

UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Bootstrap Prediction Intervals for Time Series /

Permalink

<https://escholarship.org/uc/item/24n733hr>

Author

Pan, Li

Publication Date

2013

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Bootstrap Prediction Intervals for Time Series

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

in

Mathematics

by

Li Pan

Committee in charge:

Professor Dimitris Politis, Chair
Professor Ery Arias-Castro
Professor Brendan K. Beare
Professor Anthony C. Gamst
Professor Ronghui Xu

2013

Copyright
Li Pan, 2013
All rights reserved.

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

Chair

University of California, San Diego

2013

TABLE OF CONTENTS

Signature Page	iii
Table of Contents	iv
List of Figures	vii
List of Tables	viii
Acknowledgements	x
Vita	xi
Abstract of the Dissertation	xii
Chapter 1 Bootstrap Prediction Intervals for Linear Autoregression	1
1.1 Notation and Definition	2
1.2 Forward Bootstrap Algorithm	4
1.2.1 Forward Bootstrap with Fitted Residuals	4
1.2.2 Forward Bootstrap with Predictive Residuals	6
1.2.3 Forward Studentized Bootstrap with Fitted Residuals	7
1.2.4 Forward Studentized Bootstrap with Predictive Residuals	8
1.3 Asymptotic Properties	9
1.4 Backward Bootstrap and Asymptotic Properties	13
1.4.1 Algorithm for Backward Bootstrap with Fitted Residuals	14
1.4.2 Algorithm for Backward Bootstrap with Predictive Residuals	15
1.4.3 Algorithm for Backward Studentized Bootstrap with fitted Residuals	15
1.4.4 Algorithm for Backward Studentized Bootstrap with Predictive Residuals	16
1.5 Joint Prediction intervals	17
1.6 Generalized Bootstrap prediction Intervals	18
1.7 Monte Carlo Studies	20
1.8 Alternative Approaches to Bootstrap Prediction Intervals for Linear Autoregression Model	28
1.8.1 Bootstrap Prediction Intervals Based on Studentized Predictive Root	28
1.8.2 Grigoletto's Method	29
1.8.3 A computationally faster method	30
1.8.4 Sieve/PRR Bootstrap	31
1.8.5 More Monte Carlo Studies	33

Chapter 2	Bootstrap Prediction Intervals for ARCH model	45
2.1	Bootstrap Algorithm and Asymptotic Properties	46
2.1.1	Bootstrap Algorithm Based on BWLE with Fitted Residuals	46
2.1.2	Bootstrap Algorithm Based on BWLE with Predictive Residuals	48
2.1.3	Bootstrap Algorithm Based on QMLE	49
2.1.4	Asymptotic Properties	50
2.2	Monte Carlo Studies	51
Chapter 3	Bootstrap Prediction Intervals for Nonparametric Autoregression	54
3.1	Nonparametric Autoregression with i.i.d Innovations	55
3.1.1	Bootstrap Algorithm with Fitted Residuals	55
3.1.2	Bootstrap Algorithm with Predictive Residuals	57
3.1.3	Asymptotic Properties	57
3.2	Nonparametric Autoregression Bootstrap with Heteroscedastic Inno- vations	61
3.2.1	Bootstrap Algorithm with Fitted Residuals	62
3.2.2	Bootstrap Algorithm with Predictive Residuals	63
3.2.3	Asymptotic Properties	64
3.3	Partially Studentized Method	67
3.4	Monte Carlo Studies	69
3.4.1	Simulation Results for Nonparametric Autoregression with i.i.d. Errors	69
3.4.2	Simulation Results for Nonparametric Autoregression with Het- eroscedastic Errors	73
3.5	Backward Bootstrap for Nonparametric Autoregression	79
Chapter 4	Bootstrap Prediction Intervals for Markov Processes Based on Con- ditional Distributions	81
4.1	Bootstrap Based on estimates of Transition Density	82
4.1.1	Notation and Definition	82
4.1.2	Forward Bootstrap Based on Transition Density	83
4.1.3	Backward Bootstrap Based on Transition Density	85
4.1.4	Aymptotic Properties	88
4.1.5	Simulation Results	94
4.2	Local Bootstrap	100
4.2.1	Forward Local Bootstrap	100
4.2.2	Backward Local Bootstrap	102
4.2.3	Asymptotic Properties	103
4.2.4	Simulation Results	103
4.3	Hybrid Backward Markov Bootstrap for Nonparametric Autoregression	109
4.3.1	Algorithms	110
4.3.2	Simulation	113

Chapter 5 Bootstrap Prediction Intervals for Markov Processes Based on the Model-free Method	124
5.1 Construct the Transformation	124
5.2 Algorithm for Model-free Prediction Interval:MF	126
5.3 Better Model-free Prediction Intervals:Predictive Model-free Method	127
5.4 Smoothed Model Free: SMF	128
5.5 Simulation	130
References	139

