate

$D_x(y) = P\{Y_j \leq y | X_j = x\}$ by the "local" weighted average

$$\hat{D}_x(y) = \sum_{i=1}^{n} \mathbf{1}\{Y_i \leq y\} \tilde{K}\left(\frac{d(x,x_i)}{h}\right) \tag{4.22}$$

where $\tilde{K}\left(h^{-1}d(x,x_i)\right) = K\left(h^{-1}d(x,x_i)\right)/\sum_{k=1}^{n} K\left(h^{-1}d(x,x_k)\right)$, the kernel K is a bounded, symmetric probability density with compact support, and $h > 0$ is the bandwidth parameter. For any fixed y, estimator $\hat{D}_x(y)$ is just a Nadaraya-Watson smoother of the variables $\mathbf{1}\{Y_i \leq y\}$ for $i = 1, \ldots, n$. As such, it is discontinuous as a function of y; to come up with a smooth estimator, we replace $\mathbf{1}\{Y_i \leq y\}$ by $\Lambda\left(\frac{Y_i - y}{h_0}\right)$ in Eq. (4.22), leading to the estimator

$$\bar{D}_x(y) = \sum_{i=1}^{n} \Lambda\left(\frac{Y_i - y}{h_0}\right) \tilde{K}\left(\frac{d(x,x_i)}{h}\right) \tag{4.23}$$

where h_0 is another bandwidth parameter, and $\Lambda(y) = \int_{-\infty}^{y} \lambda(s)ds$ with $\lambda(\cdot)$ being a symmetric density function that is continuous and strictly positive over its support. As a result, $\bar{D}_x(y)$ is differentiable and strictly increasing in y. Assuming Eq. (4.2) and additional regularity conditions, e.g., that as $n \to \infty$, $\max(h, h_0) \to 0$ but not too fast, Ferraty and Vieu (2006, Theorem 6.4) showed that

$$\bar{D}_x(y) \xrightarrow{a.s.} D_x(y) \text{ for any } y, \text{ and } \bar{D}_x^{-1}(\alpha) \xrightarrow{a.s.} D_x^{-1}(\alpha) \tag{4.24}$$

for any $\alpha \in [0,1]$ as long as $D_x(y)$ is strictly increasing at $y = D_x^{-1}(\alpha)$. It is conjectured that a similar consistency result can be obtained in the case of deterministic regressors that follow a regular design. Conditionally on event $S_n = \{X_j = x_j \text{ for } j = 1, \ldots, n\}$, the Y_is are non–i.i.d. but this is only because they do not have identical distributions. Since they are assumed to be continuous random variables, the probability integral transform can again be used to transform them towards "i.i.d.–ness." Hence, as in Sect. 4.2, our proposed transformation amounts to defining

$$u_i = \bar{D}_{x_i}(Y_i) \quad \text{for} \quad i = 1, \ldots, n. \tag{4.25}$$

Equation (4.24) then implies that $u_1, \ldots, u_n$ should be approximately i.i.d. Uniform (0,1) provided n is large. We can now invoke the Model-Free Prediction Principle in order to construct optimal predictors of $g(Y_f)$ where Y_f is the out-of-sample response associated with regressor value x_f, and $g(\cdot)$ is a real-valued function; for simplicity, we focus on the case $g(x) = x$. As usual, the L_2–optimal predictor of Y_f is the expected value of Y_f given x_f that is estimated in the Model-Free paradigm by

$$\Pi_{x_f} = n^{-1} \sum_{i=1}^{n} \bar{D}_{x_f}^{-1}(u_i). \tag{4.26}$$

Similarly, the Model-Free (MF) L_1–optimal predictor of $g(Y_f)$ is the median of the set $\{\bar{D}_{x_f}^{-1}(u_i),\ i=1,\ldots,n\}$. Under the Limit Model-Free (LMF) paradigm, the L_2– and L_1–optimal predictors are given by $\int_0^1 \hat{D}_{x_f}^{-1}(u)du$ and $\hat{D}_{x_f}^{-1}(1/2)$, respectively. Of course, one can also construct traditional estimators of the L_2– and L_1–optimal predictors of Y_f; these are respectively given by

$$m_{x_f} = \sum_{i=1}^{n} Y_i \tilde{K}\left(h^{-1}d(x_f, x_i)\right) \quad \text{and} \quad \bar{D}_{x_f}^{-1}(1/2).$$

Equation (4.24) shows that $\bar{D}_{x_f}^{-1}(1/2)$ is a consistent estimator of the theoretical L_1–optimal predictor $D_{x_f}^{-1}(1/2)$. Under some additional regularity conditions, Ferraty and Vieu (2006) also showed that the Nadaraya-Watson smoother m_{x_f} is consistent for $E(Y_f|X_f = x_f)$ under model (4.2). As in Sect. 4.3.2, here as well it is true that the MF, LMF, and traditional predictors are asymptotically equivalent. To elaborate,

$$m_{x_f} = \int y\, \hat{D}_{x_f}(dy) = \int_0^1 \hat{D}_{x_f}^{-1}(u)du \simeq \int_0^1 \bar{D}_{x_f}^{-1}(u)du \simeq n^{-1}\sum_{i=1}^{n} \bar{D}_{x_f}^{-1}(u_i) = \Pi_{x_f},$$

and $\text{median}\{\bar{D}_{x_f}^{-1}(u_i)\} = \bar{D}_{x_f}^{-1}(\text{median}\{u_i\}) \simeq \bar{D}_{x_f}^{-1}(1/2) \simeq \hat{D}_{x_f}^{-1}(1/2)$ since the u_is are approximately Uniform (0,1), and $\bar{D}_{x_f}^{-1}(\cdot)$ is strictly increasing.

Remark 4.7.1 All the aforementioned predictors are based on either the estimator $\bar{D}_{x_f}(\cdot)$ or $\hat{D}_{x_f}(\cdot)$ whose finite-sample accuracy crucially depends on the number of data pairs (Y_j, X_j) with regressor value that lies in the neighborhood of the point of interest x_f. If few (or none) of the regressors are found close to x_f, then nonparametric prediction will be highly inaccurate (or just plain impossible); this is where the curse of dimensionality may manifest in practice.

As already mentioned, the main advantage of the Model-Free, transformation-based approach to regression is that it allows us to go *beyond* point prediction and obtain valid predictive distributions and intervals for Y_f. To do this, however, some kind of resampling procedure is necessary in order to also capture the variance due to estimation error. For example, consider the prediction interval

$$[\hat{D}_{x_f}^{-1}(\alpha/2),\ \hat{D}_{x_f}^{-1}(1-\alpha/2)] \tag{4.27}$$

given in Ferraty and Vieu (2006, Eq. (5.10)); this interval is indeed asymptotically valid as it will contain Y_f with probability tending to the nominal $(1-\alpha)100\%$. However, interval (4.27) will be characterized by *under-coverage* in finite samples since the nontrivial variability in the estimated quantiles $\hat{D}_{x_f}^{-1}(\alpha/2)$ and $\hat{D}_{x_f}^{-1}(1-\alpha/2)$ is ignored. Having mapped the responses $Y_1,\ldots,Y_n$ onto the approximately i.i.d. variables $u_1,\ldots,u_n$, the premises of the Model-Free Prediction Principle are seen to be satisfied. Hence, the Model-Free bootstrap Algorithm 4.4.1 applies *verbatim* to the current setup of nonparametric regression with univariate response and functional regressors, and the same is true for the Limit Model-Free resampling Algorithm 4.4.2. Furthermore, the Predictive Model-Free resampling Algorithm 4.5.1 also applies *verbatim* to the current setup.

Chapter 5
Model-Free vs. Model-Based Confidence Intervals

5.1 Introduction

As in the previous two chapters, consider regression data of the type $\{(Y_t, x_t), t = 1, \ldots, n\}$. For simplicity of presentation, the regressor x_t is again assumed univariate and deterministic; the case of a multivariate regressor is handled in an identical way—see Sect. 4.7.2 for a discussion. As in the whole of Part II of the book, it will be assumed that $Y_1, \ldots, Y_n$ are independent but not identically distributed.

Let $D_x(y) = P\{Y_t \leq y | x_t = x\}$ denote the conditional distribution of the pairs (Y_t, x_t) that is assumed to be common for all t in accordance with the Model-free regression setup of Chap. 4. It may be of interest to estimate certain features, i.e., functionals, of $D_x(\cdot)$ such as the conditional mean, the conditional median, etc. For example, consider the first two conditional moments

$$\mu(x_t) = E(Y_t | x_t) \text{ and } \sigma^2(x_t) = \text{Var}(Y_t | x_t)$$

where the functions $\mu(\cdot)$ and $\sigma(\cdot)$ are considered unknown but assumed to possess some degree of smoothness (differentiability, etc.).

As discussed in Chap. 3, there are many approaches towards nonparametric estimation of the functions $\mu(\cdot)$ and $\sigma^2(\cdot)$, e.g., wavelets and orthogonal series, smoothing splines, local polynomials, and kernel smoothers. For concreteness, this chapter will also focus on one of the simplest methods, namely the Nadaraya-Watson kernel estimators (3.2) and (3.4).

Beyond point estimates of the functions $\mu(\cdot)$ and $\sigma(\cdot)$, it is important to be able to also provide interval estimates in order to have a measure of their statistical accuracy. Suppose, for example, that a practitioner is interested in the expected response to be observed at a future point x_f; a confidence interval for $\mu(x_f)$ is then desirable. Under regularity conditions, such a confidence interval can be given either via a large-sample normal approximation, or via a resampling approach; see, e.g., Freedman (1981), Härdle and Bowman (1988), Härdle and Marron (1991), Hall (1993), or Neumann and Polzehl (1998). Typical regularity conditions for the above bootstrap

D.N. Politis, *Model-Free Prediction and Regression*, Frontiers in Probability and the Statistical Sciences, DOI 10.1007/978-3-319-21347-7_5

approaches involve the assumption of an additive model with respect to independent and identically distributed (i.i.d.) errors, i.e., a model such as Eq. (3.1).

In Sect. 5.2, we revisit the usual model-based bootstrap for confidence intervals in regression adding the dimension of employing predictive as opposed to fitted residuals as developed in Chap. 3. More importantly, in Sect. 5.3 we address the problem of constructing a bootstrap confidence interval for $\mu(x_f)$—where x_f is a regressor value of interest—*without* assuming an underlying additive model. We focus attention on the parameter $\mu(x_f)$ just for simplicity and concreteness. The resampling algorithms developed in this chapter apply *verbatim* to any parameter associated with the conditional distribution of Y_f given a regressor value x_f. Other interesting parameters are the conditional variance $\sigma^2(x_f)$, the conditional median, and other quantiles of $D_{x_f}(\cdot)$.

As was the case for prediction intervals, here as well the Model-free approach is totally automatic, relieving the practitioner from the need to find an optimal transformation towards additivity and variance stabilization; this is a significant practical advantage because of the multitude of such proposed transformations, e.g. the Box/Cox power family, ACE, AVAS, etc.—see Linton et al. (1997) and the references therein. The finite-sample simulations provided in Sect. 5.4 confirm the viability and good performance of the model-free confidence intervals.

5.2 Model-Based Confidence Intervals in Regression

The usual additive model for nonparametric regression is given by Eq. (3.1) from Chap. 3 that is repeated below for convenience:

$$Y_t = \mu(x_t) + \sigma(x_t)\,\varepsilon_t \text{ for } t = 1,\ldots,n, \tag{5.1}$$

with $\varepsilon_t \sim$ i.i.d. (0,1) from an (unknown) distribution F. As before, the Nadaraya-Watson estimator of $\mu(x)$ is defined as

$$m_x = \sum_{i=1}^{n} Y_i \tilde{K}\left(\frac{x-x_i}{h}\right) \text{ with } \tilde{K}\left(\frac{x-x_i}{h}\right) = \frac{K\left(\frac{x-x_i}{h}\right)}{\sum_{k=1}^{n} K\left(\frac{x-x_k}{h}\right)}$$

where h is the bandwidth, and $K(x)$ is a symmetric kernel function satisfying $\int K(x)dx = 1$. Similarly, the Nadaraya-Watson estimator of $\sigma(x)$ is given by $s_x^2 = M_x - m_x^2$ where $M_x = \sum_{i=1}^{n} Y_i^2 \tilde{K}\left(\frac{x-x_i}{h}\right)$. As in Chap. 3, let $e_t = (Y_t - m_{x_t})/s_{x_t}$ denote the *fitted* residuals, and $\tilde{e}_t = (Y_t - m_{x_t}^{(t)})/s_{x_t}^{(t)}$ the *predictive* residuals where $m_x^{(t)}$ and $M_x^{(t)}$ denote the estimators m_x and M_x, respectively, computed from the *delete-Y_t* dataset: $\{(Y_i, x_i), i = 1,\ldots,t-1 \text{ and } i = t+1,\ldots,n\}$; also let $s_{x_t}^{(t)} = \sqrt{M_{x_t}^{(t)} - (m_{x_t}^{(t)})^2}$.

Consider the problem of constructing a confidence interval for the regression function $\mu(x_f)$ at a point of interest x_f. A normal approximation to the distribution of the estimator m_{x_f} implies an approximate $(1-\alpha)100\%$ equal-tailed, confidence interval for $\mu(x_f)$ given by:

$$[m_{x_f} + v_{x_f} \cdot z(\alpha/2),\ m_{x_f} + v_{x_f} \cdot z(1-\alpha/2)] \tag{5.2}$$

where $v_{x_f}^2 = s_{x_f}^2 \sum_{i=1}^n \tilde{K}^2(\frac{x_f - x_i}{h})$, and $z(\alpha)$ being the α–quantile of the standard normal. If the "density" (e.g., large-sample histogram) of the design points $x_1, \ldots, x_n$ can be thought to approximate a given functional shape (say, $f(\cdot)$) for large n, then Eq. (3.18) gives the useful approximation

$$\sum_{i=1}^{n} \tilde{K}^2(\frac{x_f - x_i}{h}) \sim \frac{\int K^2(x)dx}{nh\, f(x_f)}$$

provided $K(\cdot)$ is a probability density, i.e., $\int K(x)dx = 1$.

Interval (5.2) may be problematic in two respects: (a) it ignores the bias of m_x, so it must be either explicitly bias-corrected, or a suboptimal bandwidth must be used to ensure undersmoothing; and (b) it is based on a Central Limit Theorem which may not be a good finite-sample approximation if the errors are skewed and/or leptokurtic, or when the sample size is not large enough. For both above reasons, practitioners often prefer bootstrap methods over the normal approximation interval (5.2).

When using fitted residuals, the following algorithm is the well-known residual bootstrap pioneered by Freedman (1981) in a linear regression setting, and extended to nonparametric regression by Härdle and Bowman (1988), and other authors. As an alternative, we also propose the use of predictive residuals for resampling as developed in the MF/MB paradigm of Sect. 3.4; this may help alleviate the well-known phenomenon of under-coverage of bootstrap confidence intervals. Our goal is to approximate the distribution of the **confidence root**: $\mu(x_f) - m_{x_f}$ by that of its bootstrap counterpart.

The Model-based (MB) resampling algorithm goes as follows.

Algorithm 5.2.1 MB AND MF/MB RESAMPLING FOR CONFIDENCE INTERVALS FOR $\mu(x_f)$ BASED ON ROOTS

1. *Based on the* $\{(Y_t, x_t), t = 1, \ldots, n\}$ *data, construct the estimates* m_x *and* s_x *from which the fitted residuals* e_i, *and predictive residuals* $\tilde{e}_i$ *are computed for* $i = 1, \ldots, n$.
2. *For the traditional model-based bootstrap approach (MB), let* $r_i = e_i - n^{-1}\sum_j e_j$, *for* $i = 1, \ldots, n$. *For the predictive residual model-based approach (MF/MB) as in Sect. 3.4, let* $r_i = \tilde{e}_i - n^{-1}\sum_j \tilde{e}_j$, *for* $i = 1, \ldots, n$.
   a. *Sample randomly (with replacement) the residuals* $r_1, \ldots, r_n$ *to create the bootstrap pseudo-residuals* $r_1^*, \ldots, r_n^*$ *whose empirical distribution is denoted by* $\hat{F}_n^*$.
   b. *Create pseudo-data in the Y domain by letting* $Y_i^* = m_{x_i} + s_{x_i} r_i^*$ *for* $i = 1, \ldots, n$.

   c. *Based on the pseudo-data* $\{(Y_t^*, x_t), t = 1, \ldots, n\}$*, re-estimate the functions* $\mu(x)$ *and* $\sigma(x)$ *by the estimators* m_x^* *and* s_x^* *using the same methodology,*[1] *e.g. kernel smoothing with the same kernel, as the original estimators* m_x *and* s_x*.*
   d. *Calculate a replicate of the* **bootstrap confidence root***:* $m_{x_\mathrm{f}} - m_{x_\mathrm{f}}^*$*.*

3. *Steps (a)—(d) in the above are repeated B times, and the B bootstrap root replicates are collected in the form of an empirical distribution with* α*–quantile denoted by* $q(\alpha)$*.*
4. *Then, a* $(1-\alpha)100\%$ *equal-tailed confidence interval for* $\mu(x_\mathrm{f})$ *is given by:*

$$[m_{x_\mathrm{f}} + q(\alpha/2),\ m_{x_\mathrm{f}} + q(1-\alpha/2)]. \tag{5.3}$$

Note that by defining the root as $\mu(x_\mathrm{f}) - m_{x_\mathrm{f}}$ and not as the usual $m_{x_\mathrm{f}} - \mu(x_\mathrm{f})$ the cumbersome inversion of the quantiles in the bootstrap confidence interval is avoided. For example, our bootstrap confidence interval (5.3) has the $\alpha/2$ quantile in the left limit and the $1-\alpha/2$ quantile in the right limit which is more intuitive.

Interval (5.3) is of the "root" type; it is possible to construct bootstrap confidence intervals based on "studentized" roots instead. A studentized root is an asymptotically pivotal quantity whose distribution is typically well approximated by bootstrap methods. Hence, confidence intervals based on studentized roots are in general more accurate, i.e., have a coverage level closer to the nominal, as compared to unstudentized ones—see, e.g., Hall (1992).

Recall that our estimate of the variance of m_{x_f} is $v_{x_\mathrm{f}}^2 = s_{x_\mathrm{f}}^2 \sum_{i=1}^n \tilde{K}^2(\frac{x_\mathrm{f}-x_i}{h})$. Since the factor $\sum_{i=1}^n \tilde{K}^2(\frac{x_\mathrm{f}-x_i}{h})$ is the same in the real and bootstrap worlds, it is sufficient to studentize using s_{x_f} alone. The goal then would be to approximate the distribution of the **studentized root** $(\mu(x_\mathrm{f}) - m_{x_\mathrm{f}})/s_{x_\mathrm{f}}$ by that of its bootstrap counterpart.

Algorithm 5.2.2 MB AND MF/MB RESAMPLING FOR CONFIDENCE INTERVALS FOR $\mu(x_\mathrm{f})$ BASED ON STUDENTIZED ROOTS

The algorithm is the same as Algorithm 5.2.1 with one exception: in Step 1(d) the bootstrap root is defined in a studentized way, i.e., $(m_{x_\mathrm{f}} - m_{x_\mathrm{f}}^*)/s_{x_\mathrm{f}}^*$ *instead of* $m_{x_\mathrm{f}} - m_{x_\mathrm{f}}^*$*. With* $Q(\alpha)$ *denoting the* α*–quantile of the empirical distribution of the B studentized bootstrap root replicates, the resulting* $(1-\alpha)100\%$ *equal-tailed,* studentized *confidence interval for* $\mu(x_\mathrm{f})$ *is given by:*

$$[m_{x_\mathrm{f}} + Q(\alpha/2)s_{x_\mathrm{f}},\ m_{x_\mathrm{f}} + Q(1-\alpha/2)s_{x_\mathrm{f}}]. \tag{5.4}$$

As mentioned above, the studentized interval (5.4) should, in principle, be more accurate than the unstudentized interval (5.3). Nevertheless, the two intervals have almost identical performance in simulations. It turns out that the effect of studentization is mitigated by two factors: (i) the nonparametric nature of the problem in

[1] m_x^* and s_x^* can use the same bandwidth as the original estimators m_x and s_x provided these are slightly undersmoothed; otherwise, a two bandwidth trick should be used as discussed in Remark 3.5.2.

which the choice of bandwidth(s) is often the most crucial aspect, and (ii) the fact that we are already using a model that takes s_{x_f} into account by resampling *studentized* residuals; see Remark 3.6.3 for a related discussion.

Remark 5.2.1 An important feature of all bootstrap procedures is that they can handle *joint* confidence intervals, i.e., confidence *regions*, with the same ease as the univariate ones. This is especially true in regression where simultaneous confidence intervals are typicalled constructed in the form of confidence *bands*. The details are well known in the literature and are omitted due to lack of space; note that studentization is particularly helpful here as it ensures that the resulting joint confidence intervals are *balanced*, i.e., they all have (approximately) the same individual coverage level.

5.3 Model-Free Confidence Intervals Without an Additive Model

We now revisit the nonparametric regression setup but in a situation where an additive model such as Eq. (5.1) cannot be considered to hold true; we thus revert to the Model-free regression setup described in Sect. 4.2. As an example of model (5.1) not being valid, consider the setup where the skewness and/or kurtosis of Y_t depends on x_t, and thus centering and studentization will not result in "i.i.d.–ness." The dataset is still $\{(Y_t, x_t),\ t = 1, \ldots, n\}$ where the regressor x_t is univariate and deterministic, and the variables $Y_1, Y_2, \ldots$ are *independent* but not identically distributed. Recall also the definition of the conditional distribution $D_x(y) = P\{Y_f \le y | x_f = x\}$ where (Y_f, x_f) represents the random response Y_f associated with regressor x_f. Attention still focuses on constructing a Model-free interval estimate of $\mu(x_f) = E(Y_f|x_f) = \int y\, D_{x_f}(dy)$ to be compared with the model-based ones from the previous section.

As in Chap. 4, here too the default assumption is that the function $D_x(y)$ is continuous in both x and y. Consequently, we can estimate $D_x(y)$ by the local (weighted) empirical distribution $\hat{D}_x(y)$ defined in Eq. (4.4), i.e., $\hat{D}_x(y) = \sum_{i=1}^n \mathbf{1}\{Y_i \le y\}\tilde{K}\left(\frac{x-x_i}{h}\right)$. $\hat{D}_x(y)$ is a step function but we can construct the continuous in y (and differentiable) estimator $\bar{D}_x(y)$ as in Eq. (4.6), i.e.,

$$\bar{D}_x(y) = \sum_{i=1}^{n} \Lambda\left(\frac{y - Y_t}{h_0}\right) \tilde{K}\left(\frac{x - x_i}{h}\right).$$

where $\Lambda(y)$ is a (differentiable) distribution function that is strictly increasing over its support. Under regularity conditions, the two estimators $\hat{D}_x(y)$ and $\bar{D}_x(y)$ are consistent; see Eq. (4.5) and Claim 4.3.1.

As in Chap. 4, we define the variables $u_1, \ldots, u_n$ that are approximately i.i.d. and Uniform(0,1) via Eq. (4.12), i.e.,

$$u_i = \bar{D}_{x_i}(Y_i) \quad \text{for} \quad i = 1, \ldots, n.$$

Recall that the quantity

$$\hat{\Pi}_{x_f} = n^{-1} \sum_{i=1}^{n} \hat{D}_{x_f}^{-1}(u_i)$$

was proposed in Eq. (4.15) as an L_2–optimal predictor of Y_f, i.e., an approximation to the conditional expectation $\mu(x_f) = E(Y_f|x_f)$. Both $\hat{\Pi}_{x_f}$ and the closely related expression

$$\Pi_{x_f} = n^{-1} \sum_{i=1}^{n} \bar{D}_{x_f}^{-1}(u_i)$$

from Eq. (4.13) are defined as functions of the approximately i.i.d. variables $u_1, \ldots, u_n$ and hence they are amenable to the original i.i.d. bootstrap of Efron (1979). Recall Claim 4.3.1 where it was claimed that both $\hat{\Pi}_{x_f}$ and Π_{x_f} are asymptotically equivalent to the standard Nadaraya-Watson estimator m_{x_f}. Consequently, **a bootstrap procedure that works for $\hat{\Pi}_{x_f}$ and/or Π_{x_f} will by necessity also be valid for m_{x_f}**. Here it is best to think of $\hat{\Pi}_{x_f}$ and Π_{x_f} as estimators of the conditional expectation $\mu(x_f)$ instead of predictors.

Let $\hat{\mu}(x_f)$ denote our chosen estimator of $\mu(x_f) = E(Y_f|x_f)$, i.e., either m_{x_f}, $\hat{\Pi}_{x_f}$ or Π_{x_f}, or even one of the other asymptotically equivalent estimators discussed in Remark 4.3.2. Our goal is to approximate the distribution of the **confidence root**: $(\mu(x_f) - \hat{\mu}(x_f))/V_f$ by that of its bootstrap counterpart. As in the discussion leading to interval (5.4), all the aforementioned asymptotically equivalent estimators of $\mu(x_f)$ have (estimated) variance proportional to $s_{x_f}^2$; thus, V_f can be taken to either equal 1 or s_{x_f}, leading to unstudentized or studentized roots, respectively.

The Model-Free (MF) bootstrap algorithm goes as follows.

Algorithm 5.3.1 MF BOOTSTRAP FOR CONFIDENCE INTERVALS FOR $\mu(x_f)$

1. *Based on the $\{(Y_t, x_t), t = 1, \ldots, n\}$ data, construct the estimate $\bar{D}_x(\cdot)$, and use Eq. (4.12) to obtain the transformed data $u_1, \ldots, u_n$ that are approximately i.i.d. Uniform (0,1).*
   a. *Sample randomly (with replacement) the transformed data $u_1, \ldots, u_n$ to create bootstrap pseudo-data $u_1^*, \ldots, u_n^*$.*
   b. *Use the inverse transformation $\bar{D}_x^{-1}$ to create bootstrap pseudo-data in the Y domain, i.e., let $\underline{Y}_n^* = (Y_1^*, \ldots, Y_n^*)$ where $Y_t^* = \bar{D}_{x_t}^{-1}(u_t^*)$. Note that Y_t^* is paired with the original x_t design point; hence, the bootstrap dataset is $\{(Y_t^*, x_t), t = 1, \ldots, n\}$.*
   c. *Based on the pseudo-data $\{(Y_t^*, x_t), t = 1, \ldots, n\}$, re-estimate the conditional distribution $D_x(\cdot)$; denote the bootstrap estimates by $\hat{D}_x^*(\cdot)$ and $\bar{D}_x^*(\cdot)$.*
   d. *Calculate a replicate of the* **bootstrap confidence root**: $(\hat{\mu}(x_f) - \hat{\mu}^*(x_f))/V_f^*$ *where $\hat{\mu}^*(x_f)$ equals either $\int y \, \hat{D}_{x_f}^*(dy) = \int_0^1 \hat{D}_{x_f}^{*-1}(u)du$, $n^{-1}\sum_{i=1}^n \hat{D}_{x_f}^{*-1}(u_i^*)$, or $n^{-1}\sum_{i=1}^n \bar{D}_{x_f}^{*-1}(u_i^*)$ according to whether $\hat{\mu}(x_f)$ was chosen as m_{x_f}, $\hat{\Pi}_{x_f}$, or Π_{x_f}, respectively. Similarly, V_f^* is taken to either equal 1 or $s_{x_f}^*$ according to the corresponding choice for V_f; as usual, $s_{x_f}^*$ is the statistic s_{x_f} recomputed from the bootstrap dataset $\{(Y_t^*, x_t), t = 1, \ldots, n\}$.*

2. *Steps (a)—(d) in the above are repeated B times, and the B bootstrap root replicates are collected in the form of an empirical distribution with α–quantile denoted by $q(\alpha)$.*
3. *Then, the Model-Free $(1-\alpha)100\%$ equal-tailed confidence interval for $\mu(x_f)$ is*

$$[\hat{\mu}(x_f) + q(\alpha/2)V_f,\ \hat{\mu}(x_f) + q(1-\alpha/2)V_f]. \tag{5.5}$$

Remark 5.3.1 (On resampling pairs) As mentioned in Remark 4.1.1, an alternative resampling scheme in regression is bootstrapping pairs; this is typically justified under assumption (4.2), i.e., that $x_1,\ldots,x_n$ represent a realization of the random regressors $X_1,\ldots,X_n$, and that the pairs (Y_j,X_j) for $j=1,\ldots,n$ are i.i.d. The random vs. deterministic regressor dilemma is of little consequence in practice since in the former case inference is just conducted conditionally on the observed regressor values. Although bootstrapping pairs has good performance for confidence intervals and tests in semiparametric, e.g. linear, regression even in the presence of heteroscedastic errors, difficulties ensue when nonparametric regression is concerned. To see why, suppose that the goal is to set prediction intervals for $\mu(x_f)$; by resampling the i.i.d. pairs (Y_j,X_j) we run the risk of obtaining a bootstrap pseudo-sample $\{(Y_j^*,X_j^*)$ for $j=1,\ldots,n\}$ for which few of the X_j^* are found in the neighborhood of the point of interest x_f, thus making nonparametric estimation inaccurate (or just plain impossible) in the bootstrap world.

We can also define a Limit Model-Free (LMF) bootstrap algorithm. In the LMF case, there is no need to use Eq. (4.12) to create the transformed data $u_1,\ldots,u_n$; in this sense, the smooth estimator $\bar{D}_x(\cdot)$ is not needed, and the step function $\hat{D}_x(\cdot)$ suffices for the algorithm. The natural estimator of $\mu(x_f)$ associated with LMF is

$$\check{I\!I}_{x_f} = n^{-1}\sum_{i=1}^{n}\hat{D}_{x_f}^{-1}(U_i)$$

where the U_i are generated as i.i.d. Uniform (0,1). In other words, $\check{I\!I}_{x_f}$ is a Monte Carlo approximation to $\int_0^1 \hat{D}_{x_f}^{-1}(u)\,du$ which is nothing else than the Nadaraya-Watson estimator m_{x_f}. Nevertheless, the asymptotically equivalent estimators m_{x_f} or $\hat{I\!I}_{x_f}$ could also be used in the LMF procedure.

Algorithm 5.3.2 LMF BOOTSTRAP FOR CONFIDENCE INTERVALS FOR $\mu(x_f)$

1. *Based on the $\{(Y_t,x_t),t=1,\ldots,n\}$ data, construct the estimate $\hat{D}_x(\cdot)$.*
   a. *Generate bootstrap pseudo-data $u_1^*,\ldots,u_n^*$ i.i.d. from an exact Uniform $(0,1)$ distribution.*
   b. *Use the quantile inverse transformation $\hat{D}_x^{-1}$ to create bootstrap pseudo-data in the Y domain, i.e., let $\underline{Y}_n^* = (Y_1^*,\ldots,Y_n^*)$ where $Y_t^* = \hat{D}_{x_t}^{-1}(u_t^*)$.*
   c. *Based on the pseudo-data $\{(Y_t^*,x_t),t=1,\ldots,n\}$, re-estimate the conditional distribution $D_x(\cdot)$; denote the bootstrap estimate by $\hat{D}_x^*(\cdot)$.*

d. Calculate a replicate of the bootstrap confidence root: $(\hat{\mu}(x_f) - \hat{\mu}^*(x_f))/V_f^*$ *where* $\hat{\mu}^*(x_f)$ *equals either* $\int_0^1 \hat{D}_{x_f}^{*-1}(u)du$, $\int y\, \hat{D}_{x_f}^*(dy) = \int_0^1 \hat{D}_{x_f}^{*-1}(u)du$ *or* $n^{-1}\sum_{i=1}^n \hat{D}_{x_f}^{*-1}(u_i^*)$ *according to whether* $\hat{\mu}(x_f)$ *was chosen as* $\check{\Pi}_{x_f}$, m_{x_f} *or* $\hat{\Pi}_{x_f}$. *Similarly,* V_f^* *is taken to either equal 1 or* $s_{x_f}^*$ *according to the corresponding choice for* V_f.

2. *Steps (a)—(d) in the above are repeated B times, and the B bootstrap root replicates are collected in the form of an empirical distribution with* α*–quantile denoted by* $q(\alpha)$.
3. *The Limit Model-Free* $(1-\alpha)100\%$ *equal-tailed confidence interval for* $\mu(x_f)$ *is*

$$[\hat{\mu}(x_f) + q(\alpha/2)V_f,\ \hat{\mu}(x_f) + q(1-\alpha/2)V_f]. \tag{5.6}$$

Fact 5.3.1 *Both MF and LMF confidence intervals, i.e., intervals (5.5) and (5.6) respectively, were shown to be asymptotically valid under regularity conditions by Wang and Politis (2015) under the Model-free assumption (4.2). In addition, in the present setup of nonparametric regression, the LMF bootstrap can be shown to be equivalent to the local bootstrap of Shi (1991).*

As mentioned in Chap. 4, a major advantage of the LMF algorithm is that it remains valid even if $D_x(y)$ is not continuous in y, e.g. the case of discrete responses. However, LMF's downside is that the option to use "predictive" u–data is unavailable. To elaborate, recall that model-free "predictive" u–data were constructed in Sect. 4.5 as follows. Let $\bar{D}_{x_t}^{(t)}$ denote the estimator $\bar{D}_{x_t}$ as computed from the delete-Y_t dataset, i.e., $\{(Y_i, x_i),\ i = 1,\ldots,t-1 \text{ and } i = t+1,\ldots,n\}$. Now let

$$u_t^{(t)} = \bar{D}_{x_t}^{(t)}(Y_t) \quad \text{for } t = 1,\ldots,n. \tag{5.7}$$

Using the $u_t^{(t)}$ variables, we can now define *Predictive Model-Free* (PMF) confidence intervals for $\mu(x_f)$.

Algorithm 5.3.3 PMF BOOTSTRAP FOR CONFIDENCE INTERVALS FOR $\mu(x_f)$

- *The algorithm is identical to Algorithm 5.3.1 with one exception: replace the variables* $u_1,\ldots,u_n$ *with* $u_1^{(1)},\ldots,u_n^{(n)}$ *throughout the construction.*

Remark 5.3.2 Recall that the Model-free L_1–optimal predictor of Y_f is given by the median$\{\bar{D}_{x_f}^{-1}(u_i)\}$. Therefore, by analogy to Claim 4.3.1, we have:

$$\text{median}\{\bar{D}_{x_f}^{-1}(u_i)\} = \bar{D}_{x_f}^{-1}(\text{median}\{u_i\}) \simeq \bar{D}_{x_f}^{-1}(1/2)$$

since the u_is are approximately Uniform (0,1). Hence, if the practitioner wants to estimate the median (as opposed to the mean) of the conditional distribution of Y_f given x_f, then the local median $\bar{D}_{x_f}^{-1}(1/2)$ could be bootstrapped using i.i.d. resampling

in the same manner that median$\{\bar{D}_{x_f}^{-1}(u_i)\}$ can be bootstrapped. In fact, the above three Model-free resampling algorithms apply *verbatim* to *any* parameter associated with the conditional distribution of Y_f given a regressor value x_f. Examples include the aforementioned conditional median, the conditional variance $\sigma^2(x_f)$, and other quantiles of the conditional distribution of Y_f given x_f, etc.

5.4 Simulations

5.4.1 When a Nonparametric Regression Model Is True

As in Sect. 4.7.1, the building block for the simulations here as well is model (5.1) with $\mu(x) = \sin(x)$, $\sigma(x) = 1/2$, and errors ε_t i.i.d. N(0,1) or two-sided exponential (Laplace) rescaled to unit variance. Knowledge that the variance $\sigma(x)$ is constant was not used in the estimation, i.e., $\sigma(x)$ was estimated from the data. For each distribution, 500 datasets each of size $n = 100$ were created with the design points $x_1, \ldots, x_n$ being equi-spaced on $(0, 2\pi)$, and Nadaraya-Watson (N-W) estimates of $\mu(x) = E(Y|x)$ and $\sigma^2(x) = Var(Y|x)$ were computed using a normal kernel in R. On each dataset, several bootstrap methods were run; each bootstrap simulation was based on $B = 999$ replications.

Confidence intervals with nominal level 90 % were constructed using the two methods presented in Sect. 5.2: Traditional Model-Based (MB) and Predictive Residual Model-Based (MF/MB); the two methods presented in Sect. 5.3: Model-Free (MF) of Eq. (5.5), Limit Model-Free (MF) of Eq. (5.6), and Predictive Model-Free (PMF) from Algorithm 5.3.3; the NORMAL approximation interval (5.2), and the Local Bootstrap (LB) interval according to the method of Shi (1991).

In view of the discussion following Algorithm 5.2.2, all these intervals were based on unstudentized roots; intervals based on studentized roots have very similar finite-sample performance. The required bandwidths were computed by L_1 cross-validation. For each type of interval, the corresponding empirical coverage level (CVR) and average length (LEN) were recorded together with the (empirical) standard error associated with each average length.

Each method, NORMAL, MB, etc. is represented by three lines of entries in Tables 5.1 and 5.2. The first line of entries gives the empirical coverage levels (CVR) of prediction intervals calculated at several x_f points spanning the interval $(0, 2\pi)$; nominal coverage was **0.90**. The second line of entries gives the average length (LEN) of the corresponding interval; and the third line gives the standard error associated with interval length.

As already mentioned, several different estimators of $\mu(x_f)$ are asymptotically equivalent but may give some finite-sample differences. The default estimator of $\mu(x_f)$ for all methods is the Nadaraya-Watson (N-W), i.e., m_{x_f}. However, for LMF the natural estimator $\check{\Pi}_{x_f}$ was used, while for MF and PMF the estimator Π_{x_f} was

$x_f/\pi =$	0.15	0.3	0.5	0.75	1	1.25	1.5	1.7	1.85
	0.842	0.826	0.768	0.824	0.886	0.822	0.796	0.850	0.796
NORMAL	0.3852	0.3723	0.3711	0.3710	0.3710	0.3711	0.3711	0.3723	0.3852
	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002
	0.790	0.752	0.740	0.764	0.840	0.778	0.760	0.782	0.752
MB	0.3865	0.3523	0.3392	0.3598	0.3835	0.3621	0.3377	0.3498	0.3793
	0.004	0.003	0.004	0.003	0.003	0.003	0.004	0.003	0.004
	0.856	0.832	0.808	0.844	0.886	0.850	0.838	0.854	0.826
MF/MB	0.4657	0.4255	0.4134	0.4341	0.4621	0.4346	0.4104	0.4224	0.4580
	0.006	0.007	0.008	0.006	0.006	0.006	0.007	0.006	0.006
	0.846	0.816	0.800	0.830	0.886	0.840	0.832	0.824	0.812
LB	0.4403	0.4036	0.3853	0.4129	0.4405	0.4133	0.3866	0.4015	0.4336
	0.004	0.004	0.004	0.004	0.003	0.003	0.004	0.004	0.004
LMF	0.846	0.808	0.804	0.818	0.884	0.824	0.818	0.836	0.816
using	0.4398	0.4023	0.3871	0.4118	0.4382	0.4126	0.3851	0.3985	0.4330
N-W	0.004	0.004	0.004	0.004	0.003	0.003	0.004	0.004	0.004
	0.880	0.844	0.864	0.868	0.880	0.870	0.866	0.852	0.842
LMF	0.5244	0.4907	0.4706	0.5118	0.5498	0.5115	0.4679	0.4851	0.5146
	0.004	0.004	0.004	0.004	0.003	0.003	0.004	0.004	0.004
	0.862	0.836	0.828	0.848	0.888	0.852	0.840	0.842	0.826
MF	0.4579	0.4287	0.4147	0.4408	0.4698	0.4420	0.4123	0.4248	0.4494
	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003
	0.928	0.898	0.904	0.926	0.936	0.936	0.916	0.910	0.894
PMF	0.5482	0.5152	0.5005	0.5310	0.5613	0.5319	0.5006	0.5122	0.5412
	0.005	0.005	0.005	0.005	0.005	0.004	0.005	0.005	0.005

Table 5.1 Simulation results for additive model with i.i.d. Normal errors

used. In order to show the equivalence of LMF bootstrap to the Local Bootstrap (LB) we also ran the LMF algorithm using m_{x_f} as estimator—as done in LB; this is indicated by the entry: LMF using N-W.

Tables 5.1 and 5.2 summarize our findings, and contain a number of important features.

- The standard error of the reported coverage levels over the 500 replications is 0.013. Also note that—by construction—this simulation problem has some symmetry that helps us further appreciate the variability of the CVRs. For example, the expected CVRs should be the same for $x_f = 0.15\pi$ and 1.85π in all methods. So for the NORMAL case of Table 5.1, the CVR would be better estimated by

$x_f/\pi =$	0.15	0.3	0.5	0.75	1	1.25	1.5	1.7	1.85
	0.846	0.792	0.816	0.858	0.884	0.856	0.814	0.846	0.854
NORMAL	0.3820	0.3701	0.3691	0.3690	0.3691	0.3691	0.3690	0.3700	0.3820
	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002
	0.788	0.740	0.756	0.808	0.826	0.824	0.760	0.750	0.818
MB	0.3648	0.3454	0.3307	0.3545	0.3761	0.3535	0.3210	0.3355	0.3627
	0.004	0.004	0.004	0.003	0.003	0.003	0.003	0.004	0.004
	0.870	0.834	0.864	0.894	0.924	0.894	0.846	0.850	0.904
MF/MB	0.4555	0.4322	0.4119	0.4414	0.4679	0.4394	0.4006	0.4185	0.4513
	0.006	0.006	0.005	0.005	0.005	0.006	0.006	0.006	0.006
	0.866	0.790	0.828	0.850	0.908	0.870	0.832	0.818	0.880
LB	0.4242	0.4008	0.3830	0.4108	0.4373	0.4066	0.3733	0.3904	0.4188
	0.005	0.004	0.004	0.003	0.004	0.004	0.004	0.004	0.004
LMF	0.866	0.798	0.838	0.868	0.914	0.878	0.828	0.826	0.884
using	0.4210	0.3975	0.3819	0.4079	0.4333	0.4072	0.3691	0.3865	0.4180
N-W	0.004	0.004	0.004	0.003	0.004	0.004	0.004	0.004	0.004
	0.900	0.890	0.924	0.932	0.934	0.922	0.908	0.898	0.896
LMF	0.5053	0.4879	0.4648	0.5100	0.5476	0.5072	0.4546	0.4746	0.5022
	0.005	0.005	0.005	0.004	0.004	0.004	0.004	0.004	0.005
	0.868	0.806	0.848	0.890	0.910	0.876	0.836	0.862	0.872
MF	0.4119	0.3932	0.3765	0.4078	0.4404	0.4064	0.3703	0.3869	0.4124
	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003
	0.946	0.914	0.914	0.940	0.964	0.952	0.922	0.928	0.946
PMF	0.5172	0.4962	0.4808	0.5161	0.5499	0.5147	0.4737	0.4909	0.5157
	0.004	0.005	0.004	0.004	0.004	0.004	0.004	0.005	0.004

Table 5.2 Simulation results for additive model with i.i.d. Laplace errors

the average of 0.842 and 0.796, i.e., closer to 0.819; similarly, the PMF CVR for the same points could be better estimated by the average of 0.928 and 0.894, i.e., 0.911.

- The NORMAL intervals are characterized by under-coverage even when the true distribution is Normal. This under-coverage is more pronounced when $x_f = \pi/2$ or $3\pi/2$ due to the high bias of the kernel estimator at the points of a "peak" or "valley" that the normal interval (5.2) "sweeps under the carpet."
- The length of the NORMAL intervals is quite less variable than those based on bootstrap; this is not surprising since the extra randomization from the bootstrap is expected to inflate the overall variances.

- Although regression model (5.1) holds true here, the MB intervals show pronounced under-coverage; this is a phenomenon well-known in the bootstrap literature that could be alleviated by a two bandwidth trick as discussed in Remark 3.5.2.
- The MF/MB intervals are wider, and manage to partially correct the under-coverage of the MB intervals just using a single bandwidth based on cross-validation; this was to be expected since, as discussed in Chap. 3, the predictive residuals have larger scale than the fitted ones.
- The performance of MF intervals is better than that of MB intervals despite the fact that the former are constructed without making use of Eq. (5.1); hence, there is little to lose by conducting MF inference even when a model is true. However, as with the MB intervals, the MF intervals also show a tendency towards under-coverage.
- The equivalence of LB to LMF using N-W is clear in the simulations. However, the performance of either seems to be the worst among all model-free methods. By contrast, using LMF with its natural estimator leads to quite improved performance.
- The following pattern emerges, namely CVR(MF)<CVR(LMF)<CVR(PMF). The pattern CVR(MF)<CVR(PMF) is analogous to the aforementioned pattern CVR(MB)<CVR(MF/MB); it is unclear why LMF fits right in the middle.
- The PMF intervals appear to over-correct the MF under-coverage; this is especially prominent in the Laplace error case. The PMF intervals are therefore the only bootstrap intervals that are *conservative*, guaranteeing coverage of at least 90 %.

5.4.2 When a Nonparametric Regression Model Is Not True

Here, we investigate the performance of different confidence intervals in the absence of model (5.1). For easy comparison with Sect. 5.4.1, we will keep the same (conditional) mean and variance, i.e., we will generate independent Y data such that $E(Y|x) = \sin(x)$, $Var(Y|x) = 1/4$, and design points $x_1, \ldots, x_{100}$ equi-spaced on $(0, 2\pi)$. However, the error structure $\varepsilon_x = (Y - E(Y|x))/\sqrt{Var(Y|x)}$ has skewness that depends on x, thereby violating the i.i.d. assumption. For our simulation, we considered the same construction as in Sect. 4.7.2, i.e.,

$$\varepsilon_x = \frac{c_x Z + (1 - c_x)W}{\sqrt{c_x^2 + (1 - c_x)^2}} \tag{5.8}$$

where $c_x = x/(2\pi)$ for $x \in [0, 2\pi]$, and $Z \sim N(0,1)$ independent of W that is distributed as $\frac{1}{2}\chi_2^2 - 1$ to capture a changing skewness; note that $EW = 0$ and $EW^2 = 1$.

Our results are summarized in Table 5.3. The findings are qualitatively similar to those in Sect. 5.4.1. The MF/MB intervals are the undisputed winners here in terms of coverage accuracy; the overcoverage for $x_f \approx \pi$ could be attributed to the

$x_f/\pi =$	0.15	0.3	0.5	0.75	1	1.25	1.5	1.7	1.85
	0.836	0.834	0.780	0.848	0.876	0.804	0.762	0.838	0.814
NORMAL	0.3824	0.3700	0.3689	0.3688	0.3689	0.3688	0.3688	0.3700	0.3824
	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002
	0.746	0.720	0.736	0.754	0.796	0.772	0.718	0.756	0.768
MB	0.3593	0.3342	0.3146	0.3530	0.3810	0.3511	0.3287	0.3418	0.3697
	0.005	0.004	0.004	0.004	0.004	0.004	0.004	0.003	0.003
	0.808	0.798	0.806	0.850	0.884	0.872	0.822	0.846	0.850
MF/MB	0.4822	0.4463	0.4205	0.4722	0.5061	0.4393	0.4393	0.4567	0.4908
	0.009	0.009	0.009	0.009	0.010	0.008	0.009	0.009	0.009
	0.800	0.782	0.806	0.836	0.872	0.834	0.778	0.810	0.804
LB	0.4175	0.3885	0.3697	0.4098	0.4422	0.4102	0.3814	0.4006	0.4332
	0.006	0.005	0.005	0.005	0.005	0.004	0.004	0.004	0.004
LMF	0.798	0.780	0.796	0.824	0.870	0.844	0.760	0.810	0.820
using	0.4185	0.3891	0.3671	0.4114	0.4425	0.4074	0.3832	0.3997	0.4335
N-W	0.006	0.005	0.005	0.005	0.005	0.003	0.004	0.004	0.004
	0.822	0.790	0.782	0.858	0.890	0.876	0.832	0.848	0.828
LMF	0.4977	0.4713	0.4448	0.5090	0.5548	0.5056	0.4639	0.4854	0.5130
	0.007	0.005	0.005	0.006	0.005	0.004	0.005	0.004	0.004
	0.794	0.796	0.784	0.824	0.868	0.844	0.772	0.822	0.804
MF	0.4074	0.3851	0.3690	0.4028	0.4394	0.4108	0.3830	0.3983	0.4210
	0.004	0.003	0.003	0.003	0.003	0.003	0.003	0.003	0.003
	0.870	0.888	0.876	0.902	0.938	0.930	0.880	0.894	0.886
PMF	0.5167	0.4897	0.4688	0.5265	0.5509	0.5181	0.4821	0.4971	0.5214
	0.007	0.006	0.005	0.006	0.005	0.005	0.006	0.005	0.005

Table 5.3 Simulation results using regression model with non-identically distributed errors

aforementioned "bias leakage" discussed in Chap. 4. By contrast, the NORMAL and the MB bootstrap intervals show pronounced under-coverage; interestingly, these are the two methods that most practitioners use at the moment.

Acknowledgements

Many thanks are due to Liang Wang for checking/correcting the software, and re-running the simulations of Politis (2014) on the UCSD supercomputer cluster.

Part III
Dependent Data: Time Series

Chapter 6
Linear Time Series and Optimal Linear Prediction

6.1 Introduction

Consider data $Y_1, \ldots, Y_n$ arising as an observed stretch of a strictly stationary time series $\{Y_t, t \in \mathbf{Z}\}$. For simplicity, we will assume that $EY_t = 0$ which implies that the time series has been centered at expectation before our analysis. Assuming $EY_t^2 < \infty$, denote the lag-k autocovariance by $\gamma_k = Cov(Y_t, Y_{t+k})$, and the autocorrelation function (acf) at lag k by $\rho_k = \gamma_k/\gamma_0$. If $\rho_k = 0$ for all $k > 0$, then the series $\{Y_t\}$ is said to be a *white noise*, i.e., an uncorrelated sequence.

The time series $\{Y_t, t \in \mathbf{Z}\}$ will be assumed to be weakly dependent. To that effect it is common to assume that $\gamma_k \to 0$ as $k \to \infty$ fast enough so that $\sum_{k=-\infty}^{\infty} |\gamma_k| < \infty$; in this case, we can also define the *spectral density function*

$$f(w) = (2\pi)^{-1} \sum_{k=-\infty}^{\infty} \gamma_k e^{-iwk} \text{ for } w \in [-\pi, \pi],$$

i.e., the Fourier series associated with the autocovariance sequence γ_k.

A key question in time series analysis is optimal one-step-ahead prediction. Under the aforementioned finite variance assumption, it is immediate that the MSE–optimal predictor of Y_{n+1} given the data $Y_1, \ldots, Y_n$ is the conditional expectation $E(Y_{n+1}|Y_n, \ldots, Y_1)$. In the twentieth century, the simplifying assumption of Gaussianity was often made, i.e., that all finite-dimensional marginals of $\{Y_t, t \in \mathbf{Z}\}$ are (multivariate) Gaussian. Under such an assumption, the problem of optimal one-step-ahead prediction is greatly simplified since in this case $E(Y_{n+1}|Y_n, \ldots, Y_1)$ turns out to be a *linear* function of the given variables $Y_n, \ldots, Y_1$; hence, in the Gaussian case, optimal prediction is tantamount to optimal linear prediction.

D.N. Politis, *Model-Free Prediction and Regression*, Frontiers in Probability and the Statistical Sciences, DOI 10.1007/978-3-319-21347-7_6

The simplifying property of the optimal predictor being linear is shared by a class of time series that is larger than the Gaussian class. To discuss this further, recall that a time series $\{Y_t\}$ is called **linear** if it satisfies an equation of the type:

$$Y_t = \sum_{k=-\infty}^{\infty} \psi_k Z_{t-k} \tag{6.1}$$

where the coefficients ψ_k are (at least) square-summable, and the series $\{Z_t\}$ is i.i.d. with mean zero and variance $\sigma^2 > 0$. A linear time series $\{Y_t\}$ is called *causal* if $\psi_k = 0$ for $k < 0$, i.e., if

$$Y_t = \sum_{k=0}^{\infty} \psi_k Z_{t-k}. \tag{6.2}$$

Remark 6.1.1 Equation (6.2) should not be confused with the Wold decomposition that *all* purely nondeterministic, stationary time series possess—see, e.g., Brockwell and Davis (1991). In the Wold decomposition, the innovations $\{Z_t\}$ are only assumed to be a white noise and not i.i.d.; the i.i.d. assumption is of course much stronger.

A linear time series is called *invertible* if one can use Eq. (6.1) to solve for Z_t in terms of present and past Y_ts in which case we can write

$$Y_t = \sum_{k=1}^{\infty} \phi_k Y_{t-k} + Z_t; \tag{6.3}$$

a typical assumption here is that the sequence ϕ_k is absolutely summable. For causal time series, invertibility occurs when the power series $\psi(s) = \sum_{k=0}^{\infty} \psi_k s^k$ has no roots on the unit disc. Similarly, for a time series satisfying Eq. (6.3), causality occurs if the function $\phi(s) = \sum_{k=0}^{\infty} \phi_k s^k$ has no roots on the unit disc. Now it is not difficult to see that for a linear time series satisfying Eqs. (6.2) and (6.3) we have

$$E(Y_{n+1}|Y_n, Y_{n-1}, \ldots) = \sum_{k=1}^{\infty} \phi_k Y_{n+1-k}$$

where $E(Y_{n+1}|Y_n, Y_{n-1}, \ldots)$ denotes the conditional expectation given the infinite history. Hence, given the infinite past, the property of the optimal predictor being linear is shared by the class of linear time series that are *causal and invertible.*[1]

Under standard weak dependence conditions, it holds that

$$E(Y_0|Y_{-1}, Y_{-2}, \ldots, Y_{-m}) \to E(Y_0|Y_{-1}, Y_{-2}, \ldots) \text{ as } m \to \infty$$

[1] A slight generalization of this statement is possible, i.e., replacing the i.i.d. assumption for $\{Z_t\}$ with a martingale difference assumption; see, e.g., Kokoszka and Politis (2011).

for almost all sample paths of $\{Y_t,\ t<0\}$. Using the assumed stationarity of $\{Y_t\}$ we can then write

$$E(Y_{n+1}|Y_n,Y_{n-1},\ldots,Y_1) \simeq E(Y_{n+1}|Y_n,Y_{n-1},\ldots) \tag{6.4}$$

for large n, i.e.,

$$E(Y_{n+1}|Y_n,Y_{n-1},\ldots,Y_1) \simeq \sum_{k=1}^{n} \phi_k Y_{n+1-k}. \tag{6.5}$$

Note that Eq. (6.3) represents an Auto-Regressive (AR) recursion of infinite order that generalizes the well-known AR(p) recursion defined by

$$Y_t = \sum_{k=1}^{p} \phi_k Y_{t-k} + Z_t \tag{6.6}$$

for some natural number p. Under causality of the above AR(p) model, approximations (6.4) and (6.5) become equalities provided, of course that, $n \geq p$.

6.2 Optimal Linear Prediction

In general, the MSE–optimal predictor $E(Y_{n+1}|Y_n,\ldots,Y_1)$ will not necessarily be a linear function of the observed variables $Y_n,\ldots,Y_1$. For example, there is no guarantee that the time series at hand is linear, let alone causal and invertible. However, in order to compute $E(Y_{n+1}|Y_n,\ldots,Y_1)$ one needs to know (or estimate) the joint distribution of $Y_{n+1},Y_n,\ldots,Y_1$ which is unrealistic based on a sample of size n without additional simplifying assumptions.[2] Hence, more often than not, practitioners have to contend themselves with a predictor that is only optimal among all linear functions of the data $Y_n,\ldots,Y_1$.

The MSE–optimal linear predictor is given by

$$\tilde{Y}_{n+1} = \phi_1(n)Y_n + \phi_2(n)Y_{n-1} + \ldots + \phi_n(n)Y_1, \tag{6.7}$$

where the optimal coefficients $\phi_i(n)$ are computed from the normal equations

$$\phi(n) \equiv \begin{bmatrix} \phi_1(n) \\ \vdots \\ \phi_n(n) \end{bmatrix} = \Gamma_n^{-1}\gamma(n); \tag{6.8}$$

[2] Chapter 8 studies the case where the time series $\{Y_t\}$ is Markov of order p; the Markov assumption indeed allows us to estimate the joint distribution of $Y_{n+1},Y_n,\ldots,Y_1$ based on data $Y_1,\ldots,Y_n$, and to then estimate $E(Y_{n+1}|Y_n,\ldots,Y_1)$.

see Brockwell and Davis (1991, p. 167). The notation $\phi_i(n)$ makes it clear that the optimal coefficients depend on n despite the fact that they stabilize asymptotically as Eq. (6.5) claims. In Eq. (6.8), $\Gamma_n = [\gamma_{|i-j|}]_{i,j=1}^n$ is the autocovariance matrix of the data vector $\underline{Y}_n = (Y_1, \ldots, Y_n)'$, and $\gamma(n) = (\gamma_1, \ldots, \gamma_n)'$ is the vector of covariances at lags $1, \ldots, n$. Note that Eq. (6.7) represents an *oracle* predictor because the coefficients $\phi_1(n), \ldots, \phi_n(n)$ are unknown.

In practice, the coefficient vector $\phi(n) \equiv (\phi_1(n), \phi_2(n), \ldots, \phi_n(n))'$ is often truncated to its first p components in order to be consistently estimated; this procedure is equivalent to fitting an AR(p) model to the data. The resulting predictor is

$$\hat{Y}_{n+1}^{AR} = \hat{\phi}_1 Y_n + \hat{\phi}_2 Y_{n-1} + \ldots + \hat{\phi}_p Y_{n-p+1}, \tag{6.9}$$

where the coefficient vector is typically estimated by the Yule-Walker equations

$$(\hat{\phi}_1, \ldots, \hat{\phi}_p)' = \breve{\Gamma}_p^{-1} \breve{\gamma}(p). \tag{6.10}$$

In Eq. (6.10), $\breve{\gamma}_k = n^{-1} \sum_{t=1}^{n-|k|} Y_t Y_{t+|k|}$ is the sample autocovariance at lag k; we also let $\breve{\gamma}(p) = (\breve{\gamma}_1, \ldots, \breve{\gamma}_p)'$, and $\breve{\Gamma}_p = [\breve{\gamma}_{|i-j|}]_{i,j=1}^p$.

Interestingly, $\breve{\Gamma}_p$ is positive definite for any $p \leq n$ as long as $\breve{\gamma}_0 > 0$, which is a *sine qua non*. In addition, for any fixed p, $\breve{\gamma}(p)$ and $\breve{\Gamma}_p$ are consistent for their respective targets $\gamma(p)$ and Γ_p; a similar statement can be made when p is allowed to increase with n but at a slower rate, i.e., the case $p = o(n)$. Unfortunately, when p is large, problems ensue. For example, when $p = n$, Wu and Pourahmadi (2009) showed that the sample autocovariance matrix $\breve{\Gamma}_n = [\breve{\gamma}_{|i-j|}]_{i,j=1}^n$ is not a consistent estimator of matrix Γ_n, i.e., the operator norm of the difference $\breve{\Gamma}_n - \Gamma_n$ does not converge to zero. Hence, Eq. (6.10) cannot be used with $p = n$ to give a consistent estimator of the full coefficient vector $\phi(n)$.

6.3 Linear Prediction Using the Complete Process History

Instead of truncating the oracle predictor (6.7) to its first p summands, McMurry and Politis (2015) recently proposed an alternative approach to estimating all n coefficients in Eq. (6.7) which then allows for the complete process history to be used in prediction. The estimated prediction coefficients $\hat{\phi}(n) = (\hat{\phi}_1(n), \ldots, \hat{\phi}_n(n))'$ are given by the n-dimensional Yule-Walker equations:

$$\hat{\phi}(n) = (\hat{\Gamma}_n^{\star})^{-1} \hat{\gamma}(n), \tag{6.11}$$

where $\hat{\Gamma}_n^{\star}$ is a positive definite version of the $n \times n$ banded and tapered estimate of the autocovariance matrix Γ_n introduced in McMurry and Politis (2010), and $\hat{\gamma}(n)$ is the corresponding estimate of the autocovariance vector; the quantities appearing in Eq. (6.11) will be defined and discussed in the following section.

Using the estimated coefficients $\hat{\phi}(n)$, McMurry and Politis (2015) introduced the so-called Full-Sample Optimal (FSO) predictor

$$\hat{Y}_{n+1} = \hat{\phi}_1(n)Y_n + \hat{\phi}_2(n)Y_{n-1} + \ldots + \hat{\phi}_n(n)Y_1, \tag{6.12}$$

and proved its convergence to the oracle optimal predictor (6.7) under standard moment and weak dependence conditions.

Remark 6.3.1 Bickel and Gel (2011) proposed a predictor for Y_{n+1} that uses the upper-left $p \times p$ submatrix of the sample autocovariance matrix $\check{\Gamma}_n$ with $p = o(n)$. Their estimator is designed for an "on-line" prediction problem that allows for the parameters to be updated after each new observation at relatively low computational cost, and the resulting prediction for Y_{n+1} is a linear combination of $Y_n, \ldots, Y_{n-p+1}$. This is still an AR-type predictor as in Eq. (6.9) but using a larger order p than the one employed when fitting an AR model by minimizing AIC or a related criterion; see Choi (1992) for details.

6.3.1 Autocovariance Matrix Estimation

The matrix estimator of McMurry and Politis (2010) is defined as

$$\hat{\Gamma}_n = \left[\hat{\gamma}_{|i-j|}\right]_{i,j=1}^{n} \tag{6.13}$$

with

$$\hat{\gamma}_s = \kappa(|s|/l)\check{\gamma}_s \;\text{ for }\; |s| \le n, \text{ and } \hat{\gamma}(n) = (\hat{\gamma}_1, \ldots, \hat{\gamma}_n)'. \tag{6.14}$$

In the above, $\kappa(\cdot)$ can be any member of the *flat-top* family of compactly supported functions defined in Politis (2001), i.e., $\kappa(\cdot)$ is given by

$$\kappa(x) = \begin{cases} 1 & \text{if } |x| \le 1 \\ g(|x|) & \text{if } 1 < |x| \le c_\kappa \\ 0 & \text{if } |x| > c_\kappa, \end{cases} \tag{6.15}$$

where the function $g(\cdot)$ satisfies $|g(x)| < 1$, and c_κ is a constant satisfying $c_\kappa \ge 1$.

Remark 6.3.2 Wu and Pourahmadi (2009) conducted an in-depth study of a matrix estimator of the type (6.13) that uses the indicator function $\kappa(x) = \mathbf{1}_{[-1,1]}(x)$ as taper, i.e., a purely banded matrix, following up on earlier work by Bickel and Levina (2008a,b). However, as discussed by Politis (2011), and McMurry and Politis (2010, 2015), it is advantageous if the flat-top taper $\kappa(x)$ is a continuous function. A simple example of a flat-top taper with good properties is the trapezoidal, i.e.,

$$\kappa(x) = \begin{cases} 1 & \text{if } |x| \leq 1 \\ 2 - |x| & \text{if } 1 < |x| \leq 2 \\ 0 & \text{if } |x| > 2 \end{cases} \tag{6.16}$$

which was originally put forth by Politis and Romano (1995) in the related context of spectral density estimation.

Remark 6.3.3 Recently, Cai et al. (2013) proved that the matrix estimator $\hat{\Gamma}_n$ that uses the trapezoidal taper (6.16) enjoys a (slightly) improved rate of convergence as compared to the matrix estimator using the rectangular $\kappa(x) = \mathbf{1}_{[-1,1]}(x)$; this gives some theoretical substance to the aforementioned suggestion of employing a flat-top taper $\kappa(x)$ that is continuous. Recent papers showing interesting asymptotic results on estimating Γ_n include Chen et al. (2013), Cheng et al. (2015), and Xiao and Wu (2012).

Note that $\hat{\Gamma}_n$ as defined by (6.13) is consistent under regularity assumptions, and therefore asymptotically positive definite; however, in finite samples it can exhibit negative eigenvalues. In order to use it in practice, it must be corrected to positive definiteness. McMurry and Politis (2015) gave four different ways of correcting $\hat{\Gamma}_n$ to make it positive definite; these are reviewed in Sect. 6.4. The matrix $\hat{\Gamma}_n^{\star}$ appearing in Eq. (6.11) is a positive definite version of $\hat{\Gamma}_n$ where any one of the correction methods has been used.

6.3.2 Data-Based Choice of the Banding Parameter l

The matrix estimator (6.13) depends on the nonnegative banding parameter l. The flat-top tapering leaves the $2l+1$ main diagonals of the sample autocovariance matrix intact, and gradually down-weights more distant diagonals. In order to cover the possibility of the data at hand being uncorrelated, it is useful to adopt the convention that when $l = 0$, the resulting $\hat{\Gamma}_n$ matrix is given by $\check{\gamma}_0 I_n$ where I_n is the $n \times n$ identity matrix; this is equivalent to adopting that $0/0 = 0$ in the context of Eq. (6.14).

As a result, the FSO predictor of Eq. (6.12) depends on the choice of the banding parameter l. One possible approach to choosing l in a data-dependent way is the following rule, which was introduced for density and spectral density estimation in Politis (2003b).

Empirical rule for picking l. *Let $\rho_k = \gamma_k/\gamma_0$ and $\check{\rho}_k = \check{\gamma}_k/\check{\gamma}_0$. Let $\hat{l}$ be the smallest positive integer such that $|\check{\rho}_{\hat{l}+k}| < c(\log n/n)^{1/2}$ for $k = 1, \ldots, K_n$ where $c > 0$ is a fixed constant, and K_n is a positive, nondecreasing sequence that satisfies $K_n = o(\log n)$.*

McMurry and Politis (2010) further showed that the above rule produces approximately correct rates for autocovariance matrix estimation, and has good finite sample performance.

Remark 6.3.4 The empirical rule for picking l remains valid for all $c > 0$ and $1 \leq K_n \leq n$, although different choices of c and K_n can lead to different finite sample performances. Nonetheless, there are some guidelines for practically useful choices. The factor $(\log n)^{1/2}$ varies slowly; for example, if log is taken to denote base 10 logarithm, then for sample sizes between 100 and 1000, as is quite typical, $(\log n)^{1/2}$ varies between 1.41 and 1.73. Thus, if c is chosen to be around 2 and K_n about 5, Bonferroni's inequality implies that the bound $\pm c(\log n/n)^{1/2}$ can be used as the critical value for an approximate 95 % test of the null hypothesis that $\rho(\hat{l}+1),\ldots,\rho(\hat{l}+K_n)$ are all simultaneously equal to zero; values in this range seem to work well in practice.

6.4 Correcting a Matrix Towards Positive Definiteness

Under standard moment and weak dependence assumptions, e.g. assuming that the spectral density $f(w)$ is continuous and strictly positive for all $w \in [-\pi,\pi]$, the matrix $\hat{\Gamma}_n$ of Eq. (6.13) will have eigenvalues bounded away from zero with probability tending to one as $n \to \infty$. However, for finite samples, $\hat{\Gamma}_n$ may occasionally have eigenvalues that are negative and/or positive but close to zero. Since the inverse of $\hat{\Gamma}_n$ is a key element in prediction, the matrix $\hat{\Gamma}_n$ must be corrected to achieve finite-sample positive definiteness and avoid ill-conditioning. Following McMurry and Politis (2015), we now present four ways to implement such a correction.

6.4.1 Eigenvalue Thresholding

In the context of the Linear Process Bootstrap, McMurry and Politis (2010) suggested correcting the eigenvalues obtained in the spectral decomposition

$$\hat{\Gamma}_n = T_n D T_n' \tag{6.17}$$

where T_n is an orthogonal matrix, and D is diagonal with i-th entry denoted d_i. Letting $D^{\varepsilon} = \mathrm{diag}(d_1^{\varepsilon},\ldots,d_n^{\varepsilon})$ with $d_i^{\varepsilon} = \max\{d_i, \varepsilon\hat{\gamma}_0/n^{\beta}\}$, the adjusted estimate

$$\hat{\Gamma}_n^{\varepsilon} = T_n D^{\varepsilon} T_n' \tag{6.18}$$

is positive definite but maintains the same asymptotic rate of convergence as $\hat{\Gamma}_n$; in the above, $\varepsilon > 0$ and $\beta > 1/2$ are some fixed numbers. For the purposes of Linear Process Bootstrap, it had been found that the simple choices $\varepsilon = 1$ and $\beta = 1$ worked well in practice. In the matrix estimation context, however, the choice $\varepsilon = 1$ sometimes produced unstable predictions; a much larger ε, of the order of 10 or 20 (together with $\beta = 1$) seems to solve the problem.

Note that the average eigenvalue of $\check{\Gamma}_n$ equals $\check{\gamma}_0$, which is our best estimator of $\mathrm{Var}\, Y_t$; similarly, the average eigenvalue of $\hat{\Gamma}_n$ equals $\hat{\gamma}_0 = \check{\gamma}_0$. However, the threshold

correction (6.18) increases the average eigenvalue of the estimated matrix which is associated with an increased/inflated estimate of $\mathrm{Var}\, Y_t$. To see why, recall that, up to a factor of 2π, the eigenvalues of Γ_n are asymptotically given by the values of the spectral density evaluated at the Fourier frequencies; see, e.g., Gray (2005). Since the integral of the spectral density equals $\mathrm{Var}\, Y_t$, it is apparent that increasing the integral of the estimated spectral density results in an (artificially) increased estimate of $\mathrm{Var}\, Y_t$. Consequently, it is intuitive to rescale the estimate $\hat{\Gamma}_n^{\varepsilon}$ in order to ensure that its average eigenvalue remains equal to $\hat{\gamma}_0 = \check{\gamma}_0$.

Another way to motivate rescaling the corrected matrix estimate is to note that the Yule-Walker equations (6.11) should be *scale invariant*, i.e., invariant upon changes of $\mathrm{Var}\, Y_t$. In fact, they are often defined via a correlation matrix and vector instead of a covariance matrix and vector. To turn $\hat{\gamma}(n)$ into a vector of correlations, we just divide it by $\hat{\gamma}_0$. Dividing $\hat{\Gamma}_n^{\star}$ by $\hat{\gamma}_0$ should then provide a correlation matrix—hence the need for rescaling.

The rescaled estimate is thus given by

$$\hat{\Gamma}_n^{\star} = c\hat{\Gamma}_n^{\varepsilon} \text{ where } c = \hat{\gamma}_0/\bar{d}^{\varepsilon} \tag{6.19}$$

and $\bar{d}^{\varepsilon} = n^{-1}\sum_{i=1}^{n} d_i^{\varepsilon}$ is the average eigenvalue of $\hat{\Gamma}_n^{\varepsilon}$.

6.4.2 Shrinkage of Problematic Eigenvalues

Section 6.4.1 described a hard-threshold adjustment to the eigenvalues of $\hat{\Gamma}_n$ in order to render it positive definite. An alternative approach is to make the adjustment based on a positive definite estimate of Γ_n.

If the flat top weight function (6.16) is replaced by a weight function with a positive Fourier transform such as Parzen's piecewise cubic lag window, the resulting estimator $\hat{\Gamma}_n^{pd}$ will be positive definite and consistent—albeit with a slower rate of convergence than $\hat{\Gamma}_n$. Since $\hat{\Gamma}_n^{pd}$ and $\hat{\Gamma}_n$ are both Toeplitz, they are asymptotically diagonalized by the same orthogonal matrix—see, e.g., Grenander and Szegö (1958). Therefore, letting T_n be the orthogonal matrix from Eq. (6.17), the matrix defined as

$$\tilde{D} = T_n' \hat{\Gamma}_n^{pd} T_n$$

will be close to diagonal, and its diagonal entries will approximate the eigenvalues of $\hat{\Gamma}^{pd}$. Let $\tilde{d}_1, \ldots, \tilde{d}_n$ be the diagonals of $\tilde{D}$. We then produce adjusted eigenvalues $d_i^{\star}$ of D [as in (6.17)] by the following shrinkage rule. Let $d_i^{+} = \max\{d_i, 0\}$. Then

$$d_i^{\diamond} = \begin{cases} d_i & \text{if } \; d_i \geq \tilde{d}_i \\ (1-\tau_n) d_i^{+} + \tau_n \tilde{d}_i & \text{if } \; d_i < \tilde{d}_i, \end{cases} \tag{6.20}$$

where $\tau_n = c/n^a$ for constants $c > 0$ and $a > 1/2$. Let $D^\diamond$ be a diagonal matrix with diagonal elements $d_1^\diamond, \ldots, d_n^\diamond$, and define the shrinkage estimator

$$\hat{\Gamma}_n^\diamond = T_n D^\diamond T_n'$$

that is positive definite, and maintains the same asymptotic properties as $\hat{\Gamma}_n$ as long as the constant a in (6.20) is greater than $1/2$. However, if a is chosen too large, the shrinkage correction will be ineffective for small samples. Finally, note that a rescaling step as given in Eq. (6.19) must be performed here as well; hence, our final estimator is given by

$$\hat{\Gamma}_n^\star = c\hat{\Gamma}_n^\diamond \text{ where } c = \hat{\gamma}_0/\bar{d}^\diamond \tag{6.21}$$

and $\bar{d}^\diamond = n^{-1}\sum_{i=1}^n d_i^\diamond$ is the average eigenvalue of $\hat{\Gamma}_n^\diamond$.

6.4.3 Shrinkage Towards White Noise

Section 6.4.2 proposed shrinking $\hat{\Gamma}_n$ towards the positive definite estimator $\hat{\Gamma}_n^{pd}$. The shrinking was selective: only problematic eigenvalues were corrected as in the threshold method of Sect. 6.4.1. We now describe a correction that is based on shrinking the corresponding spectral density estimate toward that of a white noise with the same variance—in effect adjusting all eigenvalues; this approach provides substantial computational benefits. Note that the notion of shrinking covariance matrices towards the identity has been previously employed by Ledoit and Wolf (2003, 2004) in a different context, namely as a tool to regularize the sample covariance matrix based on a sample consisting of multiple i.i.d. copies of a random vector.

The shrinkage corrected version of $\hat{\Gamma}_n$ is given by

$$\hat{\Gamma}_n^\star = s\hat{\Gamma}_n + (1-s)\hat{\gamma}_0 I_n, \tag{6.22}$$

where I_n is the identity matrix and $s \in (0,1]$. If all the eigenvalues d_i are greater or equal to $\varepsilon\hat{\gamma}_0/n^\beta$, then we let $s = 1$. Otherwise, we let s be the maximum value that ensures that the minimum eigenvalue of $\hat{\Gamma}_n^\star$ is exactly equal to $\varepsilon\hat{\gamma}_0/n^\beta$.

Estimator (6.22) has several appealing properties. First, it keeps the estimated variance of the process fixed to $\hat{\gamma}_0$, i.e., there is no need for rescaling. Second, the shrinkage estimator $\hat{\Gamma}_n^\star$ remains banded and Toeplitz, therefore fast and memory efficient Toeplitz equation solving algorithms can be used. Third, the estimate itself does not require numerical diagonalization of $\hat{\Gamma}_n$ since s can be estimated by evaluating the corresponding spectral density estimate.

6.4.4 Shrinkage Towards a Second Order Estimate

Section 6.4.2 suggested shrinking the smaller eigenvalues of $\hat{\Gamma}_n$ towards a second order target. Section 6.4.3 introduced the idea of shrinking all the eigenvalues of $\hat{\Gamma}_n$ towards those of a white noise process. An approach that combines the most appealing features of these two methods is to shrink the whole of $\hat{\Gamma}_n$ towards a positive definite, second order estimate of Γ_n.

Let $\hat{\Gamma}_n^{pd}$ be as defined in Sect. 6.4.2, and define the corrected estimator by

$$\hat{\Gamma}_n^{\star} = s\hat{\Gamma}_n + (1-s)\hat{\Gamma}_n^{pd}. \tag{6.23}$$

The shrinkage factor $s \in [0,1]$ is chosen to raise the minimum eigenvalue of $\hat{\Gamma}_n$ as close as possible to $\varepsilon\hat{\gamma}_0/n^{\beta}$ while keeping s in the desired range.

Estimator (6.23) also has the desirable property of yielding a $\hat{\Gamma}_n^{\star}$ that is banded and Toeplitz. In addition, $\hat{\Gamma}_n^{\star}$ has no need for rescaling as it has $\check{\gamma}_0$ on the main diagonal. Finally, using the second order estimator as the target feels less arbitrary than shrinking towards white noise. However, the reason that both shrinkage methods work well, asymptotically as well as in simulations, is that the correction is a small one, i.e., s tends to one in large samples. Thus, the target is not meant to be achieved but gives only a general direction for the correction—see McMurry and Politis (2015) for more discussion.

Remark 6.4.1 Among the four correction methods, the two global shrinkage estimators, namely estimators (6.22) and (6.23), may prove especially useful in the case of very large data sets. The reason is that they both result in a banded Toeplitz matrix that can be calculated easily, stored efficiently, and inverted via fast algorithms. Recall that the system $Tb = z$ with T being Toeplitz can be solved in $O(n\log^2 n)$ time using $O(n)$ memory; see, e.g., Brent et al. (1980).

6.5 Estimating the Length n Vector $\gamma(n)$

Implicit in the n-dimensional Yule-Walker equations (6.11) is the need for consistent estimation of the length n vector of auto-covariances $\gamma(n) = (\gamma_1, \ldots, \gamma_n)'$. The vector of sample auto-covariances $\check{\gamma}(n) = (\check{\gamma}_1, \ldots, \check{\gamma}_n)'$ is not a consistent estimator of $\gamma(n)$. In fact, $\check{\gamma}(n)$ misbehaves. To see why, recall that the periodogram of the centered data vanishes at frequency zero; this implies the identity $\sum_{i=1}^{n} \check{\gamma}_i = -\check{\gamma}_0/2$ which, of course, has no reason to hold for the true γ_i.

By contrast, the flat-top weighted estimator $\hat{\gamma}(n) = (\hat{\gamma}_1, \ldots, \hat{\gamma}_n)'$ defined in Eq. (6.14) is consistent for $\gamma(n)$ in (Euclidean) norm although not positive definite. Notice that $\hat{\gamma}(n)$ is closely related to the first row of $\hat{\Gamma}_n$ which is a consistent estimator of Γ_n; the only difference is that while $\hat{\gamma}(n) = (\hat{\gamma}_1, \ldots, \hat{\gamma}_n)'$, the first row of $\hat{\Gamma}_n$ is $(\hat{\gamma}_0, \hat{\gamma}_1, \ldots, \hat{\gamma}_{n-1})'$.

Let $\hat{\Gamma}_n^\star$ denote the matrix Γ_n corrected to positive definiteness by one of the four methods discussed in the previous section. By looking at the first row of the consistent and positive definite matrix $\hat{\Gamma}_n^\star$, McMurry and Politis (2015) proposed an alternative estimator of $\gamma(n)$. To elaborate, define

$$\hat{\gamma}^\star(n) = [(\hat{\Gamma}_n^\star)_{1,2}, \cdots, (\hat{\Gamma}_n^\star)_{1,n}, 0]' \tag{6.24}$$

where $(\hat{\Gamma}_n^\star)_{i,j}$ denotes the i,j-th element of matrix $\hat{\Gamma}_n^\star$. Estimator $\hat{\gamma}^\star(n)$ is also consistent for $\gamma(n)$ in (Euclidean) norm, and has the additional property of being a positive definite sequence (by construction).

As an application, the coefficients in the FSO predictor (6.12) can be estimated via Yule-Walker equations that use $\hat{\gamma}^\star(n)$ instead of $\hat{\gamma}(n)$, i.e., letting

$$\hat{\phi}(n) = (\hat{\Gamma}_n^\star)^{-1}\hat{\gamma}^\star(n). \tag{6.25}$$

6.6 Linear Prediction Based on the Model-Free Prediction Principle

6.6.1 A First Idea: The Discrete Fourier Transform

To apply the Model-Free Prediction Principle, one must find a way to transform the time series data vector $\underline{Y}_n = (Y_1, \ldots, Y_n)'$ into an i.i.d. data vector. Recall that the autocovariance matrix of $\underline{Y}_n$ is denoted $\Gamma_n = [\gamma_{|i-j|}]_{i,j=1}^n$ which is a symmetric Toeplitz matrix. As already mentioned, a fundamental tool for time series analysis is the fact that all symmetric $n \times n$ Toeplitz matrices are approximately, i.e., asymptotically as $n \to \infty$, diagonalized by the same orthogonal matrix that has eigenvectors obtained from sinusoids sampled on a grid. This orthogonal transformation is nothing other than the Discrete Fourier Transform (DFT).

To elaborate, the DFT maps the vector $\underline{Y}_n$ to the vector $\underline{U}_n = (U_1, \ldots, U_n)'$ with j-th coordinate given by

$$U_j = \frac{1}{\sqrt{n}} \sum_{t=1}^{n} Y_t \exp(-i\lambda_j t)$$

where, as usual, $\lambda_j = 2\pi j/n$ for $j = 1, \ldots, n$ are the Fourier frequencies. Because of the aforementioned (approximate) diagonalizing property of the DFT, it now follows that the variables $U_1, \ldots, U_n$ are (approximately) uncorrelated.

Now if the time series $\{Y_t\}$ is linear, i.e., it satisfies Eq. (6.1), then it is not hard to see that the variables $U_1, \ldots, U_n$ will also be (approximately) independent with an asymptotic Gaussian distribution. Lahiri (2003) gives necessary and sufficient conditions for the asymptotic independence and normality of the DFT coefficients. Since $EU_j = 0$, a simple re-scaling would then bring us to an approximate i.i.d. setting.

The required re-scaling involves the spectral density; as shown in Brockwell and Davis (1991, Chap. 10), letting $\varepsilon_j^{(n)} = U_j/\sqrt{f(\lambda_j)}$ renders the variables $\varepsilon_1^{(n)}, \ldots, \varepsilon_n^{(n)}$ (approximately) i.i.d. Furthermore, they contain all the information needed to recapture (by re-scaling and inverse DFT) the original data vector $\underline{Y}_n$. Hence, the map from $\underline{Y}_n$ to $\underline{\varepsilon}_n^{(n)} = (\varepsilon_1^{(n)}, \ldots, \varepsilon_n^{(n)})'$ is invertible as the first premise of the Model-Free Prediction Principle requires.

However, there seems to be a difficulty with premise (b) of the Model-Free Prediction Principle as it is not clear if/how one can solve for Y_{n+1} in terms of $\underline{Y}_n$ and $\varepsilon_{n+1}^{(n+1)}$ alone. Nevertheless, this de-correlation idea can be made fruitful using a different approach; Sect. 6.6.2 gives the details that are based on a *whitening* filter.

Remark 6.6.1 The DFT variables $U_1, \ldots, U_n$ are complex-valued; an equivalent real-valued transformation is obtained by looking at the real and imaginary parts separately, i.e., letting $U_j^R = Real(U_j)$ and $U_j^I = Imag(U_j)$. Due to the symmetries $U_{n-j}^R = U_j^R$ and $U_{n-j}^I = -U_j^I$, it is apparent that the coefficients U_j^R and U_j^I for $j = 1, \ldots, \lfloor (n-1) \rfloor / 2$ carry all the necessary information required in order to recapture (by inverse DFT) the original series; see Kirch and Politis (2011) for details.

Remark 6.6.2 The periodogram $I(\lambda_j)$ is defined via the modulus squared of the DFT, i.e., $I(\lambda_j) = (2\pi)^{-1}|U_j|^2$. Note that the asymptotic independence of DFT ordinates $U_1, \ldots, U_n$ implies the asymptotic independence of periodogram ordinates $I(\lambda_1), \ldots, I(\lambda_n)$ that has served as the basis of the well-known frequency-domain bootstrap for time series proposed by Franke and Härdle (1992); see also Kreiss and Paparoditis (2012) for some current developments. A frequency-domain bootstrap that resamples the DFT instead of the periodogram was recently put forth by Kirch and Politis (2011) reviving an early idea of Hurvich and Zeger (1987).

6.6.2 Whitening and the Model-Free Linear Predictor

To start with, let us assume the working hypothesis that $\{Y_t, t \in \mathbf{Z}\}$ is a linear time series that is causal and invertible, i.e., it satisfies Eqs. (6.2) and (6.3) with respect to the innovations $\{Z_t\}$ that are i.i.d. with mean zero and variance σ^2.

Recall that the autocovariance matrix of data vector $\underline{Y}_n$ is denoted Γ_n which is a positive definite Toeplitz matrix. Consider a square-root decomposition

$$\Gamma_n = C_n C_n' \tag{6.26}$$

where C_n is positive definite. Now define the new vector $\underline{Z}_n^{(n)} = (Z_1^{(n)}, \ldots, Z_n^{(n)})'$ by

$$\underline{Z}_n^{(n)} = C_n^{-1} \underline{Y}_n. \tag{6.27}$$

Equation (6.27) is a *whitening* filter since the variables $Z_1^{(n)}, \ldots, Z_n^{(n)}$ are mean-zero, variance one, and uncorrelated, i.e., they constitute a *white noise* sample path.

Nevertheless, a stronger result is true if we insist that Eq. (6.26) is the Cholesky decomposition of Γ_n, i.e., requires that the positive definite matrix C_n is (lower) triangular. In that case, it is not hard to see that the variables $Z_1^{(n)}, \ldots, Z_n^{(n)}$ are approximately i.i.d. as the filter (6.27) gives an approximation to the inversion (6.3). In fact, the whitening filter (6.27) that uses the Cholesky decomposition of Γ_n is equivalent to the well-known innovations algorithm of Brockwell and Davis (1988); see also Rissanen and Barbosa (1969), and Pourahmadi (1999, 2011) for further applications of the Cholesky decomposition of Γ_n.

To elaborate, letting C_n be the (lower) triangular Cholesky factor of Γ_n implies $Z_j^{(n)} \simeq Z_j/\sigma$ for all $j \geq$ some j_0; the reason we have approximation instead of equality is due to edge effects in initializing the filter. Furthermore, transformation (6.27) is invertible so if we define $H_n : \underline{Y}_n \mapsto \underline{Z}_n^{(n)}$, then transformation H_n satisfies the first premise of the Model-Free Prediction Principle.[3] It is easy to see that it also satisfies premise (b) of the Model-Free Prediction Principle since the $Z_j^{(n)}$ do not depend on n for $j \geq j_0$, i.e., they are a simple sequence (up to edge effects) and not a triangular array; see Sect. 2.3.4.

In order to put the Model-Free Prediction Principle to work, we need to estimate the transformation H_m both for $m = n$ and for $m = n+1$. Recall that Sect. 6.4 developed several estimators of Γ_n that are consistent and positive definite. Let $\hat{\Gamma}_n^\star$ denote one of the two global shrinkage estimators, i.e., either estimators (6.22) or (6.23). The reason we focus on the two global shrinkage estimators is that they yield a matrix $\hat{\Gamma}_n^\star$ that is banded and Toeplitz; see Remark 6.4.1. In addition to fast computation, the banded Toeplitz property gives us an immediate way of constructing $\hat{\Gamma}_{n+1}^\star$ that is needed for transformation H_{n+1} and its inverse.

Denote by $\hat{\gamma}_{|i-j|}^\star$ the i,j-th element of $\hat{\Gamma}_n^\star$ for $i,j = 1, \ldots, n$; by construction, the sequence $\hat{\gamma}_s^\star$ for $s = 0, 1, \ldots$ is positive definite, and consistent for the true $\hat{\gamma}_s$ for $s = 0, 1, \ldots$. Hence, we define $\hat{\Gamma}_{n+1}^\star$ to be the symmetric, banded Toeplitz matrix with i,j-th element given by $\hat{\gamma}_{|i-j|}^\star$ for $i,j = 1, \ldots, n+1$. Recall that $\hat{\Gamma}_n^\star$ is banded, so $\hat{\gamma}_{|i-j|}^\star = 0$ if $|i-j| > lc_\kappa$. Thus, the two entries of $\hat{\Gamma}_{n+1}^\star$ at the upper-right and lower-left, i.e., both i,j-th entries that satisfy $|i-j| = n$ are naturally estimated by zeros.

The practical application of the Model-Free Prediction Principle in order to obtain the L_2–optimal predictor of Y_{n+1} can be summarized in the following algorithm.

[3] The fact that the $Z_j^{(n)}$ are only approximately i.i.d. is of little consequence in practice since the matrix Γ_n is unknown, and has to be estimated; hence, the practically feasible version of the $Z_j^{(n)}$ can only be expected to approximately i.i.d. anyway.

Algorithm 6.6.1 L_2-OPTIMAL MODEL-FREE LINEAR PREDICTOR

1. *Let $\hat{C}_n$ be the (lower) triangular Cholesky factor of $\hat{\Gamma}_n^\star$, and define*

$$\underline{\hat{Z}}_n = \hat{C}_n^{-1}\underline{Y}_n \text{ and hence } \underline{Y}_n = \hat{C}_n\underline{\hat{Z}}_n. \tag{6.28}$$

   Ignoring the aforementioned edge effects, we have denoted $\underline{\hat{Z}}_n = (\hat{Z}_1,\ldots,\hat{Z}_n)'$ as a simple sequence as opposed to a triangular array.
2. *Let $\underline{Y}_{n+1} = (Y_1,\ldots,Y_n,Y_{n+1})'$ that includes the unobserved Y_{n+1}, and $\underline{\hat{Z}}_{n+1} = (\hat{Z}_1,\ldots,\hat{Z}_n,\hat{Z}_{n+1})'$. Use the inverse transformation to write*

$$\underline{Y}_{n+1} = \hat{C}_{n+1}\underline{\hat{Z}}_{n+1} \tag{6.29}$$

   where $\hat{C}_{n+1}$ is the (lower) triangular Cholesky factor of $\hat{\Gamma}_{n+1}^\star$.
3. *Note that Eq. (6.29) implies that*

$$Y_{n+1} = \underline{\hat{c}}_{n+1}\underline{\hat{Z}}_{n+1} \tag{6.30}$$

   where $\underline{\hat{c}}_{n+1} = (\hat{c}_1,\ldots,\hat{c}_n,\hat{c}_{n+1})$ is the last row of $\hat{C}_{n+1}$.
4. *Recall that the prediction is carried out conditionally on $\underline{Y}_n$. Due to Eq. (6.28), the first n elements of the vector $\underline{\hat{Z}}_{n+1}$ can be treated as fixed (and known) given $\underline{Y}_n$. Then, the Model-Free approximation to the L_2–optimal predictor $E(Y_{n+1}|Y_n,\ldots,Y_1)$ is given by*

$$\hat{Y}_{n+1} = \sum_{i=1}^{n}\hat{c}_i\hat{Z}_i + \hat{c}_{n+1}\overline{\hat{Z}} \tag{6.31}$$

   where $\overline{\hat{Z}}$ is an empirical approximation to the expected value of $\hat{Z}_{n+1}$. A natural choice is to let $\overline{\hat{Z}} = n^{-1}\sum_{i=1}^{n}\hat{Z}_i$; alternatively, we can simply estimate $\overline{\hat{Z}}$ by zero using the fact that $\hat{Z}_{n+1} \simeq Z_{n+1}/\sigma$, and $E(Z_{n+1}|Y_n,\ldots,Y_1) = E(Z_{n+1}) = 0$ by assumption.

Remark 6.6.3 Recall our working hypothesis that $\{Y_t, t \in \mathbf{Z}\}$ is a linear time series that is causal and invertible. Under this hypothesis, the conditional expectation $E(Y_{n+1}|Y_n,\ldots,Y_1)$ is linear in $\underline{Y}_n$, and the same is true for its Model-Free estimate (6.31). However, if the working hypothesis of linearity is *not* true, then predictor (6.31) gives a novel approximation to the best *linear* predictor of Y_{n+1} on the basis of $\underline{Y}_n$, i.e., the orthogonal projection of Y_{n+1} onto the linear span of $(Y_n,\ldots,Y_1)$.

The performance of the Model-free predictor was investigated in simulation using some simple AR and MA models based on innovations $\varepsilon_t \sim$ i.i.d. $N(0,1)$. The models were: MA(1): $Y_t = \varepsilon_t + \theta\varepsilon_{t-1}$, and AR(1): $Y_t = \phi Y_{t-1} + \varepsilon_t$ for different choices of θ and ϕ; each model was simulated 1000 times using a sample size $n = 200$. Additional models were considered in the Rejoinder of McMurry and Politis (2015).

The notation for Tables 6.1 and 6.2 is as follows: FSO denotes the predictor (6.12); WN vs. 2o denotes that $\hat{\Gamma}_n^\star$ was obtained via shrinkage to white noise from Eq. (6.22) vs. shrinkage towards a second order estimator from Eq. (6.23);

Raw vs. Shr indicates using $\hat{\gamma}(n)$ vs. $\hat{\gamma}^{\star}(n)$ in the Yule-Walker equations, i.e., using Eq. (6.11) vs. (6.25). For simplicity, we considered the Model-Free prediction approach with $\hat{\Gamma}_n$ corrected to positive definiteness using shrinkage to white noise, i.e., the matrix $\hat{\Gamma}_n^{\star}$ used corresponded to estimator (6.22); hence, the notation MF-WN for the Model-Free method. AR denotes the AR(p) predictor (6.9) with the order p chosen by AIC minimization.

Note that the AR predictor (6.9) has been the linear prediction method of choice for the last century or so; hence, it serves as the benchmark for comparison. Interestingly, all versions of the FSO predictor seem competitive to the AR predictor under an AR(1) model, and for the most part outperform it in the case of the MA(1) model. Also notable is that the MF-WN method generates predictions that are of very similar quality to the FSO-WN-Shr approach; see Tables 6.1 and 6.2. The Model-Free approach using matrix $\hat{\Gamma}_n^{\star}$ obtained from estimator (6.23), i.e., shrinkage towards a second order estimator, would generate predictions that are of similar quality to the FSO-2o-Shr approach. Thus, it looks like the Model-Free approach in essence gives a different way to compute the FSO predictor based on $\hat{\Gamma}_n^{\star}$ in connection with the shrunk autocovariance estimator $\hat{\gamma}^{\star}(n)$ whatever the choice of $\hat{\Gamma}_n^{\star}$ might be. This is corroborated by the fact that, as mentioned before, the construction of predictor (6.31) was motivated by the Model-Free Prediction Principle but it is similar in spirit to the innovations algorithm of Brockwell and Davis (1988). The latter, however, assumes knowledge of Γ_n; the crucial difference is that the Model-Free predictor uses the consistent, positive definite estimator $\hat{\Gamma}_n^{\star}$ in place of the unknown Γ_n.

	FSO-WN-Raw	FSO-WN-Shr	FSO-2o-Raw	FSO-2o-Shr	MF-WN	AR
$\theta = -0.9$	1.0626	1.0662	1.0635	1.0629	1.0647	1.0614
$\theta = -0.5$	0.9849	0.9839	0.9892	0.9885	0.9840	0.9886
$\theta = -0.1$	0.9869	0.9869	0.9869	0.9869	0.9869	0.9939
$\theta = 0.1$	1.0314	1.0314	1.0314	1.0314	1.0314	1.0348
$\theta = 0.5$	1.0087	1.0070	1.0112	1.0106	1.0070	1.0222
$\theta = 0.9$	1.0481	1.0507	1.0460	1.0484	1.0504	1.0374

Table 6.1 Root mean square prediction errors associated with different one-step-ahead predictors: MA(1) model simulation

6.6.3 From Point Predictors to Prediction Intervals

The Model-free point predictor (6.31) was based on the matrix estimator $\hat{\Gamma}_n^{\star}$. However, it is possible to use alternative matrix estimators in Algorithm 6.6.1 as long as they are consistent for Γ_n and positive definite. One such possibility is the AR-based estimation of the matrix Γ_n discussed in the Rejoinder of McMurry and Politis (2015). To elaborate, we may fit an AR(p) model to the data based on the sample autocovariances $\check{\gamma}_0, \ldots, \check{\gamma}_p$; fitting the AR($p$) model via the Yule-Walker equations is

	FSO-WN-Raw	FSO-WN-Shr	FSO-2o-Raw	FSO-2o-Shr	MF-WN	AR
$\phi = -0.9$	1.1481	1.0968	1.0948	1.0633	1.0952	1.0091
$\phi = -0.5$	1.0121	1.0100	1.0239	1.0204	1.0102	0.9978
$\phi = -0.1$	0.9874	0.9874	0.9874	0.9874	0.9875	0.9841
$\phi = 0.1$	0.9975	0.9975	0.9975	0.9975	0.9975	0.9983
$\phi = 0.5$	1.0322	1.0298	1.0489	1.0454	1.0300	1.0093
$\phi = 0.9$	1.0942	1.0866	1.0654	1.0496	1.0849	1.0087

Table 6.2 Root mean square prediction errors associated with different one-step-ahead predictors: AR(1) model simulation

convenient as it results in a causal (and therefore stationary) model. Then, we estimate the whole autocovariance sequence by the autocovariance implied by the fitted AR model, i.e., by solving the difference equation as outlined in Brockwell and Davis (1991, Sect. 3.3); R automates this process through the `ARMAacf()` function.

Denote by $\hat{\gamma}_k^{AR}$ the lag-k autocovariance of the fitted AR model. As the autocovariance sequence of a stationary time series, the sequence $\hat{\gamma}_k^{AR}$ for $k \in \mathbf{Z}$ is positive definite. Hence if we define $\hat{\Gamma}_n^{AR}$ as the $n \times n$ Toeplitz matrix with i, j-th entry given by $\hat{\gamma}_{|i-j|}^{AR}$, it then follows that $\hat{\Gamma}_n^{AR}$ will be a positive definite estimator of matrix Γ_n. Under regularity conditions, $\hat{\Gamma}_n^{AR}$ will also be consistent (in operator norm) for Γ_n provided $p \to \infty$ but $p = o(n)$. Choosing p by minimizing the popular AIC criterion typically satisfies the above consistency requirement.

Hence, we can use $\hat{\Gamma}_n^{AR}$ and $\hat{\Gamma}_{n+1}^{AR}$ instead of $\hat{\Gamma}_n^{\star}$ and $\hat{\Gamma}_{n+1}^{\star}$, respectively, in Algorithm 6.6.1 giving rise to a Model-free point predictor of Y_{n+1} that is of AR–type, i.e. it is a linear combination of just the last p values $Y_{n-p+1}, \ldots, Y_n$ despite the fact that the given dataset $\underline{Y}_n$ is of size n. Both Model-free predictors—the one based on $\hat{\Gamma}_n^{AR}$ and the original one based on $\hat{\Gamma}_n^{\star}$—are consistent for the theoretically optimal point predictor; in fact, the AR–type Model-free predictor should be practically indistinguishable from the AR(p) predictor (6.9), i.e., the AR entry in Tables 6.1 and 6.2. The AR–type Model-free predictor can be easily extended with the purpose of constructing prediction intervals for Y_{n+1}; this is the subject of Chap. 7. The banded estimator $\hat{\Gamma}_n^{\star}$ is revisited and found useful in Chap. 9.

Acknowledgements

Chapter 6 is based on the paper: T. McMurry and D.N. Politis (2015). 'High-dimensional autocovariance matrices and optimal linear prediction' (with Discussion), *Electronic Journal of Statistics*, vol. 9, pp. 753–822. Many thanks are due to the Editor, George Michailidis, for hosting this discussion paper, and to the discussants: Xiaohui Chen, Yulia Gel, Rob Hyndman, Wilfredo Palma, and Wei-Biao Wu for their insightful comments.

Chapter 7
Model-Based Prediction in Autoregression

7.1 Introduction

Chapter 3 described in detail the construction of prediction intervals in model-based regression. An autoregressive (AR) time series model, be it linear, nonlinear, or nonparametric, bears a formal resemblance to the analogous regression model. Indeed, AR models can typically be fitted by the same methods used to estimate a regression, e.g., ordinary Least Square (LS) regression methods for parametric models, and scatterplot smoothing for nonparametric ones. The practitioner has only to be careful regarding the standard errors of the regression estimates but model-based resampling should in principle be able to capture those.

Therefore, it is not surprising that model-based resampling for regression can be extended to model-based resampling for *auto*regression. Indeed, standard errors and confidence intervals based on resampling the residuals from a fitted AR model have been one of the first bootstrap approaches for time series; see, e.g., Freedman (1984), Efron and Tibshirani (1993), and Bose (1988).

However, the situation as regards prediction intervals is not as clear; for example, the conditional nature of the predictive inference in time series poses a difficulty. There are several papers on prediction intervals for linear AR models but the literature seems scattered and there are many open questions: (a) how to implement the model-based bootstrap for prediction, i.e., how to generate bootstrap series; (b) how to construct prediction intervals given the availability of many bootstrap series already generated; (c) how to evaluate asymptotic validity of a prediction interval; and lastly (d) how to construct prediction intervals for nonlinear and nonparametric autoregressions.

Following Pan and Politis (2015), we now attempt to give some answers to the above, and thus provide a comprehensive approach towards bootstrap prediction

D.N. Politis, *Model-Free Prediction and Regression*, Frontiers in Probability and the Statistical Sciences, DOI 10.1007/978-3-319-21347-7_7

intervals for linear, nonlinear, or nonparametric autoregressions. Equation (2.2) from Chap. 2 gave a general autoregression model; for simplicity, in what follows we will not consider the possibility of having the additional regressor $\underline{x}_t$.

Therefore, the time series we will consider in this chapter are the stationary solutions of one of the following two recursions for $t \in \mathbf{Z}$:

- **AR model with homoscedastic errors**

$$Y_t = \mu(Y_{t-1}, \ldots, Y_{t-p}) + \varepsilon_t \text{ with } \varepsilon_t \sim \text{ i.i.d. } (0, \sigma^2) \tag{7.1}$$

- **AR model with heteroscedastic errors**

$$Y_t = \mu(Y_{t-1}, \ldots, Y_{t-p}) + \sigma(Y_{t-1}, \ldots, Y_{t-p})\varepsilon_t \text{ with } \varepsilon_t \sim \text{ i.i.d. } (0, 1). \tag{7.2}$$

In the above, $\mu(\cdot)$ and $\sigma(\cdot)$ are unknown; if they can be are assumed to belong to a finite-dimensional, parametric family of functions, then the above describe a linear or nonlinear AR model. If $\mu(\cdot)$ and $\sigma(\cdot)$ are only assumed to belong to a smoothness class, then the above models describe a nonparametric autoregression.

In addition, the following **causality** condition is assumed:

$$\varepsilon_t \text{ is independent from } \{Y_s, s < t\} \text{ for all } t. \tag{7.3}$$

Under either model (7.1) or (7.2), the causality assumption (7.3) ensures that

$$E(Y_t | Y_s, s < t) = \mu(Y_{t-1}, \ldots, Y_{t-p}) \tag{7.4}$$

which then gives the predictor of Y_t given $\{Y_s, s < t\}$ that is optimal with respect to Mean Squared Error (MSE) of prediction; this is a manifestation of the Markovian property of causal autoregressive models such as (7.1) and/or (7.2).

7.2 Prediction Intervals in AR Models: Laying the Foundation

7.2.1 Forward and Backward Bootstrap for Prediction

As previously mentioned, an autoregression can be formally viewed as regression. However, in prediction with an AR(p) model, linear or nonlinear, an additional difficulty is that the one-step-ahead prediction is done conditionally on the last p observed values that are themselves random.

To fix ideas, suppose $Y_1 = y_1, \ldots, Y_n = y_n$ constitutes a realization from the linear AR(1) model: $Y_t = \phi_1 Y_{t-1} + \varepsilon_t$ where $|\phi_1| < 1$ and the ε_t are i.i.d. with mean zero. Given the data, the MSE–optimal predictor of Y_{n+1} is $\phi_1 y_n$ which is approximated in practice by plugging-in an estimator, say $\hat{\phi}_1$, for ϕ_1; hence, our practical approximation to the MSE–optimal predictor of Y_{n+1} is $\hat{Y}_{n+1} = \hat{\phi}_1 y_n$.

Generating bootstrap series $Y_1^*, Y_2^*, \cdots$ from the fitted AR model enables us to capture the variability of $\hat{\phi}_1$ when the latter is re-estimated from bootstrap datasets such as $Y_1^*, \cdots, Y_n^*$. For the application to prediction intervals, note that the bootstrap also allows us to generate Y_{n+1}^* so that the statistical accuracy of the predictor $\hat{\phi}_1 Y_n$ can be gauged. However, none of these bootstrap series will have their last value Y_n^* *exactly* equal to the original value y_n as needed for prediction purposes. Herein lies the problem, since the behavior of the predictor $\hat{\phi}_1 Y_n$ needs to be captured **conditionally** on the original value $Y_n = y_n$.

To avoid this difficulty, Thombs and Schucany (1990) proposed to generate the bootstrap data $Y_1^*, \cdots, Y_n^*$ going backwards from the last value that is fixed at $Y_n^* = y_n$; this is the **backward bootstrap** method that was revisited by Breidt et al. (1995) who gave the correct algorithm for constructing the backward errors. Note that the generation of Y_{n+1}^* is still done in a forward fashion using the fitted AR model conditionally on the value Y_n.

Nevertheless, the natural way autoregressions evolve is *forward* in time, i.e., given Y_{t-1}, the next observation is generated as $Y_t = \phi_1 Y_{t-1} + \varepsilon_t$, and so on. It is intuitive to construct bootstrap procedures that run forward in time, i.e., given Y_{t-1}^*, the next bootstrap observation is given by

$$Y_t^* = \hat{\phi}_1 Y_{t-1}^* + \varepsilon_t^*; \tag{7.5}$$

as usual, the bootstrap errors ε_t^* are generated in an i.i.d. fashion from some estimate of the error distribution. Indeed, most (if not all) of the literature on bootstrap confidence intervals for AR models uses the natural time order to generate bootstrap series. It would be nice to be able to build upon this large body of work in order to construct prediction intervals. However, recall that predictive inference is to be conducted conditionally on the last value $Y_n = y_n$ in order to be able to place prediction bounds around the point predictor $\hat{\phi}_1 Y_n$. So how can one ensure that $Y_n^* = y_n$ so that $Y_{n+1}^* = \hat{\phi}_1 y_n + \varepsilon_{n+1}^*$?

Aided by the additive structure of the AR model, it is possible to "have our cake and eat it too," i.e., generate bootstrap series forward in time but also ensure that Y_{n+1}^* is constructed correctly. This procedure was called the **forward bootstrap** method for prediction intervals by Pan and Politis (2015), and comprises of the following steps:

A. Choose a starting value Y_0^* appropriately, e.g., choose it at random from one of the original data $Y_1, \cdots, Y_n$. Then, use recursion (7.5) for $t = 1, 2, \ldots, n$ in order to generate bootstrap data $Y_1^*, \cdots, Y_n^*$. Re-compute the statistic of interest (in this case $\hat{\phi}_1$) from the bootstrap data $Y_1^*, \cdots, Y_n^*$ to obtain the bootstrap statistic $\hat{\phi}_1^*$.
B. Re-define the last value in the bootstrap world, i.e., let $Y_n^* = y_n$. Compute the one-step ahead bootstrap predictor $\hat{Y}_{n+1}^* = \hat{\phi}_1^* y_n$, and also generate the future bootstrap observation $Y_{n+1}^* = \hat{\phi}_1 y_n + \varepsilon_{n+1}^*$.
C. Use the simulated distribution of the bootstrap predictive root $Y_{n+1}^* - \hat{Y}_{n+1}^*$ to estimate the true distribution of the real-world predictive root $Y_{n+1} - \hat{Y}_{n+1}$; note that it is also possible to use studentized predictive roots.

The above algorithm works because the two constituents of the prediction error $Y_{n+1} - \hat{Y}_{n+1} = (\phi_1 Y_n - \hat{\phi}_1 Y_n) + \varepsilon_{n+1}$, i.e., estimation error $(\phi_1 Y_n - \hat{\phi}_1 Y_n)$ and innovation error ε_{n+1} are independent, and the same is true in the bootstrap world. As stated above, the forward bootstrap algorithm is specific to an AR(1) model but its extension to higher-order models is straightforward and will be given in the sequel. Indeed, the forward bootstrap is the method that can be immediately generalized to nonlinear and nonparametric autoregressions as well, thus forming a unifying principle for treating all AR models.

The forward bootstrap idea has been previously used for prediction intervals in linear AR models by Masarotto (1990) and Pascual et al. (2004) but with some important differences. For example, Masarotto (1990) omits the above step B, while Pascual et al. (2004) base their prediction intervals on an analog of the bootstrap percentile method without considering predictive roots.

Remark 7.2.1 Both aforementioned bootstrap ideas, backward and forward, hinge on an i.i.d. resampling of the residuals obtained from the fitted model. In the AR(1) case, the *fitted* residuals are obtained as $\hat{\varepsilon}_t = Y_t - \hat{\phi}_1 Y_{t-1}$ for $t = 2, 3, \cdots, n$. Nevertheless, in Chap. 3 we made the case that resampling the *predictive* residuals gives more accurate prediction intervals in regression, be it linear or nonparametric. Section 7.3 defines a particular notion of predictive residuals in autoregression, and shows their potential benefit in constructing bootstrap prediction intervals.

7.2.2 Prediction Intervals for Autoregressive Processes

Let $Y_1, \cdots, Y_n$ be an observed stretch of a time series that follows a stationary autoregressive model of order p, i.e., model (7.1) or (7.2); the autoregression can be linear, nonlinear, or nonparametric. Denote by $\hat{Y}_{n+1}$ a point predictor of Y_{n+1} based on the data $\underline{Y}_n = (Y_1, \ldots, Y_n)'$. Let $\hat{V}_n^2$ be an estimate of $\mathrm{Var}(Y_{n+1} - \hat{Y}_{n+1} | Y_1, \cdots, Y_n)$ which is the conditional variance in one-step-ahead prediction. Given a bootstrap pseudo series $Y_1^*, \cdots, Y_n^*, Y_{n+1}^*$, analogs of the aforementioned quantities can be defined, i.e., $\hat{Y}_{n+1}^*$ and $\hat{V}_n^*$. Recall that bootstrap probability and expectation are usually denoted by P^* and E^*, and they are understood to be conditional on the original data $Y_1 = y_1, \cdots, Y_n = y_n$.

Our objective is to construct a prediction interval for Y_{n+1} given the data $\underline{Y}_n$. The salient point in all bootstrap algorithms discussed in this monograph is to use the bootstrap distribution of the bootstrap predictive root $Y_{n+1}^* - \hat{Y}_{n+1}^*$ or studentized predictive root $(Y_{n+1}^* - \hat{Y}_{n+1}^*)/\hat{V}_n^*$ to estimate the true distribution of the real-world predictive root $Y_{n+1} - \hat{Y}_{n+1}$ or studentized predictive root $(Y_{n+1} - \hat{Y}_{n+1})/\hat{V}_n$, respectively.

Note that in our causal autoregressive models (7.1) and (7.2) there are no extraneous regressors; rather, Y_t is regressed on its own past p values. We may then define $X_{t-1} = (Y_{t-1}, \ldots, Y_{t-p})'$ in which case Eqs. (7.1) and (7.2) can be re-written as:

- **AR model with homoscedastic errors**

$$Y_t = \mu(X_{t-1}) + \varepsilon_t \text{ with } \varepsilon_t \sim \text{ i.i.d. } (0, \sigma^2) \tag{7.6}$$

- **AR model with heteroscedastic errors**

$$Y_t = \mu(X_{t-1}) + \sigma(X_{t-1})\varepsilon_t \text{ with } \varepsilon_t \sim \text{ i.i.d. } (0, 1) \tag{7.7}$$

thus resembling very closely the regression models of Chap. 3; the main difference here is that the scatterplot pairs $\{(Y_t, X_{t-1}) \text{ for } t = p+1, \ldots, n\}$ are no longer independent.

Taking the regression analogy a bit further, recall that in Chap. 3 the future (unobserved) scatterplot pair was denoted (Y_f, x_f) which in our case would be (Y_{n+1}, X_n). Notation aside, here lies the main difficulty: we are interested in the response associated with a regressor value x_f that is fixed to the value $X_n = (Y_n, \ldots, Y_{n-p+1})'$ which is actually part of the dataset $\underline{Y}_n$.

Hence, as described in Sect. 7.2.1 on the forward vs. backward bootstrap, the tension lies between the need to create bootstrap datasets $\underline{Y}_n^* = (Y_1^*, \ldots, Y_n^*)'$ in order to capture the variability of the predictor $\hat{Y}_{n+1}$ while keeping the last p values of the bootstrap dataset fixed to the original ones. To do that, it is very convenient to base our predictive inference on Markovian models such as (7.6) and (7.7) that result into point predictors of type (7.4) that only utilize the finite history.[1]

7.2.3 Pertinent Prediction Intervals in Model-Based Autoregression

In Sect. 3.6.2, the notion of asymptotic pertinence was defined for model-based prediction intervals in regression. To extend these ideas to autoregression, consider first the homoscedastic model (7.1), and recall that Eq. (7.3) implies that the MSE–optimal predictor of Y_{n+1} given $Y_1 = y_1, \ldots, Y_n = y_n$ is $\mu(y_n, \ldots, y_{n-p+1})$. Hence we set $\hat{Y}_{n+1} = \hat{m}(y_n, \ldots, y_{n-p+1})$ where $\hat{m}(\cdot)$ is a consistent estimator of $\mu(\cdot)$. Assume that $\hat{m}(\cdot)$ has rate of convergence a_n, i.e., $a_n(\hat{m}(\cdot) - \mu(\cdot))$ has a well-defined, nontrivial asymptotic distribution where $a_n \to \infty$ as $n \to \infty$. Then, the predictive root is given by

$$Y_{n+1} - \hat{Y}_{n+1} = \varepsilon_{n+1} + A_\mu \tag{7.8}$$

where $A_\mu = \mu(y_n, \ldots, y_{n-p+1}) - \hat{m}(y_n, \ldots, y_{n-p+1}) = O_p(1/a_n)$ represents the estimation error.