LIST OF FIGURES

Figure 1.1: densities of bootstrap predictive roots vs. true predictive root . . .	27
--	----

LIST OF TABLES

Table 1.1:	Simulation Results of AR(1) with normal innovations and $\phi_1 = 0.5$	22
Table 1.2:	Simulation Results of AR(1) with normal innovations and $\phi_1 = 0.9$	23
Table 1.3:	Simulation Results of AR(1) with Laplace innovations and $\phi_1 = 0.5$	24
Table 1.4:	Simulation Results of AR(1) with Laplace innovations and $\phi_1 = 0.9$	25
Table 1.5:	Simulation Results of AR(1) with normal innovations and $\phi_1 = 0.5$	34
Table 1.6:	Simulation Results of AR(1) with Laplace innovations and $\phi_1 = 0.5$	35
Table 1.7:	Simulation Results of AR(2) model $X_t = 1.55X_{t-1} - 0.6X_{t-2} + \epsilon_t$ with normal innovations	36
Table 1.8:	Simulation Results of AR(2) model $X_t = 1.55X_{t-1} - 0.6X_{t-2} + \epsilon_t$ with Laplace innovations	37
Table 1.9:	Simulation Results of AR(1) model $X_t = 0.5X_{t-1} + \epsilon_t$ with normal innovations when $n = 100$	39
Table 1.10:	Simulation Results of AR(1) model $X_t = 0.5X_{t-1} + \epsilon_t$ with Laplace innovations when $n = 100$	40
Table 1.11:	Simulation Results of AR(20) model $X_t = 0.7X_{t-10} - 0.1X_{t-20} + \epsilon_t$ with normal innovations	42
Table 1.12:	Simulation Results of AR(20) model $X_t = 0.7X_{t-10} - 0.1X_{t-20} + \epsilon_t$ with Laplace innovations	43
Table 2.1:	Simulation Results of ARCH(1) model $X_t = \sqrt{0.5 + 0.25X_{t-1}^2}\epsilon_t$ with normal innovations	51
Table 2.2:	Simulation Results of ARCH(1) model $X_t = \sqrt{0.5 + 0.25X_{t-1}^2}\epsilon_t$ with Laplace innovations	52
Table 2.3:	Simulation Results of ARCH(2) model $X_t = \sqrt{0.1 + 0.2X_{t-2}^2}\epsilon_t$ with normal innovations	52
Table 2.4:	Simulation Results of ARCH(2) model $X_t = \sqrt{0.1 + 0.2X_{t-2}^2}\epsilon_t$ with Laplace innovations	53
Table 3.1:	nonparametric autoregression with i.i.d innovations-model 1	70
Table 3.2:	nonparametric autoregression with i.i.d innovations-model 2	71
Table 3.3:	nonparametric autoregression with i.i.d innovations-model 3	72
Table 3.4:	heteroscedastic model 4 without over-smoothed bandwidth	74
Table 3.5:	heteroscedastic model 5 without over-smoothed bandwidth	74
Table 3.6:	heteroscedastic model 4 with $g = 1.5h$	75
Table 3.7:	heteroscedastic model 4 with $g = 2h$	76
Table 3.8:	heteroscedastic model 5 with $g = 1.5h$	77
Table 3.9:	heteroscedastic model 5 with $g = 2h$	78
Table 4.1:	bootstrap based on transition density with i.i.d. errors-model 1 . .	95
Table 4.2:	bootstrap based on transition density with i.i.d. errors-model 2 . .	96
Table 4.3:	bootstrap based on transition density with i.i.d. errors-model 3 . .	97
Table 4.4:	bootstrap based on transition density with i.i.d. errors-model 4 . .	98

Table 4.5:	bootstrap based on transition density with i.i.d. errors-model 5 . . .	99
Table 4.6:	local bootstrap with i.i.d. errors-model 1	104
Table 4.7:	local bootstrap with i.i.d. errors-model 2	105
Table 4.8:	local bootstrap with i.i.d. errors-model 3	106
Table 4.9:	local bootstrap with heteroscedastic errors-model 4	107
Table 4.10:	local bootstrap with heteroscedastic errors-model 5	108
Table 4.11:	Hybrid bootstrap based on transition density for nonlinear autoregression-model 1	114
Table 4.12:	Hybrid bootstrap based on transition density for nonlinear autoregression-model 2	115
Table 4.13:	Hybrid bootstrap based on transition density for nonlinear autoregression-model 3	116
Table 4.14:	Hybrid bootstrap based on transition density for nonlinear autoregression-model 4	117
Table 4.15:	Hybrid bootstrap based on transition density for nonlinear autoregression-model 5	118
Table 4.16:	Hybrid backward local bootstrap for nonlinear autoregression-model 1	119
Table 4.17:	Hybrid backward local bootstrap for nonlinear autoregression-model 2	120
Table 4.18:	Hybrid backward local bootstrap for nonlinear autoregression-model 3	121
Table 4.19:	Hybrid backward local bootstrap for nonlinear autoregression-model 4	122
Table 4.20:	Hybrid backward local bootstrap for nonlinear autoregression-model 5	123
Table 5.1:	Model Free Method-model 1	132
Table 5.2:	Model Free Method-model 2	133
Table 5.3:	Model Free Method-model 3	134
Table 5.4:	Model Free Method-model 4	135
Table 5.5:	Model Free Method-model 5	136

ACKNOWLEDGEMENTS

I would like to acknowledge Professor Politis for his support as the chair of my committee. Through multiple drafts, his guidance has proved to be invaluable.