Similarly, the bootstrap predictive root can be written as

$$Y_{n+1}^* - \hat{Y}_{n+1}^* = \varepsilon_{n+1}^* + A_\mu^* \tag{7.9}$$

[1] However, as shown in Chap. 9, the Model-free Prediction Principle can still be applied even when a time series is not Markov.

where $A^*_\mu = \hat{m}(y_n,\ldots,y_{n-p+1}) - \hat{m}^*(y_n,\ldots,y_{n-p+1})$. As before, the model-based bootstrap should be capable of asymptotically capturing both the pure prediction error, i.e., the distribution of ε_{n+1}, and the estimation error.

Definition 7.2.1 ***Asymptotic pertinence of bootstrap prediction intervals under model*** (7.1)**.** *Consider a bootstrap prediction interval for Y_{n+1} that is based on approximating the distribution of the predictive root $Y_{n+1} - \hat{Y}_{n+1}$ of Eq.* (7.8) *by the distribution of the bootstrap predictive root $Y^*_{n+1} - \hat{Y}^*_{n+1}$ of Eq.* (7.9)*. The interval will be called* asymptotically pertinent *provided the bootstrap satisfies the following three conditions as $n \to \infty$ conditionally on $Y_{n-p+1} = y_{n-p+1}, \cdots, Y_n = y_n$.*

(i) $\sup_a |P(\varepsilon_{n+1} \le a) - P^*(\varepsilon^*_{n+1} \le a)| \xrightarrow{P} 0$*, presupposing that the error distribution is continuous.*

(ii) $|P(a_n A_\mu \le a) - P^*(a_n A^*_\mu \le a)| \xrightarrow{P} 0$ *for some sequence $a_n \to \infty$, and for all points a where the assumed nontrivial limit of $P(a_n A_\mu \le a)$ is continuous.*

(iii) ε^*_{n+1} *and* A^*_μ *are independent in the bootstrap world—as their analogs are in the real world due to the causality assumption* (7.3)*.*

Furthermore, the bootstrap prediction interval for Y_{n+1} that is based on the approximating the distribution of the studentized *predictive root $(Y_{n+1} - \hat{Y}_{n+1})/\hat{V}_n$ by the distribution of the bootstrap studentized predictive root $(Y^*_{n+1} - \hat{Y}^*_{n+1})/\hat{V}^*_n$ will be called asymptotically pertinent if, in addition to (i)—(iii) above, the following also holds:*

(iv) $\hat{V}_n/\hat{V}^*_n \xrightarrow{P} 1$.

Consider now the heteroscedastic model (7.2). Much of the above discussion carries over *verbatim*; for example, our predictor of Y_{n+1} given $Y_1 = y_1, \ldots, Y_n = y_n$ is still $\hat{Y}_{n+1} = \hat{m}(y_n,\ldots,y_{n-p+1})$. The only difference is that the predictive root now is

$$Y_{n+1} - \hat{Y}_{n+1} = \sigma(y_n,\ldots,y_{n-p+1})\varepsilon_{n+1} + A_\mu, \tag{7.10}$$

and the bootstrap predictive root is

$$Y^*_{n+1} - \hat{Y}^*_{n+1} = \hat{\sigma}(y_n,\ldots,y_{n-p+1})\varepsilon^*_{n+1} + A^*_\mu \tag{7.11}$$

where $\hat{\sigma}(\cdot)$ is the (consistent) estimator of $\sigma(\cdot)$ that is employed in the bootstrap data generation mechanism. Hence, the following definition is immediate.

Definition 7.2.2 ***Asymptotic pertinence of bootstrap prediction intervals under model*** (7.2)**.** *Consider a bootstrap prediction interval for Y_{n+1} that is based on approximating the distribution of the predictive root $Y_{n+1} - \hat{Y}_{n+1}$ of Eq.* (7.10) *by the distribution of the bootstrap predictive root $Y^*_{n+1} - \hat{Y}^*_{n+1}$ of Eq.* (7.11)*. The interval will be called* asymptotically pertinent *provided the bootstrap satisfies conditions (i)—(iii) of Definition 7.2.1 together with the additional requirement:*

(iv′) $\sigma(y_n,\ldots,y_{n-p+1}) - \hat{\sigma}(y_n,\ldots,y_{n-p+1}) \xrightarrow{P} 0$.

Furthermore, the bootstrap prediction interval for Y_{n+1} that is based on the approximating the distribution of the studentized *predictive root $(Y_{n+1} - \hat{Y}_{n+1})/\hat{V}_n$ by the distribution of the bootstrap studentized predictive root $(Y^*_{n+1} - \hat{Y}^*_{n+1})/\hat{V}^*_n$ will be called asymptotically pertinent if, in addition condition (iv) of Definition 7.2.1 also holds.*

Remark 7.2.2 Taking into account that $A_\mu = o_p(1)$ as $n \to \infty$, a simple estimator for the (conditional) variance of the predictive root $Y_{n+1} - \hat{Y}_{n+1}$ under model (7.2) is $\hat{V}_n = \hat{\sigma}(y_n, \ldots, y_{n-p+1})$. Thus, condition (iv) or Definition 7.2.1 can be re-written as

$$\hat{\sigma}(y_n, \ldots, y_{n-p+1}) - \hat{\sigma}^*(y_n, \ldots, y_{n-p+1}) \xrightarrow{P} 0,$$

i.e., it is just a bootstrap version of condition (iv′) or Definition 7.2.2. As a matter of fact, resampling in the heteroscedastic model (7.2) entails using studentized residuals. In this case, the predictive root method gives results that are almost identical to the studentized predictive root method when the simple estimator $\hat{V}_n = \hat{\sigma}(y_n, \ldots, y_{n-p+1})$ is used; see Remark 3.6.3 for an analogous discussion.

7.3 Bootstrap Prediction Intervals for Linear Autoregressions

Consider the strictly stationary, causal AR(p) model defined by the recursion

$$Y_t = \phi_0 + \sum_{j=1}^{p} \phi_j Y_{t-j} + \varepsilon_t \tag{7.12}$$

which is a special case of model (7.1) with the ε_t being i.i.d. with mean zero, variance σ^2, and distribution F_ε. The assumed causality condition (7.3) is now tantamount to $\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p \neq 0$ for $|z| \leq 1$. Denote $\underline{\phi} = (\phi_0, \phi_1, \phi_2, \cdots, \phi_p)'$ the vector of autoregressive parameters, and $\underline{\hat{\phi}} = (\hat{\phi}_0, \hat{\phi}_1, \cdots, \hat{\phi}_p)'$ and $\hat{\phi}(z) = 1 - \hat{\phi}_1 z - \cdots - \hat{\phi}_p z^p$ the respective estimates. Let $\hat{Y}_t$ be the "fitted" value of Y_t, i.e., $Y_t = \hat{\phi}_0 + \sum_{j=1}^{p} \hat{\phi}_j Y_{t-j}$; in other words, $\hat{Y}_t$ is our approximation to the MSE–optimal predictor of Y_t given the past. As before, let $X_t = (Y_t, \cdots, Y_{t-p+1})'$ be the vector of the last p observations up to Y_t.

As described in Sect. 7.2.1, the idea of forward bootstrap method is that given observations $Y_1 = y_1, \ldots, Y_n = y_n$, we can use the fitted AR recursion to generate bootstrap series "forward" in time starting from some initial conditions. This recursion stops when n bootstrap data have been generated; to generate the $(n+1)$th bootstrap point (and beyond), the recursion has to be re-started with different initial values that are fixed to be the last p original observations. The details for estimating the coefficients, generating the bootstrap pseudo-data and constructing the prediction intervals using both fitted and predictive residuals are given below in Sects. 7.3.1 and 7.3.2

7.3.1 Forward Bootstrap with Fitted Residuals

Given a sample $Y_1 = y_1, \ldots, Y_n = y_n$ from (7.12), the following are the steps needed to construct the prediction interval for future value Y_{n+1} based on the predictive root method.

Algorithm 7.3.1 FORWARD BOOTSTRAP WITH FITTED RESIDUALS (FF)

1. *Use all observations $y_1, \cdots, y_n$ to obtain the Least Squares (LS) estimators $\underline{\hat{\phi}} = (\hat{\phi}_0, \hat{\phi}_1, \cdots, \hat{\phi}_p)'$ by fitting the following linear model*

$$\begin{pmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_{p+1} \end{pmatrix} = \begin{bmatrix} 1 & y_{n-1} & \cdots & y_{n-p} \\ 1 & y_{n-2} & \cdots & y_{n-p-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & y_p & \cdots & y_1 \end{bmatrix} \begin{pmatrix} \phi_0 \\ \phi_1 \\ \vdots \\ \phi_p \end{pmatrix} + \begin{pmatrix} \varepsilon_n \\ \varepsilon_{n-1} \\ \vdots \\ \varepsilon_{p+1} \end{pmatrix}. \tag{7.13}$$

2. *For $t = p+1, \cdots, n$, compute the fitted value and fitted residuals:*

$$\hat{y}_t = \hat{\phi}_0 + \sum_{j=1}^{p} \hat{\phi}_j y_{t-j}, \text{ and } \hat{\varepsilon}_t = y_t - \hat{y}_t.$$

3. *Center the fitted residuals: let $r_t = \hat{\varepsilon}_t - \bar{\hat{\varepsilon}}$ for $t = p+1, \cdots, n$, and $\bar{\hat{\varepsilon}} = (n-p)^{-1} \sum_{p+1}^{n} \hat{\varepsilon}_t$; let the empirical distribution of r_t be denoted by $\hat{F}_n$.*
   (a) *Draw bootstrap pseudo-residuals $\varepsilon_1^*, \varepsilon_2^*, \ldots$ i.i.d. from $\hat{F}_n$.*
   (b) *To ensure stationarity of the bootstrap series, we can use an arbitrary initial condition such as $(u_1^*, \cdots, u_p^*) = (0, \ldots, 0)$, generate $n+M$ pseudo-data for some large positive integer M, and then discard the first M data. In other words, generate $\{u_t^*, t \geq p+1\}$ by the recursion:*

$$u_t^* = \hat{\phi}_0 + \sum_{j=1}^{p} \hat{\phi}_j u_{t-j}^* + \varepsilon_t^* \text{ for } t = p+1, \cdots, n+M.$$

   Finally, define $y_t^* = u_{M+t}^*$ for $t = 1, 2, \cdots, n$.
   (c) *Based on the pseudo-data $\{y_1^*, \cdots, y_n^*\}$, re-estimate the coefficients $\underline{\phi}$ by the LS estimator $\underline{\hat{\phi}}^* = (\hat{\phi}_0^*, \hat{\phi}_1^*, \cdots, \hat{\phi}_p^*)'$ as in step 1. Then compute the bootstrap predicted value*

$$\hat{y}_{n+1}^* = \hat{\phi}_0^* + \sum_{j=1}^{p} \hat{\phi}_j^* \, y_{n+1-j}^*.$$

   (d) *In order to conduct conditionally valid predictive inference,* re-define *the last p observations to match the original observed values, i.e., let $y_{n-p+1}^* = y_{n-p+1}, \cdots, y_n^* = y_n$. Then, generate the future bootstrap observation*

$$y^*_{n+1} = \hat{\phi}_0 + \sum_{j=1}^{p} \hat{\phi}_j y^*_{n+1-j} + \varepsilon^*_{n+1}.$$

(e) Calculate a bootstrap root replicate as $y^*_{n+1} - \hat{y}^*_{n+1}$.

4. *Steps (a)–(e) above are repeated B times, and the B bootstrap replicates are collected in the form of an empirical distribution whose* α*-quantile is denoted* $q(\alpha)$.
5. *Compute the predicted value* $\hat{y}_{n+1} = \hat{\phi}_0 + \sum_{j=1}^{p} \hat{\phi}_j y_{n+1-j}$.
6. *Construct the* $(1-\alpha)100\%$ *equal-tailed prediction interval for* Y_{n+1} *as*

$$[\hat{y}_{n+1} + q(\alpha/2), \hat{y}_{n+1} + q(1-\alpha/2)]. \tag{7.14}$$

Remark 7.3.1 Step 3 (b) of the above algorithm describes one method to generate a stationary stretch of a time series defined by an autoregressive (or in general Markovian) structure; the technique allows the practitioner to not worry about the initial conditions. A different approach is to generate the starting points of the autoregression from its stationary distribution, e.g., replace Step 3 (b) by (b′) below:

(b′) Let $(y^*_1, \cdots, y^*_p)$ *be chosen at random from the set of p–tuplets* $\{(y_k, \cdots, y_{k+p-1})$ *for* $k = 1, \cdots, n-p+1\}$. *Then, generate* $\{y^*_t, t \geq p+1\}$ *by the recursion:*

$$y^*_t = \hat{\phi}_0 + \sum_{j=1}^{p} \hat{\phi}_j y^*_{t-j} + \varepsilon^*_t \ \text{ for } t = p+1, \cdots, n.$$

In what follows, we will use either (or both) of these techniques in order to generate stationary autoregressive (or Markovian) time series in the bootstrap world.

Remark 7.3.2 Algorithm 7.3.1 focuses on one-step-ahead prediction for simplicity. However, it is straightforward to extend these results—as well as those in the sequel—in order to construct a prediction interval for Y_{n+h} for some $h \geq 1$ based on the data $\underline{Y}_n$. In addition, the use of resampling affords us the possibility of constructing joint, i.e., simultaneous, prediction intervals for $Y_{n+1}, \ldots, Y_{n+h}$ with prespecified coverage level; details are given in Pan and Politis (2015).

7.3.2 Forward Bootstrap with Predictive Residuals

As in Chap. 3, we may consider using predictive—as opposed to fitted—residuals for the bootstrap. We define the predictive residuals in the AR context as $\hat{\varepsilon}^{(t)}_t = y_t - \hat{y}^{(t)}_t$ where $\hat{y}^{(t)}_t$ is computed from the delete-y_t data set, i.e., the available data for the scatterplot of y_k vs. $\{y_{k-p}, \cdots, y_{k-1}\}$ over which the LS fitting that takes place excludes the single point that corresponds to $k = t$. The forward bootstrap with predictive residuals is similar to Algorithm 7.3.1 except for Step 2.

Algorithm 7.3.2 FORWARD BOOTSTRAP WITH PREDICTIVE RESIDUALS (FP)

1. *Same as step 1 of Algorithm 7.3.1.*
2. *Use the delete-y_t dataset to compute the LS estimator*

$$\underline{\hat{\phi}}^{(t)} = (\hat{\phi}_0^{(t)}, \hat{\phi}_1^{(t)}, \cdots, \hat{\phi}_p^{(t)})'$$

as in step 1, i.e., compute $\underline{\hat{\phi}}^{(t)}$ by changing regression model (7.13) as follows. Delete the row of y_t at the left-hand side of (7.13), delete the row $(1, y_{t-1}, \cdots, y_{t-p})$ in the design matrix, delete ε_t from the vector of εs, and replace $\underline{\hat{\phi}}$ by $\underline{\hat{\phi}}^{(t)}$ throughout.
Then, calculate the delete-y_t fitted values:

$$\hat{y}_t^{(t)} = \hat{\phi}_0^{(t)} + \sum_{j=1}^{p} \hat{\phi}_j^{(t)} y_{t-j}, \ \ for \ \ t = p+1, \cdots, n$$

and the predictive residuals: $\hat{\varepsilon}_t^{(t)} = y_t - \hat{y}_t^{(t)}$ for $t = p+1, \cdots, n$.

3–6. *Replace the $\hat{\varepsilon}_t$ by $\hat{\varepsilon}_t^{(t)}$; the rest is the same as in Algorithm 7.3.1.*

Remark 7.3.3 The LS estimator $\underline{\hat{\phi}}$ is asymptotically equivalent to the popular Yule-Walker (YW) estimators for fitting AR models. The advantage of YW estimators is that they almost surely lead to a causal fitted model. By contrast, the LS estimator $\underline{\hat{\phi}}$ is only asymptotically causal but it is completely scatterplot-based, and thus convenient in terms of our notion of predictive residuals. Indeed, for any bootstrap method using fitted residuals (studentized or not), e.g., the forward Algorithm 7.3.1 or the backward Algorithm 7.3.5 in the sequel, we could equally employ the Yule-Walker instead of the LS estimators. But for methods using our notion of predictive residuals, it is most convenient to be able to employ the LS estimators. If the LS estimator $\underline{\hat{\phi}}$ is causal—as it is hopefully the case—we can use either fitted or predictive residuals but will need to discard all bootstrap pseudo-series that lead to a non-causal $\underline{\hat{\phi}}^*$; this is equally important for the Backward Bootstrap discussed in Sect. 7.3.4.

7.3.3 Forward Bootstrap Based on Studentized Roots

In the previous two subsections, the forward bootstrap based on predictive roots was described. However, as already mentioned, we can use studentized predictive roots instead. The forward studentized bootstrap procedure with fitted and/or predictive residuals is similar to Algorithm 7.3.1; the only differences are in step 3(e) and 6.

The key observation is that in the causal AR(p) model (7.12), σ^2 is the variance of the MSE–optimal predictor of Y_{n+1}; this can be interpreted as either conditional or unconditional variance since the two coincide in a linear AR(p) model. Hence,

we can estimate the variance of $\hat{Y}_{n+1}$ by $\hat{\sigma}^2$, i.e., the sample variance of the fitted residuals. Also let $\hat{\sigma}^{*2}$ denote the sample variance of the bootstrap fitted residuals that are defined as $y_t^* - \hat{y}_t^*$ for $t = p+1, \cdots, n$.

Algorithm 7.3.3 FORWARD STUDENTIZED BOOTSTRAP WITH FITTED RESIDUALS (FSF)
The algorithm is the same as Algorithm 7.3.1 except for steps 3(e) and 6 that should be replaced by the following steps:

3. (e) *Calculate a studentized bootstrap root replicate as* $(y_{n+1}^* - \hat{y}_{n+1}^*)/\hat{\sigma}^*$.

6. *Construct the* $(1-\alpha)100\%$ *equal-tailed predictive interval for* Y_{n+1} *as*

$$[\hat{y}_{n+1} + \hat{\sigma} q(\alpha/2),\ \hat{y}_{n+1} + \hat{\sigma} q(1-\alpha/2)] \tag{7.15}$$

where $q(\alpha)$ *is the* α*-quantile of the empirical distribution of the B studentized bootstrap roots.*

As in Sect. 7.3.2, we can resample the predictive—as opposed to the fitted—residuals.

Algorithm 7.3.4 FORWARD STUDENTIZED BOOTSTRAP WITH PREDICTIVE RESIDUALS (FSP)

1. *Same as step 1 in Algorithm 7.3.1.*

2. *Same as step 2 in Algorithm 7.3.2*

3–6. *Replace the* $\hat{\varepsilon}_t$ *by* $\hat{\varepsilon}_t^{(t)}$*; the rest is the same as in Algorithm 7.3.3*

Remark 7.3.4 As in the regression case discussed in Chap. 3, the Fp method yields improved coverage as compared to the Ff method since predictive residuals are inflated as compared to fitted residuals. Interestingly, the FSp method is not much better than the FSf method in finite samples. The reason is that when we studentize the predictive residuals, the aforementioned inflation effect is offset by the simultaneously inflated bootstrap estimator $\hat{\sigma}^*$ in the denominator. In the Monte Carlo simulations of Sect. 7.5, we will see that the Fp, FSf, and FSp methods have similarly good performance while the Ff method is the worst, exhibiting pronounced undercoverage.

Remark 7.3.5 Under standard assumptions, all four methods Ff, Fp, FSf, and FSp yield prediction intervals that are asymptotically valid and pertinent; see Pan and Politis (2015) for details.

7.3.4 Backward Bootstrap

The difference of the backward to the forward bootstrap is in the way they generate the bootstrap pseudo-data $Y_1^*, \cdots, Y_n^*$. The idea of the backward bootstrap is to start

from the last p observations (that are given) and generate the bootstrap-pseudo data $\{Y^*_{n-p}, \cdots, Y^*_1\}$ backward in time using the backward representation

$$\phi(B^{-1})Y_t = \phi_0 + w_t$$

where B is the backward shift operator: $B^k Y_t = Y_{t-k}$, and $\{w_t\}$ is the backward noise defined by

$$w_t = \frac{\phi(B^{-1})}{\phi(B)}\varepsilon_t. \tag{7.16}$$

Thombs and Schucany (1990) generated the fitted backward residuals $\hat{w}_t = y_t - \hat{\phi}_0 - \hat{\phi}_1 y_{t+1} - \cdots - \hat{\phi}_p y_{t+p}$ for $t = 1, 2, \cdots, n-p$. Then they fixed the last p values of the data, and generated the pseudo series backwards through the following backwards recursion:

$$y^*_t = \hat{\phi}_0 + \hat{\phi}_1 y^*_{t+1} + \cdots + \hat{\phi}_p y^*_{t+p} + w^*_t \text{ for } t = n-p, n-p-1, \cdots, 1$$

with w^*_t being generated i.i.d. from $\hat{F}_w$, the empirical distribution of the (centered) $\hat{w}_t$s.

However, as pointed out by Breidt et al. (1995), although the backward errors w_ts are uncorrelated, they are not necessarily independent. Therefore, it is not advisable to resample $\{w^*_t\}$ i.i.d. from $\hat{F}_w$. Nevertheless, the forward errors ε_t are independent; so we can generate ε^*_t i.i.d. from $\hat{F}_n$. After obtaining the ε^*_ts, it is possible to generate the bootstrapped backward noise w^*_t using the bootstrap analog of (7.16), i.e.,

$$w^*_t = \frac{\hat{\phi}(B^{-1})}{\hat{\phi}(B)}\varepsilon^*_t.$$

The following algorithm for backward bootstrap with fitted residuals is identical to that of Breidt et al. (1995). However, we also describe the backward bootstrap using predictive residuals which has better finite-sample properties. In addition, we address the construction of prediction intervals via either unstudentized or studentized predictive roots.

Algorithm 7.3.5 BACKWARD BOOTSTRAP WITH FITTED RESIDUALS (BF)

1–2. Same as steps 1–2 in Algorithm 7.3.1.

3. Center the fitted residuals: define $r_t = \hat{\varepsilon}_t - \bar{\hat{\varepsilon}}$ for $t = p+1, \cdots, n$ where $\bar{\hat{\varepsilon}} = (n-p)^{-1}\sum_{p+1}^{n}\hat{\varepsilon}_t$; the empirical distribution of r_t is denoted by $\hat{F}_n$.

(a) Choose a large positive integer M and create the independent bootstrap pseudo-noise $\varepsilon^*_{-M}, \cdots, \varepsilon^*_n, \varepsilon^*_{n+1}, \cdots$ from $\hat{F}_n$; then generate the bootstrap backward noises $\{w^*_t, t = -M, \cdots, n\}$ recursively as follows:

$$w^*_t = \begin{cases} 0, & t < -M \\ \hat{\phi}_1 w^*_{t-1} + \cdots + \hat{\phi}_p w^*_{t-p} + \varepsilon^*_t - \hat{\phi}_1 \varepsilon^*_{t+1} - \cdots - \hat{\phi}_p \varepsilon^*_{t+p}, & t \geq -M \end{cases}$$

(b) *Fix the last p values,i.e.,* $y_n^* = y_n, \cdots, y_{n-p+1}^* = y_{n-p+1}$, *and then generate a bootstrap realization* $\{y_t^*\}$ *by the backward recursion:*

$$y_t^* = \begin{cases} \hat{\phi}_0 + \hat{\phi}_1 y_{t+1}^* + \cdots + \hat{\phi}_p y_{t+p}^* + w_t^* & t = n-p, n-p-1, \cdots, 1 \\ y_t & t = n, n-1, \cdots, n-p+1. \end{cases}$$

(c) *Based on the pseudo-data* $\{y_1^*, \cdots, y_n^*\}$, *re-estimate the coefficients in* $\underline{\phi}$ *by LS estimators* $\hat{\underline{\phi}}^* = (\hat{\phi}_0^*, \hat{\phi}_1^*, \cdots, \hat{\phi}_p^*)'$ *as in step 1. Then compute the bootstrap predicted value* $\hat{y}_{n+1}^* = \hat{\phi}_0^* + \sum_{j=1}^p \hat{\phi}_j^* \hat{y}_{n+1-j}^*$.

(d) *Compute the future bootstrap observation* $y_{n+1}^* = \hat{\phi}_0 + \sum_{j=1}^p \hat{\phi}_j y_{n+1-j}^* + \varepsilon_{n+1}^*$.

(e) *Calculate a bootstrap root replicate* $y_{n+1}^* - \hat{y}_{n+1}^*$.

4. *Steps (a)–(e) in the above are repeated B times, and the B bootstrap replicates are collected in the form of an empirical distribution whose* α*-quantile is denoted* $q(\alpha)$.
5. *Compute the predicted value* $\hat{y}_{n+1} = \hat{\phi}_0 + \sum_{j=1}^p \hat{\phi}_j \hat{y}_{n+1-j}$.
6. *Construct the* $(1-\alpha)100\%$ *equal-tailed predictive interval for* Y_{n+1} *as*

$$[\hat{y}_{n+1} + q(\alpha/2),\ \hat{y}_{n+1} + q(1-\alpha/2)]. \tag{7.17}$$

Algorithm 7.3.6 BACKWARD BOOTSTRAP WITH PREDICTIVE RESIDUALS (BP)

1–2. Same as steps 1–2 of Algorithm 7.3.2

3–6. Change the $\hat{\varepsilon}_t$ *into* $\hat{\varepsilon}_t^{(t)}$, *i.e., the predictive residuals defined in step 2 of Algorithm 7.3.2; the rest is the same as in Algorithm 7.3.5.*

Algorithm 7.3.7 BACKWARD STUDENTIZED BOOTSTRAP WITH FITTED RESIDUALS (BSF)

This algorithm is the same as Algorithm 7.3.5 with the exception of steps 3(e) and 6 that should be replaced by steps 3(e) and 6 of Algorithm 7.3.3.

Algorithm 7.3.8 BACKWARD BOOTSTRAP WITH PREDICTIVE RESIDUALS (BSP)

Replace the $\hat{\varepsilon}_t$ *by* $\hat{\varepsilon}_t^{(t)}$, *the predictive residuals defined in step 2 of Algorithm 7.3.2; the rest is the same as in Algorithm 7.3.7.*

Remark 7.3.6 The asymptotic validity of the backward bootstrap prediction interval based on fitted residuals, i.e., interval (7.17), has been proven by Breidt et al. (1995). It is not hard to see that the property of asymptotic pertinence also holds true here; the same is true for the backward bootstrap prediction intervals that use predictive residuals.

7.3.5 Generalized Bootstrap Prediction Intervals

Chatterjee and Bose (2005) introduced the generalized bootstrap method for estimators obtained by solving estimating equations. The LS estimators of the AR coefficients is a special case. With a bootstrapped weight $(w_{n1}, \cdots, w_{nn})$ in the estimating equations, the generalized bootstrapped estimators are obtained simply by solving the bootstrapped estimating equations. The generalized bootstrap method is computationally fast because we do not need to generate the pseudo-series; instead, we just resample the weights $(w_{n1}, \cdots, w_{nn})$ from some distribution, e.g., Multinomial$(n; 1/n, \cdots, 1/n)$.

Inspired by the idea of generalized bootstrap, Pan and Politis (2015) proposed the following bootstrap approach for bootstrap prediction intervals in linear AR models.

Algorithm 7.3.9 GENERALIZED BOOTSTRAP WITH FITTED RESIDUALS (GF)

1–2. Same as the steps in Algorithm 7.3.1.

3.(a) Calculate the bootstrapped estimator of the coefficients

$$\underline{\hat{\phi}}^* = (X'WX)^{-1}X'WY,$$

where $X = \begin{bmatrix} 1 & y_{n-1} & \cdots & y_{n-p} \\ 1 & y_{n-2} & \cdots & y_{n-p-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & y_p & \cdots & y_1 \end{bmatrix}$, $Y = \begin{pmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_{p+1} \end{pmatrix}$

and W is a diagonal matrix whose diagonal elements $(w_1, \cdots, w_{n-p})$ are sampled from Multinomial $(n-p; 1/(n-p), \cdots, 1/(n-p))$.

(b) Compute the bootstrap predicted value and future observation by

$$\hat{y}^*_{n+1} = \hat{\phi}^*_0 + \sum_{j=1}^{p} \hat{\phi}^*_j y_{n+1-j} \quad \text{and} \quad y^*_{n+1} = \hat{\phi}_0 + \sum_{j=1}^{p} \hat{\phi}_j y_{n+1-j} + \varepsilon^*_{n+1}$$

respectively; as usual, ε^*_{n+1} is a random draw from the empirical distribution of the (centered) fitted residuals. Finally, calculate the bootstrap predictive root replicate as $y^*_{n+1} - \hat{y}^*_{n+1}$.

4–6. Same as the corresponding steps from Algorithm 7.3.1.

The generalized bootstrap can also be performed using the predictive residuals.

Algorithm 7.3.10 GENERALIZED BOOTSTRAP WITH PREDICTIVE RESIDUALS (GP)

The algorithm is identical to Algorithm 7.3.9 with the following changes: replace step 2 of Algorithm 7.3.9 with step 2 of Algorithm 7.3.2, and use the predictive residuals instead of the fitted residuals in step 3(b) of Algorithm 7.3.9.

Under regularity conditions, Chatterjee and Bose (2005) proved the consistency of the Generalized bootstrap in estimating the distribution of $\sqrt{n}(\hat{\underline{\phi}} - \underline{\phi})$; it follows that both Gf and Gp prediction intervals are asymptotically pertinent.

7.4 Alternative Approaches to Bootstrap Prediction Intervals for Linear Autoregressions

Box and Jenkins (1976) proposed a widely used $(1-\alpha)100\%$ prediction interval for a Gaussian AR(p) model as

$$[\hat{y}_{n+1} + z_{\alpha/2}\hat{\sigma},\ \hat{y}_{n+1} + z_{1-\alpha/2}\hat{\sigma}], \tag{7.18}$$

where z_α is the α–quantile of a standard normal variate, and $\hat{\sigma}^2$ is an estimate of the innovation variance σ^2. This prediction interval only takes into account the variability due to the future innovation but does not account for the variability from the estimation of the model, thus it is not asymptotically pertinent; in fact, it is the analog of the naive interval (3.21). Still under the assumption of Gaussian errors, Stine (1987) proposed a resampling-based interval that manages to capture estimation variability.

To relax the often unrealistic assumption of Gaussianity, Masarotto (1990) proposed a bootstrap method to construct the prediction interval as follows. Generate a pseudo series $y_1^*, \cdots, y_n^*, \cdots, y_{n+1}^*$ from the fitted model, and generate the studentized bootstrap predictive root $r^* = (y_{n+1}^* - \hat{y}_{n+1}^*)/\hat{\sigma}^*$. Having generated B replicate values of r^*, these can be sorted in increasing order as $r_1^* \leq \cdots \leq r_B^*$. Letting $k = \lfloor B\alpha \rfloor$, Masarotto's $(1-\alpha)100\%$ prediction interval is

$$[\hat{y}_{n+1} + r_k^*\hat{\sigma},\ \hat{y}_{n+1} + r_{B-k}^*\hat{\sigma}]. \tag{7.19}$$

The main difference of the above from our studentized prediction intervals is that we make it a point—either using Backward or Forward bootstrap—to fix the last p bootstrap pseudo-values to the values present in the original series with regard to generating out-of-sample bootstrap data and/or predictors. For example, we obtain the bootstrap predicted value $\hat{y}_{n+1}^*$ and future value y_{n+1}^* in steps 3(c) and 3(d) of Algorithm 7.3.1 using the original datapoints $y_{n-p+1}, \ldots, y_n$, thus ensuring the property of asymptotic pertinence, i.e., capturing the estimation error. In essence, Masarotto's (1990) method omits step B from our Forward bootstrap procedure outlined in Sect. 7.2.1. Consequently, Masarotto's interval (7.19) is not asymptotically pertinent although it has reasonably good performance in practice; see Sect. 7.5.2 for further discussion.

In this book, the focus has been on constructing prediction intervals based on a bootstrap approximation of the distribution of predictive roots. However, some authors have chosen to construct bootstrap prediction intervals via a *percentile* method reminiscent of Efron's well-known percentile method for confidence intervals.

To elaborate, the percentile method uses the bootstrap distribution of Y^*_{n+h} to estimate the distribution of the future value Y_{n+h} directly; the following methods are all based on the percentile method.

Cao et al. (1997) proposed a computationally fast bootstrap method in order to relax the Gaussianity assumption implicit in the Box/Jenkins interval (7.18). Conditionally on the last p observations, the authors only generate the future bootstrap observation

$$y^*_{n+1} = \hat{\phi}_0 + \hat{\phi}_1 y_n + \cdots + \hat{\phi}_p y_{n+1-p} + \hat{\varepsilon}^*_{n+1} \tag{7.20}$$

instead of generating the whole bootstrap series up to y^*_{n+1}. As was the case with the Box/Jenkins interval (7.18), the prediction interval of Cao et al. (1997) does not make any attempt to capture the variability stemming from model estimation.

Alonso et al. (2002) and Pascual et al. (2004) used a different way to generate the future bootstrap values, namely they defined

$$y^*_{n+1} = \hat{\phi}^*_0 + \hat{\phi}^*_1 y_n + \cdots + \hat{\phi}^*_p y_{n+1-p} + \hat{\varepsilon}^*_{n+1}. \tag{7.21}$$

Equation (7.21) generates the future pseudo-values using the parameters $\underline{\hat{\phi}}^*$ instead of $\underline{\hat{\phi}}$ as is customary; e.g., compare with recursion (7.20). We will call the percentile interval based on (7.21), the APR/PRR bootstrap method. Note that the APR/PRR interval does consider the variability from the model estimation albeit in a slightly different fashion than usual.

7.5 Simulations: Linear AR Models

In this section, we evaluate the performance of all the ten proposed bootstrap methods, i.e., four forward methods with fitted or predictive residuals using nonstudentized or studentized predictive root (Ff, Fp, FSf, and FSp), four corresponding backward methods (Bf, Bp, BSf, and BSp) and two generalized bootstrap methods (Gf and Gp), and compare them to the aforementioned older methods discussed in Sect. 7.4, i.e., Box and Jenkins (1976), Cao et al. (1997), Alonso et al. (2002)/Pascual et al. (2004) and Masarotto (1990); the latter four are abbreviated BJ, Cao, APR/PRR, and M respectively.

7.5.1 Unconditional Coverage Level

The simulation experiment had the following parameters:

(1) AR(1) model: $Y_{t+1} = \phi_1 Y_t + \varepsilon_t$ with $\phi_1 = 0.5$.
(2) Errors ε_t i.i.d. from $N(0,1)$ or two-sided exponential (Laplace) distribution rescaled to unit variance.

(3) 500 "true" datasets each of size $n = 50$ or 100; for each "true" dataset $B = 1000$ bootstrap pseudo-series were created.
(4) Prediction intervals with nominal coverage levels of 95 and 90 %.

Simulations with different AR(1) and AR(2) models were also performed with qualitatively similar results; see Pan and Politis (2015).

For the ith "true" dataset, one of the bootstrap methods was used to create $B = 1000$ bootstrap sample paths (step 4 of the algorithms), and construct the prediction interval (step 6 of the algorithms) $[L_i, U_i]$. To assess the corresponding empirical coverage level (CVR) and average length (LEN) of the constructed interval, 1000 one-step ahead future values $y_{n+1,j} = \hat{\phi}_1 y_{ni} + \varepsilon_j^*$ for $j = 1, 2, \cdots, 1000$ were also generated, where $\hat{\phi}_1$ is the estimate from the ith "true" dataset and y_{ni} is the ith dataset's last value. Then, the empirical coverage level and length from the ith dataset are calculated as

$$CVR_i = \frac{1}{1000} \sum_{j=1}^{1000} \mathbf{1}_{[L_i, U_i]}(y_{n+1,j}) \text{ and } LEN_i = U_i - L_i$$

where $\mathbf{1}_A(x)$ is the indicator function, i.e., $\mathbf{1}_A(x)$ equals 1 or 0 according to whether $x \in A$ or not. Note that the ability to generate the future values $y_{n+1,j}$ independently from the bootstrap datasets allows us to estimate CVR_i in a more refined way as opposed to the usual 0–1 coverage.

Finally, the coverage level and length for each bootstrap method is estimated by the average $\{CVR_i\}$ and $\{LEN_i\}$ over the 500 "true" datasets, i.e.

$$CVR = \frac{1}{500} \sum_{i=1}^{500} CVR_i \text{ and } LEN = \frac{1}{500} \sum_{i=1}^{500} LEN_i.$$

Note, however, that the value of the last observation y_{ni} is different from dataset to dataset; hence, the above CVR represents an *unconditional* coverage probability, i.e., an average of the conditional coverage probability discussed in the context of asymptotic validity.

Tables 7.1 and 7.2 summarize the findings of our simulation; the entry for st.err is the standard error associated with each average length. Some important features are as follows:

- As expected, all bootstrap prediction intervals considered are characterized by some degree of under-coverage. It is encouraging that the use of predictive residuals appears to partially correct the under-coverage problem in linear autoregression as was the case in linear regression; see Sect. 3.7.2.
- The Fp, Bp, and Gp methods using predictive residuals have uniformly improved CVRs as compared to Ff, Bf, and Gf using fitted residuals. The reason is that the finite-sample empirical distribution of the predictive residuals is very much like a re-scaled (inflated) version of the empirical distribution of fitted residuals.
- The price to pay for using predictive residuals is the increased variability of the interval length; this is true for the unstudentized methods only.

	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
Ff	0.930	3.848	0.490	0.881	3.267	0.386
Fp	0.940	4.011	0.506	0.895	3.405	0.406
Bf	0.929	3.834	0.500	0.880	3.261	0.393
Bp	0.941	4.017	0.521	0.896	3.410	0.410
FSf	0.942	4.036	0.501	0.894	3.391	0.395
FSp	0.941	4.028	0.493	0.894	3.393	0.399
BSf	0.941	4.016	0.514	0.894	3.388	0.402
BSp	0.942	4.033	0.500	0.896	3.402	0.398
Gf	0.930	3.847	0.483	0.881	3.264	0.389
Gp	0.940	4.007	0.502	0.895	3.402	0.399
BJ	0.934	3.832	0.402	0.880	3.216	0.338
M	0.946	4.510	0.599	0.898	3.792	0.493
Cao	0.917	3.720	0.532	0.871	3.199	0.417
APR/PRR	0.930	3.858	0.498	0.880	3.268	0.390
$n = 100$						
Ff	0.940	3.895	0.357	0.892	3.294	0.283
Fp	0.945	3.968	0.377	0.899	3.355	0.281
Bf	0.940	3.895	0.371	0.892	3.286	0.275
Bp	0.945	3.971	0.375	0.899	3.360	0.289
FSf	0.946	3.981	0.358	0.899	3.355	0.282
FSp	0.945	3.977	0.370	0.899	3.350	0.277
BSf	0.945	3.978	0.366	0.898	3.349	0.275
BSp	0.946	3.978	0.366	0.898	3.352	0.283
Gf	0.940	3.891	0.359	0.891	3.289	0.275
Gp	0.944	3.969	0.383	0.897	3.350	0.284
BJ	0.943	3.887	0.275	0.892	3.262	0.231
M	0.948	4.514	0.430	0.898	3.793	0.348
Cao	0.936	3.853	0.392	0.888	3.262	0.291
APR/PRR	0.939	3.893	0.368	0.891	3.283	0.283

Table 7.1 Simulation Results of AR(1) with normal innovations and $\phi_1 = 0.5$

	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
Ff	0.930	4.175	0.804	0.881	3.270	0.570
Fp	0.937	4.376	0.828	0.892	3.420	0.597
Bf	0.929	4.176	0.815	0.881	3.267	0.571
Bp	0.937	4.376	0.882	0.892	3.415	0.600
FSf	0.940	4.176	0.873	0.894	3.438	0.578
FSp	0.941	4.376	0.851	0.894	3.452	0.583
BSf	0.939	4.457	0.862	0.893	3.436	0.587
BSp	0.941	4.462	0.875	0.895	3.443	0.583
Gf	0.930	4.177	0.774	0.881	3.274	0.577
Gp	0.937	4.367	0.864	0.892	3.420	0.611
BJ	0.923	3.812	0.603	0.885	3.199	0.506
M	0.942	4.827	0.960	0.897	3.817	0.692
Cao	0.921	4.065	0.863	0.873	3.197	0.605
APR/PRR	0.930	4.211	0.832	0.882	3.279	0.573
$n = 100$						
Ff	0.939	4.208	0.612	0.891	3.274	0.431
Fp	0.943	4.302	0.638	0.897	3.344	0.439
Bf	0.940	4.220	0.616	0.892	3.274	0.429
Bp	0.943	4.290	0.618	0.896	3.340	0.431
FSf	0.945	4.343	0.622	0.898	3.363	0.431
FSp	0.945	4.349	0.629	0.898	3.362	0.429
BSf	0.945	4.338	0.618	0.898	3.362	0.435
BSp	0.945	4.340	0.615	0.898	3.357	0.424
Gf	0.940	4.238	0.627	0.892	3.285	0.424
Gp	0.943	4.305	0.638	0.897	3.355	0.439
BJ	0.931	3.877	0.456	0.894	3.254	0.383
M	0.946	4.802	0.668	0.897	3.789	0.479
Cao	0.938	4.198	0.650	0.888	3.245	0.452
APR/PRR	0.940	4.226	0.628	0.892	3.282	0.434

Table 7.2 Simulation Results of AR(1) with Laplace innovations and $\phi_1 = 0.5$

- The four studentized methods have similar performance to the respective unstudentized methods using predictive residuals. Thus, using predictive residuals is not deemed necessary for the studentized methods although it does not seem to hurt; see also Remark 7.3.4.
- The coverage levels of the Gf intervals resemble those of Ff and Bf intervals. Similarly, the coverages of Gp intervals resemble those of Fp and Bp intervals.

In comparison with the older methods discussed in Sect. 7.4, the following observations are in order.

- The BJ method has similar coverage rates as APR/PRR and our Ff method when the error is normal. However, when the errors have Laplace distribution, the BJ method performs poorly.
- Our forward and backward methods with fitted residuals (Ff and Bf) outperform both Cao and APR/PRR methods. This conclusion is expected and consistent with the discussion in Sect. 7.4.
- Our methods with predictive residuals (Fp and Bp) and the studentized methods (FSf, FSp, BSf, BSp) are the best performing in terms of coverage.
- Masarotto's (M) method has similar performance to our FSf method; this was somewhat expected in view of the discussion in Sect. 7.4. Further comparison of Masarotto's method to the FSf method is given in the following subsection.

7.5.2 Conditional Coverage Level

The CVRs reported in the previous subsection gave a measure of unconditional coverage level of the different prediction intervals. Obviously, conditional validity implies unconditional validity but the converse is not necessarily true. We now investigate the conditional coverage of a subset of the methods already discussed. To do this, 500 true data series are generated in a backwards fashion fixing the last datapoint y_n to a desired value. Surprisingly, Masarotto's intervals appear to also have accurate conditional coverages as the entries of Table 7.3 suggest. Note that Table 7.3 was based on an AR(1) model with Gaussian errors but similar findings using Laplace errors were also observed.

To explain this phenomenon, we focus on the causal AR(1) model $Y_t = \phi Y_{t-1} + \varepsilon_t$ with $|\phi| < 1$. Recall that the distribution of the bootstrap predictive root depends on the value $Y_n = y_n$ because

$$y^*_{n+1} - \hat{y}^*_{n+1} = (\hat{\phi} - \hat{\phi}^*)y_n + \varepsilon^*_{n+1}. \tag{7.22}$$

Since $\hat{\phi} - \hat{\phi}^* = O_p(1/\sqrt{n})$, it is apparent that the term $(\hat{\phi} - \hat{\phi}^*)y_n$ is small compared to the error term ε^*_{n+1}; this is why using the wrong y_n—as Masarotto's method does—can still yield accurate conditional coverages. The situation is similar for studentized bootstrap roots since the first term of the numerator contains a term including y_n. Nevertheless, there seems no reason to forego using the correct y_n in the bootstrap predictive root (7.22).

$n = 100$	nominal coverage 95%			nominal coverage 90%		
$Y_n = 3$	CVR	LEN	st.err	CVR	LEN	st.err
M	0.947	4.108	0.368	0.899	3.450	0.290
FSf	0.951	4.216	0.355	0.905	3.540	0.272
FSp	0.951	4.222	0.337	0.905	3.535	0.264
Ff	0.946	4.125	0.342	0.898	3.466	0.269
Fp	0.951	4.217	0.341	0.904	3.537	0.265
$Y_n = 2$	CVR	LEN	st.err	CVR	LEN	st.err
M	0.945	4.002	0.362	0.899	3.384	0.283
FSf	0.947	4.045	0.357	0.902	3.417	0.274
FSp	0.947	4.049	0.349	0.902	3.413	0.263
Ff	0.943	3.959	0.350	0.895	3.350	0.270
Fp	0.947	4.047	0.358	0.902	3.415	0.270
$Y_n = 1$	CVR	LEN	st.err	CVR	LEN	st.err
M	0.944	3.960	0.364	0.897	3.340	0.282
FSf	0.944	3.957	0.369	0.897	3.336	0.279
FSp	0.945	3.968	0.366	0.897	3.335	0.269
Ff	0.939	3.877	0.370	0.891	3.273	0.275
Fp	0.944	3.966	0.380	0.898	3.340	0.269
$Y_n = 0$	CVR	LEN	st.err	CVR	LEN	st.err
M	0.945	3.956	0.366	0.897	3.329	0.283
FSf	0.944	3.937	0.371	0.895	3.313	0.281
FSp	0.944	3.949	0.374	0.895	3.312	0.272
Ff	0.939	3.861	0.379	0.889	3.252	0.281
Fp	0.943	3.944	0.389	0.896	3.318	0.273

Table 7.3 Conditional coverage under the AR(1) model $Y_t = 0.5Y_{t-1} + \varepsilon_t$ with normal innovations

To elaborate, Masarotto replaces the term $(\hat{\phi} - \hat{\phi}^*)y_n$ in (7.22) with $(\hat{\phi} - \hat{\phi}^*)y_n^*$ where y_n^* is random (with mean zero). If y_n is near zero and y_n^* happens to be near its mean, then the terms match well. However, there is an issue of unnecessary variability here that is manifested with slightly higher standard errors of the lengths of Masarotto's intervals and with inflated CVRs—but the CVR inflation is due to a fluke, not a *bona fide* capturing of the predictor variability. Now if y_n is large (in absolute value), there is an issue of bias in the centering of the Masarotto intervals which is again masked by the unnecessary/excess variability of the term $(\hat{\phi} - \hat{\phi}^*)y_n^*$.

All in all, adjusting the last p values of the bootstrap series to match the original ones is highly advisable in a causal, linear AR(p) model. Furthermore, it may

achieve particular importance under a nonlinear and/or nonparametric model in which the above arguments break down. Adjusting the last p values certainly becomes crucial in autoregressions with heteroscedastic errors as in Eq. (1.2) of the main paper where the scale of the error also depends on these last p values.

7.6 Bootstrap Prediction Intervals for Nonparametric Autoregression

In the last several sections, the focus was on prediction intervals for linear autoregressions. In a *nonlinear* autoregression setting, backward bootstrap methods have not been found useful mainly because it is unclear how to generate a nonlinear model such as Eq. (7.1) backwards. By contrast, extension of the four forward bootstrap methods to nonlinear—but parametric—autoregressions is straightforward; see Pan and Politis (2015) for details. In what follows, we provide some details on how to employ the forward bootstrap in order to construct bootstrap prediction intervals under a nonparametric autoregression model fitted via kernel smoothing.

7.6.1 Nonparametric Autoregression with i.i.d Errors

In this subsection, we consider a stationary and geometrically ergodic process satisfying Eq. (7.1) with the conditional mean function $\mu(\cdot)$ being unknown but assumed smooth. Given a sample $Y_1 = y_1, \cdots, Y_n = y_n$, let $x_t = (y_t, y_{t-1}, \cdots, y_{t-p+1})'$ as before.

Algorithm 7.6.1 FORWARD BOOTSTRAP WITH FITTED RESIDUALS (FF)

1. *For $x \in \mathbf{R}^p$, construct the Nadaraya-Watson kernel estimator $\hat{m}(\cdot)$ as*

$$\hat{m}(x) = \frac{\sum_{t=p}^{n-1} K(\frac{\|x-x_t\|}{h}) y_{t+1}}{\sum_{t=p}^{n-1} K(\frac{\|x-x_t\|}{h})} \tag{7.23}$$

   where $\|\cdot\|$ is a norm on $\mathbf{R}^p$, and $K(\cdot)$ is compactly supported, symmetric density function with bounded derivative. As usual, the bandwidth satisfies $h \to 0$ but $hn \to \infty$.
2. *Compute the fitted residuals: $\hat{\varepsilon}_i = y_i - \hat{m}(x_{i-1})$ for $i = p+1, \cdots, n$*
3. *Center the residuals: $\hat{r}_i = \hat{\varepsilon}_i - (n-p)^{-1} \sum_{t=p+1}^{n} \hat{\varepsilon}_t$, for $i = p+1, \cdots, n$.*
   (a) *Sample randomly (with replacement) from the values $\hat{r}_{p+1}, \cdots, \hat{r}_n$ to create bootstrap pseudo errors ε_i^* for $i = -M+p, \cdots, n+1$ where M is some large positive number.*
   (b) *Let $x_p^* = (y_{p+I}, \cdots, y_{1+I})'$ where I is generated as a discrete random variable uniform on the values $0, 1, \ldots, n-p$, and define $(y_{-M}^*, y_{-M+1}^*, \cdots,$*

$y^*_{-M+p-1})' = x^*_p$. *Then, generate* y^*_i *by the recursion:*

$$y^*_i = \hat{m}(x^*_{i-1}) + \varepsilon^*_i \quad \text{for } i = p+1, \ldots, n$$

where $x^*_t = (y^*_t, \cdots, y^*_{t-p+1})'$.

(c) *Drop the first M "burn in" observations to make sure that the starting values have an insignificant effect. Then recompute the kernel estimator* $\hat{m}^*(\cdot)$ *from the bootstrap series* $\{y^*_1, \cdots, y^*_n\}$, *i.e., let*

$$\hat{m}^*(x) = \frac{\sum_{i=p}^{n-1} K(\frac{\|x - x^*_i\|}{h}) y^*_{i+1}}{\sum_{i=p}^{n-1} K(\frac{\|x - x^*_i\|}{h})} \tag{7.24}$$

where $x^*_t = (y^*_t, y^*_{t-1}, \cdots, y^*_{t-p+1})'$.

(d) *Now fix the last p pseudo values to be the true observations, i.e., re-define* $x^*_n = x_n$, *and then calculate the bootstrap predictor*

$$\hat{y}^*_{n+1} = \hat{m}^*(x^*_n) = \hat{m}^*(x_n)$$

and the future bootstrap observation

$$y^*_{n+1} = \hat{m}(x^*_n) + \varepsilon^*_{n+1} = \hat{m}(x_n) + \varepsilon^*_{n+1}.$$

(e) *Calculate the bootstrap predictive root replicate as* $y^*_{n+1} - \hat{y}^*_{n+1}$.

4. *Steps (a)–(e) in the above are repeated B times, and the B bootstrap predictive root replicates are collected in the form of an empirical distribution whose* α*-quantile is denoted* $q(\alpha)$.
5. *Then, a* $(1-\alpha)100\%$ *equal-tailed predictive interval for* Y_{n+1} *is given by*

$$[\hat{m}(x_n) + q(\alpha/2),\ \hat{m}(x_n) + q(1-\alpha/2)]. \tag{7.25}$$

Estimating $\mu(\cdot)$ in the above could in principle be done via different smoothing methods, e.g., local polynomials, splines, etc. We employ the Nadaraya-Watson kernel estimator $\hat{m}(\cdot)$ just for simplicity and concreteness. To define the predictive residuals, however, recall that the chosen estimator must be completely scatterplot-based.

Algorithm 7.6.2 FORWARD BOOTSTRAP WITH PREDICTIVE RESIDUALS (FP)

1. *Same as step 1 of Algorithm 7.6.1.*
2. *Use the delete-*y_t *dataset as described in Sect. 7.3.2 to compute the delete-one kernel estimator*

$$\hat{m}^{(t)}(x) = \frac{\sum_{i=p+1, i\neq t}^{n} K(\frac{\|x - x_{i-1}\|}{h}) y_i}{\sum_{i=p+1, i\neq t}^{n} K(\frac{\|x - x_{i-1}\|}{h})} \quad \text{for } t = p+1, \cdots, n. \tag{7.26}$$

Then calculate the predictive residuals:

$$\hat{\varepsilon}_t^{(t)} = y_t - \hat{m}_t^{(t)}(x_{t-1}) \quad \textit{for } t = p+1, \cdots, n. \tag{7.27}$$

3–5. Replace $\hat{\varepsilon}_t$ *by* $\hat{\varepsilon}_t^{(t)}$ *in Algorithm 7.6.1; the remaining steps are the same.*

Remark 7.6.1 (Scatterplot-based cross-validation) Cross-validation can be used in order to choose the bandwidth h in estimator $\hat{m}(x)$ of Eq. (7.23), i.e., choose h that minimizes either PRESS$= \sum_{t=p+1}^{n} [\hat{\varepsilon}_t^{(t)}]^2$ or PRESAR$= \sum_{t=p+1}^{n} |\hat{\varepsilon}_t^{(t)}|$. Note that the delete-one kernel estimator $\hat{m}^{(t)}(x)$ that gives rise to the predictive residuals $\hat{\varepsilon}_t^{(t)}$ of Eq. (7.27) are completely scatterplot-based, and so is the resulting cross-validation. To further elaborate, typical time series cross-validation procedures involve deleting a whole stretch of the time series dataset; see Chap. 9 for an example. By contrast, working with scatterplot-based estimators such as $\hat{m}(x)$ allows us to carry out a delete-one cross-validation as in regular regression; for example, to construct estimator (7.26) a *single* point was deleted from the scatterplot of Y_t vs. X_{t-1}.

The studentized versions of Algorithms 7.6.1 and 7.6.2 are defined analogously to the ones in Sect. 7.3.

Algorithm 7.6.3 FORWARD STUDENTIZED BOOTSTRAP WITH FITTED RESIDUALS (FSF) OR PREDICTIVE RESIDUALS (FSP)
For FSf, define $\hat{\sigma}$ *and* $\hat{\sigma}^*$ *to be the sample standard deviation of the fitted residuals* $\hat{\varepsilon}_t$ *and bootstrap residuals* $\hat{\varepsilon}_t^*$*, respectively. For FSp, define* $\hat{\sigma}$ *and* $\hat{\sigma}^*$ *to be the sample standard deviation of the predictive residuals* $\hat{\varepsilon}_t^{(t)}$ *and their bootstrap analogs* $\hat{\varepsilon}_t^{*(t)}$*, respectively.*
Then, replace steps 3(e) and 6 of Algorithms 7.3.1 and/or 7.6.2 by the following steps:

3.(e) Calculate a studentized bootstrap root replicate as $(y_{n+1}^* - \hat{y}_{n+1}^*)/\hat{\sigma}^*$.
6. Construct the $(1-\alpha)100\%$ *equal-tailed predictive interval for* Y_{n+1} *as*

$$[\hat{y}_{n+1} + \hat{\sigma}\, q(\alpha/2),\ \hat{y}_{n+1} + \hat{\sigma}\, q(1-\alpha/2)] \tag{7.28}$$

where $q(\alpha)$ *is the* α*-quantile of the empirical distribution of the B studentized bootstrap roots.*

Under regularity conditions that include the use of a nonnegative kernel $K(\cdot)$, Franke et al. (2002) showed the consistency of the residual bootstrap in approximating the distribution of the kernel estimator $\hat{m}(\cdot)$, i.e., that

$$\sup_{y \in \mathbf{R}} |P^*\{\sqrt{nh}(\hat{m}(x) - \hat{m}^*(x)) \le y\} - P\{\sqrt{nh}(\mu(x) - \hat{m}(x)) \le y\}| \xrightarrow{P} 0 \tag{7.29}$$

under the undersmoothing condition $h = o(n^{-1/5})$. As a result, all four forward bootstrap prediction intervals in Sect. 7.6.1 are asymptotically valid and pertinent.

Remark 7.6.2 Using a nonnegative kernel $K(\cdot)$, the condition $hn^{1/5} \to c > 0$ leads to optimal smoothing in that the large-sample MSE of $\hat{m}(x)$ is minimized. In this case, however, the bias of $\hat{m}(x)$ becomes of exact order $O(1/\sqrt{hn})$ which is the order of its standard deviation, and (7.29) fails because the bootstrap cannot capture the bias term exactly. This is of course important for confidence interval construction—for which (7.29) was originally developed—and is routinely solved via one of three approaches: (a) plugging-in explicit estimates of bias in the two distributions appearing in (7.29); (b) using a bandwidth satisfying $hn^{1/5} \to 0$ leading to *undersmoothing*, i.e., making the bias of $\hat{m}(x)$ negligible as compared to the standard deviation; or (c) using the optimal bandwidth $h \sim cn^{-1/5}$ with $c > 0$ but resampling based an *oversmoothed* estimator. Either of these approaches work—the simplest being undersmoothing—but note that the problem is not as crucial for prediction intervals that remain asymptotically valid in both cases $c > 0$ or $c = 0$; in the latter case, however, asymptotic pertinence is compromised. These issues become more important in the presence of heteroscedastic errors as the following subsection shows.

7.6.2 Nonparametric Autoregression with Heteroscedastic Errors

We now consider the nonparametric autoregression model (7.2) that is driven by heteroscedastic innovations. As in Sect. 7.6.1, we use Nadaraya-Watson estimators based on a nonnegative kernel $K(\cdot)$ in order to estimate the unknown (but assumed smooth) functions μ and σ. In particular, $\hat{m}(x)$ is exactly as given in (7.23) while $\hat{\sigma}^2(x)$ is defined as

$$\hat{\sigma}^2(x) = \frac{\sum_{t=p}^{n-1} K(\frac{\|x-x_t\|}{h})(y_{t+1} - \hat{m}(x_t))^2}{\sum_{t=p}^{n-1} K(\frac{\|x-x_t\|}{h})}. \tag{7.30}$$

Remark 7.6.3 As mentioned in Remark 7.6.2, in generating the bootstrap pseudo-series it may be advantageous to use oversmoothed estimators of μ and σ that will be denoted by $\hat{m}_g$ and $\hat{\sigma}_g$, respectively; these are computed in exactly the same way as $\hat{m}$ and $\hat{\sigma}$ but using an oversmoothed bandwidth g (instead of h) that satisfies

$$g/h \to \infty \text{ with } h \sim cn^{-1/5} \text{ for some } c > 0. \tag{7.31}$$

Such over-smoothing was originally proposed for bootstrap confidence intervals in nonparametric regression by Härdle and Bowman (1988), and Härdle and Marron (1991). It can also be useful in the nonparametric AR model (7.1) with i.i.d. innovations but it is particularly helpful in the heteroscedastic model (7.2).

Algorithm 7.6.4 FORWARD BOOTSTRAP WITH FITTED RESIDUALS (FF)

1. *Construct the estimates $\hat{m}(\cdot)$ and $\hat{\sigma}^2(\cdot)$ by formulas (7.23) and (7.30).*
2. *Recall the notation $x_t = (y_t, \cdots, y_{t-p+1})'$, and compute the residuals*

$$\hat{\varepsilon}_i = \frac{y_i - \hat{m}(x_{i-1})}{\hat{\sigma}(x_{i-1})} \text{ for } i = p+1, \cdots, n. \tag{7.32}$$

3. *Center the residuals, i.e., let* $\hat{r}_i = \hat{\varepsilon}_i - (n-p)^{-1}\sum_{t=p+1}^{n}\hat{\varepsilon}_t$ *for* $i = p+1, \cdots, n$.
   (a) *Sample randomly (with replacement) from the values* $r_{p+1}, \cdots, r_n$ *to create bootstrap pseudo errors* ε_i^* *for* $i = -M+p, \cdots, 1, 2, \cdots, n+1$ *where* M *is some large positive integer.*
   (b) *Let* $x_p^* = (y_{p+I}, \cdots, y_{1+I})'$ *where* I *is generated as a discrete random variable uniform on the values* $0, 1, \ldots, n-p$, *and define* $(y_{-M}^*, y_{-M+1}^*, \cdots, y_{-M+p-1}^*)' = x_p^*$. *Then, generate* y_i^* *by the recursion:*

$$y_i^* = \hat{m}_g(x_{i-1}^*) + \hat{\sigma}_g(x_{i-1}^*)\varepsilon_i^* \quad \text{for } i = -M+p, \cdots, n \tag{7.33}$$

   where $x_t^* = (y_t^*, \cdots, y_{t-p+1}^*)'$.
   (c) *Drop the first* M *"burn in" observations to make sure that the starting values have an insignificant effect, and construct the kernel estimator* $\hat{m}^*$ *from the bootstrap series* $\{y_1^*, \cdots, y_n^*\}$ *as in* (7.24).
   (d) *Now fix the last* p *pseudo values to be the true observations, i.e., re-define* $x_n^* = x_n$, *and then calculate the future bootstrap observation*

$$y_{n+1}^* = \hat{m}_g(x_n^*) + \hat{\sigma}_g(x_n^*)\varepsilon_{n+1}^* = \hat{m}_g(x_n) + \hat{\sigma}_g(x_n)\varepsilon_{n+1}^*$$

   and the bootstrap predictor $\hat{y}_{n+1}^* = \hat{m}^*(x_n^*) = \hat{m}^*(x_n)$; *recall that* $\hat{m}^*$ *uses bandwidth* h *as the original estimator* $\hat{m}$.
   (e) *Calculate the bootstrap predictive root replicate as* $y_{n+1}^* - \hat{y}_{n+1}^*$.
4. *Steps (a)–(d) in the above are repeated* B *times, and the* B *bootstrap root replicates are collected in the form of an empirical distribution whose* α*-quantile is denoted* $q(\alpha)$.
5. *Then, a* $(1-\alpha)100\%$ *equal-tailed predictive interval for* Y_{n+1} *is given by*

$$[\hat{m}(x_n) + q(\alpha/2),\ \hat{m}(x_n) + q(1-\alpha/2)]. \tag{7.34}$$

Algorithm 7.6.5 FORWARD BOOTSTRAP WITH PREDICTIVE RESIDUALS (FP)

1. *Same as step 1 of Algorithm 7.6.4.*
2. *Use the delete-*y_t *dataset to compute the delete-one kernel estimators* $\hat{m}^{(t)}$ *by Eq.* (7.26) *and* $\hat{\sigma}^{(t)}$ *by*

$$\hat{\sigma}^{(t)}(x) = \frac{\sum_{i=p+1, i\neq t}^{n} K(\frac{\|x - x_{i-1}\|}{h})(y_i - \hat{m}^{(t)}(x_{i-1}))^2}{\sum_{i=p+1, i\neq t}^{n} K(\frac{\|x - x_{i-1}\|}{h})}. \tag{7.35}$$

   Then, calculate the predictive residuals:

$$\hat{\varepsilon}_t^{(t)} = \frac{y_t - \hat{m}^{(t)}(x_{t-1})}{\hat{\sigma}^{(t)}(x_{t-1})} \text{ for } t = p+1, \cdots, n. \tag{7.36}$$

3–5. Replace $\hat{\varepsilon}_t$ *by* $\hat{\varepsilon}_t^{(t)}$ *in Algorithm 7.6.4; the remaining steps are the same.*

Algorithms 7.6.4 and 7.6.5 can be extended to include studentized roots as in Algorithm 7.6.3, yielding prediction intervals based on Forward Studentized bootstrap with fitted residuals (FSf) or predictive residuals (FSp). As in the case of nonparametric AR model with i.i.d. errors, Franke et al. (2002) showed the consistency of the residual bootstrap in approximating the distributions of both kernel estimators $\hat{m}(\cdot)$ and $\hat{\sigma}^2(\cdot)$ under model (7.2). As a result, the four forward bootstrap prediction intervals of Sect. 7.6.2 are asymptotically valid.

Monte Carlo simulations assessing the finite-sample performance of prediction intervals based on a Forward bootstrap using the nonparametric AR models (7.1) and/or (7.2) are presented in Chap. 8 in comparison with relevant Model-free methods.

Acknowledgements

Chapter 7 is based on the paper: Pan, L. and Politis, D.N. (2015), "Bootstrap prediction intervals for linear, nonlinear and nonparametric autoregressions" (with Discussion), to appear in the *Journal of Statistical Planning and Inference*. Many thanks are due to the Editors, Anirban DasGupta and Wei-Liem Loh, for hosting this discussion paper, and to the discussants: Silvia Gonçalves, Jae Kim, Jens-Peter Kreiss, Soumen Lahiri, Dan Nordman, and Benoit Perron for their insightful comments.

Chapter 8
Model-Free Inference for Markov Processes

8.1 Introduction

Chapter 7 presented a unified approach towards prediction intervals when a time series $\{Y_t\}$ obeys an autoregressive model that is either linear/parametric or nonparametric; the aforementioned are some popular models for Markov processes. We will now expand our scope by pursuing *model-free* inference under the sole assumption that $\{Y_t\}$ is a Markov process of order p; some smoothness conditions may also be needed. Thus, as Chap. 4 studied model-free regression in contrast to the model-based regression of Chap. 3, the present chapter will investigate model-free autoregression, i.e., Markov processes, to serve as contrast to the model-based autoregression of Chap. 7.

Bootstrap methods for time series have been the subject of active investigation for the last 25 years; recent review by Kreiss and Paparoditis (2011) gave a recent review of the state-of-the-art of the literature. In particular, when the data at hand are a sample from a Markov process, several different resampling schemes have been proposed; see, e.g., Bertail and Clémencon (2006) and the references therein. With respect to constructing bootstrap confidence intervals in the setting of Markov processes, two well-known methods exist that are based on employing the conditional density and/or distribution directly. These are:

1. The bootstrap method based on kernel estimates of the transition density of the Markov processes as proposed by Rajarshi (1990); see Sect. 8.3.
2. The Local Bootstrap for Markov processes of Paparoditis and Politis (2001, 2002a); see Sect. 8.4.

Recall that two different general approaches towards building bootstrap prediction intervals with *conditional validity*—namely the Forward and Backward recursive schemes—were discussed in Chap. 7. We will address both Forward and Backward approaches towards prediction intervals using either Rajarshi's method

D.N. Politis, *Model-Free Prediction and Regression*, Frontiers in Probability and the Statistical Sciences, DOI 10.1007/978-3-319-21347-7_8

or the Local Bootstrap. Interestingly, the Local Bootstrap (LB) has already been used for the construction of prediction intervals via the Backward approach; see Paparoditis and Politis (1998).

In addition, we will introduce a third resampling option (both for prediction as well as confidence intervals) that stems from the Model-Free Prediction Principle:

3. The **Model-Free Bootstrap for Markov Processes** which is a novel resampling scheme proposed by Pan and Politis (2014); see Sect. 8.6.

As mentioned in Sect. 7.6, the Backward approach is not readily available for time series that satisfy a nonlinear and/or nonparametric autoregression. Recall that, under causality, AR models are special cases of Markov processes. Hence, in Sect. 8.5 we propose a *hybrid* approach for nonparametric autoregressions in which the forward step uses the autoregressive equation explicitly while the backward step uses one of the three aforementioned Markov bootstrap procedures.

Finally, in Sect. 8.8 we will explore the possibility of constructing confidence intervals using the Model-Free Bootstrap, and contrast them to confidence intervals obtained by Rajarshi's methods or the Local Bootstrap. All the above methods are developed in the case the random variables Y_t are continuous; the case of discrete-valued time series, e.g., finite-state Markov chains, will be dealt with in Sect. 8.9.

8.2 Prediction and Bootstrap for Markov Processes

8.2.1 Notation and Definitions

Here, and throughout the rest of Chap. 8, we assume that $Y = \{Y_t \text{ for } t \in \mathbf{Z}\}$ is a real-valued, *strictly stationary* process that is Markov of order p. Letting $X_t = (Y_t, Y_{t-1}, \cdots, Y_{t-p+1})'$, we denote[1]

$$\begin{aligned} \mathbf{F}(x) &= P[X_p \le x], \\ \mathbf{F}(y,x) &= P[Y_{p+1} \le y, X_p \le x], \\ \mathbf{F}(y|x) &= P[Y_{p+1} \le y | X_p = x], \end{aligned} \tag{8.1}$$

for $x \in \mathbf{R}^p$, $y \in \mathbf{R}$; in the above, we have used the short-hand $\{X_p \le x\}$ to denote the event: {the i-th coordinate of X_p is less or equal to the i-th coordinate of x for all $i = 1, \ldots, p$}.

Let $f(x), f(y,x), f(y|x)$ be the corresponding densities of the distributions in Eq. (8.1). We will assume throughout the chapter that these densities are with respect to Lebesgue measure. However, all our model-free methods from Sects. 8.3, 8.4, and 8.6, i.e., bootstrap based on estimated transition densities, Local Bootstrap, and

[1] The distribution of random vector X_p was denoted $\mathbf{F}$ in order to be distinguished from the limiting distribution F of the i.i.d. variables $\varepsilon_1^{(m)}, \ldots, \varepsilon_m^{(m)}$ in the Model-Free Prediction Principle.

the Model-free bootstrap, could be generalized to the case of densities taken with respect to counting measure, i.e., the case of discrete random variables; Sect. 8.9 gives the details.

Let $Y_1 = y_1, Y_2 = y_2, \cdots, Y_n = y_n$ denote the observed sample path from the Markov chain Y, and let $x_t = (y_t, \cdots, y_{t-p+1})'$. Denote by $\hat{Y}_{n+1}$ the chosen point predictor of Y_{n+1} based on the data at hand. Because of the Markov structure, this predictor will be a functional of $\hat{f}_n(\cdot|x_n)$ which is our data-based estimator of the conditional density $f(\cdot|x_n)$. For example, the L_2–optimal predictor would be given by the mean of $\hat{f}_n(\cdot|x_n)$; similarly, the L_1–optimal predictor would be given by the median of $\hat{f}_n(\cdot|x_n)$. To fix ideas in what follows we will focus on the L_2–optimal predictor, usually approximated by $\hat{Y}_{n+1} = \int y\hat{f}_n(y|x_n)dy$, with the understanding that other functionals of $\hat{f}_n(\cdot|x_n)$ can be accommodated equally well.

Remark 8.2.1 An integral such as $\int y\hat{f}_n(y|x_n)dy$ can be calculated by numerical integration, e.g., using the *adaptive quadrature* method. However, the L_2–optimal predictor can be approximated in several different ways that are asymptotically equivalent. The most straightforward alternative is a kernel smoothed estimator of the autoregression scatterplot, i.e., estimator (7.23), that has been discussed in the previous chapter. Claim 8.6.2 in what follows lists some further alternative options.

Beyond the point predictor $\hat{Y}_{n+1}$, we want to construct a prediction interval that will contain Y_{n+1} with (conditional) probability $1-\alpha$ asymptotically. Of course, asymptotic validity is a fundamental property but it does not tell the whole story. For example, one could construct an interval having as left and right end-points the $\alpha/2$ and $1-\alpha/2$ quantiles of the conditional density estimator $\hat{f}_n(\cdot|x_n)$, respectively. If $\hat{f}_n(\cdot|x_n)$ is consistent for $f_n(\cdot|x_n)$, then this interval would be asymptotically valid. Nevertheless, it would be characterized by pronounced *undercoverage* in finite samples since the nontrivial variability in the estimator $\hat{f}_n(\cdot|x_n)$ is ignored.

In order to capture the finite-sample variability involved in model estimation some kind of bootstrap algorithm is necessary. Thus, consider a bootstrap pseudo series $Y_1^*, \cdots, Y_n^*$ constructed according to one of the methods mentioned in the Introduction. Let $\hat{f}_n^*(\cdot|x_n)$ be the corresponding estimator of $f(\cdot|x_n)$ as obtained from the bootstrap data $Y_1^*, \cdots, Y_n^*$. To achieve conditional validity, we will ensure that the last p–values in the bootstrap world coincide with the last p–values in the real world, i.e., that $(Y_n^*, \cdots, Y_{n-p+1}^*)' = x_n$. Finally, we construct the predictor $\hat{Y}_{n+1}^*$ using the *same* functional, i.e., mean, median, etc., as used in the construction of $\hat{Y}_{n+1}$ in the real world but, of course, this time the functional is applied to $\hat{f}_n^*(\cdot|x_n)$. For example, the L_2–optimal predictor in the bootstrap world will be given by $\hat{Y}_{n+1}^* = \int y\hat{f}_n^*(y|x_n)dy$.

Bootstrap probabilities and expectations are usually denoted by P^* and E^*, and they are understood to be conditional on the original data $Y_1 = y_1, \cdots, Y_n = y_n$. Since our goal is conditional asymptotic validity, we will understand that P^* and E^* are also conditional on $Y_{n-p+1}^* = y_{n-p+1}, \cdots, Y_n^* = y_n$ when they are applied to

"future" events in the bootstrap world, i.e., events determined by $\{Y_s^* \text{ for } s > n\}$; this is not restrictive since we will ensure that our bootstrap algorithms satisfy this requirement.

Indeed, all prediction intervals that will be studied in this chapter are asymptotically valid under appropriate conditions. However, as mentioned earlier, it is difficult to quantify asymptotically the extent to which a prediction interval is able to capture both sources of variation, i.e., the variance associated with the new observation Y_{n+1} and the variability in estimating $\hat{Y}_{n+1}$; hence, the prediction intervals in this paper will be compared via finite-sample simulations.

8.2.2 Forward vs. Backward Bootstrap for Prediction Intervals

Consider the bootstrap sample $Y_1^*, \cdots, Y_n^*$. As mentioned in Sect. 8.2.1, in order to ensure conditional validity it would be helpful if the last p–values in the bootstrap world coincided with the last p–values in the real world, i.e., that $(Y_n^*, \cdots, Y_{n-p+1}^*)' = x_n \equiv (y_n, \cdots, y_{n-p+1})'$. For the application to prediction intervals, note that the bootstrap also allows us to generate Y_{n+1}^* so that the statistical accuracy of the predictor $\hat{Y}_{n+1}$ can be gauged. However, under a usual Monte Carlo simulation, none of the simulated bootstrap series will have their last p values exactly equal to the original data sub-vector x_n as needed for prediction purposes. Herein lies the problem, since the behavior of the predictor $\hat{Y}_{n+1}$ needs to be captured *conditionally* on the original vector x_n.

As discussed in Chap. 7, one possibility is to generate the bootstrap data $Y_1^*, \cdots, Y_n^*$ going *backwards* from the last p values that are fixed at $(Y_n^*, \cdots, Y_{n-p+1}^*)' = x_n$; this is the **backward bootstrap** method originally proposed by Thombs and Schucany (1990), and by Breidt et al. (1995) in the context of a linear AR(p) model. Note that the generation of Y_{n+1}^* must still be done in a forward fashion using the fitted AR model conditionally on the value Y_n. Going beyond the linear AR(p) model, a backward bootstrap for Markov processes was proposed by Paparoditis and Politis (1998) via the aforementioned Local Bootstrap. We will elaborate on the backward Local Bootstrap and other backward bootstrap methods for Markov processes in the sequel. A key result here is the following; see Pan and Politis (2014) for a proof.

Fact 8.2.1 *A stationary Markov process remains a stationary Markov process after a time-reversal.*

Nevertheless, the natural way Markov processes evolve is *forward* in time, i.e., one generates Y_t given $Y_{t-1}, Y_{t-2}, \ldots, Y_{t-p}$. Thus, it is intuitive to construct bootstrap procedures that run forward in time, i.e., to generate Y_t^* given $Y_{t-1}^*, Y_{t-2}^*, \ldots, Y_{t-p}^*$. Indeed, most (if not all) of the literature on bootstrap confidence intervals for linear AR models uses the natural time order to generate bootstrap series. However, recall that predictive inference is to be conducted conditionally on the last p values given by y_n in order to be able to place prediction bounds around the point predictor $\hat{Y}_{n+1}$.

In order to maintain the natural time order, i.e., generate bootstrap series forward in time, but also ensure that Y^*_{n+1} is constructed correctly, i.e., conditionally on the original x_n, Pan and Politis (2014, 2015) introduced the **forward bootstrap** method for prediction intervals that was defined in a model-based setting in Sect. 7.2.1. In describing it under a model-free context, we will use the notion of fitting a Markov model by estimating the transition density $f(y|x)$ as will be discussed in Sect. 8.3; different notions of Markov bootstrap, e.g., the Local Bootstrap, work analogously. The forward bootstrap for Markov processes comprises of the following three steps:

A. Choose a starting vector $(Y^*_{1-p}, Y^*_{2-p}, \ldots, Y^*_0)'$ in a way that is compatible to the stationary distribution of Y, e.g., choose it at random as one of the stretches (subseries) of length p found in the original data $Y_1, \cdots, Y_n$. Then, use the fitted Markov model, i.e., use the estimated transition density $\hat{f}_n(y|x)$, in order to generate bootstrap data Y^*_t recursively for $t = 1, \ldots, n$. Now re-fit the Markov model using the bootstrap data $Y^*_1, \cdots, Y^*_n$, i.e., obtain $\hat{f}^*_n(y|x)$ as an estimate of the transition density.
B. Re-define the last p values in the bootstrap world, i.e., let $(Y^*_n, \cdots, Y^*_{n-p+1})' = x_n$, and generate the future bootstrap observation Y^*_{n+1} by a random draw from density $\hat{f}^*_n(\cdot|x_n)$. Also construct the predictor $\hat{Y}^*_{n+1}$ using the *same* functional, i.e., mean, median, etc., as used in the construction of $\hat{Y}_{n+1}$ in the real world but this time the functional is applied to $\hat{f}^*_n(\cdot|x_n)$. For example, the L_2–optimal predictor in the bootstrap world will be given by $\hat{Y}^*_{n+1} = \int y \hat{f}^*_n(y|x_n)dy$.
C. Use the simulated distribution of the bootstrap predictive root $Y^*_{n+1} - \hat{Y}^*_{n+1}$ to estimate the true distribution of the real-world predictive root $Y_{n+1} - \hat{Y}_{n+1}$; it is also possible to use studentized predictive roots in this connection.

Pan and Politis (2015) found that the forward bootstrap is the method that can be immediately generalized to apply for nonlinear and/or nonparametric autoregressions as well, thus forming a unifying principle for treating all AR models. As already mentioned, for nonlinear/nonparametric autoregressions the backward bootstrap seems infeasible. Nevertheless, as will be shown in the next two sections, the backward bootstrap becomes feasible again under the more general setup of Markov process data. In Sect. 8.5 we will return briefly to the setup of a nonparametric autoregression and propose a hybrid approach in which the forward step uses the autoregressive equation explicitly while the backward step uses one of the aforementioned Markov bootstrap procedures.

8.3 Bootstrap Based on Estimates of Transition Density

Rajarshi (1990) introduced a bootstrap method that creates pseudo-sample paths of a Markov process based on an estimated transition density; this method can form the basis for a forward bootstrap procedure for prediction intervals. In what follows, the phrase "generate $z \sim f(\cdot)$" will be used as short-hand for "generate z by a random

draw from probability density $f(\cdot)$." For simplicity, we will focus on the L_2–optimal predictor as being the point predictor of choice but other predictor choices can be accommodated in the same manner.

Algorithm 8.3.1 FORWARD BOOTSTRAP BASED ON TRANSITION DENSITY

1. *Choose a probability density K on $\mathbf{R}^2$ and positive bandwidths h_1, h_2 to construct the following kernel estimators:*

$$\hat{f}_n(y,x) = \frac{1}{(n-p)h_1h_2} \sum_{i=p+1}^{n} K\left(\frac{y-y_i}{h_1}, \frac{\|x-x_{i-1}\|}{h_2}\right) \tag{8.2}$$

$$\hat{f}_n(x) = \int \hat{f}_n(y,x)dy \tag{8.3}$$

$$\hat{f}_n(y|x) = \frac{\hat{f}_n(y,x)}{\hat{f}_n(x)} \tag{8.4}$$

for all $y \in \mathbf{R}$, $x \in \mathbf{R}^p$, and where $\|\cdot\|$ is a norm on $\mathbf{R}^p$.

2. *Calculate the point predictor $\hat{y}_{n+1} = \int y\hat{f}_n(y|x_n)dy$.*

3.(a) *Generate $x_p^* = (y_p^*, \cdots, y_1^*)$ with probability density function $\hat{f}_n(\cdot)$ given by Eq. (8.3). Alternatively, let $x_p^* = (y_{p+I}, \cdots, y_{1+I})'$ where I is generated as a discrete random variable uniform on the values $0, 1, \ldots, n-p$, i.e., choose x_p^* as one of the subseries of p consecutive data points found in the original data series $y_1, \ldots, y_n$.*

(b) *Generate $y_{p+1}^* \sim \hat{f}_n(\cdot|x_p^*)$ given by (8.4).*

(c) *Repeat (b) to generate $y_{t+1}^* \sim \hat{f}_n(\cdot|x_t^*)$ for $t = p, \cdots, n-1$, where as before $x_t^* = (y_t^*, \cdots, y_{t-p+1}^*)'$.*

(d) *Construct $\hat{f}_n^*(y|x)$ in a similar way as in (8.4)—with the same kernel and bandwidths—but based on the pseudo-data $y_1^*, y_2^*, \cdots, y_n^*$ instead of the original data.*

(e) *Calculate the bootstrap point predictor $\hat{y}_{n+1}^* = \int y\hat{f}_n^*(y|x_n)dy$.*

(f) *Generate the bootstrap future value $y_{n+1}^* \sim \hat{f}_n(\cdot|x_n)$.*

(g) *Calculate the bootstrap root replicate as $y_{n+1}^* - \hat{y}_{n+1}^*$*

4. *Repeat step 3 above B times; the B bootstrap root replicates are collected in the form of an empirical distribution whose α-quantile is denoted $q(\alpha)$.*

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

$$[\hat{y}_{n+1} + q(\alpha/2),\ \hat{y}_{n+1} + q(1-\alpha/2)]. \tag{8.5}$$

Rajarshi (1990) showed the uniform consistency of the density estimators appearing in Algorithm 8.3.1, i.e., he showed

$$\sup_{x,y} |\hat{f}_n(y,x) - f(y,x)| \to 0 \text{ a.s.} \tag{8.6}$$

$$\sup_{y} |\hat{f}_n(y) - f(y)| \to 0 \text{ a.s.} \tag{8.7}$$

$$\sup_{x,y} |\hat{f}_n(y|x) - f(y|x)| \to 0 \text{ a.s.} \tag{8.8}$$

under regularity assumptions that include the following:

(β_1) $\{Y_t\}$ is an aperiodic, strictly stationary, geometrically ergodic and ϕ–mixing Markov chain;

(β_2) The densities $f(y)$, $f(y,x)$, and $f(y|x)$ are uniformly continuous and bounded, and $f(y)$ has compact support; and

(β_3) As $n \to \infty$, we have $h = h(n) \to 0$, $nh \to \infty$, and $\sum_{m=1}^{\infty} m^{k+1} h(m)^{4(k+1)} < \infty$ for some $k \geq 3$.

Equations (8.6)—(8.8) are enough to show that the prediction interval (8.5) is asymptotically valid; the same is true for the prediction interval constructed from the backward bootstrap of Algorithm 8.3.2 to be described next.

Let us define the backwards transition distribution as $\mathbf{F}_b(y|x) = P[Y_0 \leq y | X_p = x]$ with corresponding density $f_b(y|x)$. Similarly, we define the backwards joint distribution as $\mathbf{F}_b(y,x) = P[Y_0 \leq y, X_p \leq x]$ with corresponding density $f_b(y,x)$. Having observed the sample path $y_1, y_2, \cdots, y_n$ of our Markov chain Y, Fact 8.2.1 implies that the time-reversed sample path $y_n, y_{n-1}, \cdots, y_1$ can be considered as a sample path of another Markov chain with transition distribution and density given by $\mathbf{F}_b(y|x)$ and $f_b(y|x)$, respectively.

Note that the densities $f_b(y,x)$ and $f_b(y|x)$ admit kernel estimators as follows:

$$\hat{f}_{bn}(y,x) = \frac{1}{(n-p)h_1 h_2} \sum_{i=p+1}^{n} K\left(\frac{y - y_{i-p}}{h_1}, \frac{\|x - x_i\|}{h_2}\right) \tag{8.9}$$

$$\hat{f}_{bn}(y|x) = \frac{\hat{f}_{bn}(y,x)}{\hat{f}_{bn}(x)}. \tag{8.10}$$

Note that the above can be used to form an alternative estimator of the unconditional density $f(x)$, i.e.,

$$\hat{f}_{bn}(x) = \int \hat{f}_{bn}(y,x) dy;$$

we will not delve into the difference between $\hat{f}_n(x)$ and $\hat{f}_{bn}(x)$ as it is not important in what follows.

The algorithm for backward bootstrap based on transition density is very similar to that of the corresponding forward bootstrap. The only difference is in Step 3 where we generate the pseudo series $y_1^*, \cdots, y_n^*$ in a time-reversed fashion. The backward bootstrap algorithm is described below where the notation $x_t^* = (y_t^*, \cdots, y_{t-p+1}^*)'$ is again used.

Algorithm 8.3.2 BACKWARD BOOTSTRAP BASED ON TRANSITION DENSITY

1–2. Same as the corresponding steps in Algorithm 8.3.1.
3.(a) Let $x_n^* = x_n$.
(b) Generate $y_{n-p}^* \sim \hat{f}_{bn}(\cdot|x_n^* = x_n)$
(c) Repeat (b) going backwards in time to generate $y_t^* \sim \hat{f}_{bn}(\cdot|x_{t+p}^*)$ *for* $t = n-p, n-p-1, \cdots, 1$.
(d) Generate bootstrap future value $y_{n+1}^* \sim \hat{f}_n(\cdot|x_n)$. [Note: this is again going forward in time, using the forward transition density exactly as in the Forward Bootstrap Algorithm 8.3.1.]
(e) Construct $\hat{f}_n^*(y|x)$ *in a similar way as in (8.4)—with the same kernel and bandwidths—but based on the pseudo-data* $y_1^*, y_2^*, \cdots, y_n^*$ *instead of the original data.*
(f) Calculate the bootstrap root replicate as $y_{n+1}^* - \hat{y}_{n+1}^*$.
4– 5. Same as the corresponding steps in Algorithm 8.3.1.

Remark 8.3.1 (On bandwidth choice) Bandwidth choice is as difficult as it is important in practice. Rajarshi (1990) used the bandwidth choice $h = 0.9An^{-1/6}$ where $A = \min(\hat{\sigma}, \frac{IQR}{1.34})$, $\hat{\sigma}$ is the estimated standard deviation of the data, and IQR is the interquartile range. However, our simulations indicated that such a bandwidth choice typically gives prediction intervals that exhibit *overcoverage*. Note that the last requirement of assumption (β_3) implies $nh^4 \to 0$; this convergence is allowed to be very slow (when k is large) but in any case, h should be at most $O(n^{-1/4})$. Therefore, we modified the practical bandwidth choice recommendation to $h = 0.9An^{-1/4}$. Cross-validation is not recommended here as it results into an h of order $n^{-1/5}$.

8.4 The Local Bootstrap for Markov Processes

Paparoditis and Politis (2001, 2002a) proposed the Local Bootstrap for Markov processes that, in essense, generates bootstrap sample paths based on a transition distribution that is a step function as opposed to generating bootstrap sample paths based on an estimated transition density as in Sect. 8.3. In that sense, Rajarshi's (1990) method is to the Local Boostrap what the smoothed bootstrap for i.i.d. data is to Efron's (1979) original bootstrap that resamples from the empirical distribution function.

As before, let $Y_1 = y_1, \cdots, Y_n = y_n$ be the observed sample path, and let $X_t = (Y_t, Y_{t-1}, \cdots, Y_{t-p+1})'$ and $x_t = (y_t, y_{t-1}, \cdots, y_{t-p+1})'$. The aforementioned step-function estimator of the transition distribution function is given by the weighted empirical distribution

$$\tilde{\mathbf{F}}_n(y|x) = \frac{\sum_{j=p}^{n-1} \mathbf{1}_{(-\infty,y]}(y_{j+1}) W_g(x - x_j)}{\sum_{m=p}^{n-1} W_g(x - x_m)} \tag{8.11}$$

where $W_g(\cdot) = (1/g)W(\cdot/g)$ with $W(\cdot)$ being a bounded, continuous, and symmetric probability density on $\mathbf{R}^p$, and $g > 0$ is a bandwidth parameter tending to zero.

The Local Boostrap generation of pseudo-data is based on the estimated conditional distribution $\tilde{\mathbf{F}}_n(y|x)$. However, since the latter is a step function, i.e., it is the distribution of a discrete random variable, it is easier to work with the probability mass function associated with this discrete random variable.

Algorithm 8.4.1 FORWARD LOCAL BOOTSTRAP

1. *Choose a resampling kernel W and bandwidth g; here g can be selected by cross-validation. Then calculate the predictor* $\hat{y}_{n+1}$ *as*

$$\frac{\sum_{j=p}^{n-1} W_g(x_n - x_j) y_{j+1}}{\sum_{m=p}^{n-1} W_g(x_n - x_m)}.$$

2.(a) *Let* $x_p^* = (y_{p+I}, \cdots, y_{1+I})'$ *where I is generated as a discrete random variable uniform on the values* $0, 1, \ldots, n-p$.

(b) *Suppose* $y_1^*, \cdots, y_t^*$ *for* $t \geq p$ *have already been generated. Let J be a discrete random variable taking its values in the set* $\{p, \cdots, n-1\}$ *with probability mass function given by*

$$P(J = s) = \frac{W_g(x_t^* - x_s)}{\sum_{m=p}^{n-1} W_g(x_t^* - x_m)}.$$

Then, let $y_{t+1}^* = y_{J+1}$ *for* $t = p$.

(c) *Repeat (b) for* $t = p+1, p+2, \ldots$ *to generate* $y_{p+1}^*, \cdots, y_n^*$.

(d) *Calculate the bootstrap predictor* $\hat{y}_{n+1}^*$ *as*

$$\frac{\sum_{j=p}^{n-1} W_g(x_n - x_j^*) y_{j+1}^*}{\sum_{m=p}^{n-1} W_g(x_n - x_m^*)},$$

where $x_t^* = (y_t^*, \cdots, y_{t-p+1}^*)'$.

(e) *Re-define* $x_n^* = x_n$, *and then generate* $y_{n+1}^* = y_{J+1}$ *as in step (b), where J is a discrete random variable taking its values in the set* $\{p, \cdots, n-1\}$ *with probability mass function given by*

$$P(J = s) = \frac{W_g(x_n - x_s)}{\sum_{m=p}^{n-1} W_g(x_n - x_m)}.$$

(f) *Calculate the bootstrap prediction root replicate as* $y_{n+1}^* - \hat{y}_{n+1}^*$.

3. *Repeat step 2 above B times; the B bootstrap root replicates are collected in the form of an empirical distribution whose* α*-quantile is denoted* $q(\alpha)$.

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

$$[\hat{y}_{n+1} + q(\alpha/2),\ \hat{y}_{n+1} + q(1-\alpha/2)].$$

Recall Fact 8.2.1, i.e., that the time-reverse of a Markov process is also Markov; this is the necessary premise behind the Backward Local Bootstrap introduced by Paparoditis and Politis (1998) whose motivating example was a first order autoregressive process with conditionally heteroscedastic errors, i.e., the model $Y_t = \phi Y_{t-1} + \varepsilon_t \sqrt{\alpha_0 + \alpha_1 Y_{t-1}^2}$ with $\{\varepsilon_t\}$ i.i.d. (0,1). We will now show how this idea applies generally to the Markov(p) case. The Backward Local Bootstrap employs an estimate of the backward conditional distribution given by

$$\tilde{\mathbf{F}}_{bn}(y|x) = \frac{\sum_{j=1}^{n-p} \mathbf{1}_{(-\infty,y]}(y_j) W_b(x - x_{j+p})}{\sum_{m=1}^{n-p} W_b(x - x_{m+p})}. \tag{8.12}$$

Remark 8.4.1 In practice, both the forward bandwidth g as well as the backward bandwidth b can be chosen by delete-one cross-validation on the scatterplot of Y_t vs. X_{t-1} as discussed in Remark 7.6.1.

Algorithm 8.4.2 BACKWARD LOCAL BOOTSTRAP

1. Same as in the Forward Local Bootstrap of Algorithm 8.4.1.

2.(a) Set starting value $x_n^* = x_n$.

(b) Suppose x_{t+p}^* *has already been generated where* $1 \le t \le n-p$. *Let* J *be a discrete random variable taking its values in the set* $\{1, 2, \cdots, n-p\}$ *with probability mass function given by*

$$P(J = s) = \frac{W_b(x_{t+p}^* - x_{s+p})}{\sum_{m=1}^{n-p} W_b(x_{t+p}^* - x_{m+p})}.$$

Then let $y_t^* = y_J$.

(c) Repeat (b) to generate $y_{n-p}^*, \cdots, y_2^*, y_1^*$ *backwards in time, i.e., first generate* y_{n-p}^*, *then generate* y_{n-p-1}^*, *etc.*

(d) Let J *be a discrete random variable taking its values in the set* $\{p, p+1, \cdots, n-1\}$ *with probability mass function given by*

$$P(J = s) = \frac{W_b(x_n - x_s)}{\sum_{m=p}^{n-1} W_b(x_n - x_m)}.$$

Then, let $y_{n+1}^* = y_{J+1}$. [Note: this is again going forward in time exactly as in the Forward Local Bootstrap Algorithm 8.4.1.]

(e) Calculate the bootstrap predictor $\hat{y}_{n+1}^*$ *by*

$$\frac{\sum_{j=p}^{n-1} W_b(x_n - x_j^*) y_{j+1}^*}{\sum_{m=p}^{n-1} W_b(x_n - x_m^*)}.$$

(f) Calculate the bootstrap prediction root replicate as $y_{n+1}^* - \hat{y}_{n+1}^*$.

3–4. Same as the corresponding steps of the Forward Local Bootstrap of Algorithm 8.4.1.

Under regularity conditions, Paparoditis and Politis (2002a) proved that

$$\sup_{x,y} |\tilde{\mathbf{F}}_n(y|x) - \mathbf{F}(y|x)| \to 0 \text{ a.s.} \tag{8.13}$$

where the transition distribution estimator $\tilde{\mathbf{F}}(y|x)$ was defined in (8.11). This is sufficient to show that the prediction interval constructed from the Forward Local Bootstrap of Algorithm 8.4.1 is asymptotically valid; the same is true for the prediction interval constructed from the Backward Local Bootstrap of Algorithm 8.4.2.

8.5 Hybrid Backward Markov Bootstrap for Nonparametric Autoregression

In this section only, we will revert to the special case where our Markov (p) process is generated via a nonparametric autoregression, i.e., either model (7.1) or (7.2) from Chap. 7. As before, we assume that $\{Y_t\}$ is strictly stationary; we further need to assume *causality*, i.e., that ε_t is independent of $\{Y_{t-1}, Y_{t-2}, \ldots\}$ for all t. As usual, the recursions (7.1) and (7.2) are meant to run forward in time, i.e., Y_{p+1} is generated given an initial assignment for $Y_1, \ldots, Y_p$; then, Y_{p+2} is generated given its own p-past, etc.

Using the ideas presented in Sects. 8.3 and 8.4, we can now propose a *hybrid* Backward Markov Bootstrap for nonparametric autoregression models in which forward resampling is done using the model, i.e., Eq. (7.1) or (7.2), whereas the backward resampling is performed using the Markov property only; the latter can employ either resampling based on estimated (backwards) transition densities or the backward Local Bootstrap.

Algorithm 8.5.1 HYBRID BACKWARD MARKOV BOOTSTRAP BASED ON TRANSITION DENSITIES—HOMOSCEDASTIC CASE OF MODEL (7.1)

1. *Select a bandwidth h and construct the kernel estimator $\hat{m}(x)$ by Eq. (7.23), i.e.,*

$$\hat{m}(x) = \frac{\sum_{t=p}^{n-1} K(\frac{\|x-x_t\|}{h}) y_{t+1}}{\sum_{t=p}^{n-1} K(\frac{\|x-x_t\|}{h})}.$$

2. *Compute the residuals: $\hat{\varepsilon}_i = y_i - \hat{m}(x_{i-1})$ for $i = p+1, \cdots, n$.*
3. *Center the residuals: $\hat{r}_i = \hat{\varepsilon}_i - (n-p)^{-1} \sum_{t=p+1}^{n} \hat{\varepsilon}_t$ for $i = p+1, \cdots, n$; let the empirical distribution of $\hat{r}_t$ denoted by $\hat{F}_\varepsilon$.*

   (a) *Construct the backward transition density estimate $\hat{f}_{bn}$ as in Eq. (8.10).*
   (b) *Let $x_n^* = x_n$.*
   (c) *Generate $y_{n-p}^* \sim \hat{f}_{bn}(\cdot|x_n^* = x_n)$. Repeat it to generate $y_t^* \sim \hat{f}_{bn}(\cdot|x_{t+p}^*)$ for $t = n-p, \cdots, 1$ backwards in time, i.e., first for $t = n-p$, then for $t = n-p-1$, etc.*

(d) *Compute a future bootstrap observation* y^*_{n+1} *using model* (7.1)*, i.e.,*

$$y^*_{n+1} = \hat{m}(x^*_n) + \varepsilon^*_{n+1} = \hat{m}(x_n) + \varepsilon^*_{n+1}$$

where ε^*_{n+1} *is generated from* $\hat{F}_\varepsilon$*. Then re-estimate the conditional expectation based on the pseudo-data, i.e., let*

$$\hat{m}^*(x) = \frac{\sum_{i=p}^{n-1} K(\frac{\|x-x^*_i\|}{h}) y^*_{i+1}}{\sum_{i=p}^{n-1} K(\frac{\|x-x^*_i\|}{h})},$$

and compute the bootstrap predictor $\hat{y}^*_{n+1} = \hat{m}^*(x^*_n) = \hat{m}^*(x_n)$.

(e) *Calculate the bootstrap root replicate as* $y^*_{n+1} - \hat{y}^*_{n+1}$.

4. *Steps (a)-(e) in the above are repeated B times, and the B bootstrap root replicates are collected in the form of an empirical distribution whose* α*-quantile is denoted* $q(\alpha)$.
5. *Then, a* $(1-\alpha)100\%$ *equal-tailed predictive interval for* Y_{n+1} *is given by*

$$[\hat{m}(x_n) + q(\alpha/2),\ \hat{m}(x_n) + q(1-\alpha/2)].$$

Algorithm 8.5.2 HYBRID BACKWARD MARKOV BOOTSTRAP BASED ON TRANSITION DENSITIES—HETEROSCEDASTIC CASE OF MODEL (7.2)

1. *Select the bandwidth h and construct the estimates* $\hat{m}(\cdot)$ *and* $\hat{\sigma}^2(\cdot)$ *from Eq.* (7.23) *and* (7.30) *respectively.*
2. *Compute the residuals:*

$$\hat{\varepsilon}_i = \frac{y_i - \hat{m}(x_{i-1})}{\hat{\sigma}(x_{i-1})} \text{ for } i = p+1, \cdots, n.$$

3. *This step is similar to step 3 of Algorithm 8.5.1—the only difference is in part (d); here the future bootstrap observation* y^*_{n+1} *is computed from model* (7.2)*, i.e.,*

$$y^*_{n+1} = \hat{m}_g(x^*_n) + \hat{\sigma}_g(x^*_n)\varepsilon^*_{n+1} = \hat{m}_g(x_n) + \hat{\sigma}_g(x_n)\varepsilon^*_{n+1}.$$

In the above, $\hat{m}_g$ *and* $\hat{\sigma}_g$ *are* over-smoothed *estimates of* μ *and* σ *computed in the same way as* $\hat{m}$ *and* $\hat{\sigma}$ *but using a bandwidth g that is of bigger order than h.*

4– 5. *Same as the corresponding steps of Algorithm 8.5.1.*

Algorithm 8.5.3 HYBRID BACKWARD LOCAL BOOTSTRAP—HOMOSCEDASTIC CASE OF MODEL (7.1)

The algorithm is identical to Algorithm 8.5.1 with the exception of steps 3 (a) to (c) that have to be replaced by the following.

3.(a) Select a resampling bandwidth b and kernel W.

(b) Let $x_n^* = X_n$ Suppose x_{t+p}^* has already been generated for $1 \le t \le n-p$. Let J be a discrete random variable taking its values in the set $\{1, 2, \cdots, n-p\}$ with probability mass function given by

$$P(J = s) = \frac{W_b(x_{t+p}^* - x_{s+p})}{\sum_{m=1}^{n-p} W_b(x_{t+p}^* - x_{m+p})},$$

and let $y_t^* = y_J$.

(c) Repeat (b) to generate $y_{n-p}^*, \cdots, y_2^*, y_1^*$ backwards in time.

Algorithm 8.5.4 HYBRID BACKWARD LOCAL BOOTSTRAP—HETEROSCEDASTIC CASE OF MODEL (7.2)
The algorithm is the same as Algorithm 8.5.2 with the exception of steps 3 (a) to (c) that have to be performed as in Algorithm 8.5.3.

Remark 8.5.1 The hybrid Algorithms 8.5.1—8.5.4 use a model-based resampling based on fitted residuals. As discussed in Chap. 7, usage of *predictive* residuals may be preferable. According to the two models (7.1) or (7.2), the predictive residuals are respectively defined as $\hat{\varepsilon}_t^{(t)} = y_t - \hat{m}_t^{(t)}(x_{t-1})$ or $\hat{\varepsilon}_t^{(t)} = [y_t - \hat{m}_t^{(t)}(x_{t-1})]/\hat{\sigma}_t^{(t)}(x_{t-1})$ where $\hat{m}^{(t)}$ and $\hat{\sigma}^{(t)}$ are smoothing estimators calculated from the original dataset having the t-th point deleted. Finally, to define hybrid backward bootstrap intervals based on predictive residuals we just need to replace the fitted residuals $\{\hat{\varepsilon}_t\}$ in step 2 of Algorithms 8.5.1—8.5.4 by the predictive residuals $\{\hat{\varepsilon}_t^{(t)}\}$.

8.6 Prediction Intervals for Markov Processes Based on the Model-Free Prediction Principle

We now return to the setup of data from a general Markov(p) process that does not necessarily satisfy a model equation such as (7.1) or (7.2). In what follows, we will describe the **Model-Free Bootstrap for Markov Processes** introduced by Pan and Politis (2014); this is a novel approach that stems from the Model-Free Prediction Principle of Chap. 2.

As usual, the key idea is to transform a given complex dataset into one that is i.i.d., and therefore easier to handle. Instead of generating one-step ahead pseudo-data by some estimated conditional distribution, e.g., the transition density given in Eq. (8.4) or the transition distribution function given in Eq. (8.11), the Model-Free Bootstrap resamples the transformed i.i.d. data, and then transforms them back to obtain the desired one-step ahead prediction.

Note that the bootstrap based on kernel estimates of the transition density of Sect. 8.3, and the Local Bootstrap of Sect. 8.4 can also be considered model-free

methods as they apply in the absence of a model equation such as (7.1) or (7.2). The term Model-Free Bootstrap specifically refers to the transformation-based approach to inference stemming from the Model-Free Prediction Principle.

8.6.1 Theoretical Transformation

Let Y be a stationary Markov process of order p, and $X_{t-1} = (Y_{t-1}, \cdots, Y_{t-p})'$. Given $X_{t-1} = x \in \mathbf{R}^p$, we denote the conditional distribution of Y_t as

$$D_x(y) = P(Y_t \leq y | X_{t-1} = x). \tag{8.14}$$

This is the same distribution appearing in Eq. (8.1); changing the notation will help us differentiate between the different methods, and draws an analogy with the Model-free regression notions of Chap. 4.

For some positive integer $i \leq p$, we also define the distributions with partial conditioning as follows

$$D_{x,i}(y) = P(Y_t \leq y | X_{t-1}^{(i)} = x) \tag{8.15}$$

where $X_{t-1}^{(i)} = (Y_{t-1}, \cdots, Y_{t-i})'$ and $x \in \mathbf{R}^i$. In this notation, we can denote the unconditional distribution as $D_{x,0}(y) = P(Y_t \leq y)$ which does not depend on x. Throughout this section, we assume that, for any fixed x and i, the function $D_{x,i}(\cdot)$ is continuous and invertible over its support.

A transformation from our Markov(p) dataset $Y_1, \cdots, Y_n$ to an i.i.d. dataset $\eta_1, \cdots, \eta_n$ can now be constructed as follows. Let

$$\eta_1 = D_{x,0}(Y_1);\ \eta_2 = D_{X_1^{(1)},1}(Y_2);\ \eta_3 = D_{X_2^{(2)},2}(Y_3);\ \cdots;\ \eta_p = D_{X_{p-1}^{(p-1)},p-1}(Y_p) \tag{8.16}$$

$$\text{and}\ \ \eta_t = D_{X_{t-1}}(Y_t)\ \ \text{for}\ \ t = p+1, p+2, \cdots, n. \tag{8.17}$$

Note that the transformation from the vector $(Y_1, \cdots, Y_m)'$ to the vector $(\eta_1, \cdots, \eta_m)'$ is one-to-one and invertible for any natural number m by construction. Hence, the event $\{Y_1 = y_1, \cdots, Y_t = y_m\}$ is identical to the event $\{\eta_1 = \zeta_1, \cdots, \eta_t = \zeta_m\}$ when the construction of ζ_t follows (8.16) and (8.17), i.e.,

$$\zeta_1 = D_{x,0}(y_1);\ \zeta_2 = D_{x_1^{(1)},1}(y_2);\ \zeta_3 = D_{x_2^{(2)},2}(y_3);\ \cdots;\ \zeta_p = D_{x_{p-1}^{(p-1)},p-1}(y_p) \tag{8.18}$$

$$\text{and}\ \ \zeta_t = D_{x_{t-1}}(y_t)\ \ \text{for}\ \ t = p+1, p+2, \cdots, n \tag{8.19}$$

where $x_{t-1} = (y_{t-1}, \cdots, y_{t-p})'$ and $x_{t-1}^{(i)} = (y_{t-1}, \cdots, y_{t-i})'$.

It is not difficult to see that the random variables $\eta_1, \cdots, \eta_n$ are i.i.d. Uniform (0,1); in fact, this is just an application of Rosenblatt's (1952) transformation in the case of Markov(p) sequences. For example, the fact that η_1 is Uniform (0,1) is simply due to the probability integral transform. Now for $t > p$, we have

$$\begin{aligned} P(\eta_t \le z | \eta_{t-1} = \zeta_{t-1}, \ldots, \eta_1 = \zeta_1) &= P(\eta_t \le z | Y_{t-1} = y_{t-1}, \ldots, Y_1 = y_1) \\ &= P(\eta_t \le z | X_{t-1} = x_{t-1}, Y_{t-p-1} = y_{t-p-1}, \ldots, Y_1 = y_1) \end{aligned}$$

by the discussion preceding Eq. (8.18). Letting x be a short-hand for x_{t-1}, we have:

$$\begin{aligned} P(\eta_t \le z | \eta_{t-1} = \zeta_{t-1}, \ldots, \eta_1 = \zeta_1) =& P(D_x(Y_t) \le z | X_{t-1} = x, Y_{t-p-1} = y_{t-p-1}, \ldots,) \\ =& P(D_x(Y_t) \le z | X_{t-1} = x) \text{ [by Markov property]} \\ =& P(Y_t \le D_x^{-1}(z) | X_{t-1} = x) \\ =& D_x(D_x^{-1}(z)) \\ =& z \text{ [which is uniform, and not depending on } x\text{]}. \end{aligned}$$

Hence, for $t > p$, $P(\eta_t \le z | \eta_{t-1} = \zeta_{t-1}, \ldots, \eta_1 = \zeta_1) = z$, i.e., η_t is a random variable that is independent of its own past and has a Uniform (0,1) distribution. The same is true for η_t with $1 < t < p$; the argument is similar to the above but using the $D_{x,t}(\cdot)$ distribution instead of $D_x(\cdot)$. All in all, it should be clear that the random variables $\eta_1, \cdots, \eta_n$ are i.i.d. Uniform (0,1).

8.6.2 Estimating the Transformation from Data

To estimate the theoretical transformation from data, we would need to estimate the distributions $D_{x,i}(\cdot)$ for $i = 0, 1, \ldots, p-1$ and $D_x(\cdot)$. Note, however, that $D_{x,i}(\cdot)$ for $i < p$ can—in principle—be computed from $D_x(\cdot)$ since the latter uniquely specifies the whole distribution of the stationary Markov process. Hence, it should be sufficient to just estimate $D_x(\cdot)$ from our data. Another way of seeing this is to note that the p variables in Eq. (8.16) can be considered as "edge effects" or "initial conditions"; the crucial part of the transformation is given by Eq. (8.17), i.e., the one based on $D_x(\cdot)$.

Given observations $Y_1 = y_1, \cdots, Y_n = y_n$, we can estimate $D_x(y)$ by local averaging methods such as the kernel estimator

$$\hat{D}_x(y) = \frac{\sum_{i=p+1}^{n} \mathbf{1}\{y_i \le y\} K(\frac{\|x - x_{i-1}\|}{h})}{\sum_{k=p+1}^{n} K(\frac{\|x - x_{k-1}\|}{h})}. \tag{8.20}$$

In the above, $\hat{D}_x(y)$ is a step function in y. It is possible to use linear interpolation on this step function to produce an estimate $\tilde{D}_x(y)$ that is piecewise linear and strictly increasing (and therefore invertible); see Pan and Politis (2014) for details. However,

in the interest of conciseness, we will go straight to the construction of a smooth, i.e., differentiable, distribution estimator $\bar{D}_x(y)$.

As in Chap. 4, let $\Lambda(\cdot)$ be a cumulative distribution function that is absolutely continuous and strictly increasing over its support; h_0 is a positive bandwidth parameter. Define the smooth estimator

$$\bar{D}_x(y) = \frac{\sum_{i=p+1}^{n} \Lambda(\frac{y-y_i}{h_0}) K(\frac{\|x-x_{i-1}\|}{h})}{\sum_{k=p+1}^{n} K(\frac{\|x-x_{k-1}\|}{h})}; \tag{8.21}$$

consequently, the transformed data $\{v_t \text{ for } t = p+1, \cdots, n\}$ can be calculated by

$$v_t = \bar{D}_{y_{t-1}}(x_t); \tag{8.22}$$

it then follows that $v_t \approx \eta_t$ where η_t was defined in Sect. 8.6.1.

Claim 8.6.1 *Under regularity conditions, including absolute continuity of $D_x(y)$ in y for all x, the sequence $\{v_t$ for $t = p+1, \cdots, n\}$ is approximately i.i.d. Uniform (0,1).*

As in Claim 4.2.1, the word "approximately" in the above should be interpreted as "asymptotically" for large n; again note that $v_{p+1}, \ldots, v_n$ represent the n-th row of a triangular array although this is not explicitly denoted. Hence, the goal of transforming our observed data $y_1, \cdots, y_n$ to a realization of a sequence of (approximately) i.i.d. random variables v_t has been achieved; note that the "initial conditions" $v_1, \ldots, v_p$ were not explicitly generated in the above as they are not needed in the Model-free bootstrap algorithms.

The Model-free bootstrap algorithm for Markov processes goes as follows.