Chapter 1, 2 and 3, in full, are currently being prepared for the following paper prepared for future submission for publication: Bootstrap prediction intervals for linear, nonlinear and nonparametric autoregressions. Pan,Li; Politis, Dimitris. The dissertation author was the primary investigator and author of this material.

Chapter 4 and 5, in full, are currently being prepared for the following paper prepared for future submission for publication: Bootstrap prediction intervals for Markov Processes. Pan,Li; Politis, Dimitris. The dissertation author was the primary investigator and author of this material.

I would also like to acknowledge everyone who has contributed to the LaTeX style files from which this work is derived. Many thanks are due to them.

VITA

2009	B. S. in Mathematics and Applied Mathematics <i>cum laude</i> , Fudan University, Shanghai, China
2010	M. S. in Statistics, University of California, San Diego
2013	Ph. D. in Mathematics, University of California, San Diego

ABSTRACT OF THE DISSERTATION

Bootstrap Prediction Intervals for Time Series

by

Li Pan

Doctor of Philosophy in Mathematics

University of California, San Diego, 2013

Professor Dimitris Politis, Chair

We construct bootstrap prediction intervals for linear autoregressions, non-linear autoregressions, nonparametric autoregressions and Markov processes. Several forward and backward bootstrap methods using predictive residuals and fitted residuals are introduced and applied to those time series. We describe exact algorithms for these different models and show that the bootstrap intervals properly estimate the distribution of the future values. In simulations using standard time series models, we compare the prediction intervals of different methods with regards to coverage level and length of interval.

Chapter 1

Bootstrap Prediction Intervals for Linear Autoregression

Stine(1987) used a bootstrap method to estimate the prediction mean squared error of the estimated linear predictor of an $AR(p)$ model with i.i.d. Gaussian errors. Relaxing the assumption of Gaussian errors, Thombs and Schucany(1990) proposed a first backward and then forward bootstrap method to find prediction intervals for linear autoregressions conditioned on the last p observations. This backward bootstrap method was revisited by Breidt, Davis and Dunsmuir(1995) who gave the correct algorithm of finding the backward errors. Masarotto(1990) first proposed a forward bootstrap method based on the studentized predictive root to get the prediction intervals for $AR(p)$ models and Pascual *et al.* (2004) proposed another forward bootstrap method and apply it to the prediction intervals for both autoregression and ARMA model.

In this chapter we first give our algorithms for constructing forward bootstrap prediction intervals using fitted residuals and predictive residuals, and then prove the consistency of the prediction root method for prediction intervals. We then study the corresponding backward methods. We improve the Breidt, Davis and Dunsmuir(1995)'s method by introducing the predictive residuals. In simulation, we will see that the method with predictive residuals improves the coverage level significantly

comparing to the method with fitted residuals. This result is not unexpected since a similar phenomenon occurs in linear regression, cf. Politis (2010,2013).

In the last section of this chapter, we will review other methods to construct bootstrap prediction intervals, for example, the bootstrap prediction interval based on generating the standardized bootstrap prediction error proposed by Masarotto(1990) and Grigoletto(1998), the computationally faster method proposed by Cao *et al.*(1997), the sieve bootstrap prediction interval proposed by Alonso, Peña and Romo(2002), and the bootstrap prediction intervals for ARIMA model proposed by Pascual, Romo and Ruiz(2004), and compare their approaches with ours.

1.1 Notation and Definition

Let $X_1, \dots, X_n$ be variables from the strictly stationary causal AR(p) model defined by the following recursion

$$X_t = \phi_0 + \sum_{j=1}^p \phi_j X_{t-j} + \epsilon_t \quad (1.1)$$

where $\phi(z) = 1 - \phi_1 z - \dots - \phi_p z^p \neq 0$ for $|z| \leq 1$ and $\{\epsilon_t\}$ are i.i.d. random variables with mean zero, variance σ^2 and distribution F_ϵ . $\underline{\phi} = (\phi_0, \phi_1, \phi_2, \dots, \phi_p)'$ is the vector of autoregressive parameters. Let $Y_t = (X_t, X_{t-1}, \dots, X_{t-p+1})', t = p, p+1, \dots, n$ is the vector of the last p observations up to X_t . Let $\hat{\underline{\phi}} = (\hat{\phi}_0, \hat{\phi}_1, \dots, \hat{\phi}_p)'$ be the vector of estimators of the coefficients, and $\hat{X}_t$ be the fitted value of X_t . X_{n+h} is the h -step ahead future value that we want to predict.