Algorithm 8.6.1 MODEL-FREE (MF) BOOTSTRAP PREDICTION INTERVALS

1. *Use Eq. (8.22) to obtain the transformed data $v_{p+1}, \cdots, v_n$.*
2. *Calculate $\hat{y}_{n+1}$, the point predictor of y_{n+1}, by*

$$\hat{y}_{n+1} = \frac{1}{n-p} \sum_{t=p+1}^{n} \bar{D}_{x_n}^{-1}(v_t). \tag{8.23}$$

3.(a) *Resample randomly (with replacement) the transformed variables $v_{p+1}, \cdots, v_n$ to create the pseudo-data $v_{-M}^*, v_{-M+1}^*, \cdots, v_0^*, v_1^*, \cdots, v_{n-1}^*, v_n^*$ and v_{n+1}^* for some large positive integer M.*

(b) *Let $x_p^* = (y_{p+I}, \cdots, y_{1+I})'$ where I is generated as a discrete random variable uniform on the values $0, 1, \ldots, n-p$.*

(c) *Generate the bootstrap pseudo-data $y_t^* = \bar{D}_{x_{t-1}^*}^{-1}(v_t^*)$ for $t = -M+p, \cdots, n$.*

(d) *Calculate the bootstrap future value $y_{n+1}^* = \bar{D}_{x_n}^{-1}(v_{n+1}^*)$.*

(e) *Calculate the bootstrap predictor $\hat{y}_{n+1}^* = \frac{1}{n-p} \sum_{t=p+1}^{n} \bar{D}_{x_n}^{*-1}(v_t^*)$ where*

$$\bar{D}_x^*(y) = \frac{\sum_{i=p+1}^{n} \Lambda(\frac{y-y_i^*}{h_0}) K(\frac{\|x-x_{i-1}^*\|}{h})}{\sum_{k=p+1}^{n} K(\frac{\|x-x_{k-1}^*\|}{h})}.$$

(f) *Calculate the bootstrap root $y_{n+1}^* - \hat{y}_{n+1}^*$.*

4. *Repeat step 3 above B times; the B bootstrap root replicates are collected in the form of an empirical distribution whose α-quantile is denoted $q(\alpha)$.*
5. *The $(1-\alpha)100\%$ equal-tailed prediction interval for Y_{n+1} is given by*

$$[\hat{y}_{n+1} + q(\alpha/2),\ \hat{y}_{n+1} + q(1-\alpha/2)].$$

Bandwidth choices here are as in Remark 4.2.4, i.e., $h_0 = h^2$ where h is chosen via the cross-validation procedure discussed in Remark 7.6.1.

Remark 8.6.1 Algorithm 8.6.1 is in effect a *Forward bootstrap* algorithm for prediction intervals according to the discussion of Sect. 8.2. Constructing a *backward bootstrap* analog of Algorithm 8.6.1 is straightforward based on the Markov property of the time-reversed process as in Fact 8.2.1. One would just need a reverse construction of the theoretical transformation of Sect. 8.6.1. To elaborate on the latter, we would instead let $\eta_t = G_{X_{t+p}}(Y_t)$ for $t = n-p, n-p-1, \ldots, 1$ where $G_x(y) = P(Y_t \le y | X_{t+p} = x)$ is the backwards analog of $D_x(y)$; the η_t for $t = n, \ldots, n-p+1$ can be generated using the backwards analogs of $D_{x,i}(y)$. The details are straightforward and are omitted especially since the finite-sample performance of the two approaches is practically indistinguishable.

As mentioned in Remark 8.2.1, there exist different approximations to the conditional expectation which serves as the L_2–optimal predictor. The usual one is the kernel smoothed estimator (7.23) but Eq. (8.23) gives an alternative approximation; we have used it in Algorithm 8.6.1 because it follows directly from the Model-Free Prediction Principle. However, the two approximations are asymptotically equivalent, and thus can be used interchangeably. To see why, note that

$$\frac{1}{n-p}\sum_{t=p+1}^{n} \bar{D}_{x_n}^{-1}(u_t) \simeq \int_0^1 \bar{D}_{x_n}^{-1}(u)du \simeq \int y\hat{f}_n(y|x_n)dy \simeq \hat{m}(x_n) \tag{8.24}$$

where

$$\hat{m}(x_n) = \int_0^1 \hat{D}_{x_n}^{-1}(u)du \simeq \frac{1}{n-p}\sum_{t=p+1}^{n} \hat{D}_{x_n}^{-1}(u_t); \tag{8.25}$$

as usual, $\hat{D}_x^{-1}(\cdot)$ indicates the *quantile inverse* of the step-function $\hat{D}_x(\cdot)$. By analogy to Claim 4.3.1, we can also state the following:

Claim 8.6.2 *Under regularity conditions, all the quantities appearing in Eqs.* (8.24) *and* (8.25) *are asymptotically equivalent, i.e., the difference between any two of these quantities is $o_p(1/(hn))$.*

Remark 8.6.2 Recall that $\hat{D}_x(y)$ is a local average estimator, i.e., averaging the indicator $\mathbf{1}\{y_i \le y\}$ over data blocks X_t that are close to x. If a given x is outside the range of the data blocks X_t, then obviously estimator $\hat{D}_x(y)$ cannot be constructed, and the same is true for $\bar{D}_x(y)$. Similarly, if x is at the edges of the range of X_t, e.g., within h of being outside the range, then $\hat{D}_x(y)$ and $\bar{D}_x(y)$ will not be very accurate. Step 1 of Algorithm 8.6.1 can then be modified to drop the v_is that are obtained from an y_i whose x_{i-1} is within h of the boundary; see Chap. 4 for a related discussion.

From Eq. (8.20), we see that the conditional distribution of main interest is $D_{x_n}(y) = P(Y_t \leq y | X_{t-1} = x_n)$ which is estimated by

$$\bar{D}_{x_n}(y) = \frac{\sum_{i=p+1}^{n} \Lambda(\frac{y-y_i}{h_0}) K(\frac{\|x_n - x_{i-1}\|}{h})}{\sum_{k=p+1}^{n} K(\frac{\|x_n - x_{i-1}\|}{h})}.$$

Since y_{n+1} is not observed, the above estimated conditional distribution treats the "scatterplot" pair (x_n, y_{n+1}) as an "out-of-sample" pair. To mimic this situation in the model-free setup, we can use the trick as in Chap. 4, i.e., to calculate an estimate of $D_{x_n}(y)$ based on a dataset that excludes the pair (x_{t-1}, y_t) for $t = p+1, \cdots, n$. The corresponding delete-one estimator is defined as

$$\bar{D}_x^{(t)}(y_t) = \frac{\sum_{i=p+1, i\neq t}^{n} \Lambda(\frac{y_t - y_i}{h_0}) K(\frac{\|x_{t-1} - x_{i-1}\|}{h})}{\sum_{k=p+1, k\neq t} K(\frac{\|x_{t-1} - x_{k-1}\|}{h})} \text{ for } t = p+1, \cdots, n$$

that is used to construct the transformed data:

$$v_t^{(t)} = \bar{D}_{x_{t-1}}^{(t)}(y_t) \text{ for } t = p+1, \cdots, n; \tag{8.26}$$

this leads to the Predictive Model-Free bootstrap algorithm.

Algorithm 8.6.2 PREDICTIVE MODEL-FREE (PMF) PREDICTION INTERVALS
The algorithm is identical to Algorithm 8.6.1 after substituting $v_{p+1}^{(p+1)}, \cdots, v_n^{(n)}$ *in place of* $v_{p+1}, \cdots, v_n$.

As in Chap. 4, we can also devise a Limit Model-Free bootstrap scheme that is not affected from the boundary issues mentioned in Remark 8.6.2. To do so, we estimate $D_x^{-1}(\cdot)$ by the quantile inverse $\hat{D}_x^{-1}(\cdot)$. There is no need to estimate the smooth function $D_x(\cdot)$ *per se* as there is no need to generate the "uniformized" data $v_{p+1}, \cdots, v_n$. Otherwise, the LMF algorithm is similar to Algorithm 8.6.1.

Algorithm 8.6.3 LIMIT MODEL-FREE (LMF) PREDICTION INTERVALS

1. *Calculate* $\hat{y}_{n+1}$, *the point predictor of* y_{n+1}, *by*

$$\hat{y}_{n+1} = \frac{1}{n-p} \sum_{t=p+1}^{n} \hat{D}_{x_n}^{-1}(v_t)$$

2.(a) *For some large positive integer M, generate the pseudo-data* $v_{-M}^*, v_{-M+1}^*, \cdots, v_0^*, v_1^*, \cdots, v_{n-1}^*, v_n^*$ *and* v_{n+1}^* *as i.i.d. Uniform (0,1).*

(b) *Let* $x_p^* = (y_{p+I}, \cdots, y_{1+I})'$ *where I is generated as a discrete random variable uniform on the values* $0, 1, \ldots, n-p$.

(c) *Generate the bootstrap pseudo-data* $y_t^* = \hat{D}_{x_{t-1}^*}^{-1}(v_t^*)$ *for* $t = -M+p, \cdots, n$.

(d) *Calculate the bootstrap future value* $y_{n+1}^* = \hat{D}_{x_n}^{-1}(v_{n+1}^*)$.

(e) *Calculate the bootstrap predictor* $\hat{y}_{n+1}^* = \frac{1}{n-p} \sum_{t=p+1}^{n} \hat{D}_{x_n}^{*-1}(v_t^*)$ *where* $\hat{D}_x^*(y)$ *is the step function estimator* $D_x(y)$ *as computed from the bootstrap pseudo-data* $y_1^*, \ldots, y_n^*$.

(f) Calculate the bootstrap root $y^*_{n+1} - \hat{y}^*_{n+1}$.

3. *Repeat step 2 above B times; the B bootstrap root replicates are collected in the form of an empirical distribution whose* α*-quantile is denoted* $q(\alpha)$.
4. *The* $(1-\alpha)100\%$ *equal-tailed prediction interval for* Y_{n+1} *is given by*

$$[\hat{y}_{n+1} + q(\alpha/2),\ \hat{y}_{n+1} + q(1-\alpha/2)].$$

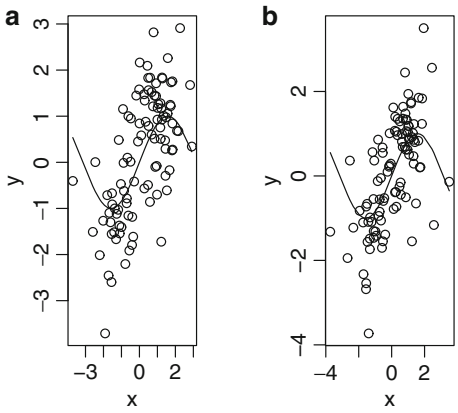

Fig. 8.1 Typical scatterplots of y_t vs. x_t (where $x_t = y_{t-1}$) with the conditional (true) mean function superimposed. (**a**) Data from Model 1; (**b**) Data from Model 2 [Both with normal errors and $n = 100$]

8.7 Finite-Sample Performance of Model-Free Prediction Intervals

Monte Carlo simulations were conducted to assess the performance of the prediction intervals proposed in this chapter through average coverage level (CVR) and length (LEN). The following models were chosen in order to generate Markov processes (of order $p = 1$).

- Model 1: $Y_{t+1} = \sin(Y_t) + \varepsilon_{t+1}$
- Model 2: $Y_{t+1} = \sin(Y_t) + \sqrt{0.5 + 0.25Y_t^2}\ \varepsilon_{t+1}$

where the errors $\{\varepsilon_t\}$ were i.i.d. $N(0,1)$ or Laplace rescaled to unit variance. Sample sizes n =50, 100, and 200 were considered, and both 90 % and 95 % prediction intervals were constructed.

For each model, 500 datasets of size n were generated. Figure 8.1 shows typical scatterplots of Y_t vs. $X_t (= Y_{t-1})$ from Models 1 and 2 based on normal errors and $n = 100$. For each dataset, one of the bootstrap methods was used to create B bootstrap sample paths and B one-step ahead future values denoted by $Y_{n+1,j}$ for

$j = 1, 2, \cdots, B$; for computational reasons, we chose $B = 250$ but in a real-data application (with a single dataset) it is advisable to let B be at least 1000.

Replicates of the bootstrap prediction interval (L_i, U_i) were constructed for $i = 1, 2, \cdots, 500$, and coverage level and length were estimated by

$$CVR = \frac{1}{500}\sum_{i=1}^{500} CVR_i \text{ and } LEN = \frac{1}{500}\sum_{i=1}^{500} LEN_i$$

where

$$CVR_i = \frac{1}{B}\sum_{j=1}^{B} \mathbf{1}_{[L_i, U_i]}(X_{n+1,j}) \text{ and } LEN_i = U_i - L_i.$$

Note that for the i-th dataset of size n (where $i = 1, 2, \cdots, 500$), the prediction interval (L_i, U_i) of Y_{n+1} was constructed given $Y_n = y_{ni}$, where y_{ni} is the last observation from the i-th dataset. The values of these y_{ni}s are different for each i; therefore, the above CVRs are an estimate of *unconditional*, i.e., average, coverage level.

Some further details are as follows:

- For the bootstrap approach based on the transition density, we chose $K(x, y) = k_1(x)k_1(y)$, where $k_1(x)$ is the standard normal density. As suggested in Remark 8.3.1, we chose $h = 0.9An^{-1/4}$ where $A = min(\hat{\sigma}, \frac{IQR}{1.34})$, $\hat{\sigma}$ is the estimated standard deviation of the data, and IQR is the sample interquartile range.
- The kernel W in the Local Bootstrap method was the normal kernel, and the forward and backward bandwidth g and b were chosen by cross-validation.
- In the hybrid backward bootstrap procedure for nonparametric autoregression, the estimation bandwidth h for nonparametric bootstrap and the resampling bandwidth b for Local Bootstrap were all selected by cross-validation based on corresponding regression kernel estimators. As above, the resampling bandwidth b for the backward bootstrap based on the transition density was chosen as $0.9An^{-1/4}$.
- The model-based and hybrid bootstrap methods employed a two-bandwidth trick in the case of data from heteroscedastic Model 2 as discussed in Sect. 7.6.2. In particular, the choice $g = 2h$ was made where h is chosen by cross-validation[2]; doubling the original bandwidth h is a simple rule-of-thumb used in previous work in nonparametric regression with i.i.d. errors.
- For the Model-free Bootstrap, $\Lambda(\cdot)$ was chosen as the standard normal cumulative distribution function restricted on $[-2, 2]$. As already mentioned, the bandwidth h for kernel $K(\cdot)$ was chosen via cross-validation, and the smoothing bandwidth h_0 was set to $h_0 = h^2$.

Tables 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, and 8.7 summarize all the simulation results. The first two lines of each table are the simulation results using the model-based bootstrap procedures discussed in Sect. 7.6, i.e., fitting a nonparametric AR model to the data, and resampling the residuals; entries `nonpara-f` and `nonpara-p` denote resampling the fitted vs. predictive residuals, respectively, i.e., the methods

[2] If the choice of bandwidth g is not over-smoothed, e.g., if $g = h$, then the resulting prediction intervals exhibit profound under-coverage; see Pan and Politis (2015) for discussion.

referred to as Ff and Fp in Sect. 7.6. These intervals are based on unstudentized roots, since as alluded to in Remark 7.2.2, the studentized root intervals FSf and FSp were found to have identical performance as their unstudentized counterparts Ff and Fp.

Lines 3 and 4 of each table are the simulation results using bootstrap based on transition density discussed in Sect. 8.3; the forward and backward methods are denoted `trans-forward` and `trans-backward`, respectively. Lines 5 and 6 are the results using the Local Bootstrap discussed in Sect. 8.4; the forward and backward methods are denoted `LB-forward` and `LB-backward`, respectively.

Lines 7 to 10 are the simulation results using the hybrid backward Markov bootstrap for nonparametric autoregression of Sect. 8.5. Notation `hybrid-trans-f` and `hybrid-trans-p` denote that the backward generating mechanism uses an estimator of transition density while the generation of Y^*_{n+1} is done via model-based resampling the fitted vs. predictive residuals, respectively. Similarly, notation `hybrid-LB-f` and `hybrid-LB-p` denote that the backward generating mechanism is done by Local Bootstrap while the generation of Y^*_{n+1} is done via model-based resampling the fitted vs. predictive residuals, respectively.

The last two lines of each table give the results using the Model-free Bootstrap of Sect. 8.6. Both the basic Model-Free method of Algorithm 8.6.1 as well as the Predictive Model-Free method of Algorithm 8.6.2 were used; the notation is `MF` and `PMF`, respectively.

Some general comments on the simulations are as follows:

- As expected, when the sample size is increased, then the coverage level accuracy is improved and the standard deviation associated with each interval length (denoted by st.dev. in the tables) is decreased.
- The model-based nonparametric and/or hybrid methods with predictive residuals outperform the respective ones with fitted residuals. Especially when the sample size is not large enough, using predictive residuals significantly improves the coverage level.
- The standard deviations of interval lengths are quite large for the model with heteroscedastic errors using the model-based nonparametric and/or hybrid approaches.
- The forward and backward methods have similar performances in both the bootstrap based on transition density and the Local Bootstrap.
- The Predictive Model-free (PMF) method improves the coverage level of the basic MF bootstrap at the cost of higher variability.

Comparing all the simulation results from the Tables 8.1, 8.2, 8.3, and 8.4 it is apparent that with data generated by the model with homoscedastic errors, the nonparametric model-based and hybrid methods—especially the methods with predictive residuals—have better performance, particularly in view of their smaller standard deviation of the interval length; this should not be surprising since model-based methods should have an advantage when the model is true—as it is the case here.

normal innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.dev.	CVR	LEN	st.dev.
nonpara-f	0.913	3.824	0.519	0.855	3.223	0.424
nonpara-p	0.941	4.207	0.536	0.893	3.547	0.425
trans-forward	0.931	4.126	0.760	0.886	3.544	0.667
trans-backward	0.931	4.130	0.757	0.887	3.555	0.677
LB-forward	0.910	3.885	0.778	0.862	3.337	0.685
LB-backward	0.911	3.920	0.795	0.863	3.355	0.676
Hybrid-trans-f	0.908	3.822	0.522	0.852	3.230	0.432
Hybrid-trans-p	0.935	4.181	0.583	0.889	3.553	0.470
Hybrid-LB-f	0.914	3.782	0.525	0.860	3.199	0.433
Hybrid-LB-p	0.938	4.136	0.592	0.892	3.496	0.463
MF	0.892	3.627	0.717	0.843	3.131	0.619
PMF	0.939	4.293	0.828	0.893	3.614	0.709
$n = 100$	CVR	LEN	st.dev.	CVR	LEN	st.dev.
nonpara-f	0.927	3.860	0.393	0.873	3.255	0.310
nonpara-p	0.943	4.099	0.402	0.894	3.456	0.317
trans-forward	0.942	4.137	0.627	0.901	3.535	0.519
trans-backward	0.942	4.143	0.621	0.900	3.531	0.519
LB-forward	0.930	3.980	0.625	0.886	3.409	0.511
LB-backward	0.932	4.001	0.605	0.886	3.411	0.508
Hybrid-trans-f	0.921	3.822	0.412	0.868	3.241	0.335
Hybrid-trans-p	0.936	4.045	0.430	0.889	3.441	0.341
Hybrid-LB-f	0.923	3.815	0.430	0.869	3.226	0.343
Hybrid-LB-p	0.937	4.018	0.433	0.890	3.414	0.338
MF	0.916	3.731	0.551	0.869	3.221	0.489
PMF	0.946	4.231	0.647	0.902	3.471	0.530
$n = 200$	CVR	LEN	st.dev.	CVR	LEN	st.dev.
nonpara-f	0.938	3.868	0.272	0.886	3.263	0.219
nonpara-p	0.948	4.012	0.283	0.899	3.385	0.231
trans-forward	0.944	4.061	0.501	0.902	3.472	0.415
trans-backward	0.944	4.058	0.507	0.902	3.470	0.424
LB-forward	0.937	3.968	0.530	0.891	3.369	0.439

Table 8.1 Model 1: $Y_{t+1} = \sin(Y_t) + \varepsilon_{t+1}$ with normal innovations.

LB-backward	0.937	3.979	0.551	0.893	3.383	0.448
Hybrid-trans-f	0.932	3.838	0.359	0.880	3.238	0.290
Hybrid-trans-p	0.942	3.977	0.360	0.893	3.358	0.281
Hybrid-LB-f	0.932	3.798	0.336	0.882	3.228	0.272
Hybrid-LB-p	0.942	3.958	0.338	0.895	3.356	0.265
MF	0.924	3.731	0.464	0.877	3.208	0.387
PMF	0.946	4.123	0.570	0.899	3.439	0.444

Table 8.1 (continued)

Interestingly, all our model-free methods seem to be competitive with the model-based methods—even in our simulation in which an AR model actually holds true—with the PMF method being the most prominent. What is surprising is that for data arising from the model with heteroscedastic errors, several of the model-free bootstrap methods have better coverage level and smaller variability compared to the benchmark model-based nonparametric AR resampling; this finding is corroborated by simulations in Pan and Politis (2014) based on three additional models.

8.8 Model-Free Confidence Intervals in Markov Processes

As already mentioned, the Model-Free Bootstrap for Markov Processes is a novel resampling scheme that follows from the Model-Free Prediction Principle. Section 8.6 described the application of the Model-Free Bootstrap for the construction of prediction intervals.

In what follows, we will show how the Model-Free Bootstrap can be used to construct confidence intervals for parameters associated with the conditional distribution $D_x(y) = P(Y_t \le y | X_{t-1} = x)$ where $X_t = (Y_t, \cdots, Y_{t-p+1})'$. For concreteness, we will focus on the conditional expectation function $\mu(x) = E(Y_t | X_{t-1} = x)$ as the parameter of interest but the algorithms remain true *verbatim* for other functionals of $D_y(\cdot)$, e.g., the conditional median, the conditional variance, etc.

Fix some $x \in \mathbf{R}^p$; the goal is to construct a $(1-\alpha)100\%$ confidence interval for $\mu(x)$ based on the Markov(p) dataset $Y_1, \ldots, Y_n$. To do this, we will need to approximate the sampling distribution of the **root:** $\mu(x) - \hat{m}(x)$ by the distribution of the bootstrap root: $\hat{m}(x) - \hat{m}^*(x)$.

The Model-free Bootstrap algorithm for confidence intervals is a simplified version of Algorithm 8.6.1.

Algorithm 8.8.1 MODEL-FREE (MF) BOOTSTRAP CONFIDENCE INTERVALS

1. *Use Eq. (8.22) to obtain the transformed data* $v_{p+1}, \cdots, v_n$.
2. *Calculate* $\hat{m}(x)$*, the estimator of* $\mu(x)$*. Here,* $\hat{m}(x)$ *can be the Nadaraya-Watson smoother of Eq.* (7.23) *or one of its aforementioned asymptotically equivalent forms.*

Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.dev.	CVR	LEN	st.dev.
nonpara-f	0.912	4.103	0.885	0.854	3.187	0.612
nonpara-p	0.935	4.504	0.890	0.888	3.561	0.641
trans-forward	0.913	4.072	1.033	0.873	3.429	0.894
trans-backward	0.913	4.081	1.021	0.873	3.441	0.897
LB-forward	0.902	4.036	1.138	0.856	3.313	0.935
LB-backward	0.905	4.046	1.103	0.861	3.324	0.847
Hybrid-trans-f	0.905	4.070	0.925	0.848	3.174	0.633
Hybrid-trans-p	0.926	4.425	0.931	0.878	3.507	0.635
Hybrid-LB-f	0.913	4.081	0.934	0.857	3.184	0.623
Hybrid-LB-p	0.929	4.447	0.973	0.882	3.498	0.655
MF	0.891	3.715	0.963	0.846	3.084	0.764
PMF	0.930	4.442	1.126	0.887	3.620	0.972
$n = 100$	CVR	LEN	st.dev.	CVR	LEN	st.dev.
nonpara-f	0.933	4.161	0.648	0.879	3.218	0.452
nonpare-p	0.944	4.430	0.658	0.896	3.445	0.470
trans-forward	0.925	4.236	1.027	0.885	3.424	0.763
trans-backward	0.926	4.239	1.024	0.885	3.437	0.764
LB-forward	0.923	4.153	0.935	0.878	3.323	0.714
LB-backward	0.923	4.189	0.986	0.879	3.356	0.724
Hybrid-trans-f	0.925	4.056	0.702	0.872	3.174	0.495
Hybrid-trans-p	0.939	4.370	0.748	0.891	3.418	0.513
Hybrid-LB-f	0.927	4.094	0.687	0.876	3.202	0.493
Hybrid-LB-p	0.938	4.310	0.731	0.891	3.400	0.512
MF	0.910	3.846	0.856	0.864	3.106	0.623
PMF	0.941	4.544	0.965	0.896	3.542	0.738
$n = 200$	CVR	LEN	st.dev.	CVR	LEN	st.dev.
nonpara-f	0.937	4.122	0.460	0.885	3.198	0.329
nonpara-p	0.943	4.275	0.455	0.895	3.341	0.341
trans-forward	0.928	4.184	0.914	0.884	3.307	0.619
trans-backward	0.929	4.202	0.904	0.883	3.299	0.619
LB-forward	0.928	4.140	0.838	0.883	3.274	0.586
LB-backward	0.929	4.142	0.850	0.884	3.298	0.610
Hybrid-trans-f	0.931	4.041	0.509	0.880	3.172	0.388

Table 8.2 Model 1: $Y_{t+1} = \sin(Y_t) + \varepsilon_{t+1}$ with Laplace innovations.

Hybrid-trans-p	0.939	4.221	0.544	0.891	3.307	0.385
Hybrid-LB-f	0.932	4.041	0.560	0.881	3.170	0.405
Hybrid-LB-p	0.940	4.204	0.552	0.892	3.302	0.395
MF	0.921	3.895	0.684	0.873	3.103	0.491
PMF	0.942	4.447	0.868	0.894	3.407	0.629

Table 8.2 (continued)

3.(a) *Resample randomly (with replacement) the transformed variables* $v_{p+1},\cdots,v_n$ *to create the pseudo-data* $v^*_{-M}, v^*_{-M+1},\cdots,v^*_0, v^*_1,\cdots,v^*_{n-1}, v^*_n$ *for some large positive integer* M.

(b) *Let* $(y^*_{-M},\cdots,y^*_{-M+p-1})' = (y_{1+I},\cdots,y_{p+I})'$ *where* I *is generated as a discrete random variable uniform on the values* $0,1,\ldots,n-p$*; let* $x^*_{-M+p-1} = (y^*_{-M+p-1},\ldots,y^*_{-M})$.

(c) *Generate* $y^*_t = \bar{D}^{-1}_{x^*_{t-1}}(v^*_t)$ *for* $t = -M+p,\cdots,n$.

(d) *Calculate the bootstrap estimator* $\hat{m}^*(x)$ *which is the same estimator as* $\hat{m}(x)$ *but computed from the bootstrap data* $y^*_1,\ldots,y^*_n$.

(e) *Calculate the bootstrap root* $\hat{m}(x) - \hat{m}^*(x)$.

4. *Repeat step 3 above B times; the B bootstrap root replicates are collected in the form of an empirical distribution whose* α*-quantile is denoted* $q(\alpha)$.

5. *The* $(1-\alpha)100\%$ *equal-tailed confidence interval for* $\mu(x)$ *is given by*

$$[\hat{m}(x) + q(\alpha/2),\ \hat{m}(x) + q(1-\alpha/2)].$$

Bandwidth choices here are as in Remark 4.2.4, i.e., h is chosen via cross-validation, and then let $h_0 = h^2$. As in Remark 8.6.2, the above algorithm can be modified to drop the v_is that are obtained from an Y_is whose X_{i-1} is within h of the boundary.

We can also define an analog of the Predictive Model-Free Algorithm 8.6.2.

Algorithm 8.8.2 PREDICTIVE MODEL-FREE (PMF) CONFIDENCE INTERVALS
The algorithm is identical to Algorithm 8.8.1 after substituting the variables $v^{(p+1)}_{p+1}$, $\cdots, v^{(n)}_n$ *obtained from Eq.* (8.26) *in place of* $v_{p+1},\cdots,v_n$.

Finally, the following is a version of the Limit Model-Free Algorithm 8.6.3 tailored for the construction of confidence intervals.

Algorithm 8.8.3 LIMIT MODEL-FREE (LMF) BOOTSTRAP CONFIDENCE INTERVALS

1. *Calculate* $\hat{m}(x)$*, the estimator of* $\mu(x)$*. Here,* $\hat{m}(x)$ *can be the Nadaraya-Watson smoother of Eq.* (7.23) *or one of its aforementioned asymptotically equivalent forms.*

normal innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.dev.	CVR	LEN	st.dev.
nonpara-f	0.864	3.112	1.276	0.813	2.671	1.089
nonpara-p	0.904	3.645	1.538	0.850	2.967	1.218
trans-forward	0.927	3.477	0.802	0.882	2.983	0.716
trans-backward	0.928	3.495	0.832	0.882	2.992	0.722
LB-forward	0.914	3.424	0.833	0.867	2.893	0.717
LB-backward	0.914	3.451	0.803	0.867	2.924	0.692
Hybrid-trans-f	0.865	3.169	1.212	0.812	2.721	1.050
Hybrid-trans-p	0.897	3.639	1.494	0.840	2.978	1.148
Hybrid-LB-f	0.862	3.146	1.256	0.808	2.686	1.070
Hybrid-LB-p	0.895	3.639	1.637	0.840	2.956	1.180
MF	0.890	3.073	0.712	0.842	2.653	0.630
PMF	0.931	3.687	0.836	0.891	3.078	0.699
$n = 100$	CVR	LEN	st.dev.	CVR	LEN	st.dev.
nonpara-f	0.894	3.015	0.926	0.843	2.566	0.783
nonpara-p	0.922	3.318	1.003	0.868	2.744	0.826
trans-forward	0.943	3.421	0.629	0.901	2.928	0.561
trans-backward	0.943	3.439	0.648	0.901	2.930	0.573
LB-forward	0.938	3.425	0.628	0.895	2.908	0.553
LB-backward	0.937	3.410	0.616	0.894	2.903	0.549
Hybrid-trans-f	0.888	2.997	0.873	0.833	2.564	0.747
Hybrid-trans-p	0.914	3.301	0.980	0.855	3.726	0.792
Hybrid-LB-f	0.888	2.988	0.877	0.834	2.553	0.748
Hybrid-LB-p	0.916	3.301	0.996	0.858	2.727	0.796
MF	0.921	3.119	0.551	0.874	2.679	0.476
PMF	0.951	3.587	0.620	0.908	2.964	0.513
$n = 200$	CVR	LEN	st.dev.	CVR	LEN	st.dev.
nonpara-f	0.903	2.903	0.774	0.848	2.537	0.647
nonpara-p	0.921	3.164	0.789	0.863	2.636	0.654
trans-forward	0.943	3.428	0.627	0.901	2.921	0.548

Table 8.3 Model 2: $Y_{t+1} = \sin(Y_t) + \sqrt{0.5 + 0.25Y_t^2}\varepsilon_{t+1}$ with normal innovations.

trans-backward	0.943	3.430	0.633	0.901	2.921	0.552
LB-forward	0.942	3.425	0.578	0.898	2.894	0.483
LB-backward	0.941	3.406	0.562	0.895	2.858	0.462
Hybrid-trans-f	0.892	2.991	0.789	0.836	2.541	0.652
Hybrid-trans-p	0.908	3.147	0.816	0.849	2.631	0.663
Hybrid-LB-f	0.892	2.953	0.763	0.837	2.520	0.643
Hybrid-LB-p	0.911	3.148	0.810	0.853	2.628	0.663
MF	0.926	3.167	0.504	0.879	2.707	0.420
PMF	0.947	3.481	0.574	0.900	2.890	0.473

Table 8.3 (continued)

2. *(a) For some large positive integer M, generate the pseudo-data* $v^*_{-M}, v^*_{-M+1}, \cdots, v^*_0, v^*_1, \cdots, v^*_{n-1}, v^*_n$ *as i.i.d. from Uniform (0,1).*
   (b) Let $(y^*_{-M}, \cdots, y^*_{-M+p-1})' = (y_{1+I}, \cdots, y_{p+I})'$ *where I is generated as a discrete random variable uniform on the values* $0, 1, \ldots, n-p$
   (c) Generate $y^*_t = \hat{D}^{-1}_{x^*_{t-1}}(v^*_t)$ *for* $t = -M+p, \cdots, n$.
   (d) Calculate the bootstrap estimator $\hat{m}^*(x)$ *which is the same estimator as* $\hat{m}(x)$ *but computed from the bootstrap data* $y^*_1, \ldots, y^*_n$.
   (e) Calculate the bootstrap root $\hat{m}(x) - \hat{m}^*(x)$.
3. *Repeat step 3 above B times; the B bootstrap root replicates are collected in the form of an empirical distribution whose* α*-quantile is denoted* $q(\alpha)$.
4. *The* $(1-\alpha)100\%$ *equal-tailed LMF confidence interval for* $\mu(x)$ *is given by*

$$[\hat{m}(x) + q(\alpha/2),\ \hat{m}(x) + q(1-\alpha/2)].$$

Remark 8.8.1 The above three Model-Free Bootstrap methods were discussed in terms of the concrete application of constructing a confidence interval for $\mu(x)$ pointwise, i.e., for a given x. However, constructing simultaneous confidence intervals for $\{\mu(x)$ with $x \in S\}$ for any finite set S is immediate as with any bootstrap method. If the set S consists of points on a fine grid that span an interval, say $[a_1, a_2]$, then the assumed smoothness of μ can be used to turn the aforementioned simultaneous confidence intervals into a confidence band for $\{\mu(x)$ for $x \in [a_1, a_2]\}$.

8.8.1 Finite-Sample Performance of Confidence Intervals

Monte Carlo simulations were conducted to assess the performance of the Model-Free bootstrap confidence intervals through average coverage level (CVR) and length (LEN), and compare them to Rajarshi's (1990) bootstrap based on transition

Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.dev.	CVR	LEN	st.dev.
nonpara-f	0.862	3.173	1.676	0.811	2.532	1.230
nonpara-p	0.904	3.881	2.078	0.849	2.878	1.354
trans-forward	0.910	3.359	0.964	0.870	2.828	0.847
trans-backward	0.911	3.369	0.953	0.871	2.839	0.841
LB-forward	0.904	3.423	1.029	0.864	2.809	0.818
LB-backward	0.907	3.433	0.983	0.864	2.798	0.766
Hybrid-trans-f	0.866	3.209	1.562	0.814	2.595	1.195
Hybrid-trans-p	0.899	3.848	1.980	0.841	2.895	1.340
Hybrid-LB-f	0.866	3.194	1.674	0.814	2.572	1.245
Hybrid-LB-p	0.901	3.875	2.093	0.846	2.894	1.391
MF	0.893	3.093	0.871	0.846	2.539	0.707
PMF	0.930	3.774	1.094	0.888	3.017	0.857
$n = 100$	CVR	LEN	st.dev.	CVR	LEN	st.dev.
nonpara-f	0.895	3.197	1.270	0.843	2.521	0.909
nonpara-p	0.921	3.662	1.515	0.866	2.740	0.967
trans-forward	0.926	3.518	0.952	0.887	2.867	0.779
trans-backward	0.926	3.526	0.963	0.886	2.872	0.789
LB-forward	0.924	3.482	0.839	0.877	2.762	0.654
LB-backward	0.924	3.531	0.871	0.881	2.810	0.676
Hybrid-trans-f	0.885	3.149	1.232	0.830	2.499	0.890
Hybrid-trans-p	0.910	3.576	1.468	0.849	2.707	0.990
Hybrid-LB-f	0.888	3.153	1.201	0.835	2.514	0.906
Hybrid-LB-p	0.913	3.580	1.470	0.854	2.699	1.026
MF	0.912	3.177	0.724	0.864	2.559	0.555
PMF	0.943	3.880	0.892	0.899	2.950	0.605
$n = 200$	CVR	LEN	st.dev.	CVR	LEN	st.dev.
nonpara-f	0.905	3.028	0.955	0.851	2.395	0.747
nonpara-p	0.921	3.285	1.029	0.864	2.514	0.776
trans-forward	0.932	3.492	0.915	0.890	2.783	0.714
trans-backward	0.932	3.494	0.920	0.890	2.780	0.721

Table 8.4 Model 2: $Y_{t+1} = \sin(Y_t) + \sqrt{0.5 + 0.25Y_t^2}\varepsilon_{t+1}$ with Laplace innovations.

LB-forward	0.932	3.425	0.717	0.888	2.724	0.580
LB-backward	0.933	3.477	0.735	0.888	2.751	0.592
Hybrid-trans-f	0.894	2.998	1.006	0.837	2.386	0.768
Hybrid-trans-p	0.910	3.241	1.040	0.850	2.494	0.768
Hybrid-LB-f	0.897	3.006	0.961	0.841	2.397	0.765
Hybrid-LB-p	0.911	3.229	1.036	0.852	2.481	0.752
MF	0.926	3.224	0.648	0.878	2.559	0.463
PMF	0.945	3.716	0.846	0.898	2.812	0.562

Table 8.4 (continued)

densities, and the Local Bootstrap of Paparoditis and Politis (2001, 2002a). As in Sect. 8.7, the following models were chosen in order to generate Markov processes (of order $p = 1$).

- Model 1: $Y_{t+1} = \sin(Y_t) + \varepsilon_{t+1}$
- Model 2: $Y_{t+1} = \sin(Y_t) + \sqrt{0.5 + 0.25Y_t^2}\,\varepsilon_{t+1}$

where the errors $\{\varepsilon_t\}$ were i.i.d. $N(0,1)$ or Laplace rescaled to unit variance. Five hundred datasets were generated from each model with $n = 200$.

The main bandwidth for each method was chosen either by cross-validation or by the rule-of-thumb formula: $h = 0.9An^{-1/4}$ discussed in Remark 8.3.1 where $A = min(\hat{\sigma}, \frac{IQR}{1.34})$, $\hat{\sigma}$ is the estimated standard deviation of the data, and IQR is the sample interquartile range. The kernels K and Λ had a normal shape; the smoothing bandwidth for Λ in the basic Model-Free method was taken to be $h_0 = h^2$ as before.

Tables 8.5 and 8.6 show empirical CVRs of confidence intervals for $\mu(x)$ for different values of x using all the Model-Free Bootstrap methods, Local Bootstrap (LB) and Rajarshi's method (RAJ). Tables 8.7 and 8.8 show the average lengths of the confidence intervals of Tables 8.5 and 8.6, while Tables 8.9 and 8.10 show the respective standard deviations of interval lengths.

Some conclusions are as follows:

- Coverage levels are closer to the nominal when x is close to zero; this was to be expected since the point clouds of Fig. 8.1 are centered around zero. Consequently, the kernel smoothers work with a higher effective sample size when x is close to zero, leading to better approximations.
- Going from MF to PMF one obtains better coverage level but larger variability of interval length.
- The Local Bootstrap has better coverage levels than MF but not as good as the ones from PMF. The best coverage levels are associated with the PMF and RAJ methods, with LMF being a close second.
- In cases where the PMF and RAJ methods led to similar coverage levels, it is observed that the PMF intervals have smaller (average) interval length; this is highly desirable, and indicates that the intervals are centered more accurately.

- The two bandwidth choice methods, cross-validation and the rule-of-thumb formula, lead to comparable coverage levels; however, the latter is associated with significantly bigger interval length which is rather striking.

	cross-validation bandwidth					rule-of-thumb bandwidth				
Model 1										
x	LMF	MF	PMF	LB	RAJ	LMF	MF	PMF	LB	RAJ
$-2\pi/3$	0.868	0.854	0.870	0.862	0.884	0.872	0.860	0.892	0.876	0.880
$-\pi/2$	0.918	0.898	0.910	0.910	0.918	0.920	0.912	0.928	0.914	0.924
$-\pi/6$	0.928	0.908	0.928	0.922	0.936	0.912	0.900	0.932	0.924	0.936
0	0.942	0.926	0.932	0.938	0.948	0.916	0.898	0.920	0.926	0.926
$\pi/6$	0.926	0.904	0.912	0.920	0.932	0.912	0.888	0.924	0.904	0.924
$\pi/2$	0.878	0.868	0.882	0.870	0.892	0.880	0.858	0.898	0.878	0.892
$2\pi/3$	0.850	0.844	0.858	0.846	0.864	0.878	0.878	0.886	0.872	0.884
Model 2										
$-2\pi/3$	0.800	0.794	0.806	0.798	0.832	0.868	0.856	0.854	0.854	0.856
$-\pi/2$	0.894	0.876	0.896	0.890	0.908	0.896	0.900	0.908	0.904	0.906
$-\pi/6$	0.918	0.896	0.906	0.902	0.914	0.918	0.904	0.924	0.914	0.920
0	0.962	0.940	0.942	0.954	0.958	0.932	0.914	0.940	0.930	0.944
$\pi/6$	0.922	0.896	0.922	0.910	0.920	0.934	0.918	0.944	0.936	0.946
$\pi/2$	0.884	0.872	0.884	0.876	0.898	0.880	0.864	0.900	0.882	0.892
$2\pi/3$	0.838	0.806	0.836	0.824	0.850	0.858	0.858	0.874	0.862	0.856

Table 8.5 Empirical coverage level (CVR) of confidence intervals for $\mu(x)$ for different x values; simulation with normal errors, $n = 200$, and nominal coverage 95 %.

8.9 Discrete-Valued Markov Processes

Up till now in this chapter it has been assumed that Y_t and $X_t = (Y_t, Y_{t-1}, \cdots, Y_{t-p+1})'$ have densities with respect to Lebesgue measure, i.e., they represent continuous random variables. In this subsection, we will instead assume that the Markov(p) chain Y_t that takes values on a discrete set $S \subset \mathbf{R}$, i.e., $S = \{s_1, \ldots, s_d\}$. If d is finite, then a finite-state Markov chain ensues; otherwise, the state space is countable.

	cross-validation bandwidth					rule-of-thumb bandwidth				
Model 1										
x	LMF	MF	PMF	LB	RAJ	LMF	MF	PMF	LB	RAJ
$-2\pi/3$	0.860	0.848	0.870	0.870	0.884	0.874	0.862	0.894	0.874	0.866
$-\pi/2$	0.892	0.872	0.898	0.894	0.908	0.888	0.868	0.910	0.888	0.892
$-\pi/6$	0.928	0.900	0.914	0.924	0.928	0.918	0.874	0.934	0.908	0.920
0	0.940	0.914	0.934	0.926	0.932	0.928	0.898	0.938	0.934	0.934
$\pi/6$	0.946	0.932	0.952	0.942	0.956	0.936	0.904	0.952	0.934	0.938
$\pi/2$	0.892	0.864	0.904	0.876	0.900	0.896	0.868	0.910	0.892	0.882
$2\pi/3$	0.868	0.870	0.888	0.874	0.884	0.908	0.896	0.910	0.890	0.900
Model 2										
$-2\pi/3$	0.810	0.804	0.826	0.822	0.842	0.844	0.856	0.852	0.848	0.856
$-\pi/2$	0.866	0.842	0.880	0.832	0.844	0.872	0.860	0.898	0.866	0.870
$-\pi/6$	0.914	0.878	0.906	0.886	0.902	0.928	0.890	0.944	0.922	0.934
0	0.954	0.918	0.946	0.944	0.958	0.936	0.908	0.956	0.936	0.936
$\pi/6$	0.946	0.944	0.950	0.916	0.928	0.948	0.906	0.964	0.944	0.944
$\pi/2$	0.868	0.842	0.888	0.844	0.860	0.886	0.880	0.908	0.896	0.890
$2\pi/3$	0.830	0.826	0.848	0.830	0.840	0.870	0.876	0.884	0.868	0.872

Table 8.6 Empirical coverage level (CVR) of confidence intervals for $\mu(x)$ for different x values; simulation with Laplace errors, $n = 200$, and nominal coverage 95 %.

8.9.1 Transition Densities and Local Bootstrap

Definition (8.1) holds true *verbatim*; however, now the density notation $f(x)$, $f(y,x), f(y|x)$ will be reserved for the probability mass functions associated with $\mathbf{F}(x) = P[X_p \le x], \mathbf{F}(y,x) = P[Y_{p+1} \le y, X_p \le x]$, and $\mathbf{F}(y|x) = P[Y_{p+1} \le y | X_p = x]$, respectively. As such, they can be estimated by counting as opposed to smoothing. In other words, Eqs. (8.2)—(8.4) are replaced by

$$\hat{f}_n(y,x) = \frac{1}{(n-p)} \sum_{i=p+1}^{n} \mathbf{1}\{y_i = y,\ x_{i-1} = x\} \tag{8.27}$$

$$\hat{f}_n(x) = \sum_{y \in S} \hat{f}_n(y,x) \tag{8.28}$$

$$\hat{f}_n(y|x) = \frac{\hat{f}_n(y,x)}{\hat{f}_n(x)} \tag{8.29}$$

for all $y \in S$ and $x \in S^p$.

	cross-validation bandwidth					rule-of-thumb bandwidth				
Model 1										
x	LMF	MF	PMF	LB	RAJ	LMF	MF	PMF	LB	RAJ
$-2\pi/3$	0.843	0.829	0.861	0.850	0.887	0.995	0.979	1.020	0.993	1.013
$-\pi/2$	0.661	0.637	0.675	0.667	0.703	0.756	0.727	0.783	0.752	0.773
$-\pi/6$	0.523	0.501	0.537	0.527	0.565	0.564	0.537	0.589	0.562	0.588
0	0.512	0.491	0.527	0.516	0.550	0.547	0.521	0.573	0.548	0.570
$\pi/6$	0.521	0.501	0.536	0.525	0.561	0.560	0.534	0.589	0.559	0.582
$\pi/2$	0.659	0.639	0.675	0.664	0.702	0.752	0.724	0.785	0.750	0.774
$2\pi/3$	0.837	0.818	0.860	0.839	0.877	0.976	0.957	1.014	0.974	0.997
Model 2										
$-2\pi/3$	1.060	1.038	1.053	1.075	1.093	1.542	1.545	1.523	1.514	1.493
$-\pi/2$	0.685	0.664	0.692	0.709	0.742	0.907	0.886	0.935	0.904	0.908
$-\pi/6$	0.411	0.395	0.425	0.418	0.466	0.443	0.419	0.480	0.444	0.468
0	0.386	0.371	0.400	0.390	0.439	0.403	0.382	0.440	0.402	0.428
$\pi/6$	0.410	0.394	0.425	0.415	0.462	0.445	0.419	0.484	0.443	0.468
$\pi/2$	0.678	0.657	0.692	0.695	0.731	0.900	0.871	0.935	0.894	0.902
$2\pi/3$	1.018	1.004	1.036	1.050	1.057	1.477	1.483	1.497	1.469	1.450

Table 8.7 Average length of confidence intervals of Table 8.5.

The discrete-valued Forward bootstrap based on transition densities is very similar to Algorithm 8.3.1 replacing Eqs. (8.2)—(8.4) with Eqs. (8.27)—(8.29). Its favorable properties for the purpose of confidence interval construction were shown by Raïs (1994); the application to predictive distributions is new, and goes along the lines of Algorithm 2.4.4. Since we cannot use predictive roots, we attempt to capture the estimation variability by generating the bootstrap future value $y^{**}_{n+1} \sim \hat{f}^*_n(\cdot|x_n)$ instead of $\hat{f}_n(\cdot|x_n)$. Due to the caveats discussed in Sect. 2.4.4, we do not attempt the construction of prediction intervals here.

Algorithm 8.9.1 DISCRETE-VALUED FORWARD BOOTSTRAP BASED ON TRANSITION DENSITY

1. Compute the estimators $\hat{f}(y,x), \hat{f}(x), \hat{f}(y|x)$ *from Eqs.* (8.27)—(8.29).

2.(a) Generate $x^*_p = (y^*_p, \cdots, y^*_1)$ *with probability density function* $\hat{f}_n(\cdot)$ *given by Eq.* (8.28)*; alternatively, let* $x^*_p = (y^*_{p+J}, \cdots, y_{1+J})$ *where* J *is generated as a discrete random variable uniform on the values* $0, 1, \ldots, n-p$.

(b) Generate $y^*_{p+1} \sim \hat{f}_n(\cdot|x^*_p)$ *given by* (8.29).

	cross-validation bandwidth					rule-of-thumb bandwidth				
Model 1										
x	LMF	MF	PMF	LB	RAJ	LMF	MF	PMF	LB	RAJ
$-2\pi/3$	0.893	0.862	0.927	0.932	0.945	1.211	1.186	1.307	1.195	1.192
$-\pi/2$	0.671	0.640	0.699	0.697	0.725	0.864	0.819	0.942	0.860	0.863
$-\pi/6$	0.497	0.470	0.513	0.510	0.546	0.558	0.517	0.612	0.554	0.577
0	0.485	0.460	0.501	0.495	0.526	0.541	0.502	0.600	0.541	0.557
$\pi/6$	0.496	0.470	0.513	0.505	0.542	0.559	0.517	0.620	0.557	0.580
$\pi/2$	0.673	0.637	0.700	0.699	0.726	0.857	0.817	0.949	0.855	0.859
$2\pi/3$	0.900	0.867	0.939	0.937	0.947	1.182	1.165	1.298	1.174	1.170
Model 2										
$-2\pi/3$	1.159	1.116	1.152	1.164	1.163	1.792	1.795	1.817	1.753	1.725
$-\pi/2$	0.730	0.688	0.736	0.746	0.772	1.067	1.035	1.146	1.050	1.035
$-\pi/6$	0.405	0.382	0.423	0.407	0.455	0.447	0.413	0.514	0.444	0.464
0	0.382	0.360	0.400	0.380	0.430	0.405	0.372	0.474	0.403	0.423
$\pi/6$	0.409	0.385	0.430	0.409	0.458	0.451	0.415	0.527	0.449	0.468
$\pi/2$	0.719	0.681	0.741	0.733	0.757	1.028	1.004	1.125	1.014	0.998
$2\pi/3$	1.139	1.109	1.158	1.181	1.168	1.751	1.782	1.819	1.725	1.698

Table 8.8 Average length of confidence intervals of Table 8.6.

(c) *Repeat (b) to generate* $y^*_{t+1} \sim \hat{f}_n(\cdot|x^*_t)$ *for* $t = p, \cdots, n-1$, *where as before* $x^*_t = (y^*_t, \cdots, y^*_{t-p+1})'$.

(d) *Construct* $\hat{f}^*_n(y|x)$ *in a similar way as in (8.29) but based on the pseudo-data* $y^*_1, y^*_2, \cdots, y^*_n$ *instead of the original data.*

(e) *Generate the bootstrap future value* $y^{**}_{n+1} \sim \hat{f}^*_n(\cdot|x_n)$.

3. *Repeat step 2 above B times; the B bootstrap replicates of the bootstrap future value* y^{**}_{n+1} *are collected in the form of an empirical distribution which is our estimate of the predictive distribution for* Y_{n+1}. *The* mode *of this predictive distribution can be used as a point predictor for* Y_{n+1}.

The discrete-valued Backward bootstrap algorithm is similar to Algorithm 8.9.1 with one exception: the pseudo-data $y^*_1, y^*_2, \cdots, y^*_n$ are generated in a backwards fashion as in Algorithm 8.3.2 but based on a backwards transition density estimated by counting as opposed to smoothing.

	cross-validation bandwidth					rule-of-thumb bandwidth				
Model 1										
x	LMF	MF	PMF	LB	RAJ	LMF	MF	PMF	LB	RAJ
$-2\pi/3$	0.233	0.232	0.239	0.266	0.247	0.249	0.248	0.234	0.249	0.227
$-\pi/2$	0.144	0.134	0.150	0.170	0.152	0.145	0.141	0.144	0.141	0.133
$-\pi/6$	0.079	0.073	0.089	0.090	0.084	0.069	0.066	0.068	0.067	0.067
0	0.071	0.066	0.086	0.079	0.070	0.058	0.057	0.057	0.059	0.055
$\pi/6$	0.079	0.075	0.092	0.101	0.089	0.065	0.063	0.067	0.063	0.063
$\pi/2$	0.161	0.156	0.175	0.170	0.161	0.154	0.149	0.153	0.153	0.143
$2\pi/3$	0.272	0.274	0.288	0.268	0.250	0.230	0.227	0.227	0.226	0.212
Model 2										
$-2\pi/3$	0.374	0.377	0.364	0.441	0.437	0.493	0.511	0.452	0.478	0.433
$-\pi/2$	0.188	0.186	0.193	0.276	0.252	0.222	0.220	0.220	0.223	0.200
$-\pi/6$	0.064	0.058	0.079	0.091	0.087	0.057	0.056	0.064	0.057	0.056
0	0.053	0.050	0.066	0.066	0.061	0.045	0.044	0.046	0.044	0.045
$\pi/6$	0.058	0.056	0.070	0.072	0.060	0.057	0.056	0.061	0.058	0.056
$\pi/2$	0.189	0.189	0.203	0.242	0.227	0.218	0.218	0.213	0.223	0.199
$2\pi/3$	0.310	0.330	0.340	0.386	0.351	0.431	0.446	0.409	0.435	0.396

Table 8.9 Standard deviation of interval length from Table 8.5.

Remark 8.9.1 (Discrete-valued Local Bootstrap) Recall that the only difference between bootstrap based on estimates of transition density and the Local Bootstrap was that the latter used a step function estimator of the transition distribution (similar to $\hat{D}_x(\cdot)$) whereas the former used a smooth (differentiable) function estimator (similar to $\bar{D}_x(\cdot)$). But in the case of discrete-valued Markov process, smoothing is not recommended; even Algorithm 8.9.1 uses probability mass functions. Hence, in the case of discrete-valued data, the Local Bootstrap is *identical* to the bootstrap based on transition density given in Algorithm 8.9.1.

8.9.2 Model-Free Bootstrap

As alluded to in Remark 8.9.1, here too it is sufficient to avoid the smoothing step. In the language of Sect. 8.6, this just means that we shun the smooth estimator $\bar{D}_x(y)$, and focus instead on estimator $\hat{D}_x(y)$ from Eq. (8.20) using a bandwidth h that is extremely close to zero, i.e., define

	cross-validation bandwidth					rule-of-thumb bandwidth				
Model 1										
x	LMF	MF	PMF	LB	RAJ	LMF	MF	PMF	LB	RAJ
$-2\pi/3$	0.256	0.250	0.255	0.335	0.304	0.460	0.454	0.448	0.454	0.405
$-\pi/2$	0.166	0.154	0.171	0.191	0.168	0.233	0.222	0.237	0.234	0.214
$-\pi/6$	0.077	0.068	0.083	0.098	0.091	0.095	0.086	0.100	0.090	0.087
0	0.062	0.056	0.068	0.080	0.069	0.074	0.068	0.085	0.076	0.071
$\pi/6$	0.073	0.069	0.083	0.083	0.076	0.093	0.088	0.104	0.092	0.088
$\pi/2$	0.188	1.173	0.190	0.227	0.203	0.259	0.255	0.273	0.261	0.244
$2\pi/3$	0.305	0.318	0.301	0.325	0.291	0.434	0.457	0.465	0.447	0.401
Model 2										
$-2\pi/3$	0.521	0.503	0.491	0.541	0.503	0.744	0.758	0.714	0.733	0.689
$-\pi/2$	0.314	0.260	0.266	0.320	0.291	0.386	0.381	0.396	0.375	0.332
$-\pi/6$	0.076	0.060	0.077	0.077	0.070	0.081	0.076	0.101	0.080	0.075
0	0.078	0.049	0.062	0.061	0.062	0.059	0.055	0.084	0.058	0.056
$\pi/6$	0.074	0.065	0.086	0.077	0.069	0.083	0.079	0.102	0.084	0.076
$\pi/2$	0.307	0.251	0.264	0.295	0.249	0.380	0.397	0.388	0.373	0.335
$2\pi/3$	0.522	0.502	0.484	0.615	0.546	0.747	0.786	0.735	0.769	0.701

Table 8.10 Standard deviation of interval length of Table 8.6.

$$\hat{D}_x(y) = \frac{1}{n-p} \sum_{i=p+1}^{n} \mathbf{1}\{y_i \le y,\ x_i = x\}. \tag{8.30}$$

Note that the transformed, i.e., "uniformized," variables v_t can no longer be calculated because the probability integral transform does not work for discrete data. Nevertheless, the Limit Model-Free Algorithm from Sect. 2.4.3 comes to our rescue, coupled with the ideas discussed in Sect. 2.4.4. The discrete-valued (forward) Limit Model-free Bootstrap for Markov processes goes as follows.

Algorithm 8.9.2 DISCRETE-VALUED LIMIT MODEL-FREE (LMF) BOOTSTRAP

1. *Compute* $\hat{D}_x(y)$ *from Eq.* (8.30).
2.(a) *For some large positive integer* M, *generate* $v^*_{-M}, v^*_{-M+1}, \cdots, v^*_0, v^*_1, \cdots, v^*_{n-1}$, v^*_n *and* v^*_{n+1} *as i.i.d. Uniform (0,1).*

(b) *Let* $(y^*_{-M}, \cdots, y^*_{-M+p-1})' = (y_{1+I}, \cdots, y_{p+I})'$ *where* I *is generated as a discrete random variable uniform on the values* $0, 1, \ldots, n-p$; *let* $x^*_{-M+p-1} = (y^*_{-M+p-1}, \ldots, y^*_{-M})$.

(c) *Generate* $y_t^* = \hat{D}^{-1}_{x^*_{t-1}}(v_t^*)$ *for* $t = -M+p, \cdots, n$.

(d) *Re-compute the estimator* $\hat{D}^*_x(y)$ *from Eq.* (8.30) *applied to the bootstrap pseudo-data* $y_1^*, \ldots, y_n^*$.

(e) *Calculate the bootstrap future value* $y^{**}_{n+1} = \hat{D}^{*-1}_{x_n}(v^*_{n+1})$.

3. *Repeat step 2 above B times; the B bootstrap future values* y^{**}_{n+1} *are collected in the form of an empirical distribution which is our estimate of the predictive distribution for* Y_{n+1}. *The mode of this predictive distribution can be used as a point predictor for* Y_{n+1}.

Remark 8.9.2 All the bootstrap algorithms that were developed in Sect. 8.8 for the construction of confidence intervals apply *verbatim* in the case of $\{Y_t\}$ being a discrete-valued Markov process. The only thing that is different, is that the generation of bootstrap pseudo-data $y_1^*, \ldots, y_n^*$ must follow the directions of the present section, i.e., must be done based on transition probabilities estimated by counting (as opposed to smoothing), and/or on the distribution estimator $\hat{D}_x(y)$ as given in Eq. (8.30).

Acknowledgements

Chapter 8 is based on the working paper: L. Pan and D.N. Politis, "Bootstrap prediction intervals for Markov processes," Dept. of Economics, UCSD, retrievable from: http://escholarship.org/uc/item/7555757g; the paper has been accepted to appear in the *CSDA Annals of Computational and Financial Econometrics* in 2015. Many thanks are due to CSDA Editor, E.J. Kontoghiorghes, and to Li Pan for compiling the software and running extensive simulations for the paper and the chapter. Additional simulations on the performance of Model-free confidence intervals can be found in: L. Pan and D.N. Politis, "Model-Free Bootstrap for Markov processes," in *Proceedings of the 60th World Statistics Congress–ISI2015*, Rio de Janeiro, Brazil, July 26–31, 2015.

Chapter 9
Predictive Inference for Locally Stationary Time Series

9.1 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 cannot 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, 1988) and Dahlhaus (1997, 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 and Davis (1991). Having done that, the "classical" decomposition simplifies to the following model with additive trend, i.e.,

$$Y_t = \mu(t) + W_t \tag{9.1}$$

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

$$Y_t = \mu(t) + \sigma(t) W_t. \tag{9.2}$$

D.N. Politis, *Model-Free Prediction and Regression*, Frontiers in Probability and the Statistical Sciences, DOI 10.1007/978-3-319-21347-7_9

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 (9.2), it is also assumed that $\mathrm{Var}(W_t) = 1$. As usual, the deterministic functions $\mu(\cdot)$ and $\sigma(\cdot)$ are unknown but assumed to belong to a class of functions that is either finite-dimensional (parametric) or not; 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 9.1.1 (Quantifying smoothness) To analyze locally stationary series it is sometimes useful to map the index set $\{1,\ldots,n\}$ onto the interval $[0,1]$. In that respect, consider two functions $\mu_{[0,1]} : [0,1] \mapsto \mathbf{R}$ and $\sigma_{[0,1]} : [0,1] \mapsto (0,\infty)$, and let

$$\mu(t) = \mu_{[0,1]}(a_t) \quad \text{and} \quad \sigma(t) = \sigma_{[0,1]}(a_t) \tag{9.3}$$

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 Sect. 9.2.2 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 (9.1) and (9.2) are considered general and flexible—especially when $\mu(\cdot)$ and $\sigma(\cdot)$ are not parametrically specified—and have been studied extensively; see, e.g., Zhang and Wu (2011), and Zhou and Wu (2009, 2010). However, it may be that the skewness and/or kurtosis of Y_t changes with t, in which case centering and studentization alone cannot render the problem stationary. To see why, note that under model (9.2), $EY_t = \mu(t)$ and $\mathrm{Var}\, Y_t = \sigma^2(t)$; hence,

$$W_t = \frac{Y_t - \mu(t)}{\sigma(t)} \tag{9.4}$$

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 $\mathrm{Corr}(Y_t, Y_{t+m})$ changes smoothly (and slowly) with t. Notably, models (9.1) and (9.2) 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 and prediction of nonstationary time series that does not rely on simple additive models[1] such as (9.1) and (9.2). Fortunately, the Model-free Prediction Principle gives us the tools 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 here is to be able to construct an

[1] An alternative approach to prediction that does not rely on models such as (9.2) is given using wavelet representations of locally stationary processes; see, e.g., Fryzlewicz et al. (2003), and Antoniadis et al. (2006).

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 Sect. 9.3. 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 and Model-free, are novel in the prediction literature.

9.2 Model-Based Inference

Throughout Sect. 9.2, we will assume model (9.2)—that includes model (9.1) as a special case—together with a nonparametric assumption on smoothness of $\mu(\cdot)$ and $\sigma(\cdot)$ as described in Remark 9.1.1.

9.2.1 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 (9.2), we have

$$E(Y_{n+1}|\underline{Y}_n) = \mu(n+1) + \sigma(n+1)E(W_{n+1}|\underline{Y}_n). \tag{9.5}$$

For $j < J$, define $\mathscr{F}_j^J(Y)$ to be the *information set* $\{Y_j, Y_{j+1}, \ldots, Y_J\}$, also known as σ–field, and note that the information sets $\mathscr{F}_{-\infty}^t(Y)$ and $\mathscr{F}_{-\infty}^t(W)$ 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.,

$$E(W_{n+1}|\underline{Y}_n) \simeq E(W_{n+1}|Y_s, s \le n) = E(W_{n+1}|W_s, s \le n) \simeq E(W_{n+1}|\underline{W}_n) \tag{9.6}$$

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 Chap. 8, or approximating $E(W_{n+1}|\underline{W}_n)$ by a linear function of $\underline{W}_n$ as in Chap. 6, i.e., contend ourselves with the best linear predictor of W_{n+1} denoted by $\bar{E}(W_{n+1}|\underline{W}_n)$. Taking the latter approach, the L_2–optimal *linear* predictor of W_{n+1} based on $\underline{W}_n$ is

$$\bar{E}(W_{n+1}|\underline{W}_n) = \phi_1(n)W_n + \phi_2(n)W_{n-1} + \ldots + \phi_n(n)W_1, \tag{9.7}$$

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 Chap. 6. Alternatively, the L_2–optimal linear predictor of W_{n+1} can be

obtained by fitting a (causal) AR(p) model to the data $W_1, \ldots, W_n$ with p chosen by minimizing AIC or a related criterion; this would entail fitting the model:

$$W_t = \phi_1 W_{t-1} + \phi_2 W_{t-2} + \cdots + \phi_p W_{t-p} + V_t \tag{9.8}$$

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

$$\bar{E}(W_{n+1}|\underline{W}_n) = \phi_1 W_n + \phi_2 W_{n-1} + \cdots + \phi_p W_{n-p+1}. \tag{9.9}$$

As discussed in Sect. 6.6.3, the two methods for constructing $\bar{E}(W_{n+1}|\underline{W}_n)$ are closely related; in fact, predictor (9.7) coincides with the above AR–type predictor if the matrix Γ_n is the one implied by the fitted AR(p) model (9.8). We will use the AR–type predictor in the sequel because it additionally affords us the possibility of resampling based on model (9.8).

9.2.2 Trend Estimation and Practical Prediction

To construct the L_2–optimal predictor (9.5), 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 and Vieu (1992), Kim and Cox (1996), or Li and 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 and 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 9.1.1 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; see Fan and Gijbels (1996), Fan and Yao (2003), or Li and Racine (2007).

Remark 9.2.1 (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 (a) in predictor (9.5), the estimands involve $\mu_{[0,1]}(1)$ and $\sigma_{[0,1]}(1)$ as just mentioned, and (b) to compute $\bar{E}(W_{n+1}|\underline{W}_n)$ in Eq. (9.7) 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. (9.4) 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 $\hat{W}_1, \ldots, \hat{W}_n$ cannot 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.

Continuing the analogy to model-based regression studied in Chap. 3, 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 new bandwidth parameter b will be assumed to satisfy

$$b \to \infty \text{ as } n \to \infty \text{ but } b/n \to 0, \tag{9.10}$$

i.e., b is analogous to the product hn in the regression example of Chap. 3. 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

$$\hat{\mu}(t) = \sum_{i=1}^{t} Y_i \, \hat{K}\left(\frac{t-t_i}{b}\right) \text{ and } \hat{M}(t) = \sum_{i=1}^{t} Y_i^2 \, \hat{K}(\frac{t-t_i}{b}) \tag{9.11}$$

   where

$$\hat{\sigma}(t) = \sqrt{\hat{M}_t - \hat{\mu}(t)^2} \text{ and } \hat{K}\left(\frac{t-t_i}{b}\right) = \frac{K(\frac{t-t_i}{b})}{\sum_{k=1}^{t} K(\frac{t-t_k}{b})}. \tag{9.12}$$

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

$$\hat{W}_t = \frac{Y_t - \hat{\mu}(t)}{\hat{\sigma}(t)} \text{ for } t = b+1, \ldots, n. \tag{9.13}$$

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

$$\tilde{\mu}(t) = \sum_{i=1}^{t-1} Y_i \, \tilde{K}\left(\frac{t-t_i}{b}\right) \text{ and } \tilde{M}(t) = \sum_{i=1}^{t-1} Y_i^2 \, \tilde{K}(\frac{t-t_i}{b}) \tag{9.14}$$

   where

$$\tilde{\sigma}(t) = \sqrt{\tilde{M}_t - \tilde{\mu}(t)^2} \text{ and } \tilde{K}\left(\frac{t-t_i}{b}\right) = \frac{K(\frac{t-t_i}{b})}{\sum_{k=1}^{t-1} K(\frac{t-t_k}{b})}. \tag{9.15}$$

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

$$\tilde{W}_t = \frac{Y_t - \tilde{\mu}(t)}{\tilde{\sigma}(t)} \text{ for } t = b+1, \ldots, n. \tag{9.16}$$

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

$$\hat{\mu}(t) = \frac{\sum_{j=1}^{t} w_j Y_j}{\sum_{j=1}^{t} w_j + n^{-2}} \text{ and } \hat{M}(t) = \frac{\sum_{j=1}^{t} w_j Y_j^2}{\sum_{j=1}^{t} w_j + n^{-2}} \tag{9.17}$$

where

$$w_j = K(\frac{t-t_j}{b})\,[s_{t,2} - (t-t_j)s_{t,1}]\,, \tag{9.18}$$

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. (9.17) is just to ensure the denominator is not zero; see Fan (1993). Equation (9.12) then yields $\hat{\sigma}(t)$, and Eq. (9.13) yields $\hat{W}_t$.

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

$$\tilde{\mu}(t) = \frac{\sum_{j=1}^{t-1} w_j Y_j}{\sum_{j=1}^{t-1} w_j + n^{-2}} \quad \text{and} \quad \tilde{M}(t) = \frac{\sum_{j=1}^{t-1} w_j Y_j^2}{\sum_{j=1}^{t-1} w_j + n^{-2}} \tag{9.19}$$

where

$$w_j = K(\frac{t-t_j}{b})\,[s_{t-1,2} - (t-t_j)s_{t-1,1}]\,. \tag{9.20}$$

Equation (9.15) then yields $\tilde{\sigma}(t)$, and Eq. (9.16) 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 (9.5). 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 Sect. 9.2.1.

Remark 9.2.2 (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 $t \le n$—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 $t \le n$ 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—see Chap. 3. Therefore, our preference is to use the predictive quantities $\tilde{\mu}(t)$, $\tilde{\sigma}(t)$, and $\tilde{W}_t$ throughout the predictive modeling.

Remark 9.2.3 (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 (suboptimal) predictor $\hat{Y}_{k+1} \simeq \hat{\mu}(k+1)$ and base $PRESS(b)$ or $PRESAR(b)$ on this approximation.

9.2.3 Model-Based Predictors and Prediction Intervals

To go from point prediction to prediction intervals, some form of resampling is required. Since model (9.2) 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 (9.2). To generate a stationary bootstrap pseudo-series $\{W_t^*\}$, two popular time series resampling methods are (a) the stationary bootstrap of Politis and Romano (1994), and (b) the AR bootstrap which entails treating the V_t appearing in Eq. (9.8) as if they were i.i.d., performing an i.i.d. bootstrap on them, and then generating $\{W_t^*\}$ via the recursion (9.8) 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 Sect. 9.2.1, and it is more amenable to the construction of prediction intervals; see, e.g., Chap. 7. In addition, Kreiss et al. (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. (9.8) are not truly 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 Sect. 7.3 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 $t = 1, \ldots, n$. Then, our model-based approximation to the L_2–optimal point predictor of Y_{n+1} is

$$\Pi = \check{\mu}(n+1) + \check{\sigma}(n+1) \left[\hat{\phi}_1 \check{W}_n + \cdots + \hat{\phi}_p \check{W}_{n-p+1} \right] \tag{9.21}$$

where $\hat{\phi}_1, \ldots, \hat{\phi}_p$ are the Yule-Walker estimators of $\phi_1, \ldots, \phi_p$ appearing in Eq. (9.8). As in Chap. 2, 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 9.2.1 MODEL-BASED 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" $\check{W}_{b+1},\ldots,\check{W}_n$ using model (9.2).*
2. *Fit the AR(p) model (9.8) to the series $\check{W}_{b+1},\ldots,\check{W}_n$ (with p selected by AIC minimization), and obtain the Yule-Walker estimators $\hat{\phi}_1,\ldots,\hat{\phi}_p$, and the error proxies*

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

3. a. *Let $\check{V}^*_t$ for $t = 1,\ldots,n,n+1$ be drawn randomly with replacement from the set $\{\check{\check{V}}_t \text{ for } t = p+b+1,\ldots,n\}$ 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*

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

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

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

   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"[2] $\check{W}^*_{b+1},\ldots,\check{W}^*_n$ 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*

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

   [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—see Chap. 7.]

   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 $Y^*_{n+1} - \Pi^*$.*

[2] For simplicity, we assume that the bootstrap estimators $\check{\mu}^*(\cdot)$ and $\check{\sigma}^*(\cdot)$ are based on the same window width b used in the real world.

4. *Steps (a)—(f) in the above are repeated a large number of times (say B times), and the B bootstrap root replicates are collected in the form of an empirical distribution whose α–quantile is denoted by $q(\alpha)$.*
5. *Finally, a $(1-\alpha)100\%$ equal-tailed prediction interval for Y_{n+1} is given by*

$$[\Pi + q(\alpha/2),\ \Pi + q(1-\alpha/2)]. \tag{9.22}$$

It is easy to see that prediction interval (9.22) is asymptotically valid (conditionally on $Y_1, \ldots, Y_n$) provided: (i) estimators $\breve{\mu}(n+1)$ and $\breve{\sigma}(n+1)$ 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 $\breve{\mu}(\cdot)$ and $\breve{\sigma}(\cdot)$ correspond to one of the abovementioned four methods (NW vs. LL, regular vs. predictive), then provision (i) is satisfied under standard conditions including the bandwidth condition (9.10). 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 $\breve{\mu}(\cdot)$, $\breve{\sigma}(\cdot)$ and $\hat{\phi}_1, \hat{\phi}_2, \ldots$. Since this finite-sample variability vanishes asymptotically, the performance of a prediction interval such as (9.22) must be gauged by finite-sample simulations.

9.3 Model-Free Inference

Model (9.2) is a general 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. For some $m \geq 1$, let $\mathscr{L}(Y_t, Y_{t-1}, \ldots, Y_{t-m+1})$ denote 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 (9.2) in what follows, we still want to employ nonparametric smoothing for estimation; thus, we must assume that $\mathscr{L}(Y_t, Y_{t-1}, \ldots, Y_{t-m+1})$ changes smoothly (and slowly) with t.

Remark 9.3.1 (Quantifying smoothness–model-free case) As in Remark 9.1.1, 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

$$D_t^{(m)}(\underline{s}) = D_{a_t}^{[0,1]}(\underline{s}) \text{ for } t = 1, \ldots, n. \tag{9.23}$$

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 9.1.1, 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 $\mathscr{L}(Y_t, Y_{t-1}, \ldots, Y_{t-m+1})$ is to assume that

$$(Y_t, Y_{t-1}, \ldots, Y_{t-m+1})' \stackrel{\mathscr{L}}{=} \mathbf{f}_t(W_t, W_{t-1}, \ldots, W_{t-m+1}) \tag{9.24}$$

for some function $\mathbf{f}_t(w)$ that is smooth in both arguments t and w, and some strictly stationary and weakly dependent, e.g., strong mixing, univariate time series $\{W_t\}$. In the above, the symbol $\stackrel{\mathscr{L}}{=}$ denotes equality in distribution, i.e., the left-hand side of Eq. (9.24) has the same probability law as the right-hand side. Note that model (9.2) is a special case of Eq. (9.24) with $m = 1$, the function $\mathbf{f}_t(w)$ being affine/linear in w, and $\stackrel{\mathscr{L}}{=}$ replaced by usual equality. Thus, for concreteness and easy comparison with the model-based case of Eq. (9.2), we will focus in the sequel on the case $m = 1$; Sect. 9.3.7 discusses how to handle the case $m > 1$.

9.3.1 Constructing the Theoretical Transformation

Hereafter, adopt the setup of Eq. (9.24) with $m = 1$, and let

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

denote the first marginal distribution of time series $\{Y_t\}$. Throughout Sect. 9.3, the default assumption will be that $D_t(y)$ is (absolutely) continuous in y for all t. We now define new variables via the probability integral transform, i.e., let

$$U_t = D_t(Y_t) \text{ for } t = 1, \ldots, n; \tag{9.25}$$

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 Chap. 2. Letting Φ denote the cumulative distribution function (cdf) of the standard normal distribution, we define

$$Z_t = \Phi^{-1}(U_t) \text{ for } t = 1, \ldots, n; \tag{9.26}$$

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,[3] 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:

$$\underline{\varepsilon}_n = C_n^{-1} \underline{Z}_n. \tag{9.27}$$

It follows that the entries of $\underline{\varepsilon}_n = (\varepsilon_1, \ldots, \varepsilon_n)'$ are uncorrelated standard normal. Assuming that the random variables $Z_1, \ldots, Z_n$ are *jointly* normal,[4] this can be further strengthened 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. (9.25), (9.26) and (9.27), are invertible; hence, the composite transformation $H_n : \underline{Y}_n \mapsto \underline{\varepsilon}_n$ is invertible as well.

9.3.2 Kernel Estimation of the "Uniformizing" Transformation

We first focus on estimating the "uniformizing" part of the transformation, i.e., Eq. (9.25). 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:

$$\hat{D}_t(y) = \sum_{i=1}^{T} \mathbf{1}\{Y_{t_i} \le y\} \tilde{K}\left(\frac{t - t_i}{b}\right) \tag{9.28}$$

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 (9.28) is *one-sided* for the same reasons discussed in Remark 9.2.1. Since $\hat{D}_t(y)$ is a step function in y, a smooth estimator can be defined as:

$$\bar{D}_t(y) = \sum_{i=1}^{T} \Lambda\left(\frac{y - Y_{t_i}}{h_0}\right) \tilde{K}\left(\frac{t - t_i}{b}\right) \tag{9.29}$$

[3] 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.

[4] The joint normality of $Z_1, \ldots, Z_n$ follows immediately if one assumes that the stationary process $\{W_t\}$ appearing in Eq. (9.24) is Gaussian; but even without this additional assumption, it is difficult to construct examples where the joint normality of the Z_ts may break down.

where h_0 is a secondary bandwidth. Furthermore, as in Sect. 9.2.2, 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 $\bar{D}_t(y)$.

Remark 9.3.2 (On choice of the two bandwidths) To choose the main bandwidth b for either $\hat{D}_t(y)$ or $\bar{D}_t(y)$, predictive cross-validation may be used but it must be adapted to the time series prediction setting, i.e., as in Remark 9.2.3. Define $h = b/n$, and recall that in the analogous regression problem in Chap. 4 the optimal rates $h_0 \sim n^{-2/5}$ and $h \sim n^{-1/5}$ were suggested in connection with the nonnegative kernel K; this led to the practical recommendation of letting $h_0 = h^2$ with h chosen by cross-validation. Similarly here, the recommendation is to choose b by the time series cross-validation of Remark 9.2.3, and then let $h_0 = b^2/n^2$.

9.3.3 *Local Linear Estimation of the "Uniformizing" Transformation*

Note that the kernel estimator $\hat{D}_t(y)$ defined in Eq. (9.28) 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. (9.29) 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 $t = n+1$. Let $\tilde{D}_t(y)$ denote the local linear estimator of $D_t(y)$ based on either the indicator variables $\mathbf{1}\{Y_i \le y\}$ or the smoothed variables $\Lambda(\frac{y-Y_i}{h_0})$. Keeping y fixed, $\tilde{D}_t(y)$ has good behavior, e.g., smaller bias than either $\hat{D}_t(y)$ or $\bar{D}_t(y)$. However, there is no guarantee that $\tilde{D}_t(y)$ is a proper distribution function 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 and 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 rather than our setting where Y_t is regressed on t. Coupled with the fact that we are interested in the boundary case, the equation yielding the adjusted Nadaraya-Watson weights does not admit a solution. Hansen (2004) has proposed a different, 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:

$$\hat{D}_t^{LL}(y) = \frac{\sum_{i=1}^T w_i^\diamond \mathbf{1}(Y_i \le y)}{\sum_{i=1}^T w_i^\diamond} \quad \text{and} \quad \bar{D}_t^{LL}(y) = \frac{\sum_{i=1}^T w_i^\diamond \Lambda(\frac{y-Y_i}{h_0})}{\sum_{i=1}^T w_i^\diamond}. \tag{9.30}$$

The weights $w_i^\diamond$ are defined by

$$w_i^\diamond = \begin{cases} 0 & \text{when } \hat{\beta}(t-t_i) > 1 \\ w_i(1-\hat{\beta}(t-t_i)) & \text{when } \hat{\beta}(t-t_i) \le 1 \end{cases} \tag{9.31}$$

where

$$w_i = \frac{1}{b} K(\frac{t-t_i}{b}) \text{ and } \hat{\beta} = \frac{\sum_{i=1}^{T} w_i(t-t_i)}{\sum_{i=1}^{T} w_i(t-t_i)^2}. \quad (9.32)$$

As with Eqs. (9.28) and (9.29), we can let $T = t$ or $T = t-1$ in the above, leading to a **fitted vs. predictive** local linear estimators of $D_t(y)$, smoothed or unsmoothed.

9.3.4 Estimation of the Whitening Transformation

To implement the whitening transformation (9.27), it is necessary to estimate Γ_n, i.e., the $n \times n$ covariance matrix of the random vector $\underline{Z}_n = (Z_1, \ldots, Z_n)'$ where the Z_t are the normal random variables defined in Eq. (9.26). As discussed in the analogous problem in Sect. 9.2.1, 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 autocovariances defined as $\breve{\gamma}_k = n^{-1} \sum_{t=1}^{n-|k|} Z_t Z_{t+|k|}$ for $|k| < n$.

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 $k = 0, 1, \ldots, p-1$. For $k \geq p$, $\hat{\gamma}_k^{AR}$ 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 and 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. Our final estimator of Γ_n will be $\hat{\Gamma}_n^{\star}$ which is 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 Sect. 6.4.

Estimating the "uniformizing" transformation $D_t(\cdot)$ and the whitening transformation 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 $\hat{\Gamma}_{n+1}^{AR}$ be the $(n+1) \times (n+1)$ covariance matrix associated with the fitted AR(p) model.
B′. Denote by $\hat{\gamma}_{|i-j|}^{\star}$ the i, j element of $\hat{\Gamma}_n^{\star}$ for $i, j = 1, \ldots, n$. Then, define $\hat{\Gamma}_{n+1}^{\star}$ to be the symmetric, banded Toeplitz $(n+1) \times (n+1)$ matrix with ij element given by $\hat{\gamma}_{|i-j|}^{\star}$ when $|i-j| < n$. Recall that $\hat{\Gamma}_n^{\star}$ is banded, so it is only natural to assign zeros to the two ij elements of $\hat{\Gamma}_{n+1}^{\star}$ that satisfy $|i-j| = n$.

Consider the "augmented" vectors $\underline{Y}_{n+1} = (Y_1, \ldots, Y_{n+1})'$, $\underline{Z}_{n+1} = (Z_1, \ldots, Z_{n+1})'$, and $\underline{\varepsilon}_{n+1} = (\varepsilon_1, \ldots, \varepsilon_{n+1})'$ 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 $H_{n+1}^{-1} : \underline{\varepsilon}_{n+1} \mapsto \underline{Y}_{n+1}$. 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.

i. Let
$$\underline{Z}_{n+1} = C_{n+1}\underline{\varepsilon}_{n+1} \tag{9.33}$$
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
$$Z_{n+1} = \underline{c}_{n+1}\underline{\varepsilon}_{n+1} \tag{9.34}$$
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
$$U_{n+1} = \Phi(Z_{n+1}). \tag{9.35}$$

iii. Finally, define
$$Y_{n+1} = D_{n+1}^{-1}(U_{n+1}); \tag{9.36}$$
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 $H_{n+1}^{-1} : \underline{\varepsilon}_{n+1} \mapsto \underline{Y}_{n+1}$.

9.3.5 Model-Free Point Predictors and Prediction Intervals

In the previous sections, it was shown how to 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 Eqs. (9.34), (9.35) and (9.36) 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
$$Y_{n+1} = g_{n+1}(\varepsilon_{n+1}) \;\text{ where }\; g_{n+1}(x) = D_{n+1}^{-1}\left(\Phi\left(\sum_{i=1}^{n} c_i\varepsilon_i + c_{n+1}x\right)\right). \tag{9.37}$$
Equation (9.37) is the predictive equation associated with 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 9.3.1 MODEL-FREE (MF) POINT PREDICTORS AND PREDICTION INTERVALS FOR Y_{n+1}

1. *Construct $U_1,\ldots,U_n$ by Eq. (9.25) with $D_t(\cdot)$ estimated by either $\bar{D}_t(\cdot)$ or $\bar{D}_t^{LL}(\cdot)$; for the latter, use the respective formulas with $T=t$.*
2. *Construct $Z_1,\ldots,Z_n$ by Eq. (9.26), and use the methods of Sect. 9.3.4 to estimate Γ_n by either $\hat{\Gamma}_n^{AR}$ or $\hat{\Gamma}_n^{\star}$.*
3. *Construct $\varepsilon_1,\ldots,\varepsilon_n$ by Eq. (9.27), and let $\hat{F}_n$ denote their empirical distribution.*
4. *The Model-free L_2–optimal point predictor of Y_{n+1} is then given by*

$$\hat{Y}_{n+1} = \int x g_{n+1}(x) d\hat{F}_n(x) = \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i g_{n+1}(\varepsilon_i) \tag{9.38}$$

   where the function g_{n+1} is defined in the predictive equation (9.37) with $D_{n+1}(\cdot)$ being again estimated by either $\bar{D}_{n+1}(\cdot)$ or $\bar{D}_{n+1}^{LL}(\cdot)$, both 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 Bootstrap of Algorithm 2.4.1 based on either the L_2– or L_1–optimal point predictor.*

Remark 9.3.3 Note that Eq. (9.38) gives an approximation to the *bona fide* L_2–optimal predictor of Y_{n+1} without resorting to the L_2–optimal *linear* predictor (9.7) as in the model-based case.

Algorithm 9.3.1 used the construction of $\bar{D}_t(\cdot)$ or $\bar{D}_t^{LL}(\cdot)$ with $T=t$; using $T=t-1$ instead, leads to the following *predictive* version of the algorithm.

Algorithm 9.3.2 PREDICTIVE MODEL-FREE (PMF) POINT PREDICTORS AND PREDICTION INTERVALS FOR Y_{n+1}

The algorithm is identical to Algorithm 9.3.1 except for using $T=t-1$ instead of $T=t$ in the construction of $\bar{D}_t(\cdot)$ and $\bar{D}_t^{LL}(\cdot)$.

Remark 9.3.4 Under a model-free setup of a locally stationary time series, Paparoditis and Politis (2002b) 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 et al. (2003, 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. The LMF Algorithm is simpler than Algorithm 9.3.2 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 $\hat{D}_t(y)$ or $\hat{D}_t^{LL}(y)$ with the understanding that their inverse must be a *quantile* inverse; recall that the quantile inverse of a distribution $D(y)$ is defined as $D^{-1}(\beta)=\inf\{y \text{ such that } D(y)\geq\beta\}$.

Algorithm 9.3.3 LIMIT MODEL-FREE (LMF) POINT PREDICTORS AND PREDICTION INTERVALS FOR Y_{n+1}

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

$$\hat{Y}_{n+1}=\int xg_{n+1}(x)d\Phi(x) \tag{9.39}$$

   where the function g_{n+1} is defined in the predictive equation (9.37) where $D_{n+1}(\cdot)$ is estimated by either $\hat{D}_{n+1}(\cdot)$ or $\hat{D}_{n+1}^{LL}(\cdot)$, both with $T=t-1$.
2. *In practice, the integral (9.39) can be approximated by Monte Carlo, i.e.,*

$$\int xg_{n+1}(x)d\Phi(x)\simeq\frac{1}{M}\sum_{i=1}^{M}x_ig_{n+1}(x_i) \tag{9.40}$$

   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 Bootstrap of Algorithm 2.4.3 based on either the L_2– or L_1–optimal point predictor.*

Remark 9.3.5 Interestingly, there is a closed-form (approximate) solution for the LMF L_1–optimal point predictor of Y_{n+1} that can also be used in Step 5 of Algorithm 9.3.1. 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:

$$\begin{aligned}&Median\left(Z_{n+1}|\mathscr{F}_1^n(Z)\right)\simeq Median\left(Z_{n+1}|\mathscr{F}_{-\infty}^n(Z)\right)\\&=Median\left(Z_{n+1}|\mathscr{F}_{-\infty}^n(Y)\right)\simeq Median\left(Z_{n+1}|\mathscr{F}_1^n(Y)\right).\end{aligned}$$

Now Eqs. (9.35) and (9.36) imply that $Y_{n+1} = D_{n+1}^{-1}(\Phi(Z_{n+1}))$. 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

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

$$\simeq D_{n+1}^{-1}(\Phi(Median(Z_{n+1}|\mathscr{F}_1^n(Z)))) = D_{n+1}^{-1}(\Phi(E(Z_{n+1}|\mathscr{F}_1^n(Z)))), \quad (9.41)$$

the latter being due to the symmetry of the normal distribution of Z_{n+1} given $\mathscr{F}_1^n(Z)$. But, as in Eq. (9.7), we have $E(Z_{n+1}|\mathscr{F}_1^n(Z)) = \phi_1(n)Z_n + \phi_2(n)Z_{n-1} + \ldots + \phi_n(n)Z_1$ where $(\phi_1(n), \cdots, \phi_n(n))' = \Gamma_n^{-1}\gamma(n)$. Plugging-in either $\bar{D}_{n+1}(\cdot)$ or $\bar{D}_{n+1}^{LL}(\cdot)$ in place of $D_{n+1}(\cdot)$ in Eq. (9.41), and also employing consistent estimates of Γ_n and $\gamma(n)$ completes the calculation. As discussed in Sect. 9.3.4, 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 Chap. 6 for details.

Remark 9.3.6 (Robustness of LMF approach) The LMF approach focuses completely on the predictive equation (9.37) 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 9.3.3. 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 9.3.3 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—provided a consistent estimate of Γ_n can still be constructed.

9.3.6 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*. In terms of Eq. (9.24), stationarity follows if $\mathbf{f}_t$ does not depend on t. In this case, however, 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 9.3.3, we can estimate all occurrences of $D_t(y)$ by the regular (non-local) empirical distribution

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

Similarly, for the purposes of Algorithm 9.3.1 we can estimate all occurrences of $D_t(y)$—which is assumed smooth—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 Remark 9.3.2, the optimal rate is $h_0 \sim n^{-2/5}$ when the estimand $D_t(y)$ is sufficiently smooth in y.

9.3.7 Local Stationarity in a Higher-Dimensional Marginal

The success of the theoretical transformation of Sect. 9.3.1 in transforming the data vector $\underline{Y}_n$ to the vector of i.i.d. components $\underline{\varepsilon}_n$ hinges on Eq. (9.24) with $m = 1$ implying that the driving force behind the nonstationarity of $\{Y_t\}$ is a time-varying first marginal $D_t(\cdot)$. However, as already mentioned, it is possible to have prominent time-varying features in the mth marginal. For example, it may be that the autocorrelation $\text{Corr}(Y_t, Y_{t+m})$ varies smoothly with t; the latter can be empirically checked by estimating $\text{Corr}(Y_t, Y_{t+m})$ over different subsamples of the data, and checking whether they are (significantly) different from each other. In addition, if Eq. (9.24) holds with $m > 1$, the instantaneous transformation (9.26) might fail to create $Z_1, \ldots, Z_n$ that are jointly normal; this can be diagnosed by performing a normality test, e.g., Shapiro-Wilk test, or other diagnostic, e.g., a quantile plot, on selected linear combinations of m consecutive components of the random vector $(Z_1, \ldots, Z_n)'$. Interestingly, there is a single solution to address both potential issues, namely blocking the time series as discussed in Remark 2.3.2. Having identified an $m \geq 1$ for which the above problems do not manifest themselves, i.e., a plausible m for which Eq. (9.24) may hold, one would then create blocks of data by defining $B_t = (Y_t, \ldots, Y_{t+m-1})'$ for $t = 1, \ldots, q$ with $q = n - m + 1$. Focusing on the multivariate time series dataset $\{B_1, \ldots, B_q\}$, let $D_t^{(m)}(\cdot)$ denote the first marginal distribution function of vector B_t which will be assumed to be (absolutely) continuous for each t. Furthermore, Eq. (9.24) implies that $D_t^{(m)}(\cdot)$ varies smoothly (and slowly) with t as discussed in Remark 9.3.1. Using the Rosenblatt (1952) transformation, we can map B_t to a random vector V_t that has components i.i.d. Uniform (0,1), and then perform the Gaussian transformation and whitening as required by the Model-Free Principle. Thus, in the general case when the time series $\{Y_t\}$ satisfies Eq. (9.24) with $m \geq 1$, 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 $EZ_t Z_{t+k}'$ for $k = 0, 1, \ldots$, and use it to "whiten" the sequence $Z_1, \ldots, Z_q$, i.e., to map it (in a one-to-one way) to the i.i.d. sequence $\zeta_1, \ldots, \zeta_q$; here, $\zeta_t \in \mathbf{R}^m$ is a random vector having components that are i.i.d. $N(0,1)$.

In Step 2 above, the mth dimensional Rosenblatt transformation is based on the mth marginal $D_t^{(m)}(\cdot)$ which is unknown but can be estimated using a local average or local linear estimator, i.e., a multivariate analog of $\bar{D}_t(\cdot)$ and $\bar{D}_t^{LL}(\cdot)$; to avoid the curse of dimensionality here, it is imperative that m is of smaller order of magnitude than the sample size n. Regarding Step 4, standard methods exist to estimate the (matrix) autocovariance of Z_t with Z_{t+k}; see, e.g., Jentsch and 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 Chap. 2; the details are straightforward.

Acknowledgements

Many thanks are due to Srinjoy Das and Stathis Paparoditis for helpful discussions.

Part IV
Case Study: Model-Free Volatility Prediction for Financial Time Series

Chapter 10
Model-Free vs. Model-Based Volatility Prediction

10.1 Introduction

Let $\{P_t, t \in \mathbf{Z}\}$ denote a financial time series of prices, i.e., P_t may denote a stock price, stock index, or foreign exchange rate at time t; the time t can run daily, weekly, or calculated at different (discrete) intervals. From the price series $\{P_t, t \in \mathbf{Z}\}$ we can define the financial *returns* time series $\{Y_t, t \in \mathbf{Z}\}$ by $Y_t = (P_t - P_{t-1})/P_t$, i.e., Y_t denotes the relative price change from time $t-1$ to time t; consequently, the percentage return as time t is just $100Y_t$.

The returns series $\{Y_t\}$ will be assumed (strictly) stationary with mean zero which—from a practical point of view—implies that trends and other nonstationarities have been successfully removed. Figure 10.1 shows the returns from three illustrative datasets, a foreign exchange rate, a stock index, and a stock price; it is apparent that the returns are typically small numbers, i.e., for the most part $|Y_t| < 0.10$ with the exception of a prominent outlier at -0.20 corresponding to the market crash of October 1987. Another way of seeing this is to note that the ratio P_{t-1}/P_t is close to one. Recall the Taylor series expansion of the natural logarithm $\log x \simeq x - 1$ for x close to one, from which it follows that

$$Y_t = 1 - \frac{P_{t-1}}{P_t} \simeq -\log \frac{P_{t-1}}{P_t} = \log P_t - \log P_{t-1}; \tag{10.1}$$

this is why the Y_ts are sometimes called "logarithmic" returns. Throughout this chapter, we will assume that the observed data consist of the returns $Y_1, \ldots, Y_n$ which implies that the price series $P_0, P_1, \ldots, P_n$ must have been previously available.

Bachelier's (1900) pioneering Ph.D. thesis put forth the Gaussian random walk model for (the logarithm of) stock market prices. Because of approximation (10.1), the implication of Bachelier's proposal was that the returns series $\{Y_t\}$ can be modelled as independent, identically distributed (i.i.d.) random variables with Gaussian

D.N. Politis, *Model-Free Prediction and Regression*, Frontiers in Probability and the Statistical Sciences, DOI 10.1007/978-3-319-21347-7_10

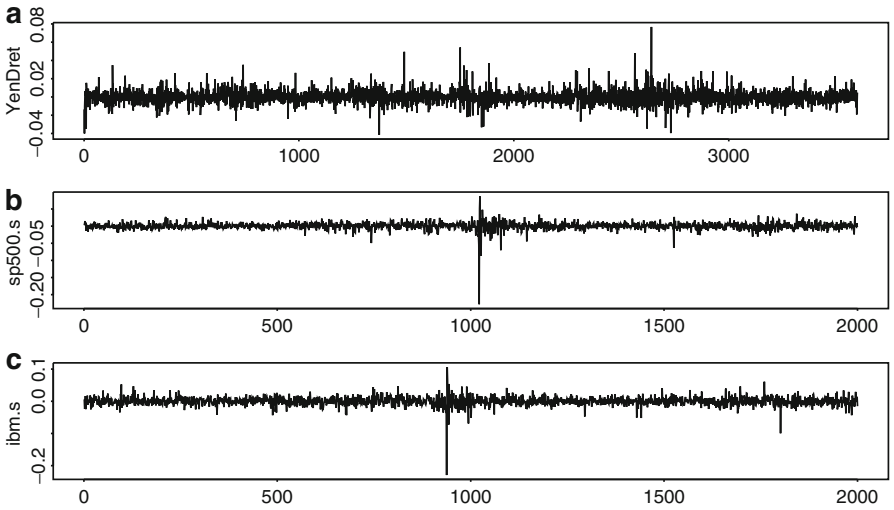

Fig. 10.1 (**a**) Plot of the daily Yen/Dollar returns from December 31, 1987 up to August 1, 2002; (**b**) plot of the daily S&P500 stock index returns from October 1, 1983 to August 30, 1991; (**c**) plot of the daily returns of the IBM stock price from February 1, 1984 to December 31, 1991

$N(0, \sigma^2)$ distribution. Although Bachelier's thesis was not so well-received by his Ph.D. committee, his work went on to serve as the foundation for financial modeling for a good part of the twentieth century.

The assumption of Gaussianity was challenged in the 1960s when it was noticed that the distribution of returns seemed to have fatter tails than the normal; see, e.g., Fama (1965). The adoption of some non-normal, heavy-tailed distribution for the returns seemed—at the time—to be the solution. However, in the early paper of Mandelbrot (1963) the phenomenon of "volatility clustering" was pointed out, i.e., the fact that days with high volatility are clustered together and the same is true for days with low volatility; this is effectively negating the assumption of independence of the returns in the implication that the absolute values (or squares) of the returns are positively correlated.

For example, Fig. 10.1b depicts the daily returns of the S&P500 index from October 1, 1983 to August 30, 1991; the extreme values associated with the aforementioned crash of October 1987 are very prominent in the plot. Figure 10.2a is a "correlogram" of the S&P500 returns, i.e., a plot of the estimated autocorrelation function (acf); the plot is consistent with the hypothesis of uncorrelated returns. By contrast, the correlogram of the squared returns of Fig. 10.2b shows some significant correlations thus lending support to the "volatility clustering" hypothesis.

The celebrated ARCH (Autoregressive Conditional Heteroscedasticity) models of Engle (1982) were designed to capture the phenomenon of volatility clustering by postulating a particular structure of dependence for the time series of squared returns $\{Y_t^2\}$. A typical ARCH(p) model is described by an equation of the type:

$$Y_t = Z_t \sqrt{a + \sum_{i=1}^{p} a_i Y_{t-i}^2} \tag{10.2}$$

where the series $\{Z_t\}$ is assumed to be i.i.d. $N(0,1)$ and p is an integer indicating the order of the model.

Let $\mathscr{F}_n$ be a short-hand for the observed information set, i.e., $\mathscr{F}_n = \{Y_t, 1 \le t \le n\}$ which was previously denoted $\mathscr{F}_1^n(Y)$. Note that under the above ARCH(p) model, the L_2–optimal predictor of Y_{n+1}^2 based on $\mathscr{F}_n$ is given by

$$E(Y_{n+1}^2 | \mathscr{F}_n) = a + \sum_{i=1}^{p} a_i Y_{n+1-i}^2. \tag{10.3}$$

The conditional expectation $E(Y_{n+1}^2 | \mathscr{F}_n)$ is commonly referred to as the **volatility** (although the same term is sometimes also used for its square root).

Volatility clustering as captured by model (10.2) does indeed imply a marginal distribution for the $\{Y_t\}$ returns that has heavier tails than the normal. However, model (10.2) can account only partly for the degree of heavy tails empirically found in the distribution of returns, and the same is true for the Generalized ARCH (GARCH) models of Bollerslev (1986); see Bollerslev et al. (1992) or Shephard (1996) for a review. For example, the market crash of October 1987 is still an outlier (six standard deviations away) even after the best ARCH/GARCH model is employed; see Nelson (1991).

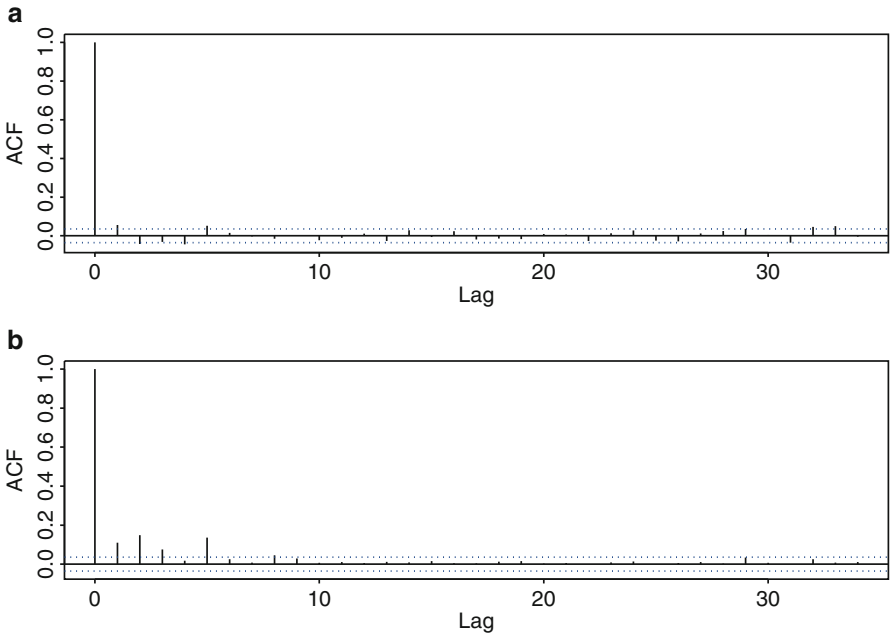

Fig. 10.2 (**a**) Correlogram of S&P500 returns. (**b**) Correlogram of S&P500 squared returns

Consequently, researchers and practitioners have been resorting to ARCH models with heavy-tailed errors. A popular assumption for the distribution of the $\{Z_t\}$ is the t-distribution with degrees of freedom empirically chosen to match the apparent degree of heavy tails in the residuals; see Shephard (1996) and the references therein.

Nevertheless, this situation is not satisfactory since the choice of a t-distribution seems quite arbitrary. Trying to model the excess kurtosis by an arbitrarily chosen heavy-tailed distribution seems to bring us full-circle back to the 1960s. Perhaps the real issue is that a simple and neat parametric model such as (10.2) could not be expected to perfectly capture the behavior of a complicated real-world phenomenon such as the evolution of financial returns that—almost by definition of market "efficiency"—ranks at the top in terms of difficulty of modeling/prediction.

As a more realistic alternative, one may resort to our Model-free prediction approach in trying to understand this complex type of data. In what follows, a normalizing and variance–stabilizing transformation (NoVaS, for short) for financial returns series will be defined and analyzed. As will be apparent, the NoVaS transformation is a straightforward application of the principle laid out in Sect. 2.3.2, i.e., using a transformation to normality as a stepping-stone towards a transformation to "i.i.d.–ness." Furthermore, the suggestion of Sect. 2.3.5 is taken into account, namely that the Model-free practitioner could/should use all the modeling know-how associated with the problem at hand. Since the state-of-the-art of modeling financial returns is to employ ARCH/GARCH models, these can serve as the starting point in concocting the desired transformation. A preliminary announcement of the NoVaS transformation was given in Politis (2003a), and a full treatment in Politis (2007a).

Remark 10.1.1 Throughout the chapter, the term "volatility prediction" will be taken to mean "prediction of squared returns." Estimating the conditional expectation $E(Y_{n+1}^2|\mathscr{F}_n)$ is briefly discussed in Remark 10.5.2. Note that this quantity is nonrandom given the data $Y_1,\ldots,Y_n$; hence, the term "estimation" is more appropriate than "prediction." As mentioned in Eq. (10.3), these two problems are interrelated since the MSE–optimal predictor of Y_{n+1}^2 is nothing other than $E(Y_{n+1}^2|\mathscr{F}_n)$, so an estimate of the latter must invariably be constructed.

10.2 Three Illustrative Datasets

Throughout the chapter, we focus on three representative datasets of daily returns taken from a foreign exchange rate, a stock price, and a stock index; a description of our main datasets is as follows:

- **Example 1: Foreign exchange rate.** Daily returns from the Yen vs. Dollar exchange rate from January 1, 1988 to August 1, 2002; the data were downloaded from Datastream. A plot of the returns is shown in Fig. 10.1a; the sample size is 3600 (weekends and holidays are excluded).

- **Example 2: Stock index.** Daily returns of the S&P500 stock index from October 1, 1983 to August 30, 1991; the data are available as part of the `garch` module in Splus. A plot of the returns is shown in Fig. 10.1b; the sample size is 2000.
- **Example 3: Stock price.** Daily returns of the IBM stock price from February 1, 1984 to December 31, 1991; the data are again available as part of the `garch` module in Splus. A plot of the returns is shown in Fig. 10.1c; the sample size is 2000.

The phenomenon of volatility clustering is quite apparent in the three returns series of Fig. 10.1. Note, in particular, the extreme volatility and outlying values around the mid-point of Fig. 10.1b and slightly before the mid-point of Fig. 10.1c; those points of time correspond to the aforementioned market crash of October 1987.

Returning to the ARCH model (10.2), it should be stressed that this is not just a model for the conditional variance; the ARCH model is a model for the *whole* data generating process (DGP) of the data series Y_t. In this connection, observe that the one-step-ahead ARCH-based predictor for Y_{n+1} given $\mathscr{F}_n = \{Y_t, t \le n\}$ is trivial, i.e., zero, essentially due to the random sign of Z_t in Eq. (10.2).

Nevertheless, the litmus test for a model is its predictive ability. Since the ARCH cannot predict the signed return Y_{n+1}, it should at least have some predictive ability for the squared returns. Recall also that if $\{Y_t\}$ follows the ARCH(p) model (1), then $\{Y_t^2\}$ follows an AR(p) model—see, e.g., Gouriéroux (1997) or Francq and Zakoian (2011). To see why, define $U_t = Y_t^2 - a - \sum_{i=1}^{p} a_i Y_{t-i}^2$; using Hilbert space projection arguments, it is now immediate that the time series $\{U_t\}$ constitutes a second-order stationary, mean-zero white noise with finite variance.[1] We are then led to the AR(p) model:

$$Y_t^2 = a + \sum_{i=1}^{p} a_i Y_{t-i}^2 + U_t. \tag{10.4}$$

which is driven by innovations that constitute a white noise although not i.i.d. Technically speaking, the U_ts are not even a martingale difference since the conditional variance $E(U_t^2|\mathscr{F}_n)$ is not constant; see, e.g., Kokoszka and Politis (2011). Nonetheless, the linear predictor associated with this AR(p) model should have some predictive power for the squared returns, and this predictor is identical to the usual ARCH model predictor of the volatility.

It is intuitive to also consider an Auto-Regressive Moving Average (ARMA) model on the squared returns; this idea is closely related to the GARCH(p,q) models of Bollerslev (1986). Among these, the GARCH(1,1) model is by far the

[1] The aforementioned Hilbert space projection arguments require that U_t has a finite second moments, i.e., that $EY_t^4 < \infty$; however, as the case will be made later on, the existence of a finite fourth moment for Y_t should not be taken for granted.

most popular, and typically forms the benchmark for modeling financial returns. The GARCH(1,1) model is described by the equation:

$$Y_t = h_t Z_t \quad \text{with} \quad h_t^2 = C + AY_{t-1}^2 + Bh_{t-1}^2; \tag{10.5}$$

where the $\{Z_t\}$s are i.i.d. (0,1), and the parameters A, B, C are assumed nonnegative. The quantity $h_t^2 = E(Y_t^2 | \mathscr{F}_{t-1})$ is the volatility as defined in Eq. (10.3).

Back-solving in the right-hand-side of Eq. (10.5), it is easy to see that the GARCH(1,1) model is tantamount to the ARCH model (10.2) with $p = \infty$ and the following identifications:

$$a = \frac{C}{1-B}, \quad \text{and} \quad a_i = AB^{i-1} \quad \text{for} \quad i = 1, 2, \ldots \tag{10.6}$$

In fact, under some conditions, all GARCH(p,q) models have ARCH(∞) representations similar to the above; see, e.g., Gouriéroux (1997, Chap. 4.1.5). So, in some sense, the only advantage GARCH models may offer over the simpler ARCH is *parsimony*, i.e., achieving the same quality of model-fitting with fewer parameters. Nevertheless, if one is to impose a certain *structure* on the ARCH parameters, then the effect is the same; the exponential structure of Eq. (10.6) is a prime such example.

The above ARCH/GARCH models constitute a beautiful attempt to capture the phenomenon of volatility clustering in a simple equation while at the same time implying a marginal distribution for the $\{Y_t\}$ returns that has heavier tails than the normal. Viewed differently, the ARCH(p) and/or GARCH (1,1) model may be considered as attempts to "normalize" the returns, i.e., to reduce the problem to a model with normal residuals (the Z_ts). In that respect though the ARCH(p) and/or GARCH (1,1) models are only partially successful as empirical work suggests that ARCH/GARCH residuals often exhibit heavier tails than the normal; the same is true for ARCH/GARCH spin-off models such as the EGARCH—see Bollerslev et al. (1992) or Shephard (1996) for a review. Nonetheless, the goal of normalization is most worthwhile and it is indeed achievable as will be shown in the sequel.

The literature on volatility prediction is already quite large and appears to be continuously expanding. The articles by Poon and Granger (2003), and Andersen et al. (2006) provide comprehensive reviews of the subject. We further mention here some papers that are related to the problem at hand: Barndorff-Nielsen et al. (1996) for an early treatment of forecasting volatility; Meddahi (2001) for an eigenfunction volatility modeling approach; Hansen et al. (2003) on selecting volatility models; Andersen et al. (2004) on analytic evaluation of volatility forecasts; Hansen and Lunde (2005, 2006) for comparing forecasts of volatility models against the standard GARCH(1,1) model and for consistent ranking of volatility models; Koopman et al. (2005) and Patton (2011) for volatility forecast evaluation; and Ghysels et al. (2006) for predicting volatility using data sampled at different frequencies.

10.3 Normalization and Variance-Stabilization

10.3.1 Definition of the NoVaS Transformation

Following the principle laid out in Sect. 2.3.2, we will now try to find a transformation to map the dataset $Y_1, \ldots, Y_n$ to a Gaussian one. Our starting point is the ARCH model (10.2), under which the quantity

$$\frac{Y_t}{\sqrt{a + \sum_{i=1}^{p} a_i Y_{t-i}^2}} \tag{10.7}$$

is thought of as perfectly normalized and variance–stabilized as it is assumed to be i.i.d. $N(0,1)$. From an applied statistics point of view, the above ratio can be interpreted as an attempt to "studentize" the return Y_t by dividing with a (time-localized) measure of the standard deviation of Y_t.

Nevertheless, there seems to be no reason—other than coming up with a neat model—to exclude the value of Y_t from an empirical, causal estimate of the standard deviation of Y_t; recall that a causal estimate is one involving present and past data only, i.e., the data $\{Y_s, s \le t\}$. Hence, we may define the new "studentized" quantity

$$W_{t,a} := \frac{Y_t}{\sqrt{\alpha s_{t-1}^2 + a_0 Y_t^2 + \sum_{i=1}^{p} a_i Y_{t-i}^2}} \quad \text{for} \quad t = p+1, p+2, \ldots, n; \tag{10.8}$$

in the above, s_{t-1}^2 is an estimator of $\sigma_Y^2 = Var(Y_1)$ based on the data up to (but not including[2]) time t; under the zero mean assumption for Y_1, the natural estimator is $s_{t-1}^2 = (t-1)^{-1} \sum_{k=1}^{t-1} Y_k^2$.

Equation (10.8) describes our proposed normalizing and variance–stabilizing transformation (**NoVaS**, for short) under which the data series $\{Y_t\}$ is mapped to the new series $\{W_{t,a}\}$. The order $p(\ge 0)$ and the vector of nonnegative parameters $(\alpha, a_0, \ldots, a_p)$ are chosen by the practitioner with the twin goals of normalization and variance stabilization in mind that will be made more precise shortly.

The NoVaS equation (10.8) can be re-arranged to yield:

$$Y_t = W_{t,a} \sqrt{\alpha s_{t-1}^2 + a_0 Y_t^2 + \sum_{i=1}^{p} a_i Y_{t-i}^2}. \tag{10.9}$$

Formally, the only real difference between the NoVaS equation (10.9) and the ARCH equation (10.2) is the presence of the term Y_t^2 paired with the coefficient a_0. Replacing the term a in Eq. (10.2) by the term αs_{t-1}^2 in (10.9) is only natural

[2] The reason for not including time t in the variance estimator is for purposes of notational clarity as well as the easy identifiability of the effect of the coefficient a_0 associated with Y_t^2 in the denominator of Eq. (10.8).

since the former has—by necessity—units of variance; in other words, the term a in Eq. (10.2) is not scale invariant, whereas the term α in (10.9) is.

Equation (10.9) is very useful but should not be interpreted as a "model" for the $\{Y_t\}$ series; rather, the focus should remain on Eq. (10.8) and the effort to render the transformed series $\{W_{t,a}, t = p+1, p+2, \cdots\}$ close—in some sense to be described shortly—to behaving like the standard normal ideal. In some sense, Eq. (10.9) is analogous to Eq. (2.4) from the Model-free Prediction Principle; the analogy would become exact if it were further shown that the random variables $W_{t,a}$ are i.i.d.—see Sect. 10.5.2.

A further note of caution on viewing Eq. (10.9) as a "model" comes from the observation that *exact* normality is not even feasible for the series $\{W_{t,a}\}$ since the latter comprises of bounded random variables; to see this, note that

$$\frac{1}{W_{t,a}^2} = \frac{\alpha s_{t-1}^2 + a_0 Y_t^2 + \sum_{i=1}^p a_i Y_{t-i}^2}{Y_t^2} \geq a_0$$

if all the parameters are nonnegative. Therefore,

$$|W_{t,a}| \leq 1/\sqrt{a_0} \tag{10.10}$$

almost surely, assuming of course that $a_0 \neq 0$. However, with a_0 chosen small enough, the boundedness of the $\{W_{t,a}\}$ series is effectively (and practically) not noticeable. This phenomenon is analogous to the fact that financial returns are modeled by distributions that have both left and right heavy tails despite being hard-bounded from below; note that a stock or index cannot lose more than 100 % of its value.