Politis(2010) gave the definition of 'predictive roots' in regression. Following is our definition for 'predictive roots' in linear autoregression,

Definition 1.1.1.

(i) *predictive root and studentized predictive root*

The predictive root is the error in h -step ahead prediction, i.e.,

$$X_{n+h} - \hat{X}_{n+h},$$

where $\hat{X}_{n+h}$ is the predictor of X_{n+h} based on the data $X_1, \dots, X_n$. Since $\hat{X}_{n+h}$ is a function of the data, let $\hat{X}_{n+h} = \Pi(X_1, \dots, X_n)$. Let $\hat{V}_n^2$ be an estimate of the error variance in h -step ahead prediction, then the studentized predictive root is

$$\frac{X_{n+h} - \hat{X}_{n+h}}{\hat{V}_n},$$

Also $\hat{V}_n$ is a function of the data, let $\hat{V}_n = V(X_1, \dots, X_n)$.

(ii) bootstrap predictive root and studentized bootstrap predictive root

Consider a bootstrap pseudo series $X_1^*, \dots, X_n^*$, the bootstrap predictive root is defined as:

$$X_{n+h}^* - \hat{X}_{n+h}^*,$$

and the studentized bootstrap predictive root is defined as:

$$\frac{X_{n+h}^* - \hat{X}_{n+h}^*}{\hat{V}_n^*}.$$

Here $\hat{X}_{n+h}^* = \Pi(X_1^*, \dots, X_n^*)$ and $\hat{V}_n^* = V(X_1^*, \dots, X_n^*)$ are functions of pseudo data, where Π and V are the same functions as above.

Definition 1.1.2. Asymptotic validity and correctness

Let L_n, U_n be functions of the data $X_1, \dots, X_n$. The interval $[L_n, U_n]$ will be called a $(1 - \alpha)100\%$ asymptotically valid prediction interval for X_{n+h} given $X_{n-p+1} = x_{n-p+1}, \dots, X_n = x_n$ if

$$P(L_n \leq X_{n+h} \leq U_n) \rightarrow 1 - \alpha \text{ as } n \rightarrow \infty$$

for any $\{x_{n-p+1}, \dots, x_n\}$.

The above probability P should be interpreted as conditional on X_{n+h} given $X_{n-p+1} = x_{n-p+1}, \dots, X_n = x_n$. So the definition indicates conditional validity.

In addition, if the interval $[L_n, U_n]$ also captures the variability resulting from estimation error, we call $[L_n, U_n]$ a $(1 - \alpha)100\%$ asymptotically valid and correct prediction interval. If the variability from the estimation error is partially captured, then the prediction interval can be called asymptotically valid and partially correct.

The salient point in all bootstrap algorithms is that we use the distribution of the (studentized) bootstrap predictive root to estimate the distribution of the (studentized) predictive root.

Remark 1.1.1. *If one can show that*

$$\sup_a |P(X_{n+h} - \hat{X}_{n+h} \leq a) - P^*(X_{n+h}^* - \hat{X}_{n+h}^* \leq a)| \rightarrow 0 \text{ in probability,}$$

here the P^ can be interpreted as the conditional probability given the observations $\{x_1, \dots, x_n\}$. Then standard results imply that the quantiles of $P^*(X_{n+h}^* - \hat{X}_{n+h}^* \leq a)$ can be used to approximate the quantiles of $P(X_{n+h} - \hat{X}_{n+h} \leq a)$, thus leading to asymptotically valid prediction intervals.*

If one wants to approximate the distribution of the studentized predictive root by its bootstrap analog, one should show

$$\sup_a |P(\frac{X_{n+h} - \hat{X}_{n+h}}{\hat{V}_n} \leq a) - P^*(\frac{X_{n+h}^* - \hat{X}_{n+h}^*}{\hat{V}_n^*} \leq a)| \rightarrow 0 \text{ in probability.}$$

To ensure the correctness, we need to show that the distribution of the bootstrap estimators is able to mimic the distribution of estimators of interest.

1.2 Forward Bootstrap Algorithm

The idea of forward bootstrap method is that given observations $\{x_1, \dots, x_n\}$, we use a certain recursion to generate bootstrap series "forward" in time with initial conditions $(x_1, \dots, x_p)$.

The details for estimating the coefficients, generating the bootstrap pseudo-data and constructing the prediction intervals using both the fitted and predictive residuals are given in the following two sections.

1.2.1 Forward Bootstrap with Fitted Residuals

Given a sample $\{x_1, \dots, x_n\}$, following are steps needed to construct the prediction interval for future value X_{n+h} .