10.3.2 Choosing the Parameters of NoVaS

In choosing the order p (≥ 0) and the parameters $\alpha, a_0, \ldots, a_p$ the twin goals of normalization and variance stabilization of the transformed series $\{W_{t,a}\}$ are first taken into account. Secondarily, the NoVaS parameters may be further optimized with a specific criterion in mind, e.g., optimal volatility prediction; this approach is expanded upon in the next section. We now focus on the primary goals of normalization and variance stabilization.

The target of variance stabilization is easier and—given the assumed structure of the return series—amounts to constructing a local estimator of scale for studentization purposes; for this reason we require

$$\alpha \geq 0, \;\; a_i \geq 0 \;\; \text{for all} \;\; i \geq 0, \;\; \text{and} \;\; \alpha + \sum_{i=0}^{p} a_i = 1. \tag{10.11}$$

Equation (10.11) has the interesting implication that the $\{W_{t,a}\}$ series can be assumed to have an (unconditional) variance that is (approximately) unity. Nevertheless, note that p and $\alpha, a_0, \ldots, a_p$ must be carefully chosen to achieve a degree

of conditional homoscedasticity as well; to do this, one must necessarily take p small enough—as well as α small enough or even equal to zero—so that a local (as opposed to global) estimator of scale is obtained. An additional intuitive—but not obligatory—constraint may involve monotonicity:

$$a_i \geq a_j \quad \text{if} \quad 1 \leq i < j \leq p. \tag{10.12}$$

It is practically advisable that a simple structure for the a_i coefficients is employed satisfying (10.11) and perhaps also (10.12). The simplest such example is to let $\alpha = 0$ and $a_i = 1/(p+1)$ for all $0 \leq i \leq p$; this specification will be called the **'simple'** NoVaS transformation, and involves only one parameter, namely the order p, to be chosen by the practitioner. Another example is given by the **exponential** decay NoVaS, where $\alpha = 0$ and $a_i = c'e^{-ci}$ for all $0 \leq i \leq p$. The exponential scheme involves choosing two parameters: p and $c > 0$ since c' is determined by (10.11); nevertheless, the parameter p is now of secondary importance—see Sect. 10.3.4. The Simple and Exponential NoVaS schemes are most intuitive as they correspond to the two popular time series methods of obtaining a "local," one-sided average, namely a moving average (of the last $p+1$ values) and "exponential smoothing"; see, e.g., Hamilton (1994).

Subject to the variance stabilization condition (10.11)—together with (10.12) if desirable—one then proceeds to choose (the parameters needed to identify) p and $\alpha, a_0, a_1, \ldots, a_p$ with the optimization goal of making the $\{W_{t,a}\}$ transformed series as close to normal as possible. To quantify this target, one can use minimize a (pseudo)distance measuring departure of the transformed data from normality; see, e.g., Sect. 2.3.2. In order to render joint distributions of the $\{W_{t,a}\}$ series more normal, one may also apply the (pseudo)distance minimization idea to a few specific linear combinations of $W_{t,a}$ random variables; more details are given in the next subsection.

However, in view of the bound (10.10), one must be careful to ensure that the $\{W_{t,a}\}$ variables have a large enough range such that the boundedness is not seen as spoiling the normality. Thus, we also require

$$\frac{1}{\sqrt{a_0}} \geq C \quad \text{i.e.,} \quad a_0 \leq 1/C^2 \tag{10.13}$$

for some appropriate C of the practitioner's choice. Recalling that 99.7 % of the mass of the $N(0,1)$ distribution is found in the range ± 3, the simple choice $C = 3$ can be suggested; this choice seems to work reasonably well—at least for the usual samples sizes.

10.3.3 Simple NoVaS Algorithm

We now give specific algorithms for optimizing the NoVaS transformation in the two previously mentioned examples, Simple and Exponential NoVaS. First note

that it is a matter of common practice to assume that the distribution of financial returns is *symmetric* (at least to a first approximation); therefore, the skewness of financial returns is often ignored. In contrast, the kurtosis is typically very large, indicating a heavy-tailed distribution. The above claims, i.e., approximate symmetry and heavy tails, are confirmed by Fig. 10.3 where histograms and Q-Q plots for our three returns series are presented.

Hence, the kyrtosis can serve as a simple (pseudo)distance measuring the departure of a (non-skewed) dataset from normality. Let $KURT_n(Y)$ denote the empirical kurtosis of data $\{Y_t, t = 1, \ldots, n\}$, i.e.,

$$KURT_n(Y) = \frac{n^{-1}\sum_{t=1}^{n}(Y_t - \bar{Y})^4}{(n^{-1}\sum_{t=1}^{n}(Y_t - \bar{Y})^2)^2}$$

where $\bar{Y} = n^{-1}\sum_{t=1}^{n} Y_t$ is the sample mean. For our three datasets, Yen/Dollar, S&P500 and IBM, the empirical kurtosis was 10.1, 94.0, and 38.3, respectively.

Note that the only free parameter in Simple NoVaS is the order p; therefore, the Simple NoVaS transformation will be denoted by $W_{t,p}^S$.

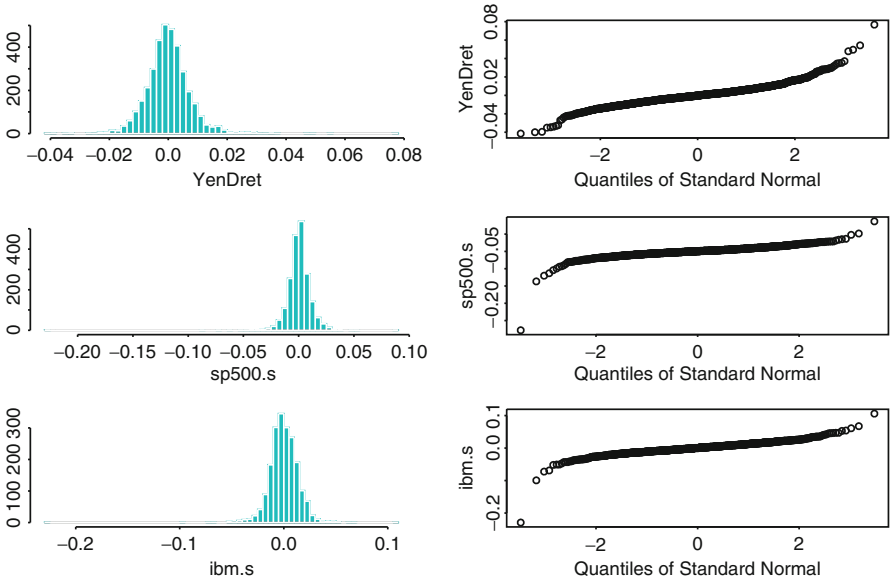

Fig. 10.3 Histograms and Q-Q plots for the three returns series of Fig. 10.1

Algorithm 10.3.1 ALGORITHM FOR SIMPLE NOVAS

1. *Let* $\alpha = 0$ *and* $a_i = 1/(p+1)$ *for all* $0 \le i \le p$.
2. *Pick* p *such that* $|KURT_n(W_{t,p}^S) - 3|$ *is minimized.*

Step 2 in the above was described as an optimization problem for mathematical concreteness. Nevertheless, it could be better understood as a moment matching, i.e.,

2′. Pick p such that $KURT_n(W^S_{t,p}) \simeq 3$

where of course the value 3 for kurtosis corresponds to the Gaussian distribution.

Remark 10.3.1 [Feasibility of normalization by moment matching.] To see that the moment matching goal is a feasible one, note first that for $p = 0$ we have $a_0 = 1$, $W^S_{t,0} = sign(Y_t)$, and $KURT_n(W^S_{t,0}) = 1$. On the other hand, it is to be expected that for large p, $KURT_n(W^S_{t,p})$ will be bigger than 3. As a matter of fact, the law of large numbers implies that for increasing values of p, $KURT_n(W^S_{t,p})$ will tend to the "true" kurtosis of the random variable Y_1 which is understood to be quite large (and may even be infinite—see the discussion in Sect. 10.4.1). Therefore, viewing $KURT_n(W^S_{t,p})$ as a (smooth) function of p, the intermediate value theorem would suggest that, for an intermediate value of p, the level 3 can always be (approximately) attained; this is actually what happens in practice.

Thus, to actually carry out the search for the optimal p in the Simple NoVaS Algorithm, one sequentially computes $KURT_n(W^S_{t,p})$ for $p = 1, 2, \cdots$, stopping when $KURT_n(W^S_{t,p})$ first hits or just passes the value 3. Interestingly, $KURT_n(W^S_{t,p})$ is typically an increasing function of p which makes this scheme very intuitive; see Fig. 10.4a.

The above simple algorithm seems to work remarkably well. A caveat, however, is that the range condition (10.13) might not be satisfied. If this is the case, the following "range–adjustment" step can be added to Algorithm 10.3.1.

3. *If p (and* a_0*) as found above are such that (10.13) is not satisfied, then increase p accordingly; in other words, redefine p to be the smallest integer such that* $1/(p+1) \leq 1/C^2$*, and let* $a_i = 1/(p+1)$ *for all* $0 \leq i \leq p$.

It goes without saying that this range–adjustment should be used with restraint, that is, the choice of C in (10.13) should be reasonably small, as it effectively over-rides the data-dependent character of choosing p. The conservative choice of letting $C = 3$ seem to work well in practice; see Remark 10.3.2 for an example.

Figure 10.4 gives an illustration of the Simple NoVaS algorithm for the Yen/Dollar dataset. The top panel of Fig. 10.4 shows a plot of $KURT_n(W^S_{t,p})$ as a function of p; the monotonic increase of $KURT_n(W^S_{t,p})$ is apparent, rendering the NoVaS algorithm easy to implement. Notably, $KURT_n(W^S_{t,p})$ is closest to 3 for $p = 9$; actually, $KURT_n(W^S_{t,9}) = 3.03$. Interestingly, the data-dependent choice $p = 9$ seems very stable; estimating p over different subsamples of the Yen/Dollar dataset typically yielded the value 9 ± 1 even for subsamples with length one tenth of $n = 3600$; see the bottom panel of Fig. 10.4.

The optimal Simple NoVaS transformed series $\{W^S_{t,9}\}$ for the Yen/Dollar dataset is plotted in Fig. 10.5a. Although $\{W^S_{t,9}\}$ is related in a simple way to the original data of Fig. 10.1a, the regions of "volatility clustering" corresponding to the $\{Y_t\}$ series are hardly (if at all) discernible in the plot of the NoVaS series $\{W^S_{t,9}\}$.

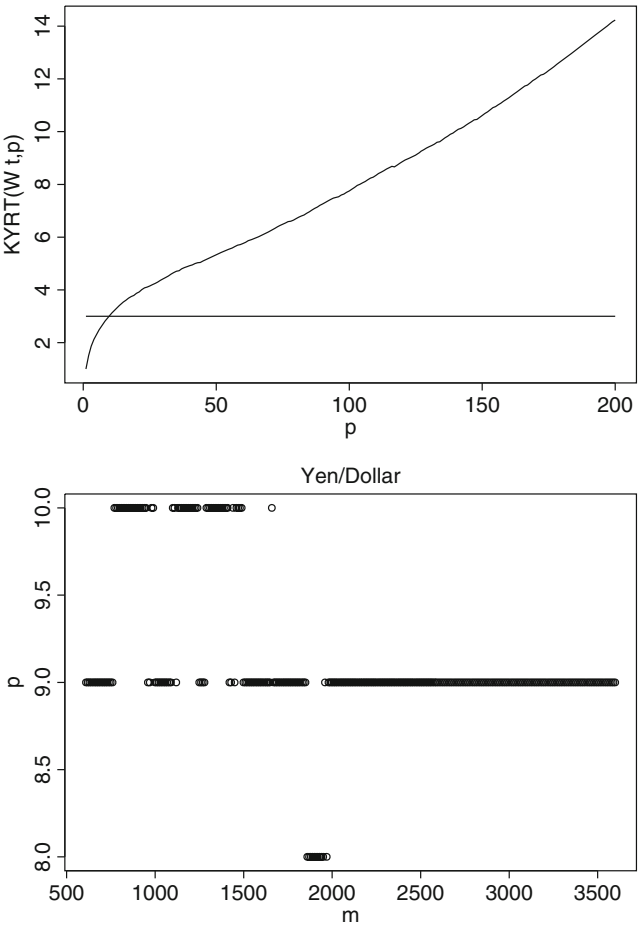

Fig. 10.4 Illustration of the Simple NoVaS algorithm for the Yen/Dollar dataset. *Top panel*: plot of $KURT_n(W^S_{t,p})$ as a function of p with the *solid line* indicating the Gaussian kurtosis of 3; here the full sample size $n = 3600$ is used. *Bottom panel*: optimal values of p in Simple NoVaS calculated from subseries $Y[1:m]$ are plotted against subsample size m; here Y is the Yen/Dollar dataset

Similar calculations were performed for our other two datasets; the optimal p values were 13 for the IBM dataset, and 11 for the S&P500 dataset. Figure 10.5 depicts plots of the Simple NoVaS transformed series for the three datasets of Fig. 10.1. The variance stabilization effect is quite apparent; in particular, note that the market crash of October 1987 is hardly (if at all) noticeable in Fig. 10.5b, c. A comparison with Fig. 10.1 is quite striking.

Figure 10.6 shows histograms and Q-Q plots for the three NoVaS series of Fig. 10.5. Comparing Fig. 10.6 to Fig. 10.3, it is visually apparent that the goal of normalization has been largely achieved. The histograms look quite normal

and the Q-Q plots look quite straight; there is no indication of heavy-tails and/or outlying values in Fig. 10.6, i.e., no "left-over" kurtosis to account for. A formal Kolmogorov-Smirnov test of the hypothesis that the transformed series are normal confirms the conclusions of the visual inspection of the three Q-Q plots; the P-values were found to be 0.1654, 0.3638, and 0.4646 for the Yen/Dollar, S&P500 and IBM datasets, respectively, so the hypothesis of normality is not rejected.

Remark 10.3.2 [On range adjustment] Focusing again on the Yen/Dollar data, note that—perhaps not surprisingly—the lowest P-value in Kolmogorov-Smirnov testing is associated with the smallest value of p for Simple NoVaS; recall that $p = 9$ for Yen/Dollar, whereas p was found to be 11 for S&P500, and 13 for IBM. A low value of p arising from our kurtosis matching algorithm may indicate an adverse effect of truncation to our normalization goal, and a subsequent need for range adjustment. Still the value $p = 9$ is high enough to yield an effective range of the Yen/Dollar NoVaS transform $\{W^S_{t,9}\}$ series of about 3.16 which is acceptable in terms of Eq. (10.13) being satisfied with $C = 3$. The higher values of p in connection with the S&P500 and IBM datasets correspond to ranges of about 3.3 and 3.6, respectively, indicating even less of a need for possible range adjustement.

Remark 10.3.3 [Normalization of joint distributions] In the Simple NoVaS algorithm, the target was fourth moment matching of $W^S_{t,p}$ to the corresponding Gaussian moment, i.e., to obtain $KURT_n(W^S_{t,p}) \simeq 3$; this procedure has the goal of (approximately) normalizing the marginal distribution of $W^S_{t,p}$. Interestingly, this simple procedure seems to somehow be also effective in normalizing joint distributions, e.g., the joint distribution of $W^S_{t,p}$ and its lagged version $W^S_{t-1,p}$, which is a highly desirable objective. Table 10.1 gives the sample kurtosis of the series $\tilde{W}^S_{t,9,i} = W^S_{t,9} + \lambda_i W^S_{t-1,9}$ (in the case of the Yen/Dollar dataset) for different values of λ_i. Notably, all the entries of Table 10.1 are close to the nominal value of 3 supporting the claim of approximate normalization of the *joint* distribution of the pair $(W^S_{t,9}, W^S_{t-1,9})$. However, if one wanted to *ensure* that some joint distributions are also normalized—at least as far as fourth moments are concerned—then the moment matching criterion of the algorithm can be modified. To fix ideas, consider the target of normalizing the joint distribution of $W^S_{t,p}$ and $W^S_{u,p}$. The Cramér-Wold device suggests that we simultaneously consider some linear combinations of the type:

$$\tilde{W}^S_{t,p,i} = W^S_{t,p} + \lambda_i W^S_{u,p} \quad \text{for} \quad i = 1, \ldots, K,$$

where the λ_i's are some chosen constants as in Table 10.1. The Simple NoVaS algorithm is then altered to focus on the kurtosis of $\tilde{W}^S_{t,p,i}$ instead of that of $W^S_{t,p}$; to elaborate, the last step of the simple NoVaS algorithm would read:

$2''$. *Pick p such that* $\max_i |KURT_n(\tilde{W}^S_{t,p,i}) - 3|$ *is minimized.*

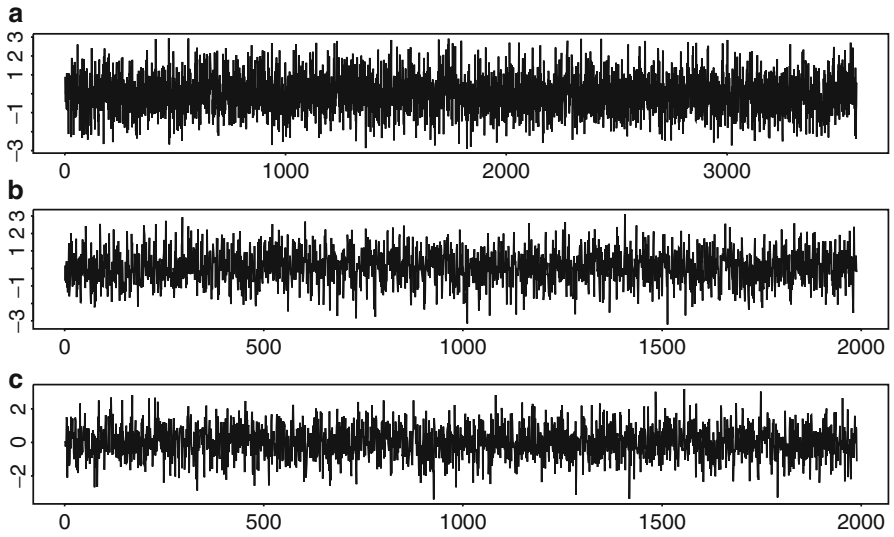

Fig. 10.5 Plots of the Simple NoVaS transformed series corresponding to the three datasets of Fig. 10.1

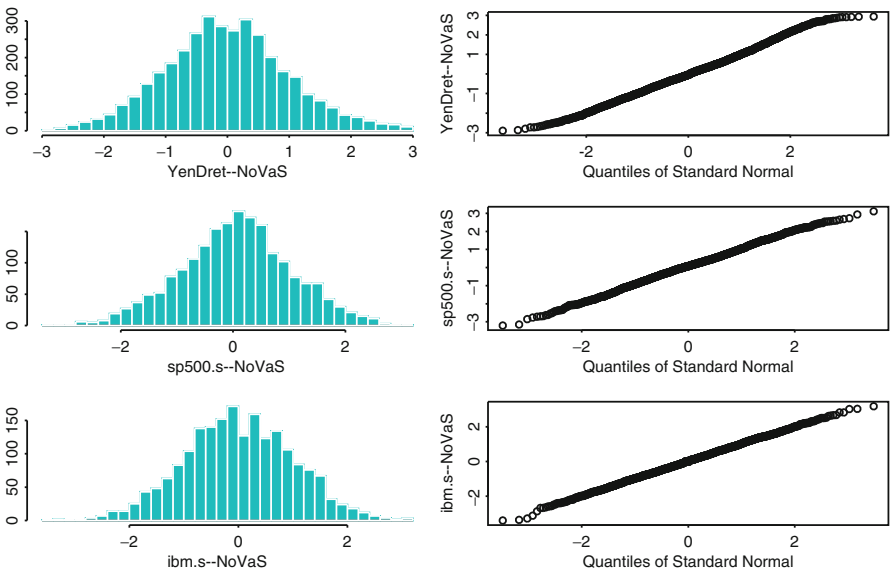

Fig. 10.6 Histograms and Q-Q plots for the three NoVaS series of Fig. 10.5

λ_i	-4	-1	-0.5	0	0.5	1	4
$KURT_n(\tilde{W}^S_{t,9,i})$	2.92	2.89	2.98	3.03	3.03	3.10	3.12

Table 10.1 (Yen/Dollar example) Sample kurtosis of the linear combination $\tilde{W}^S_{t,9,i} = W^S_{t,9} + \lambda_i W^S_{t-1,9}$ for different values of λ_i

10.3.4 Exponential NoVaS Algorithm

In the Exponential NoVaS, to specify all the a_is, one just needs to specify the two parameters p and $c > 0$, in view of (10.11). However, because of the exponential decay, the parameter p is now of secondary significance as the following algorithm suggests; thus, we may concisely denote the Exponential NoVaS transformation by $W^E_{t,c}$. Indeed, p will be determined given c, and the practitioner's choice of threshold/tolerance level; see the discussion below.

Algorithm 10.3.2 ALGORITHM FOR EXPONENTIAL NOVAS

1. *Let p take a very high starting value, e.g., let $p \simeq n/4$ or $n/5$. Then, let $\alpha = 0$ and $a_i = c' e^{-ci}$ for all $0 \le i \le p$, where $c' = 1/\sum_{i=0}^{p} e^{-ci}$ by Eq. (10.11).*
2. *Pick $c > 0$ in such a way that $|KURT_n(W^E_{t,c}) - 3|$ is minimized.*

Technically, the above search is for $c \in (0, \infty)$ which appears formidable; what makes this minimization problem well-behaved is that we know that high values of c cannot plausibly be solutions. To see why, note that if c is large, then $a_i \approx 0$ for all $i > 0$ and $W^E_{t,c} = Y_t$ which has kurtosis much larger than 3.

It is apparent that the search for the optimal c will be practically conducted over a discrete grid of c–values spanning an interval of the type $(0, b]$ for some b of the order of one (say). A practical way to narrow in on the optimal c value is to run two grid searches, one coarse followed by a fine one: (i) use a coarse grid search over the whole interval $(0, b]$, and denote $\tilde{c}_0$ the minimizer over the coarse grid search; and (ii) run a fine grid search over a neighborhood of $\tilde{c}_0$.

Let c_0 denote the resulting minimizer from the above algorithm. If needed, the following range–adjustement step may be added.

3. *If c_0 as found above is such that (10.13) is not satisfied, then decrease c stepwise (starting from c_0) over the discrete grid until (10.13) is satisfied.*

Finally, the value of p must be trimmed for efficiency of usage of the available sample; to do this we can simply discard the a_i coefficients that are close to zero, i.e., those that fall below a certain threshold/tolerance level ε which is the practitioner's choice. A threshold value of $\varepsilon = 0.01$ is reasonable in connection with the a_i which—as should be stressed—are normalized to sum to one.

4. *Trim the value of p by a criterion of the type: if $a_i < \varepsilon$, then let $a_i = 0$. If i_0 is the smallest integer such that $a_i < \varepsilon$ for all $i \ge i_0$, then let $p = i_0$ and* renormalize *the a_is so that their sum (for $i = 0, 1, \ldots, i_0$) equals one.*

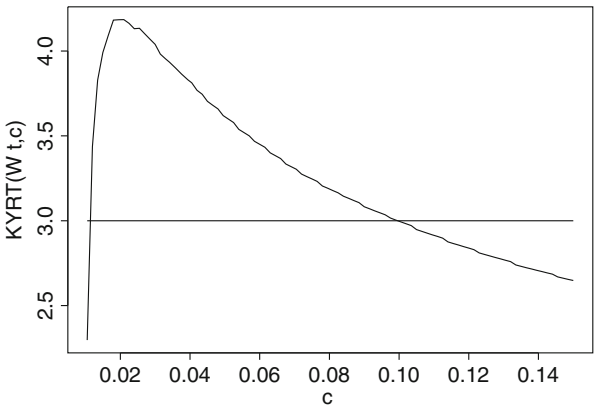

Fig. 10.7 Illustration of the Exponential NoVaS algorithm for the Yen/Dollar dataset: plot of $KURT_n(W_{t,c}^E)$ as a function of c; the *solid line* indicates the Gaussian kurtosis of 3

An illustration of the Exponential NoVaS algorithm is given for the Yen/Dollar dataset. Figure 10.7 is a plot of $KURT_n(W_{t,c}^E)$ as a function of c. Except for values of c very close to zero, $KURT_n(W_{t,c}^E)$ seems to be monotonically decreasing hitting the value 3 for $c \simeq 0.097$. Nevertheless, the behavior of $KURT_n(W_{t,c}^E)$ for c close to zero is not a fluke; rather it is a predictable outcome of our truncation/clipping of all coefficients that are less than ε (which was equal to 0.01 for the purposes of Fig. 10.7). If a very low value for ε is used—say even that ε is set to zero—then the plot of $KURT_n(W_{t,c}^E)$ would be decreasing for all values of c.

To further elaborate, note that Fig. 10.7 indicates $KURT_n(W_{t,c}^E)$ hitting the value 3 for another value of c as well, namely for $c \simeq 0.011$. Figure 10.8 shows a plot of the exponential coefficients a_i versus the index $i = 1, \ldots, p$ for the two values of c suggested by Fig. 10.7; due to the truncation effect with ε= 0.01, we have $c \simeq 0.011$ corresponding to $p = 10$, while $c \simeq 0.097$ corresponds to $p = 22$. Note that the ultra-slow decay of the a_i coefficients in the case $c \simeq 0.011$, combined with the truncation effect at $p = 10$, makes the Exponential NoVaS with $c \simeq 0.011$ very similar to a Simple NoVaS with $p = 10$; this is because the exponential coefficients decay so slowly that are close to being constant for $i = 1, \ldots, p$.

To sum up: a plot with shape such as Fig. 10.7 is typical when a nonzero ε is used, suggesting that the function $|KURT_n(W_{t,c}^E) - 3|$ may have two values of c minimizing it. The higher of those two c values is the *bona fide* exponential decay constant. The lower of the two c values is typically associated with very slow decay of the exponential coefficients which—after truncation at the pth term—appear almost constant, thus approximating the Simple NoVaS coefficients based on p terms. Hence, a plot with shape such as Fig. 10.7 is doubly informative as it can give both the Exponential and the Simple NoVaS solutions.

Analogs of Figs. 10.5 and 10.6 can be constructed using the Exponential NoVaS algorithm on our three datasets; they are not given here to save space as they are visually very similar to the Simple NoVaS results of Figs. 10.5 and 10.6. The optimal c values were: 0.066 (with $p = 27$) for the IBM dataset, and 0.079 (with $p = 24$) for the S&P500 dataset.

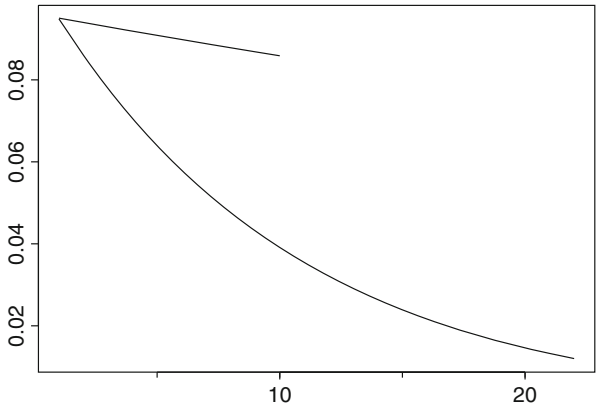

Fig. 10.8 Plot of the exponential coefficients a_i versus the index $i = 1, \ldots, p$ for the two values of c suggested by Fig. 10.7; note that $c \simeq 0.011$ corresponds to $p = 10$, while $c \simeq 0.097$ corresponds to $p = 22$

As in Remark 10.3.3, for the Exponential NoVaS as well we could focus on moment matching for the linear combinations of $W_{t,c}^E$ of $W_{u,c}^E$ (say) instead of $W_{t,c}^E$. In addition, the Exponential NoVaS algorithm could be extended to include a sum of two or more exponentials, i.e., a situation where $a_i = c' e^{-ci} + d' e^{-di} \cdots$. The generalization may well include higher order moment matching and/or looking at linear combinations of higher order lags.

Remark 10.3.4 [Alternative optimization criteria] Instead of kurtosis matching, there exist different optimization criteria that can alternatively be employed in order to choose the NoVaS parameters both in Exponential and Simple NoVaS. For example, one can instead optimize the Shapiro and Wilk (1965) normality test score associated with the variables $W_{t,a}$. Interestingly, Shapiro-Wilk optimization gives optimization results that are practically indistinguishable from moment matching. The intuitive reason is that the non-normality in financial returns appears to be mainly due to the heavy tails; see Politis and Thomakos (2008, 2012) for more details.

10.4 Model-Based Volatility Prediction

10.4.1 Some Basic Notions: L_1 vs. L_2

In this section, we consider the problem of prediction of Y_{n+1}^2 based on the observed past $\mathcal{F}_n = \{Y_t, 1 \le t \le n\}$. Under the zero mean assumption, a first predictor is given by a simple empirical estimator of the (unconditional) variance σ_Y^2 of the series

$\{Y_t, 1 \le t \le n\}$, for example, $s_n^2 = n^{-1}\sum_{k=1}^{n} Y_k^2$; this will serve as our "benchmark" for comparisons.

The above predictor is quite crude as it implicitly assumes that the squared returns $\{Y_t^2, 1 \le t \le n\}$ are independent which is typically not true. As a matter of fact, the basic premise regarding financial returns is that they are dependent although uncorrelated—hence the typical assumption of nonlinear/non-normal models in that respect. For example, Fig. 10.9a confirms that for the Yen/Dollar dataset the returns indeed appear uncorrelated. However, the squared returns appear to be correlated even for lags as high as 25 days; see Fig. 10.9b.

An immediate improvement over the above naive benchmark should thus be obtainable by a simple forecasting method such as "exponential smoothing"; see, e.g., Hamilton (1994). In our context, the exponential smoothing predictor of Y_{n+1}^2 is of the form $\sum_{k=1}^{q} \delta^k Y_{n-k}^2 / \sum_{j=1}^{q} \delta^j$, where δ is a number in $(0,1)$ and q an appropriate practical truncation limit.

Remark 10.4.1 The specification of q in exponential smoothing resembles closely the choice of p in our Exponential NoVaS algorithm. Note, however, that in Exponential NoVaS, the technique of exponential weighting was used with the purpose of constructing a (local) estimate of the variance of Y_t for the subsequent NoVaS

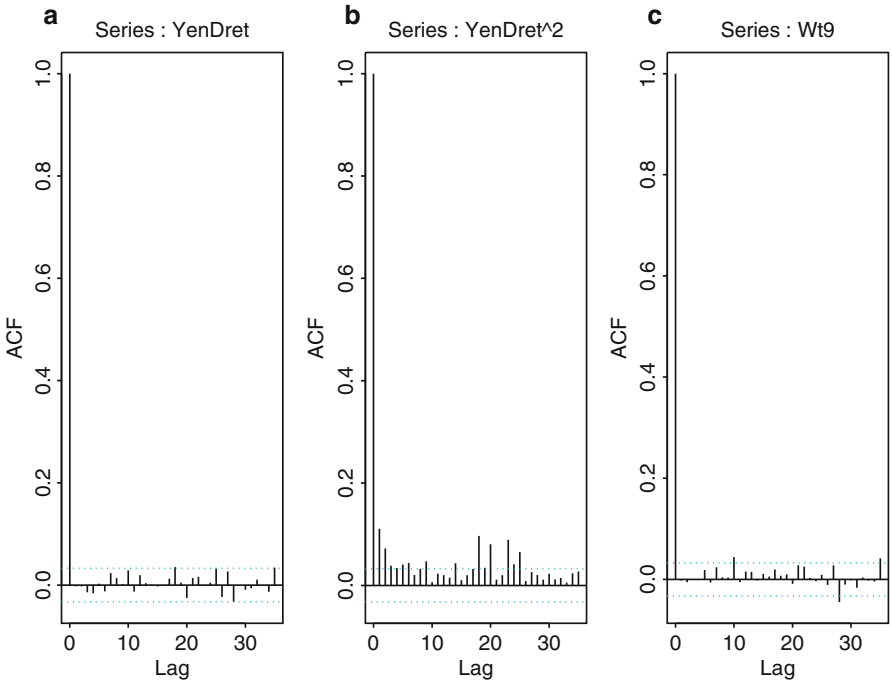

Fig. 10.9 (Yen/Dollar example) (**a**) Correlogram of the returns series $\{Y_t\}$; (**b**) correlogram of the squared returns $\{Y_t^2\}$; (**c**) correlogram of the optimal Simple NoVaS series $\{W_{t,9}^S\}$

Method	Yen/Dollar	S&P500	IBM
Exponential Smoothing with CV	1.065	1.151	1.198
Eq. (10.14)—AR fit with AIC	1.013	1.002	1.034
Eq. (10.3)—GARCH(1,1) with normal errors	1.029	1.094	1.117
Eq. (10.3)—GARCH(1,1) with t–errors	1.051	1.102	1.139

Table 10.2 Entries give the empirical Mean Squared Error (MSE) of prediction of squared returns relative to benchmark; note that the MSE of prediction achieved by the benchmark was 2.96e-008, 1.70e-006, 1.78e-006 in the three cases Yen/Dollar, S&P500, IBM, respectively [Predictor type: conditional mean]

studentization. By contrast, in the present section exponential smoothing is considered in its usual function of forecasting/predicting the (future) value of the random variable Y_{n+1}^2.

The "discount" factor δ in exponential smoothing is typically chosen by a cross-validation (CV) step with the forecasting goal in mind; that is, δ is chosen to minimize the Mean Squared Error (MSE) of prediction within the available dataset. The CV method is intuitive and ubiquitous; for example, it is built-in in many statistical software packages such as the ITSM time series software accompanying the highly influential book by Brockwell and Davis (1991). Despite its appeal, however, rigorous analysis of the performance of the CV method is difficult and has been lacking in the literature. A notable exception is the paper by Gijbels et al. (1999) where the performance of the CV method is successfully analyzed under a model of the type: deterministic trend function plus error. Unfortunately, such a model cannot be reasonably assumed in connection with our (squared) returns series. Consequently, the results in the first row of Table 10.2 should not come as a surprise. The exponential smoothing predictor is seen to perform very poorly; in fact, it is performing quite worse than our naive benchmark predictor which is our estimate of the (unconditional) variance $\sigma_Y^2 = EY_t^2$.

Note that the "exponential smoothing" predictor is linear in the variables $\{Y_t^2, 1 \leq t \leq n\}$ but the coefficients in the linear combination are not chosen according to an optimality criterion. As a matter of fact, exponential smoothing is analogous to fitting an MA(1) model to the squared returns. However, as discussed in Chap. 6, linear prediction has been traditionally approached by fitting AR models of appropriately high order. Thus, a further step in constructing a good predictor of Y_{n+1}^2 may be to fit an AR(r) model to the (de-meaned) squared returns $\{Y_t^2, 1 \leq t \leq n\}$ with the order r determined by minimizing the aforementioned AIC criterion. Denoting by ϕ_i the fitted AR coefficients leads to a linear predictor of Y_{n+1}^2 which is of the form

$$(1 - \sum_{i=1}^{r} \phi_i) s_n^2 + \sum_{i=1}^{r} \phi_i Y_{n+1-i}^2. \tag{10.14}$$

It should be noted though that this linear predictor is typically suboptimal since the series $\{Y_t^2\}$ is generally non-normal and nonlinear. However, the main reason that Eq. (10.14) may give a poor predictor in practice is the following: the correlogram of the squared returns $\{Y_t^2, 1 \le t \le n\}$ does *not* give an accurate estimation of the true correlation structure mainly due to the underlying heavy tails (and nonlinearities); see, e.g., Resnick et al. (1999). For example, using the AIC criterion to pick the order r in connection with the squared Yen/Dollar returns yields $r = 26$; this is not surprising in view of the correlogram of Fig. 10.9b, but it is hard to seriously entertain a model of such high order for this type of data. An experienced researcher might instead fit a low order AR or ARMA model in this situation.

As mentioned before, fitting an MA(1) model for prediction is closely related to the exponential smoothing forecaster. Interestingly, fitting an ARMA(1,1) to the squared returns is in the spirit of a GARCH(1,1) model since the GARCH(1,1) predictor of Y_{n+1}^2 has the same form as predictor (10.14) with the ϕ_i coefficients following the structure of an ARMA(1,1) model.

The GARCH(1,1) model (10.5) is the most popular among the GARCH(p,q) models as it is believed to achieve the most parsimonious fit for financial returns data. As previously mentioned, the ARCH family is a subset of the GARCH family since an ARCH(p) model is equivalent to a GARCH$(p,0)$; in addition, a GARCH(p,q) model is equivalent to an ARCH(∞) with a special structure for its a_i coefficients—see, e.g., Francq and Zakoian (2011).

In order to compare the different predictors of squared returns, we will use two popular performance measures: Mean Squared Error (MSE) of prediction and Mean Absolute Deviation (MAD) of prediction both relative to the benchmark; these are of course nothing other than the L_2 and L_1 norms of the prediction error, respectively, divided by the corresponding L_2 or L_1 norm of the benchmark's prediction error. Hansen et al. (2003) have also compared volatility predictions using both L_2 and L_1 loss functions.

Table 10.2 reports the L_2 prediction performance of the aforementioned predictors, namely exponential smoothing with CV, the linear model (10.14) with order chosen by minimizing the AIC, and the GARCH(1,1) with normal and t–errors (the latter having degrees of freedom estimated from the data). It is apparent that the performance of all methods is rather poor as they seem to perform worse even than our naive benchmark. In particular, the performance of the GARCH(1,1) predictor is abysmal, be it with normal or t errors.

Due to empirical results such as those in Table 10.2, it had been widely believed around the turn of the twenty-first century that ARCH/GARCH models are characterized by "poor out-of-sample forecasting performance vis-a-vis daily squared returns"; see Andersen and Bollerslev (1998) and the references therein. To further quote Andersen and Bollerslev (1998): "numerous studies have suggested that ARCH and stochastic volatility models provide poor volatility forecasts." As a remedy, Andersen and Bollerslev (1998) defined the notion of "latent" volatility based on an assumed underlying continuous-time diffusion structure, and showed that

Method	Yen/Dollar	S&P500	IBM
Exponential Smoothing with CV	1.053	1.127	1.032
Eq. (10.14)—AR fit with AIC	0.929	0.913	0.883
Eq. (10.3)—GARCH(1,1) with normal errors	0.912	0.979	0.838
Eq. (10.3)—GARCH(1,1) with t–errors	0.924	0.974	0.849

Table 10.3 Entries give the empirical Mean Absolute Deviation (MAD) of prediction of squared returns relative to benchmark; note that the MAD of prediction achieved by the benchmark was 1.56e-004, 2.33e-004 in the three cases Yen/Dollar, S&P500, IBM, respectively [Predictor type: conditional mean]

ARCH/GARCH models are successful in predicting future "latent" volatility instead. Nevertheless, the entries of Table 10.3 on the L_1 prediction performance tell a different story, namely that all aforementioned predictors—with the exception of exponential smoothing—outperform the benchmark when errors are measured in the L_1 norm!

10.4.2 Do Financial Returns Have a Finite Fourth Moment?

To see why such a big discrepancy exists between the two performance measures, L_1 and L_2, we return to our data. Let $VAR_k(Y)$ and $KURT_k(Y)$ denote the empirical (sample) variance and kurtosis of dataset Y up to time k, i.e., $\{Y_1, \ldots, Y_k\}$. By the (strong) law of large numbers, as k increases, $VAR_k(Y)$ should tend to the variance of the random variable Y_1 be that infinite or not. Similarly, $KURT_k(Y)$ should tend to the kurtosis of Y_1 be that infinite or not. Thus, plotting $VAR_k(Y)$ and $KURT_k(Y)$ as functions of k one may be able to visually gauge whether Y_1 has finite second and/or fourth moments; this is done in Fig. 10.10 for the Yen/Dollar dataset.

It appears that the Yen/Dollar dataset has finite variance as the plot in Fig. 10.10a seems to converge. Nevertheless, it seems that it may well have an infinite fourth moment as the plot in Fig. 10.10b seems to diverge with each extreme value "jolt." The same conclusions, namely finite variance but infinite fourth moment, seem to also apply to our other two datasets—and indeed to several other financial returns data; see, e.g., Politis (2004, 2007a).

Therefore, it is hardly surprising that the L_2 measure of prediction performance yields unintuitive results: the MSE of predicting Y_{n+1}^2 is essentially a fourth moment, and the data suggest that fourth moments may well be infinite! It is unreasonable to use an L_2 measure of performance in a setup where L_2 norms may not exist.

In addition, note that the GARCH predictions for Tables 10.2 and 10.3 were performed—as customary—using the estimated volatility as a predictor of Y_{n+1}^2,

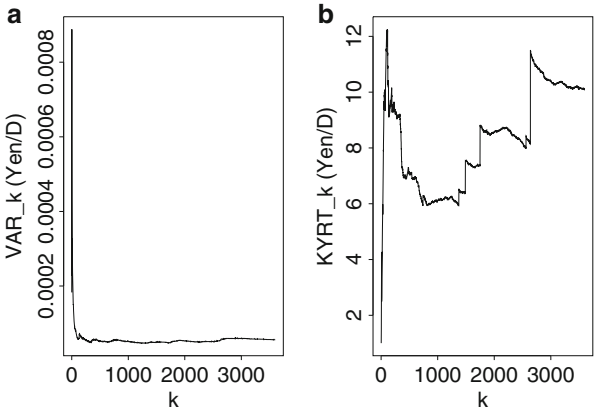

Fig. 10.10 (Yen/Dollar example) (**a**) Plot of $VAR_k(Y)$ as a function of k; (**b**) plot of $KURT_k(Y)$ as a function of k

i.e., the GARCH predictor is $h_n^2 = C + AY_n^2 + B\hat{h}_n^2$ after plugging-in estimates of the GARCH parameters C,A, and B. But h_n^2 is the conditional expectation of Y_{n+1}^2 (given $\mathscr{F}_n$) which is the L_2–optimal predictor of Y_{n+1}^2. Since the evidence leans heavily against the existence of finite fourth moments for financial returns, it seems pointless to use a predictor that is optimal for a criterion that is not well-defined.

Under the objective of L_1–optimal prediction, the optimal predictor is the conditional median—*not* the conditional expectation. Under an ARCH(p) model, the L_1–optimal predictor of Y_{n+1}^2 is given by

$$Median\left(Y_{n+1}^2|\mathscr{F}_n\right) = (a + \sum_{i=1}^{p} a_i Y_{n+1-i}^2) Median(Z_{n+1}^2); \tag{10.15}$$

note that $Median(Z_{n+1}^2) \simeq 0.45$ if $Z_t \sim N(0,1)$, whereas $Median(Z_{n+1}^2) \simeq 0.53$ if $Z_t \sim t_5$.

Furthermore, the aforementioned equivalence of GARCH(1,1) with an ARCH(∞) implies that Eq. (10.15) would also give the L_1–optimal GARCH(1,1) predictor of Y_{n+1}^2 by allowing $p = \infty$, and letting the ARCH coefficients $a, a_1, a_2, \ldots$ follow the

Method	Yen/Dollar	S&P500	IBM
Eq. (10.15)—GARCH(1,1) with normal errors	0.790	0.914	0.835
Eq. (10.15)—GARCH(1,1) with t–errors	0.797	0.901	0.844

Table 10.4 Entries give the empirical Mean Absolute Deviation (MAD) of prediction of squared returns relative to benchmark; Eq. (10.15) is coupled with a very large choice of p and Eq. (10.6) [Predictor type: conditional median.]

structure given by Eq. (10.6). Table 10.4 shows the L_1 prediction performance of our two GARCH(1,1) models using the optimal L_1 predictor (10.15) in connection with a very large choice of p and Eq. (10.6).

As expected, using the correct predictor leads to ameliorated performance; compare Table 10.4 to Table 10.3. In particular, *both* GARCH(1,1) models outperform the lineal predictor (10.14) in the L_1 sense. Interestingly, assuming the t distribution for the errors does not seem to give any appreciable advantage—if at all— over the customary normal errors assumption.

To conclude, by contrast to what was widely believed, ARCH/GARCH models do have predictive validity for the squared returns. However, to appreciate and take advantage of this one must: (a) use a more meaningful measure of prediction such as L_1, and (b) use the proper predictor, i.e., the conditional median in the L_1 case.

In the sequel we will focus exclusively on the L_1 measure of prediction performance. Although we have seen that GARCH models do have reasonable predictive validity, we will show how we can obtain even better volatility prediction using the Model-free approach, i.e., the NoVaS transformation.

Remark 10.4.2 [On "honest" predictions] All predictions reported in this chapter are "honest" in the sense that to predict Y_{t+1}^2, only information set $\{Y_1, \ldots, Y_t\}$ was used in estimating the particulars of the predictor, be it GARCH parameters, AIC for AR–fitting, the exponential smoothing constant, or NoVaS parameters in what follows. Furthermore, we did not follow the usual practice of splitting the dataset in half, estimating parameters from the first half and predicting the second half; this is unrealistic as parameters would/should be updated constantly in practice. For the purposes of our numerical work, however, it was deemed unnecessary—and computationally too expensive—to update the parameters daily. The updating was performed every $n/10$ days for a daily dataset of size n. However, for the S&P500 dataset the updating was only done every $n/7$ days due to convergence issues in the numerical MLEs involved in GARCH modeling.

10.5 Model-Free Volatility Prediction

We now revisit the volatility prediction problem through the viewpoint of the Model-free Prediction Principle of Chap. 2. Note that the NoVaS transformation developed in Sect. 10.3 was shown to empirically transform the data into a Gaussian time series. Hence, the Gaussian stepping stone of Sect. 2.3.2 has already been established; as will be shown in the sequel, the transformation towards i.i.d.–ness is straightforward.

10.5.1 Transformation Towards i.i.d.–Ness

As already mentioned, the NoVaS transformation is the Gaussian stepping stone towards i.i.d.–ness. Thus, suppose that the NoVaS parameters, i.e., the order $p(\geq 0)$ and the parameters $\alpha, a_0, \ldots, a_p$ have already been chosen. We can re-arrange the NoVaS equation (10.8) to yield:

$$Y_t^2 = \frac{W_{t,a}^2}{1 - a_0 W_{t,a}^2} \left(\alpha s_{t-1}^2 + \sum_{i=1}^{p} a_i Y_{t-i}^2 \right) \quad \text{for } t = p+1, \ldots, n \tag{10.16}$$

and

$$Y_t = \frac{W_{t,a}}{\sqrt{1 - a_0 W_{t,a}^2}} \sqrt{\alpha s_{t-1}^2 + \sum_{i=1}^{p} a_i Y_{t-i}^2} \quad \text{for } t = p+1, \ldots, n. \tag{10.17}$$

Following the Model-free Prediction Principle, the one-step ahead prediction problem can be defined as follows. Let $g(\cdot)$ be some (measurable) function of interest; examples include $g_0(x) = x$, $g_1(x) = |x|$, and $g_2(x) = x^2$, the latter being the function of interest for volatility prediction. From Eq. (10.17) it follows that the predictive (given $\mathscr{F}_n$) distribution of $g(Y_{n+1})$ is identical to the distribution of the random variable

$$g\left(A_n \frac{W}{\sqrt{1 - a_0 W^2}} \right) \tag{10.18}$$

where $A_n = \sqrt{\alpha s_n^2 + \sum_{i=1}^{p} a_i Y_{n+1-i}^2}$ is treated as a constant given the past $\mathscr{F}_n$, and the random variable W has the same distribution as the conditional (on $\mathscr{F}_n$) distribution of the random variable $W_{n+1,a}$.

Therefore, our best (in an L_1 sense) prediction of $g(Y_{n+1})$ given $\mathscr{F}_n$ is given by the median of the conditional (given $\mathscr{F}_n$) distribution of $g(Y_{n+1})$, i.e.,

$$\widehat{g(Y_{n+1})} := Median \left(g\left(A_n \frac{W_{n+1,a}}{\sqrt{1 - a_0 W_{n+1,a}^2}} \right) | \mathscr{F}_n \right) \tag{10.19}$$

Specializing to the case of interest, i.e., volatility prediction and the function $g_2(x) = x^2$ yields the NoVaS predictor:

$$\widehat{Y_{n+1}^2} = \mu_2 A_n^2 \tag{10.20}$$

where

$$\mu_2 = Median \left(\frac{W_{n+1,a}^2}{1 - a_0 W_{n+1,a}^2} | \mathscr{F}_n \right).$$

Let $IC = \{Y_1, \ldots, Y_p\}$ denote the initial conditions, and note that the information set $\mathscr{F}_n = \{Y_t, 1 \le t \le n\}$ is equivalent to $\tilde{\mathscr{F}}_n = \{W_{t,a}, p < t \le n\}$ coupled with the set of initial conditions. Due to the assumed stationarity and subject to a usual weak dependence condition—such as strong mixing—on the series $\{Y_t\}$, it also follows that the series $\{W_{t,a}\}$ is stationary and weakly dependent; this implies that $W_{n+1,a}$ will be approximately independent of the initial conditions provided n is quite larger than p; hence, the following approximation can be used, namely

$$\mu_2 = Median\left(\frac{W_{n+1,a}^2}{1 - a_0 W_{n+1,a}^2} | \tilde{\mathscr{F}}_n, IC\right) \simeq Median\left(\frac{W_{n+1,a}^2}{1 - a_0 W_{n+1,a}^2} | \tilde{\mathscr{F}}_n\right) \quad (10.21)$$

in connection with the predictor given in Eq. (10.20).

Our task now is significantly simplified: find the predictive distribution of the random variable $W_{n+1,a}$ based on its own recent past $\tilde{\mathscr{F}}_n$. But—by construction—$W_{t,a}$ should be approximately equal to a normal random variable. In addition, as mentioned in Sect. 10.3, the joint distributions of the series $\{W_{t,a}, t = p+1, \ldots, n\}$ are also typically normalized by the NoVaS transformation. Thus, the series $\{W_{t,a}, t = p+1, \ldots, n\}$ may be thought of as an Gaussian time series in which case optimal prediction is tantamount to optimal linear prediction as discussed in Chap. 6.

Since $\{W_{t,a}, t = p+1, \ldots, n\}$ is a Gaussian time series, it can be conveniently (and accurately) modeled by fitting a causal AR(q) model with a high enough q. Denote by c_i the coefficients, and ε_t the innovations of the fitted AR(q) model, i.e.,

$$\varepsilon_t = W_{t,a} - \sum_{i=1}^{q} c_i W_{t-i,a}. \quad (10.22)$$

By the Hilbert space projection theorem, it follows that the innovations ε_t constitute a mean zero white noise with variance denoted by σ_ε^2. However, due to the aforementioned normality of joint distributions, a stronger result is true, i.e.,

$$\text{the innovations } \varepsilon_t \text{ are i.i.d. } N(0, \sigma_\varepsilon^2). \quad (10.23)$$

Hence, the goal of the Model-free Prediction Principle, i.e., transforming the dataset $\{Y_1, \ldots, Y_n\}$ into an i.i.d. dataset, namely the dataset $\{\varepsilon_t \text{ for } t = p+1, \ldots, n\}$, has now been achieved. Furthermore, the transformation is invertible, i.e., given the initial conditions $\{Y_1, \ldots, Y_p\}$, one can back-transform the dataset $\{\varepsilon_t \text{ for } t = p+1, \ldots, m\}$ into $\{Y_1, \ldots, Y_m\}$ for any $m > p$; Eq. (10.17) is the key to the inverse transformation after the $\{W_{t,a}\}$ are procured based on the AR(q) model (10.22). Therefore, the Model-free Prediction Principle can be immediately invoked in order to construct optimal point (and interval) predictors for $g(Y_{n+1})$.

10.5.2 Volatility Prediction Using NoVaS

The AR(q) model (10.22) implies that our approximation to the best linear predictor of $W_{n+1,a}$ given $\tilde{\mathcal{F}}_n$ is

$$\hat{W}_{n+1,a} = \sum_{i=1}^{q} c_i W_{n-i+1,a}. \tag{10.24}$$

Furthermore, under this Gaussian structure, the whole conditional distribution of $W_{n+1,a}$ given $\tilde{\mathcal{F}}_n$ would be Gaussian with mean equal to $\hat{W}_{n+1,a}$ given in Eq. (10.24), and *constant* variance σ_ε^2, i.e., σ_ε^2 not depending on $\tilde{\mathcal{F}}_n$. Recall that fitting an AR(q) model, e.g., by the Durbin-Levinson algorithm, also gives an estimate of the prediction error variance σ_ε^2 which is the aforementioned variance of the innovations ε_t.

The simplified expression (10.21) still represents an unknown quantity but it could easily be approximated by Monte Carlo, for example using the normal predictive density that has mean given by (10.24) and variance σ_ε^2. Recall, however, that this normal density should be truncated to an effective range of $\pm 1/\sqrt{a_0}$. This procedure would be in the spirit of the Limit Model-Free philosophy of Remark 2.2.4. Note that a very large number of Monte Carlo replications would be required due to the heavy tails of the distribution of $W^2/(1-a_0W^2)$. In addition, it should be stressed that the normal (conditional or unconditional) density for $W_{n+1,a}$ is only an approximation; thus, it may be better to estimate μ_2 empirically from the data without resort to the normal distribution, i.e., using the Model-free Prediction Principle.

Nevertheless, a simpler scenario emerges with regards to financial returns data: the correlogram of the series $\{W_{t,a}, t = p+1,\ldots,n\}$ typically indicates no significant correlations; see, e.g., the Yen/Dollar Simple NoVaS correlogram of Fig. 10.9c. For completeness, we consider both cases below.

- CASE I: THE NOVAS SERIES $\{W_{t,a}\}$ APPEARS TO BE UNCORRELATED. In this case we can infer that the series $\{W_{t,a}\}$ is not only uncorrelated but also independent, i.e., that the c_i coefficients in equation model (10.22) are all zero, and $W_{t,a} = \varepsilon_t$ is its own innovation. Hence, the conditional (on $\tilde{\mathcal{F}}_n$) distribution of $W_{n+1,a}$ would equal the unconditional distribution of $W_{n+1,a}$, and we may estimate μ_2 by the sample median, i.e., let

$$\hat{\mu}_2 = median\{\frac{W_{t,a}^2}{1-a_0W_{t,a}^2};\ t = p+1, p+2,\ldots,n\} \tag{10.25}$$

  and subsequently predict Y_{n+1}^2 by

$$\hat{\mu}_2 A_n^2. \tag{10.26}$$

- CASE II: THE NOVAS SERIES $\{W_{t,a}\}$ APPEARS CORRELATED. Although in all our examples the NoVaS series $\{W_{t,a}, t = p+1,\ldots,n\}$ turned out to be effectively uncorrelated, one cannot preclude the possibility that for other datasets the series $\{W_{t,a}\}$ may exhibit some correlations; in that case, the c_i coefficients in

Eq. (10.22) are not all zero, and a slightly more complicated procedure is required in order to estimate μ_2. First, the predictive residuals must be collected from the data; to do this, let $e_t = W_{t,a} - \hat{W}_{t,a}$ for $t = r+1,\ldots,n$ where $r = \max(p,q)$. Then the conditional (on $\tilde{\mathscr{F}}_n$) distribution of $W_{n+1,a}$ may be approximated by the empirical distribution of the points $\{e_t + \hat{W}_{n+1,a};\ t = r+1,\ldots,n\}$, i.e., by the empirical distribution of the predictive residuals shifted to give it mean $\hat{W}_{n+1,a}$. In that case we would estimate μ_2 by

$$\hat{\mu}_2 = median\{\frac{(e_t + \hat{W}_{n+1,a})^2}{1 - a_0(e_t + \hat{W}_{n+1,a})^2};\ t = r+1, r+2,\ldots,n\} \tag{10.27}$$

and again predict Y^2_{n+1} by Eq. (10.26). Note that the ratio in Eq. (10.25) is always positive and finite since its denominator is bigger than zero by Eq. (10.10). Because of the approximate nature of obtaining the predictive residuals, the same is not necessarily true for the denominator of Eq. (10.27). However, the sample median is robust against such anomalies and would trim away negative values and/or infinities of the ratio found in Eq. (10.27).