Algorithm 1.2.1. *Forward bootstrap with fitted residuals(Ff)*

1. Use all observations $x_1, \dots, x_n$ to obtain the least-squares estimators of the autoregressive coefficients: $\hat{\underline{\phi}} = (\hat{\phi}_0, \hat{\phi}_1, \dots, \hat{\phi}_p)'$ through the following linear model

$$\begin{pmatrix} x_n \\ x_{n-1} \\ \vdots \\ x_{p+1} \end{pmatrix} = \begin{bmatrix} 1 & x_{n-1} & \cdots & x_{n-p} \\ 1 & x_{n-2} & \cdots & x_{n-p-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_p & \cdots & x_1 \end{bmatrix} \begin{pmatrix} \phi_0 \\ \phi_1 \\ \vdots \\ \phi_p \end{pmatrix} + \begin{pmatrix} \epsilon_n \\ \epsilon_{n-1} \\ \vdots \\ \epsilon_{p+1} \end{pmatrix}. \quad (1.2)$$

2. Compute the fitted value and fitted residuals:

$$\hat{x}_t = \hat{\phi}_0 + \sum_{j=1}^p \hat{\phi}_j x_{t-j},$$

$$\hat{\epsilon}_t = x_t - \hat{x}_t,$$

$$t \in (p+1, \dots, n).$$

3. Center the fitted residuals: let $r_t = \hat{\epsilon}_t - \bar{\hat{\epsilon}}$ for $t = p+1, \dots, n$, and $\bar{\hat{\epsilon}} = (n-p)^{-1} \sum_{p+1}^n \hat{\epsilon}_t$; let the empirical distribution of r_t be denoted by $\hat{F}_n$.

(a) Draw the bootstrap pseudo residuals ϵ_t^* , $t \geq 1$ from $\hat{F}_n$

(b) Let $u_1^* = x_1, \dots, u_p^* = x_p$, then generate $\{u_t^*, t \geq p+1\}$ by the recursion:

$$u_t^* = \hat{\phi}_0 + \sum_{j=1}^p \hat{\phi}_j u_{t-j}^* + \epsilon_t^*, \text{ for } t = p+1, \dots, n+m.$$

To ensure stationarity of the pseudo data, we generate $n+m$ pseudo-data for some large positive m and then discard the first m data. So

$$x_t^* = u_{m+t}^*, \text{ for } t = 1, 2, \dots, n.$$

- (c) Based on the pseudo-data $\{x_1^*, \dots, x_n^*\}$, re-estimate the coefficients $\underline{\phi}$ by least-squares estimator $\hat{\underline{\phi}}^* = (\hat{\phi}_0^*, \hat{\phi}_1^*, \dots, \hat{\phi}_p^*)'$ as in step 1. Then compute the future bootstrap predicted values $\hat{x}_{n+1}^*, \dots, \hat{x}_{n+h}^*$ by the recursion:

$$\hat{x}_{n+t}^* = \hat{\phi}_0^* + \sum_{j=1}^p \hat{\phi}_j^* \hat{x}_{n+t-j}^* \text{ for } t = 1, \dots, h.$$

Where $\hat{x}_{n+t-j}^* = x_{n+t-j}$ when $t \leq j$

- (d) For AR(p) model, it is well known that the distribution of the future value X_{n+h} , conditional on all past observations, is the same as the distribution of X_{n+h} , given the last p observations. In order to get the conditional distribution of future value X_{n+h} , we fix the last p observations to the true observed values. i.e., $x_{n-p+1}^* = x_{n-p+1}, \dots, x_n^* = x_n$. Then we compute the future bootstrap observations $x_{n+1}^*, x_{n+2}^*, \dots, x_{n+h}^*$ by the recursion:

$$x_{n+t}^* = \hat{\phi}_0 + \sum_{j=1}^p \hat{\phi}_j x_{n+t-j}^* + \epsilon_{n+t}^*, \text{ for } t = 1, 2, \dots, h$$

- (e) Calculate a bootstrap root replicate as

$$x_{n+h}^* - \hat{x}_{n+h}^*$$

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 future values $\hat{x}_{n+1}, \dots, \hat{x}_{n+h}$ by following recursion:

$$\hat{x}_{n+t} = \hat{\phi}_0 + \sum_{j=1}^p \hat{\phi}_j \hat{x}_{n+t-j} \text{ for } t = 1, \dots, h.$$

Note that when $t \leq j$, $\hat{x}_{n+t-j} = x_{n+t-j}$

6. Construct the $(1 - \alpha)100\%$ equal-tailed predictive interval for X_{n+h} as

$$[\hat{x}_{n+h} + q(\alpha/2), \hat{x}_{n+h} + q(1 - \alpha/2)]. \quad (1.3)$$

1.2.2 Forward Bootstrap with Predictive Residuals

The fitted residuals $\epsilon_t = x_t - \hat{x}_t$ can be viewed as an effort to the transform the X -data towards "i.i.d.-ness". Motivated by Politis(2010), we consider another transformation: predictive residuals $\hat{\epsilon}_t^{(t)} = x_t - \hat{x}_t^{(t)}$, where $\hat{x}_t^{(t)}$ is computed from the delete- x_t data set, so the available data for the LS scatter plot of x_k vs. $\{x_{k-p}, \dots, x_{k-1}\}$ excludes this pair $(\{x_{t-p}, \dots, x_{t-1}\}, x_t)$.

The forward bootstrap procedure with predictive residuals is similar with that listed in section 1.2.1. The only difference is in step 2.

Algorithm 1.2.2. *Forward bootstrap with predictive residuals(Fp)*

1' same as step 1 in Algorithm 1.2.1.

2' Use the delete- x_t data set to compute the least square estimator:

$$\underline{\hat{\phi}}^{(t)} = (\hat{\phi}_0^{(t)}, \hat{\phi}_1^{(t)}, \dots, \hat{\phi}_p^{(t)})'$$

as in step 1. i.e., compute $\underline{\hat{\phi}}^{(t)}$ by changing regression model (1.2) as follows:

Delete the row of x_t in left hand side of (1.2), delete the row $(1, x_{t-1}, \dots, x_{t-p})$ in the design matrix of right hand side of (1.2) and change the $\underline{\hat{\phi}}$ to $\underline{\hat{\phi}}^{(t)}$ in right hand side .

Then we calculate the delete- x_t fitted values:

$$\hat{x}_t^{(t)} = \hat{\phi}_0^{(t)} + \sum_{j=1}^p \hat{\phi}_j^{(t)} x_{t-j}, t = p+1, \dots, n$$

and the predictive residuals: $\hat{\epsilon}_t^{(t)} = x_t - \hat{x}_t^{(t)}, t = p+1, \dots, n$

3-6' Change the $\hat{\epsilon}_t$ into $\hat{\epsilon}_t^{(t)}$; the rest is the same in Algorithm 1.2.1.

1.2.3 Forward Studentized Bootstrap with Fitted Residuals

In the previous two sections we have introduced the bootstrap methods using prediction roots. We can use studentized prediction roots instead of prediction roots, see Definition 1.1.1 and Remark 1.1.1. Inspired by Masarotto(1990), we propose our studentized bootstrap methods with fitted residuals. The forward bootstrap procedure with predictive residuals is similar with that listed in section 1.2.1. The only differences is in step 3(e) and 6.

Algorithm 1.2.3. *Forward Studentized bootstrap with fitted residuals(FSf)*

Change the 3(e) and 6 of Algorithm 1.2.1 into the following steps:

3(e) Calculate a studentized bootstrap root replicate as

$$\frac{x_{n+h}^* - \hat{x}_{n+h}^*}{\hat{\sigma}^* (\sum_{j=0}^{h-1} \hat{\psi}_j^{*2})^{1/2}},$$

where $\hat{\psi}_j^*$, $j = 0, 1, \dots$ are the coefficients of the power series $\hat{\psi}^*(B) = \hat{\phi}^{*-1}(B)$ and $\hat{\sigma}^*$ is the standard error of residuals $x_t^* - \hat{x}_t^*$, $t = p+1, \dots, n$.

6 Construct the $(1 - \alpha)100\%$ equal-tailed predictive interval for X_{n+h} as

$$[\hat{x}_{n+h} + \hat{\sigma} (\sum_{j=0}^{h-1} \hat{\psi}_j^2)^{1/2} q(\alpha/2), \hat{x}_{n+h} + \hat{\sigma} (\sum_{j=0}^{h-1} \hat{\psi}_j^2)^{1/2} q(1 - \alpha/2)]. \quad (1.4)$$

Where $q(\alpha)$ is the α -quantile of the empirical distribution of B collected studentized bootstrap roots, $\hat{\sigma}$ is the standard error of the fitted residuals $\hat{\epsilon}_t$ for $t = p+1, \dots, n$ and $\hat{\psi}_j$, $j = 0, 1, \dots$ are the coefficients of the power series $\hat{\psi}(B) = \hat{\phi}^{-1}(B)$

Remark 1.2.1. For all the algorithms introduced above, in step 3(c) and 3(d), we fix the last p values in pseudo series as observed values in calculating the future bootstrap predicted values and future bootstrap observations. If we calculate the future bootstrap predicted values and observations without fixing the last p values of pseudo series in Algorithm 1.2.3, then this algorithm is exactly the same as the method proposed by Masarotto(1990).

1.2.4 Forward Studentized Bootstrap with Predictive Residuals

Now let's use predictive residuals instead of the fitted residuals, following is the algorithm.

Algorithm 1.2.4. Forward Studentized bootstrap with predictive residuals(FSp)

1 Same as step 1 in Algorithm 1.2.1.

2 Same as step 2' in Algorithm 1.2.2

3-6 Change the $\hat{\epsilon}_t$ into $\hat{\epsilon}_t^{(t)}$; the rest is the same as Algorithm 1.2.3

Remark 1.2.2. The two methods(FSf and FSp) using studentized bootstrap root introduced above require more calculation in finding the $\hat{\sigma}$, $\hat{\sigma}^*$ and $\hat{\psi}_j^*$. However, in simulation we will see that the method with fitted residuals benefit a lot from studentizing the bootstrap root.