Remark 10.5.1 We can generalize the previous discussion to an interesting class of prediction functions $g(\cdot)$ as in Eq. (10.18), namely the power family where $g(x) = x^k$ for some fixed k, and the power–absolute value family where $g(x) = |x|^k$. Let $g_k(x)$ denote either the function x^k or the function $|x|^k$; then Eq. (10.19) suggests that our best predictor of $g_k(Y_{n+1})$ given $\mathscr{F}_n$ is $\widehat{g_k(Y_{n+1})} = \mu_k A^k_n$, where

$$\mu_k = Median\left(g_k\left(\frac{W_{n+1,a}}{\sqrt{1 - a_0 W^2_{n+1,a}}}\right)|\mathscr{F}_n\right).$$

Note that if $Y^4_t = \infty$, then we cannot claim L_1–optimality of μ_k when $k \geq 4$; however, summarizing a predictive distribution by its median is a reasonable thing to do.

As before, μ_k can be estimated by an appropriate sample median. Let us consider Case I and II separately. Under Case I, we estimate μ_k by

$$\hat{\mu}_k = median\{g_k\left(\frac{W_{t,a}}{\sqrt{1 - a_0 W^2_{t,a}}}\right);\ t = p+1, p+2,\ldots,n\}$$

whereas under Case II the estimator becomes

$$\hat{\mu}_k = median\{g_k\left(\frac{e_t + \hat{W}_{n+1,a}}{\sqrt{1 - a_0(e_t + \hat{W}_{n+1,a})^2}}\right);\ t = r+1, r+2,\ldots,n.$$

Remark 10.5.2 (Estimating the volatility $E(Y^2_{n+1}|\mathscr{F}_n)$.) Under Case I, i.e., after empirically showing that the $W_{t,a}$ variables are (approximately) uncorrelated and hence independent, it is straightforward to construct a Model-free estimate of the

Method	Yen/Dollar	S&P500	IBM
Simple NoVaS	0.802	0.818	0.848
Exponential NoVaS	0.778	0.843	0.850

Table 10.5 Entries give the empirical Mean Absolute Deviation (MAD) of prediction of squared returns relative to benchmark [Predictor type: conditional median]

conditional expectation $E(Y_{n+1}^2|\mathscr{F}_n)$. In this case, Eq. (10.16) implies that $E(Y_{n+1}^2|\mathscr{F}_n) = A_n^2 E\left(\frac{W_{t,a}^2}{1-a_0 W_{t,a}^2}\right)$; a natural estimate thereof is

$$\frac{A_n^2}{n-p}\sum_{t=p+1}^{n}\left(\frac{W_{t,a}^2}{1-a_0 W_{t,a}^2}\right)$$

which has validity, e.g., consistency, under the sole assumption that Y_t has a finite second moment conditionally on $\mathscr{F}_n$ (and therefore unconditionally as well).

10.5.3 Optimizing NoVaS for Volatility Prediction

In the previous section, the methodology for volatility prediction based on NoVaS was put forth. Using this methodology the L_1 prediction performance of the Simple and Exponential NoVaS was quantified and tabulated in Table 10.5. In the foreign exchange data, Exponential NoVaS offers some improvement over Simple NoVaS; the situation is reversed in the S&P500 example. In comparison to the two optimal GARCH(1,1) predictors of Table 10.4, both NoVaS methods appear to have an advantage in the Yen/Dollar and S&P500 examples. In the case of IBM returns, all four methods (the two GARCH and two NoVaS) perform comparably.

It is interesting to note that the NoVaS methodology performs competitively in volatility prediction despite its extreme parsimony: both Simple and Exponential NoVaS have just *one* free parameter (p and c, respectively—since the p in Exponentials NoVaS is determined by the tolerance level ε). By contrast, the GARCH(1,1) with normal errors has three free parameters whereas the GARCH(1,1) with t–errors has four—the fourth being the degrees of freedom for the t-distribution.

The single free parameter in Simple and Exponential NoVaS was identified using the kurtosis matching ideas of Sect. 10.3.2. Nevertheless, one can entertain more general NoVaS schemes with two (or more) free parameters. In such setups, one (or more) of the parameters can be identified by kurtosis matching (of the data or lagged linear combinations thereof). The remaining free parameters can then be identified by specific optimality criteria of interest, e.g., optimal volatility prediction; see option (C) in Sect. 2.3.5.

Although many different multi-parameter NoVaS schemes can be devised, we now elaborate on the possibility of a nonzero value for the parameter α in (10.8) in connection with the Simple and Exponential NoVaS. We thus define the Generalized Simple (GS) and Generalized Exponential (GE) NoVaS denoted by $W^{GS}_{t;p,\alpha}$ and $W^{GE}_{t;c,\alpha}$ indicating their respective two free parameters; both are based on Eq. (10.8).

The search is performed using a grid of possible values for α, say $\alpha_1, \alpha_2, \ldots, \alpha_K$, that span a subset of the interval [0,1]. In picking the grid values, note that the kurtosis matching goal may only be possible with small values of α; else, the intermediate value argument of Remark 10.3.1 may fail. For instance, choosing $\alpha = 1$ implies $a_i = 0$ for all i due to Eq. (10.11); in this case, $W_{t,a} \equiv Y_t$, i.e., no transformation is effected.

Algorithm 10.5.1 ALGORITHM FOR GENERALIZED SIMPLE NOVAS

A. For $k = 1, \ldots, K$ perform the following steps.

1. *Let $\alpha = \alpha_k$ and $a_i = (1-\alpha_k)/(p+1)$ for all $0 \le i \le p$ so that Eq. (10.11) is satisfied while all the coefficients $a_0, a_1, \ldots, a_p$ are the same.*
2. *Denote by p_k the minimizer of $|KURT_n(W^{GS}_{t,p}) - 3|$ over values of $p = 1, 2, \ldots$.*
3. *If p_k (and a_0) as found above are such that Eq. (10.13) is not satisfied, then increase p_k accordingly, i.e., re-define $p_k = \lfloor 1 + C^2(1-\alpha_k) \rfloor$, and let $a_i = (1-\alpha_k)/(p_k+1)$ for all $0 \le i \le p_k$ by Eq. (10.11); here, $\lfloor x \rfloor$ denotes the integer part of x.*

B. Finally, compare the transformations $\{W^{GS}_{t;p_k,\alpha_k},\ k = 1, \ldots, K\}$ in terms of their volatility prediction performance, and pick the model with optimal performance.

Algorithm 10.5.2 ALGORITHM FOR GENERALIZED EXPONENTIAL NOVAS

A. For $k = 1, \ldots, K$ perform the following steps.

1. *Let p take a very high starting value, e.g., let $p \simeq n/4$ or $n/5$. Then, let $\alpha = \alpha_k$ and $a_i = c' e^{-ci}$ for all $0 \le i \le p$, where $c' = (1-\alpha_k)/\sum_{i=0}^{p} e^{-ci}$ by Eq. (10.11).*
2. *Pick c in such a way that $|KURT_n(W^{GE}_{t;c,\alpha_k}) - 3|$ is minimized, and denote by c_k the minimizing value.*[3]
3. *Trim the value of p to some value p_k as before: if $a_i < \varepsilon$, then set $a_i = 0$. Thus, if $a_i < \varepsilon$, for all $i \ge i_k$, then let $p_k = i_k$, and renormalize the a_is so that their sum (for $i = 0, 1, \ldots, p_k$) equals $1 - \alpha_k$ by Eq. (10.11).*

B. Finally, compare the transformations $\{W^{GE}_{t;c_k,\alpha_k},\ k = 1, \ldots, K\}$ in terms of their volatility prediction performance, and pick the model with optimal performance.

An illustration of the Generalized Simple and Exponential NoVaS algorithms in connection with our three main datasets is presented in Tables 10.6 and 10.8 where

[3] As before, if c_k is such that (10.13) is not satisfied, then decrease it stepwise over its discrete grid until (10.13) is satisfied.

α	Yen/Dollar	S&P500	IBM
0.0	0.802	0.818	0.848
0.1	0.799	0.792	0.836
0.2	0.798	0.796	0.829
0.3	**0.797**	0.792	0.824
0.4	0.800	**0.788**	0.822
0.5	0.803	0.792	0.822
0.6	0.805	0.796	0.820
0.7	0.804	0.788	**0.815**

Table 10.6 Entries give the empirical Mean Absolute Deviation (MAD) of prediction of squared returns relative to benchmark using the Generalized Simple NoVaS with parameter α; the minimum MAD is given with boldface

α	Yen/Dollar	S&P500	IBM
0.0	9	11	13
0.1	8	9	11
0.2	7	7	10
0.3	6	7	9
0.4	6	6	8
0.5	5	5	8
0.6	5	4	7
0.7	5	4	6

Table 10.7 Entries give the optimal value of p from kurtosis matching in the Generalized Simple NoVaS with parameter α; note that the values in this table were computed using the whole available sample sizes

for each different value of α, the L_1 volatility prediction performance is given. Tables 10.7 and 10.9 give the optimal NoVaS parameters associated with different values of α.

The results in Tables 10.6 and 10.8 are very interesting. Firstly, the L_1 measure appears convex[4] in α making the minimization very intuitive; a unique value of the optimal α (given in bold-face font) is easily found in each of the three datasets.[5] Secondly, although all three datasets seem to benefit from a nonzero value of α, the importance of α differs according to the type of data involved: the Yen/Dollar

[4] Note that for $\alpha = 1$, the corresponding entries of Tables 10.6 and 10.8 would be equal to 1 due to Eq. (10.11).

[5] The only exception is the S&P500 column of Table 10.6; this can be attributed to random error since the discrepancies are small.

α	Yen/Dollar	S&P500	IBM
0.0	0.778	0.843	0.850
0.1	**0.776**	0.832	0.838
0.2	0.778	0.823	0.826
0.3	0.780	0.817	0.822
0.4	0.780	0.810	0.819
0.5	0.782	0.806	0.813
0.6	0.787	**0.804**	**0.809**
0.7	0.796	0.811	0.809

Table 10.8 Entries give the empirical Mean Absolute Deviation (MAD) of prediction of squared returns relative to benchmark using the Generalized Exponential NoVaS with parameter α; the minimum MAD is given with boldface

α	Yen/Dollar	S&P500	IBM
0.0	0.0965	0.0789	0.0660
0.1	0.1111	0.0926	0.0756
0.2	0.1290	0.1084	0.0892
0.3	0.1540	0.1320	0.1042
0.4	0.1882	0.1610	0.1285
0.5	0.2354	0.2097	0.1608
0.6	0.3167	0.2818	0.2133
0.7	0.4649	0.4227	0.3149

Table 10.9 Entries give the optimal exponent c from kurtosis matching in the Generalized Exponential NoVaS with parameter α; note that the values in this table were computed using the whole available sample sizes

series is not so sensitive on the value of the parameter α; the S&P500 index is more sensitive, while the single stock price (IBM) is the most sensitive. Recalling that the IBM case was the only case where NoVaS did not give a definite advantage over GARCH, it is now apparent that *Generalized* NoVaS (Simple or Exponential) outperforms the optimal GARCH(1,1) predictors of Table 10.4 in all three of our datasets.

From Tables 10.7 and 10.9 it is apparent that, as α increases, c increases accordingly, and p decreases. All (p, α) combinations in Table 10.7, and all (c, α) combinations in Table 10.9 were equally successful in normalizing the NoVaS data in terms of achieving a kurtosis of about 3. Finally, note that the values in Tables 10.7 and 10.9 were computed using the whole available sample sizes, whereas the

results in Tables 10.6 and 10.8 were based on "honest" predictions as described in Remark 10.4.2, i.e., prediction of Y_{t+1}^2 was only based on information set $\mathscr{F}_t$ with model updating happening periodically.

10.5.4 Summary of Data-Analytic Findings on Volatility Prediction

- Because of the lack of finite fourth moments, the MSE is not a good measure of performance of prediction of squared returns; see Tables 10.2 and 10.3 where the *same* predictors are compared with respect to MSE and MAD, respectively.
- Using the L_1/MAD measure of performance, it is apparent that GARCH models *do* have predictive validity for the squared returns.
- As expected, once the L_1 setting is assumed, using the optimal predictor of Eq. (10.15) gives an appreciable difference; compare Table 10.3 to Table 10.4.
- Once the optimal GARCH predictor is used, it seems to make little difference whether the normal or t distribution is assumed for the errors; see Table 10.4.
- Simple NoVaS seems comparable to Exponential NoVaS in volatility prediction, and they both outperform the best GARCH(1,1) predictors in the foreign exchange and stock index data examples—see Table 10.5.
- Generalized Simple and Exponential NoVaS gives appreciable improvements over either Simple or Exponential NoVaS; the most significant improvement is seen in the IBM dataset—see Tables 10.6 and 10.8.
- All in all, Generalized NoVaS (Simple and/or Exponential) outperforms the optimal GARCH(1,1) predictors of Table 10.4 in all *three* of our datasets.

10.6 Model-Free Prediction Intervals for Financial Returns

In order to discuss the construction of prediction intervals, we will focus hereafter on Case I of Sect. 10.5.2, i.e., the case where the NoVaS series $\{W_{t,a}\}$ appears to be uncorrelated; in this case, the series $W_{t,a}$ for $t = p+1, \ldots, n$ is tantamount to the i.i.d. series $\varepsilon_t^{(n)}$ featuring in the Model-free Prediction Principle of Chap. 2.

Recall that our best (in an L_1 sense) prediction of $g(Y_{n+1})$ given $\mathscr{F}_n$ was given in Eq. (10.19), i.e.,

$$\widehat{g(Y_{n+1})} = Median\left(g\left(A_n \frac{W_{n+1,a}}{\sqrt{1-a_0 W_{n+1,a}^2}}\right)|\mathscr{F}_n\right)$$

$$= Median\left(g\left(A_n \frac{W_{n+1,a}}{\sqrt{1-a_0 W_{n+1,a}^2}}\right)\right)$$

where the second equality is due to the independence in the series $W_{t,a}$. The above can be seen as a corollary of premise (d) of the Model-free Prediction Principle which also gives a preliminary approximation to the predictive distribution of $g(Y_{n+1})$ given $\mathscr{F}_n$ in the form of the empirical distribution of the random variables $\{g\left(A_n \frac{W_{t,a}}{\sqrt{1-a_0 W_{t,a}^2}}\right) \text{ for } t = p+1,\ldots,n\}$.

However, as discussed in Remark 2.2.3, this "plug-in" empirical distribution ignores the variability of estimated parameters in the construction of the NoVaS transformation; to incorporate this variability, Model-free resampling is needed as well. Note that the point predictor $\widehat{g(Y_{n+1})}$ is a function only[6] of $Y_n,\ldots,Y_{n-p+1}$, i.e., is a predictor of the type of a (nonlinear) AR model or Markov process of order p. Hence, to develop the relevant resampling algorithms, we can borrow some ideas from Chaps. 7 and 8; in particular, we will adopt the "forward" bootstrap methodology.

The basic Model-free (MF) bootstrap algorithm for prediction intervals in the setting of financial returns goes as follows.

Algorithm 10.6.1 MF BOOTSTRAP PREDICTION INTERVALS FOR $g(Y_{n+1})$

1. *Use one of the NoVaS algorithms (Simple vs. Exponential, Generalized or not, etc.) to obtain the transformed data* $\{W_{t,a}$ *for* $t = p+1,\ldots,n\}$ *that are assumed to be approximately i.i.d.*[7] *Let* p, α *and* a_i *denote the fitted NoVaS parameters.*
2. *Calculate* $\widehat{g(Y_{n+1})}$, *the point predictor of* $g(y_{n+1})$, *as the median of the set* $\{g\left(A_n \frac{W_{t,a}}{\sqrt{1-a_0 W_{t,a}^2}}\right)$ *for* $t = p+1,\ldots,n\}$; *recall that* $A_n = \sqrt{\alpha s_n^2 + \sum_{i=1}^p a_i Y_{n+1-i}^2}$.

3.(a) *Resample randomly (with replacement) the transformed variables* $\{W_{t,a}$ *for* $t = p+1,\ldots,n\}$ *to create the pseudo-data* $W_{p+1}^*,\cdots,W_{n-1}^*,W_n^*$ *and* W_{n+1}^*.

(b) *Let* $(Y_1^*,\ldots,Y_p^*)' = (Y_{1+I},\cdots,Y_{p+I})'$ *where* I *is generated as a discrete random variable uniform on the values* $0,1,\ldots,n-p$.

(c) *Generate the bootstrap pseudo-data* Y_t^* *for* $t = p+1,\ldots,n$ *using Eq.* (10.17), *i.e., let*

$$Y_t^* = \frac{W_t^*}{\sqrt{1-a_0 W_t^{*2}}} \sqrt{\alpha s_{t-1}^{*2} + \sum_{i=1}^p a_i Y_{t-i}^{*2}} \text{ for } t = p+1,\ldots,n \tag{10.28}$$

where $s_{t-1}^{*2} = (t-1)^{-1}\sum_{k=1}^{t-1} Y_k^{*2}$.

(d) *Based on the bootstrap data* $Y_1^*,\ldots,Y_n^*$, *re-estimate the NoVaS transformation yielding parameters* p^*, α^*, $a_0^*, a_1^*,\ldots,a_{p^*}^*$. *Let* $A_n^* = \sqrt{\alpha^* s_n^2 + \sum_{i=1}^{p^*} a_i^* Y_{n+1-i}^2}$, *and calculate the bootstrap predictor* $\widehat{g(Y_{n+1}^*)}$ *as the median of the set*

[6] In the case of Generalized NoVaS (Simple or Exponential), $\widehat{g(Y_{n+1})}$ is also a function of s_n^2 which, however, converges to EY_t^2 for large n; hence, it can be treated as constant for all practical purposes.

[7] Otherwise, a further transformation step will be required as discussed in Case II of Sect. 10.5.2.

$$\{g\left(A_n^* \frac{W_{t,a}}{\sqrt{1-a_0^* W_{t,a}^2}}\right) \quad for \ t = p+1,\ldots,n\} \tag{10.29}$$

using the convention[8] *that when* $1 - a_0^* W_{t,a}^2 \leq 0$, *we assign* $\frac{1}{\sqrt{1-a_0^* W_{t,a}^2}} = \infty$.

(e) Calculate the bootstrap future value Y_{n+1}^* *as*

$$Y_{n+1}^* = \frac{W_{n+1}^*}{\sqrt{1-a_0 W_{n+1}^{*2}}} \sqrt{\alpha s_n^2 + \sum_{i=1}^{p} a_i Y_{n-i+1}^2}. \tag{10.30}$$

(f) Calculate the bootstrap root: $g(Y_{n+1}^*) - \widehat{g(Y_{n+1}^*)}$.

4. *Repeat step 3 above B times; the B bootstrap root replicates are collected in the form of an empirical distribution whose* α*-quantile is denoted* $q(\alpha)$.
5. *The* $(1-\alpha)100\%$ *equal-tailed prediction interval for* $g(Y_{n+1})$ *is given by*

$$[\widehat{g(Y_{n+1})} + q(\alpha/2),\ \widehat{g(Y_{n+1})} + q(1-\alpha/2)].$$

Note that the last p values from the *original* data, i.e., $Y_{n-p+1},\ldots,Y_n$, are used in both the creation of the bootstrap predictor in Eq. (10.29) and bootstrap future value in Eq. (10.30); this is in accordance with the "forward" bootstrap methodology of Chaps. 7 and 8 but also with the general Model-free Bootstrap described in Algorithm 2.4.1.

An LMF version of Algorithm 10.6.1 can also be devised; it would amount to replacing Step 3 (a) by:
(a′) Generate $W_{p+1}^*,\cdots,W_{n-1}^*,W_n^*$ *and* W_{n+1}^* *as i.i.d. from a* $N(0,1)$ *distribution truncated to* $\pm 1/\sqrt{a_0}$.

10.7 Time-Varying NoVaS: Robustness Against Structural Breaks

Up to this point, the series of financial returns has been assumed to be strictly stationary. Nevertheless, if the data $Y_1,\ldots,Y_n$ span a long-time interval, e.g., 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.

[8] This is because the original NoVaS data satisfies $|W_{t,a}| \leq 1/\sqrt{a_0}$ but a_0^* might turn out bigger (or smaller) than a_0. Alternatively, one can base Eq. (10.29) on the NoVaS transformed series $W_{t,a}^*$ that corresponds to the bootstrap data $Y_1^*,\ldots,Y_n^*$, or on a Monte Carlo experiment using a $N(0,1)$ distribution truncated to $\pm 1/\sqrt{a_0^*}$. All these options are practically indistinguishable as far as taking the median is concerned, and Eq. (10.29) is the most straightforward.

Instead, one can assume a slowly-changing stochastic structure, i.e., a locally stationary model as discussed in Chap. 9. Indeed, the theory of time–varying ARCH (TV-ARCH) processes was developed to capture such a phenomenon; see Dahlhaus and Subba Rao (2006). The analysis of a time–varying ARCH/GARCH model can be based on the premise of local stationarity. For example, in order to predict $g(Y_{t+1})$ based on $\mathscr{F}_t$ via a time–varying GARCH(1,1) model, we can simply fit model (10.5) using as data the subseries $Y_{t-b+1}, \ldots, Y_t$. Here, the window size b should be large enough so that accurate estimation of the GARCH parameters is possible based on the subseries $Y_{t-b+1}, \ldots, Y_t$ but small enough so that such a subseries can plausibly be considered stationary.

In a similar vein, we can predict $g(Y_{t+1})$ by fitting one of the NoVaS algorithms (Simple vs. Exponential, Generalized or not) just using the "windowed" data $Y_{t-b+1}, \ldots, Y_t$. In so doing, we are constructing a **time–varying NoVaS** (TV-NoVaS) transformation. In numerical work, Politis and Thomakos (2008, 2012) showed that NoVaS fitting can be done more efficiently than GARCH fitting by (numerical) MLE. Thus, it is expected that TV-NoVaS may be able to capture a changing stochastic structure in a more flexible manner; stated in different term, the window size b required for accurate NoVaS fitting should be smaller than the one required for accurate GARCH fitting.

We investigate this conjecture in a small simulation experiment. Before describing it, note that an alternative form of nonstationarity is due to the possible presence of structural breaks, i.e., change points, occuring at some isolated time points. Mikosch and Starica (2004), and Starica and Granger (2005) show the interesting effects that an undetected change point may have on our interpretation and analysis of ARCH/GARCH modeling.[9] Hence, in the simulation that follows, we also include a structural break model in order to see the effect of an undetected change point on the performance of TV-NoVaS and TV-GARCH volatility predictors.

For the simulation, 500 datasets $\underline{Y}_n = (Y_1, \ldots, Y_n)'$ were constructed using either a TV-GARCH or a change point GARCH (CP-GARCH); these were defined using the standard GARCH model $Y_{t+1} = h_{t+1} Z_{t+1}$ with $h_{t+1}^2 = C + AY_t^2 + Bh_t^2$ as building block with $C = 10^{-5}$. The i.i.d. errors Z_t are commonly assumed to have a Student t_5 distribution; instead, we use the simple assignment $Z_t \sim$ i.i.d. $N(0,1)$ in the simulation in order to facilitate the convergence of the numerical (Gaussian) MLE in fitting the TV-GARCH model.

CP-GARCH: For $t \leq n/2$, let $A = 0.10$ and $B = 0.73$; for $t > n/2$, let $A = 0.05$ and $B = 0.93$. These values are close to the ones used by Mikosch and Starica (2004).

TV-GARCH: The value of A decreases as a linear function of t, starting at $A = 0.10$ for $t = 1$, and ending at $A = 0.05$ for $t = n$. At the same time, the value of B increases as a linear function of t, starting at $B = 0.73$ for $t = 1$, and ending at $B = 0.93$ for $t = n$.

[9] Kokoszka and Leipus (2000), and Berkes et al. (2004) have studied the detection/estimation of change points in ARCH/GARCH modeling.

The difference between the two models, CP-GARCH and TV-GARCH, is an abrupt vs. smooth transition spanning the same values. Some more information on the simulation follows.

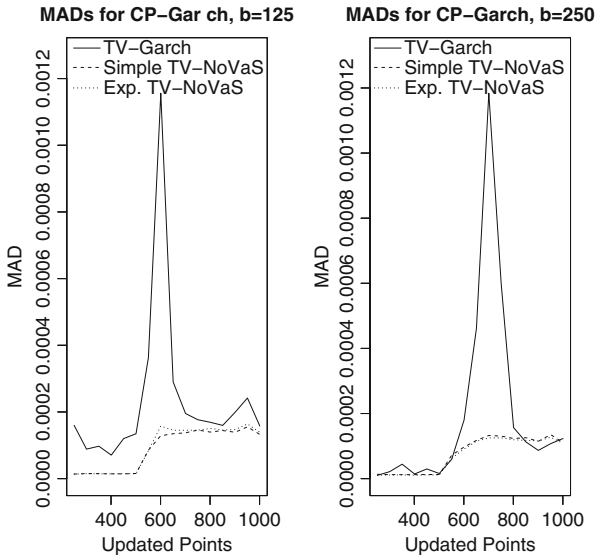

Fig. 10.11 MAD of prediction of squared returns obtained by fitting TV-GARCH vs. TV-NoVaS; data from CP-GARCH model

- The prediction method employed was the conditional median obtained either from a TV-GARCH model fitted by windowed Gaussian MLE, or via TV-NoVaS (Simple or Exponential); in either case, two window sizes were tried out, namely $b = 125$ or 250.
- The sample size was $n = 1001$ corresponding to about 4 years of daily data; so the choices $b = 125$ and 250 correspond to 6 and 12 months, respectively.
- Training period for all methods was 250, i.e., the experiment amounted to predicting Y_{t+1}^2 from the "windowed" data $Y_{t-b+1}, \ldots, Y_t$ for $t = 250, 251, \ldots, 1000$.
- Updating (re-estimation) of all methods would ideally be for each $t = 250, 251, \ldots, 1000$. To save computing time, updating in the simulation was only performed for t being an integer multiple of 50. In fairness, the performance of predictions was recorded and compared *only* at the moment of updating the model, i.e., at time points $250, 300, 350, \ldots, 1000$.

Figure 10.11 shows the MAD of volatility prediction of TV-GARCH as compared to the MAD of TV-NoVaS (Simple or Exponential) with data from model CP-GARCH for the 16 time points where the updating and prediction occured, i.e., the time points $250, 300, 350, \ldots, 1000$. Each point in the figure gives the absolute value of the prediction error at the update time point averaged over the 500

replications; the left panel depicts the case $b = 125$ while the right panel depicts the case $b = 250$. Figure 10.12 is similar but using data generated by a TV-GARCH model instead.

Some conclusions are as follows:

- Time points $250, 300, 350, 400, 450$, and 500 in the left panel of Fig. 10.11 corroborate the aforementioned fact that NoVaS (Simple or Exponential) beats GARCH for prediction of squared returns even if the data generating model is (stationary) GARCH as long as the sample size available for model-fitting is small—equal to 125 in this case. The corresponding points in the right panel of Fig. 10.11 indicate that GARCH manages to do as well as (or better than)[10] NoVaS when the effective sample size is increased to 250.
- Figure 10.11 shows that the change point at $t = 500$ wreaks havoc in GARCH model fitting and the associated predictions; this adds another dimension to the observations of Mikosch and Starica (2004). By contrast, both NoVaS methods seem to adapt immediately to the new regime that occurs after the unknown/undetected change point.
- Figure 10.12 shows that TV-NoVaS (Simple or Exponential) beats TV-GARCH for prediction of squared returns even when the data generating model is TV-GARCH. Not only is the MAD of prediction of TV-NoVaS just a small fraction of that of TV-GARCH, but the wild swings associated with the latter indicate the inherent instability of GARCH model-fitting; this instability is prominent even in this simplistic case where the errors have a true Gaussian distribution, and Gaussian MLE is used for estimating just the three GARCH parameters.
- As seen in both Figs. 10.11 and 10.12, the performance of Simple NoVaS is practically indistinguishable from that of Exponential NoVaS although upon closer look the latter appears to be marginally better.

Acknowledgements

Sections 10.1–10.5 are based on the paper: D.N. Politis, "Model–free vs. model–based volatility prediction," *J. Financial Econometrics*, vol. 5, no. 3, pp. 358–389, 2007. Many thanks are due to René Garcia and Eric Renault, Founding Editors of the Journal of Financial Econometrics, for hosting this paper, and to Oxford University Press for their kind permission to reuse some of the material. Note that the entries of Tables 5 and 6 in the abovementioned paper were incorrect due to a software bug discovered by C. Meggiris. Many thanks are due to Jie Chen for re-working all the numerical examples of the chapter using the corrected software, and for running the simulation presented in Sect. 10.7. Finally, note that the whole of Chap. 10 focused

[10] Note that here GARCH is fitted by Gaussian MLE with only three free parameters; in the more realistic case of four parameter MLE using the t distribution—the fourth parameter being the degrees of freedom—GARCH underperforms compared to NoVaS even with a sample size of 350; see Politis and Thomakos (2008, 2012).

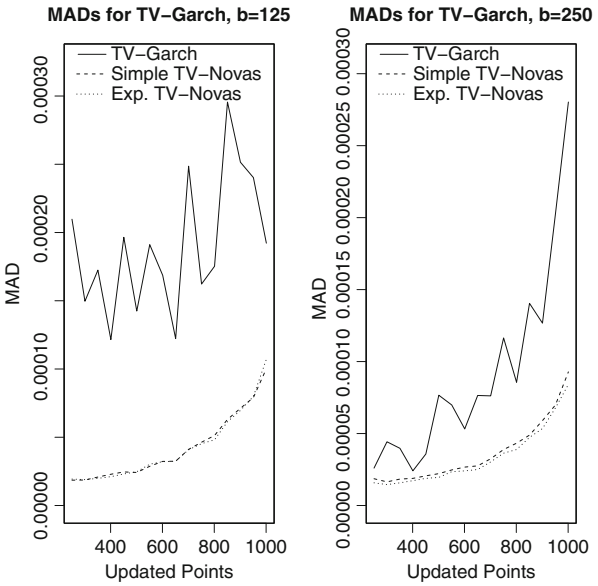

Fig. 10.12 MAD of prediction of squared returns obtained by fitting TV-GARCH vs. TV-NoVaS; data from TV-GARCH model

on analyzing a univariate series of financial returns via the Model-free approach, i.e., the NoVaS transformation; a multivariate version of NoVaS has been recently studied by Thomakos et al. (2015) with application to capturing the time-evolution of conditional correlations.

References

Akritas MG, VanKeilegom I (2001) Non-parametric estimation of the residual distribution. Scand J Stat 28(3):549–567

Allen DM (1971) Mean square error of prediction as a criterion for selecting variables. Technometrics 13:469–475

Allen DM (1974) The relationship between variable selection and data augmentation and a method for prediction. Technometrics 16:1307–1325

Alonso AM, Peña D, Romo J (2002) Forecasting time series with sieve bootstrap. J Stat Plan Infer 100(1):1–11

Altman NS (1992) An introduction to kernel and nearest-neighbor nonparametric regression. Am Stat 46(3):175–185

Andersen TG, Bollerslev T (1998) Answering the sceptics: yes, standard volatility models do provide accurate forecasts. Int Econ Rev 39(4):885–905

Andersen TG, Bollerslev T, Christoffersen PF, Diebold FX (2006) Volatility and correlation forecasting. In: Elliott G, Granger CWJ, Timmermann A (eds) Handbook of economic forecasting. North-Holland, Amsterdam, pp 778–878

Andersen TG, Bollerslev T, Meddahi N (2004) Analytic evaluation of volatility forecasts. Int Econ Rev 45:1079–1110

Atkinson AC (1985) Plots, transformations and regression. Clarendon Press, Oxford

Antoniadis A, Paparoditis E, Sapatinas T (2006) A functional wavelet-kernel approach for time series prediction. J R Stat Soc Ser B 68(5):837–857

Bachelier L (1900) Theory of speculation. Reprinted in Cootner PH (ed) The random character of stock market prices. MIT Press, Cambridge, MA, pp 17–78, 1964

Barndorff-Nielsen OE, Nielsen B, Shephard N, Ysusi C (1996) Measuring and forecasting financial variability using realized variance with and without a model. In: Harvey AC, Koopman SJ, Shephard N (eds) State space and unobserved components models: theory and applications. Cambridge University Press, Cambridge, pp 205–235

Bartlett MS (1946) On the theoretical specification of sampling properties of autocorrelated time series. J R Stat Soc Suppl 8:27–41

D.N. Politis, *Model-Free Prediction and Regression*, Frontiers in Probability and the Statistical Sciences, DOI 10.1007/978-3-319-21347-7

Beran R (1990) Calibrating prediction regions. J Am Stat Assoc 85:715–723
Berkes I, Gombay E, Horvath L, Kokoszka P (2004) Sequential change-point detection in GARCH (p, q) models. Econ Theory 20(6):1140–1167
Bertail P, Clémencon S (2006) Regenerative block bootstrap for Markov chains. Bernoulli 12(4):689–712
Bickel P, Gel YR (2011) Banded regularization of autocovariance matrices in application to parameter estimation and forecasting of time series. J R Stat Soc Ser B 73(5):711–728
Bickel P, Levina E (2008a) Regularized estimation of large covariance matrices. Ann Stat 36:199–227
Bickel P, Levina E (2008b) Covariance regularization via thresholding. Ann Stat 36:2577–2604
Bickel P, Li B (2006) Regularization in statistics. Test 15(2):271–344
Bollerslev T (1986) Generalized autoregressive conditional heteroscedasticity. J Econ 31:307–327
Bollerslev T, Chou R, Kroner K (1992) ARCH modelling in finance: a review of theory and empirical evidence. J Econ 52:5–60
Bose A (1988) Edgeworth correction by bootstrap in autoregressions. Ann Stat 16:1345–1741
Bose A, Chatterjee S (2002) Comparison of bootstrap and jackknife variance estimators in linear regression: second order results. Stat Sin 12:575–598
Box GEP (1976) Science and statistics. J Am Stat Assoc 71(356):791–799
Box GEP, Cox DR (1964) An analysis of transformations. J R Stat Soc Ser B 26:211–252
Box GEP, Draper NR (1987) Empirical model-building and response surfaces. Wiley, New York
Box GEP, Jenkins GM (1976) Time series analysis, control, and forecasting. Holden Day, San Francisco
Breidt FJ, Davis RA, Dunsmuir W (1995) Improved bootstrap prediction intervals for autoregressions. J Time Ser Anal 16(2):177–200
Breiman L (1996) Bagging predictors. Mach Learn 24:123–140
Breiman L, Friedman J (1985) Estimating optimal transformations for multiple regression and correlation. J Am Stat Assoc 80:580–597
Brent RP, Gustavson FG, Yun DYY (1980) Fast solution of Toeplitz systems of equations and computation of Padé approximants. J Algorithm 1(3):259–295
Brockwell PJ, Davis RA (1991) Time series: theory and methods, 2nd edn. Springer, New York
Brockwell PJ, Davis RA (1988) Simple consistent estimation of the coefficients of a linear filter. Stoch Process Appl 22:47–59
Bühlmann P, van de Geer S (2011) Statistics for high-dimensional data. Springer, New York
Cai TT, Ren Z, Zhou HH (2013) Optimal rates of convergence for estimating Toeplitz covariance matrices. Probab Theory Relat Fields 156(1–2):101–143

Cao R, Febrero-Bande M, Gonzalez-Manteiga W, Prada-Sanchez JM, GarcIa-Jurado I (1997) Saving computer time in constructing consistent bootstrap prediction intervals for autoregressive processes. Commun Stat Simul Comput 26(3):961–978

Carmack PS, Schucany WR, Spence JS, Gunst RF, Lin Q, Haley RW (2009) Far casting cross-validation. J Comput Graph Stat 18(4):879–893

Carroll RJ, Ruppert D (1988) Transformations and weighting in regression. Chapman and Hall, New York

Caroll RJ, Ruppert D (1991) Prediction and tolerance intervals with transformation and/or weighting. Technometrics 33(2):197–210

Chatterjee S, Bose A (2005) Generalized bootstrap for estimating equations. Ann Stat 33:414–436

Chen X, Xu M, Wu W-B (2013) Covariance and precision matrix estimation for high-dimensional time series. Ann Stat 41(6):2994–3021

Cheng T-ZF, Ing C-K, Yu S-H (2015) Inverse moment bounds for sample autocovariance matrices based on detrended time series and their applications. Linear Algebra Appl (to appear)

Choi B-S (1992) ARMA model identification. Springer, New York

Cox DR (1975) Prediction intervals and empirical Bayes confidence intervals. In: Gani J (eds) Perspectives in probability and statistics. Academic, London, pp 47–55

Dahlhaus R (1997) Fitting time series models to nonstationary processes. Ann Stat 25(1):1–37

Dahlhaus R (2012) Locally stationary processes. In: Handbook of statistics, vol 30. Elsevier, Amsterdam, pp 351–412

Dahlhaus R, Subba Rao S (2006) Statistical inference for time-varying ARCH processes. Ann Stat 34(3):1075–1114

Dai J, Sperlich S (2010) Simple and effective boundary correction for kernel densities and regression with an application to the world income and Engel curve estimation. Comput Stat Data Anal 54(11):2487–2497

DasGupta A (2008) Asymptotic theory of statistics and probability. Springer, New York

Davison AC, Hinkley DV (1997) Bootstrap methods and their applications. Cambridge University Press, Cambridge

Dawid AP (2004) Probability, causality, and the empirical world: a Bayes-de Finetti-Popper-Borel synthesis. Stat Sci 19(1):44–57

Devroye L (1981) Laws of the iterated logarithm for order statistics of uniform spacings. Ann Probab 9(6):860–867

Dowla A, Paparoditis E, Politis DN (2003) Locally stationary processes and the local block bootstrap. In: Akritas MG, Politis DN (eds) Recent advances and trends in nonparametric statistics. Elsevier, Amsterdam, pp 437–444

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

Draper NR, Smith H (1998) Applied regression analysis, 3rd edn. Wiley, New York

Efron B (1979) Bootstrap methods: another look at the jackknife. Ann Stat 7:1–26

Efron B (1983) Estimating the error rate of a prediction rule: improvement on cross-validation. J Am Stat Assoc 78:316–331
Efron B (2014) Estimation and accuracy after model selection. J Am Stat Assoc 109:991–1007
Efron B, Tibshirani RJ (1993) An introduction to the bootstrap. Chapman and Hall, New York
Engle R (1982) Autoregressive conditional heteroscedasticity with estimates of the variance of UK inflation. Econometrica 50:987–1008
Fama EF (1965) The behaviour of stock market prices. J Bus 38:34–105
Fan J (1993) Local linear regression smoothers and their minimax efficiencies. Ann Stat 21(1):196–216
Fan J, Gijbels I (1996) Local polynomial modelling and its applications. Chapman and Hall, New York
Fan J, Yao Q (2003) Nonlinear time series: nonparametric and parametric methods. Springer, New York
Ferraty F, Vieu P (2006) Nonparametric functional data analysis. Springer, New York
Francq C, Zakoian JM (2011) GARCH models: structure, statistical inference and financial applications. Wiley, New York
Franke J, Härdle W (1992) On bootstrapping kernel spectral estimates. Ann Stat 20:121–145
Franke J, Kreiss J-P, Mammen E (2002) Bootstrap of kernel smoothing in nonlinear time series. Bernoulli 8(1):1–37
Freedman DA (1981) Bootstrapping regression models. Ann Stat 9:1218–1228
Freedman D (1984) On bootstrapping two-stage least-squares estimates in stationary linear models. Ann Stat 12:827–842
Fryzlewicz P, Van Bellegem S, Von Sachs R (2003) Forecasting non-stationary time series by wavelet process modelling. Ann Inst Stat Math 55(4):737–764
Gangopadhyay AK, Sen PK (1990) Bootstrap confidence intervals for conditional quantile functions. Sankhya Ser A 52(3):346–363
Geisser S (1971) The inferential use of predictive distributions. In: Godambe BP, Sprott DS (eds) Foundations of Statistical Inference. Holt, Rinehart and Winston, Toronto, pp 456–469
Geisser S (1975) The predictive sample re-use method with applications. J Am Stat Assoc 70:320–328
Geisser S (1993) Predictive inference: an introduction. Chapman and Hall, New York
Gijbels I, Pope A, Wand MP (1999) Understanding exponential smoothing via kernel regression. J R Stat Soc Ser B 61:39–50
Ghysels E, Santa-Clara P, Valkanov R (2006) Predicting volatility: getting the most out of return data sampled at different frequencies. J Econ 131(1–2):59–95
Gouriéroux C (1997) ARCH models and financial applications. Springer, New York
Gray RM (2005) Toeplitz and circulant matrices: a review. Commun Inf Theory 2(3):155–239

Grenander U, Szegö G (1958) Toeplitz forms and their applications, vol 321. University of California Press, Berkeley
Hahn J (1995) Bootstrapping quantile regression estimators. Econ Theory 11(1):105–121
Hall P (1992) The bootstrap and edgeworth expansion. Springer, New York
Hall P (1993) On Edgeworth expansion and bootstrap confidence bands in nonparametric curve estimation. J R Stat Soc Ser B 55:291–304
Hall P, Wehrly TE (1991) A geometrical method for removing edge effects from kernel type nonparametric regression estimators. J Am Stat Assoc 86:665–672
Hall P, Wolff RCL, Yao Q (1999) Methods for estimating a conditional distribution function. J Am Stat Assoc 94(445):154–163
Hamilton JD (1994) Time series analysis. Princeton University Press, Princeton, NJ
Hampel FR (1973) Robust estimation, a condensed partial survey. Z Wahrscheinlichkeitstheorie verwandte Gebiete 27:87–104
Hansen BE (2004) Nonparametric estimation of smooth conditional distributions. Working paper, Department of Economics, University of Wisconsin
Hansen PR, Lunde A (2005) A forecast comparison of volatility models: does anything beat a GARCH (1,1)? J Appl Econ 20:873–889
Hansen PR, Lunde A (2006) Consistent ranking of volatility models. J Econ 131:97–121
Hansen PR, Lunde A, Nason JM (2003) Choosing the best volatility models: the model confidence set approach. Oxf Bull Econ Stat 65:839–861
Härdle W (1990) Applied nonparametric regression. Cambridge University Press, Cambridge
Härdle W, Bowman AW (1988) Bootstrapping in nonparametric regression: local adaptive smoothing and confidence bands. J Am Stat Assoc 83:102–110
Härdle W, Marron JS (1991) Bootstrap simultaneous error bars for nonparametric regression. Ann Stat 19:778–796
Härdle W, Vieu P (1992) Kernel regression smoothing of time series. J Time Ser Anal 13:209–232
Härdle W, Hall P, Marron JS (1988) How far are automatically chosen regression smoothing parameters from their optimum? (with discussion). J Am Stat Assoc 83:86–95
Hart JD (1997) Nonparametric smoothing and lack-of-fit tests. Springer, New York
Hart JD, Yi S (1998) One-sided cross-validation. J Am Stat Assoc 93(442):620–631
Hastie T, Loader C (1993) Local regression: automatic kernel carpentry. Stat Sci 8(2):120–143
Hastie T, Tibshirani R, Friedman JH (2009) The elements of statistical learning: data mining, inference, and predictions, 2nd edn. Springer, New York
Hocking RR (1976) The analysis and selection of variables in linear regression. Biometrics 31(1):1–49
Hong Y (1999) Hypothesis testing in time series via the empirical characteristic function: a generalized spectral density approach. J Am Stat Assoc 94:1201–1220
Hong Y, White H (2005) Asymptotic distribution theory for nonparametric entropy measures of serial dependence. Econometrica 73(3):837–901

Horowitz J (1998) Bootstrap methods for median regression models. Econometrica 66(6):1327–1351

Huber PJ (1973) Robust regression: asymptotics, conjectures and Monte Carlo. Ann Stat 1:799–821

Hurvich CM, Zeger S (1987) Frequency domain bootstrap methods for time series. Technical Report, New York University, Graduate School of Business Administration

Jentsch C, Politis DN (2015) Covariance matrix estimation and linear process bootstrap for multivariate time series of possibly increasing dimension. Ann Stat 43(3):1117–1140

Kim TY, Cox DD (1996) Bandwidth selection in kernel smoothing of time series. J Time Ser Anal 17:49–63

Kirch C, Politis DN (2011) TFT-Bootstrap: resampling time series in the frequency domain to obtain replicates in the time domain. Ann Stat 39(3):1427–1470

Koenker R (2005) Quantile regression. Cambridge University Press, Cambridge

Kokoszka P, Leipus R (2000) Change-point estimation in ARCH models. Bernoulli 6(3):513–539

Kokoszka P, Politis DN (2011) Nonlinearity of ARCH and stochastic volatility models and Bartlett's formula. Probab Math Stat 31:47–59

Koopman SJ, Jungbacker B, Hol E (2005) Forecasting daily variability of the S&P 100 stock index using historical, realised and implied volatility measurements. J Empir Finance 12:445–475

Kreiss J-P, Paparoditis E (2011) Bootstrap methods for dependent data: a review. J Korean Stat Soc 40(4):357–378

Kreiss J-P, Paparoditis E (2012) The hybrid wild bootstrap for time series. J Am Stat Assoc 107:1073–1084

Kreiss J-P, Paparoditis E, Politis DN (2011) On the range of validity of the autoregressive sieve bootstrap. Ann Stat 39(4):2103–2130

Kuhn M, Johnson K (2013) Applied predictive modeling. Springer, New York

Künsch H (1989) The jackknife and the bootstrap for general stationary observations. Ann Stat 17:1217–1241

Lahiri SN (2003) A necessary and sufficient condition for asymptotic independence of discrete Fourier transforms under short-and long-range dependence. Ann Stat 31(2):613–641

Ledoit O, Wolf M (2003) Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. J Empir Financ 10(5):603–621

Ledoit O, Wolf M (2004) Honey, I shrunk the sample covariance matrix. UPF economics and business working paper 691

Lei J, Robins J, Wasserman L (2013) Distribution free prediction sets. J Am Stat Assoc 108:278–287

Li Q, Racine JS (2007) Nonparametric econometrics. Princeton University Press, Princeton

Linton OB, Chen R, Wang N, Härdle W (1997) An analysis of transformations for additive nonparametric regression. J Am Stat Assoc 92:1512–1521

Linton OB, Sperlich S, van Keilegom I (2008) Estimation of a semiparametric transformation model. Ann Stat 36(2):686–718
Loader C (1999) Local regression and likelihood. Springer, New York
Mandelbrot B (1963) The variation of certain speculative prices. J Bus 36:394–419
Maronna RA, Martin RD, Yohai VJ (2006) Robust statistics: theory and methods. Wiley, New York
Masarotto G (1990) Bootstrap prediction intervals for autoregressions. Int J Forecast 6(2):229–239
Masry E, Tjøstheim D (1995) Nonparametric estimation and identification of nonlinear ARCH time series. Econ Theory 11:258–289
McCullagh P, Nelder J (1983) Generalized linear models. Chapman and Hall, London
McMurry T, Politis DN (2008) Bootstrap confidence intervals in nonparametric regression with built-in bias correction. Stat Probab Lett 78:2463–2469
McMurry T, Politis DN (2010) Banded and tapered estimates of autocovariance matrices and the linear process bootstrap. J Time Ser Anal 31:471–482 [Corrigendum: J Time Ser Anal 33, 2012]
McMurry T, Politis DN (2015) High-dimensional autocovariance matrices and optimal linear prediction (with discussion). Electr J Stat 9:753–822
Meddahi N (2001) An eigenfunction approach for volatility modeling. Technical report, CIRANO Working paper 2001s–70, University of Montreal
Mikosch T, Starica C (2004) Changes of structure in financial time series and the GARCH model. Revstat Stat J 2(1):41–73
Nadaraya EA (1964) On estimating regression. Theory Probab Appl 9:141–142
Nelson D (1991) Conditional heteroscedasticity in asset returns: a new approach. Econometrica 59:347–370
Neumann M, Polzehl J (1998) Simultaneous bootstrap confidence bands in nonparametric regression. J Nonparametr Stat 9:307–333
Olive DJ (2007) Prediction intervals for regression models. Comput Stat Data Anal 51:3115–3122
Pagan A, Ullah A (1999) Nonparametric econometrics. Cambridge University Press, Cambridge
Pan L, Politis DN (2014) Bootstrap prediction intervals for Markov processes. Discussion paper, Department of Economics, University of California, San Diego. Retrievable from http://escholarship.org/uc/item/7555757g. Accepted for publication in CSDA Annals of Computational and Financial Econometrics
Pan L, Politis DN (2015) Bootstrap prediction intervals for linear, nonlinear and nonparametric autoregressions (with discussion). J Stat Plan Infer (to appear)
Paparoditis E, Politis DN (1998) The backward local bootstrap for conditional predictive inference in nonlinear time series. In: Lipitakis EA (ed) Proceedings of the 4th Hellenic-European conference on computer mathematics and its applications (HERCMA'98). Lea Publishing, Athens, pp 467–470
Paparoditis E, Politis DN (2001) A Markovian local resampling scheme for nonparametric estimators in time series analysis. Econ Theory 17(3):540–566

Paparoditis E, Politis DN (2002a) The local bootstrap for Markov processes. J Stat Plan Infer 108(1):301–328

Paparoditis E, Politis DN (2002b) Local block bootstrap. C R Acad Sci Paris Ser I 335:959–962

Pascual L, Romo J, Ruiz E (2004) Bootstrap predictive inference for ARIMA processes. J Time Ser Anal 25(4):449–465

Patel JK (1989) Prediction intervals: a review. Commun Stat Theory Methods 18:2393–2465

Patton AJ (2011) Volatility forecast evaluation and comparison using imperfect volatility proxies. J Econ 160(1):246–256

Politis DN (1998) Computer-intensive methods in statistical analysis. IEEE Signal Process Mag 15(1):39–55

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

Politis DN (2003a) A normalizing and variance-stabilizing transformation for financial time series. In: Akritas MG, Politis DN (eds) Recent advances and trends in nonparametric statistics. Elsevier, Amsterdam, pp 335–347

Politis DN (2003b) Adaptive bandwidth choice. J Nonparametr Stat 15(4–5):517–533

Politis DN (2004) A heavy-tailed distribution for ARCH residuals with application to volatility prediction. Ann Econ Finance 5:283–298

Politis DN (2007a) Model-free vs. model-based volatility prediction. J Financ Econ 5(3):358–389

Politis DN (2007b) Model-free prediction, vol LXII. Bulletin of the International Statistical Institute, Lisbon, pp 1391–1397

Politis DN (2010) Model-free model-fitting and predictive distributions. Discussion paper, Department of Economics, University of California, San Diego. Retrievable from: http://escholarship.org/uc/item/67j6s174

Politis DN (2011) Higher-order accurate, positive semi-definite estimation of large-sample covariance and spectral density matrices. Econ Theory 27(4):703–744

Politis DN (2013) Model-free model-fitting and predictive distributions (with discussion). Test 22(2):183–250

Politis DN (2014) Bootstrap confidence intervals in nonparametric regression without an additive model. In: Akritas MG, Lahiri SN, Politis DN (eds) Proceedings of the first conference of the international society for nonParametric statistics. Springer, New York, pp 271–282

Politis DN, Romano JP (1992) A general resampling scheme for triangular arrays of alpha-mixing random variables with application to the problem of spectral density estimation. Ann Stat 20:1985–2007

Politis DN, Romano JP (1994) The stationary bootstrap. J Am Stat Assoc 89(428):1303–1313

Politis DN, Romano JP (1995) Bias-corrected nonparametric spectral estimation. J Time Ser Anal 16(1):67–104

Politis DN, Romano JP, Wolf M (1999) Subsampling. Springer, New York

Politis DN, Thomakos D (2008) Financial time series and volatility prediction using NoVaS transformations. In: Rapach DE, Wohar ME (eds) Forecasting in the presence of structural breaks and model uncertainty. Emerald Group Publishing, Bingley, pp 417–447

Politis DN, Thomakos DD (2012) NoVaS transformations: flexible inference for volatility forecasting. In: Chen X, Swanson N (eds) Recent advances and future directions in causality, prediction, and specification analysis: essays in honor of Halbert L. White Jr. Springer, New York, pp 489–528

Pourahmadi M (1999) Joint mean-covariance models with applications to longitudinal data: unconstrained parameterisation. Biometrika 86(3):677–690

Pourahmadi M (2011) Modeling covariance matrices: the GLM and regularization perspectives. Stat Sci 26(3):369–387

Poon S, Granger C (2003) Forecasting volatility in financial markets: a review. J Econ Lit 41:478–539

Priestley MB (1965) Evolutionary spectra and non-stationary processes. J R Stat Soc Ser B 27:204–237

Priestley MB (1988) Nonlinear and nonstationary time series analysis. Academic, London

Raïs N (1994) Méthodes de rééchantillonnage et de sous-échantillonnage pour des variables aléatoires dépendantes et spatiales. Ph.D. thesis, University of Montreal

Rajarshi MB (1990) Bootstrap in Markov sequences based on estimates of transition density. Ann Inst Stat Math 42:253–268

Resnick S, Samorodnitsky G, Xue F (1999) How misleading can sample ACF's of stable MA's be? (Very!) Ann Appl Probab 9(3):797–817

Rissanen J, Barbosa L (1969) Properties of infinite covariance matrices and stability of optimum predictors. Inf Sci 1:221–236

Rosenblatt M (1952) Remarks on a multivariate transformation. Ann Math Stat 23:470–472

Ruppert D, Cline DH (1994) Bias reduction in kernel density estimation by smoothed empirical transformations. Ann Stat 22:185–210

Schmoyer RL (1992) Asymptotically valid prediction intervals for linear models. Technometrics 34:399–408

Schucany WR (2004) Kernel smoothers: an overview of curve estimators for the first graduate course in nonparametric statistics. Stat Sci 19:663–675

Seber GAF, Lee AJ (2003) Linear regression analysis, 2nd edn. Wiley, New York

Shao J, Tu D (1995) The Jackknife and bootstrap. Springer, New York

Shapiro SS, Wilk M (1965) An analysis of variance test for normality (complete samples). Biometrika 52:591–611

Shephard N (1996) Statistical aspects of ARCH and stochastic volatility. In: Cox DR, Hinkley DV, Barndorff-Nielsen OE (eds) Time series models in econometrics, finance and other fields. Chapman and Hall, London, pp 1–67

Shi SG (1991) Local bootstrap. Ann Inst Stat Math 43:667–676

Shmueli G (2010) To explain or to predict? Stat Sci 25:289–310

Starica C, Granger C (2005) Nonstationarities in stock returns. Rev Econ Stat 87(3):503–522

Stine RA (1985) Bootstrap prediction intervals for regression. J Am Stat Assoc 80:1026–1031
Stine RA (1987) Estimating properties of autoregressive forecasts. J Am Stat Assoc 82:1072–1078
Stone M (1974) Cross-validatory choice and assessment of statistical predictions. J R Stat Soc Ser B 39:144–147
Thombs LA, Schucany WR (1990) Bootstrap prediction intervals for autoregression. J Am Stat Assoc 85:486–492
Tibshirani R (1988) Estimating transformations for regression via additivity and variance stabilization. J Am Stat Assoc 83:394–405
Tibshirani R (1996) Regression shrinkage and selection via the Lasso. J R Stat Soc Ser B 58(1):267–288
Thomakos DD, Klepsch J, Politis DN (2015) Multivariate NoVaS and inference on conditional correlations. Working paper, Department of Economics, University of California, San Diego
Wang L, Brown LD, Cai TT, Levine M (2008) Effect of mean on variance function estimation in nonparametric regression. Ann Stat 36:646–664
Wang L, Politis DN (2015) Asymptotic validity of bootstrap confidence intervals in nonparametric regression without an additive model. Working paper, Department of Mathematics, University of California, San Diego
Watson GS (1964) Smooth regression analysis. Sankhya Ser A 26:359–372
Wolf M, Wunderli D (2015) Bootstrap joint prediction regions. J Time Ser Anal 36(3):352–376
Wolfowitz J (1957) The minimum distance method. Ann Math Stat 28:75–88
Wu S, Harris TJ, McAuley KB (2007) The use of simplified or misspecified models: linear case. Can J Chem Eng 75:386–398
Wu W-B, Pourahmadi M (2009) Banding sample autocovariance matrices of stationary processes. Stat Sin 19(4):1755–1768
Xiao H, Wu W-B (2012) Covariance matrix estimation for stationary time series. Ann Stat 40(1):466–493
Zhang T, Wu W-B (2011) Testing parametric assumptions of trends of nonstationary time series. Biometrika 98(3):599–614
Zhou Z, Wu W-B (2009) Local linear quantile estimation for non-stationary time series. Ann Stat 37:2696–2729
Zhou Z, Wu W-B (2010) Simultaneous inference of linear models with time-varying coefficients. J R Stat Soc Ser B 72:513–531