In the simulation at the end of this section, we will see that advantage of using the studentized root is that we do not need to use the predictive residuals and achieve the same performance as our Fp and Bp methods.

Remark 1.2.3. Our Fp method improved the Ff method through predictive residuals since Ff method is characterized by under-coverage(see the Monte Carlo studies in Section 1.7) and predictive residuals are inflated as compared to fitted residuals. However, if we studentized the residuals, the inflated effect is offset by the simultaneously inflated bootstrap estimator $\hat{\sigma}^*$. In simulation, we will see that our Fp and FSf methods have similar performances and FSp method is not necessary.

1.3 Asymptotic Properties

Theorem 1.3.1. Let $\{X_j\}$ be a stationary causal $AR(p)$ process defined by (1.1) with $E[\epsilon] = 0$, $\text{var}(\epsilon) = \sigma^2 > 0$ and $E|\epsilon|^4 < \infty$. Let $\{x_1, \dots, x_n\}$ denote a realization from $\{X_j\}$. Then as $n \rightarrow \infty$, the conditional bootstrap distribution of $\sqrt{n}(\hat{\phi}^* - \hat{\phi})$ and the distribution of $\sqrt{n}(\hat{\phi} - \phi)$ converge to the same limit in probability, and $|P^*(X_{n+m}^* \leq x) - P(X_{n+m} \leq x)| \xrightarrow{d} 0$ for $m > 0$, where P^* is the conditional probability given the data.

Proof. Using the argument along the lines described in Freedman(1984), Theorem 4.1 together with formulas (4.14),(4.20) of Freedman(1984) ensure that the conditional distribution of $\sqrt{n}(\hat{\phi}^* - \hat{\phi})$ and the distribution of $\sqrt{n}(\hat{\phi} - \phi)$ converge to the same limit. See also Kreiss(1997).

Note that the last p values $X_{n-p+1}^*, \dots, X_n^*$ are fixed to be the $x_{n-p+1}, \dots, x_n$ and

$$\begin{aligned} X_{n+1}^* &= \begin{pmatrix} 1 & Y_n' \end{pmatrix} \underline{\hat{\phi}} + \epsilon_{n+1}^* \\ X_{n+1} &= \begin{pmatrix} 1 & Y_n' \end{pmatrix} \underline{\phi} + \epsilon_{n+1}. \end{aligned}$$

Freedman(1984) proved that $\hat{F}_\epsilon \rightarrow F_\epsilon$. i.e., $\epsilon^* \xrightarrow{d} \epsilon$. Apply this result and Slutsky's Lemma, to prove by induction that X_{n+m}^* and X_{n+m} have the same limit distribution for any $m > 0$. $\square$

Lemma 1.3.2. $\hat{\epsilon}_t - \epsilon_t^{(t)} = O_p(\frac{1}{n})$

Proof. Let $M = \begin{bmatrix} 1 & X_{n-1} & \cdots & X_{n-p} \\ 1 & X_{n-2} & \cdots & X_{n-p-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & X_p & \cdots & X_1 \end{bmatrix}$, $V = \begin{pmatrix} X_n \\ X_{n-1} \\ \vdots \\ X_{p+1} \end{pmatrix}$;

M_t be the matrix after deleting the row $Z_t' = (1, X_{t-1}, \dots, X_{t-p})$ in M , V_t be the vector after deleting X_t in V . Then

$$\begin{aligned} \underline{\hat{\phi}} &= (M'M)^{-1}M'V \\ &= [(M_t'M_t)^{-1}Z_tZ_t' + I_{p+1}]^{-1}(M_t'M_t)^{-1}(M_t'V_t + Z_tX_t) \\ &= [I_{p+1}(1 + O_p(\frac{1}{n}))]^{-1}(M_t'M_t)^{-1}M_t'V_t(1 + O_p(\frac{1}{n})) \\ &= \underline{\hat{\phi}}^{(t)} + O_p(\frac{1}{n}) \end{aligned}$$

Here $X_i = O_p(1)$, $i = 1, 2, \dots, n$. Z_tZ_t' is a $(p+1) \times (p+1)$ matrix whose entries are of $O_p(1)$ while $M_t'M_t$ is a $(p+1) \times (p+1)$ matrix whose entries are the summation of n bounded in probability terms, then $Z_tZ_t' = M_t'M_t \cdot O_p(\frac{1}{n})$. Similarly $Z_tX_t = M_t'V_t \cdot O_p(\frac{1}{n})$.

$$\text{So } \hat{\epsilon}_t - \epsilon_t^{(t)} = Z_t'(\underline{\hat{\phi}}^{(t)} - \underline{\hat{\phi}}) = O_p(\frac{1}{n}). \quad \square$$

Theorem 1.3.3. Let $\hat{F}_n$ be the empirical distribution of fitted residuals $\{\hat{\epsilon}_t\}$, F_ϵ be the distribution of errors ϵ satisfying the assumptions of Theorem 1.3.1 and $\sup_x |F_\epsilon''(x)| < \infty$.

$$\sup_x |\hat{F}_n(x) - F_\epsilon(x)| = O_p(1/\sqrt{n}),$$

Proof. Let $F_n = \frac{1}{n-p} \sum_{t=p+1}^n 1_{\{\epsilon_t \leq x\}}$ be the empirical distribution of the errors. Similarly, $\hat{F}_n = \frac{1}{n-p} \sum_{t=p+1}^n 1_{\{\hat{\epsilon}_t \leq x\}}$. Boldin(1982) proved $\sup_x \sqrt{n} |\hat{F}_n(x) - F_n(x)| \rightarrow 0$ in probability. By the central limit theorem of empirical distribution, we have $F_n(x) = F_\epsilon(x) + O_p(1/\sqrt{n})$ for any x .

□

Lemma 1.3.4. *Let $\tilde{F}_n$ be the empirical distribution of predictive residuals. Under the same assumptions of Theorem 1.3.3*

$$d_2(\tilde{F}_n, \hat{F}_n) = O_p(1/n).$$

d_2 is the Mallows distance.

Proof. Let ϵ_t^* is sampled from $\hat{F}_n$, $\tilde{\epsilon}_t$ is sampled from $\tilde{F}_n$, and ϵ_t^* , $\tilde{\epsilon}_t$ have the same distribution as $\hat{\epsilon}_J$ and $\epsilon_J^{(t)}$, where J taking values uniformly from $(p+1, \dots, n)$. Apply Lemma 1.3.2, we have

$$\begin{aligned} d_2(\hat{F}_n, \tilde{F}_n) &\leq (E(\hat{\epsilon}_J - \epsilon_J^{(t)})^2)^{1/2} \\ &= \left(\frac{1}{n-p} \sum_{t=p+1}^n (\hat{\epsilon}_t - \epsilon_t^{(t)})^2 \right)^{1/2} = O_p(1/n), \end{aligned}$$

□

Note that

$$\begin{aligned} X_{n+h}^* - \hat{X}_{n+h}^* &= \begin{pmatrix} 1 & Y_{n+h-1}^* \end{pmatrix}' (\hat{\underline{\phi}} - \underline{\hat{\phi}}^*) + \epsilon_{n+h}^* \\ X_{n+h} - \hat{X}_{n+h} &= \begin{pmatrix} 1 & Y_{n+h-1} \end{pmatrix}' (\underline{\hat{\phi}} - \hat{\underline{\phi}}) + \epsilon_{n+h} \end{aligned}$$

Let $Z_{nh} = \begin{pmatrix} 1 & Y_{n+h-1} \end{pmatrix}' (\underline{\hat{\phi}} - \underline{\phi})$ and $Z_{nh}^* = \begin{pmatrix} 1 & Y_{n+h-1}^* \end{pmatrix}' (\underline{\hat{\phi}}^* - \underline{\hat{\phi}})$. Let F_ϵ be the distribution function of ϵ_t and $\hat{F}_\epsilon$ be the distribution of $\hat{\epsilon}_t$. Assume F_ϵ and $\hat{F}_\epsilon$ are twice continuously differentiable, (if empirical distribution is used, it need to be smoothed first) and $F'_\epsilon = f_\epsilon$, $\hat{F}'_\epsilon = \hat{f}_\epsilon$. Let $\hat{F}$ be the distribution of fitted residuals, then the empirical distribution of the fitted residuals $\hat{F}_n = \hat{F} + O_p(1/\sqrt{n})$ by the property of empirical distribution and the smoothed estimator of the distribution of fitted residuals $\hat{F}_\epsilon = \hat{F} + O_p(1/\sqrt{n})$, cf. Berg and Politis(2009) and

Elharfaoui and Harel(2008). So The difference between $\hat{F}_n$ and $\hat{F}_\epsilon$ is of $O_P(1/\sqrt{n})$. Together with $|\hat{F}_n(\alpha) - F_\epsilon(\alpha)| = O_p(1/\sqrt{n})$ as showed in Theorem 1.3.3, we have $|F_\epsilon - \hat{F}_\epsilon| = O_p(1/\sqrt{n})$.

$$\begin{aligned}
& P(X_{n+h} - \hat{X}_{n+h} \leq \alpha | Y_n = y_n) \\
&= \int_{-\infty}^{\infty} P(\epsilon_{n+h} \leq \alpha + z) dF_{Z_{nh}}(z) \\
&= F_\epsilon(\alpha) + f_\epsilon(\alpha) E(Z_{nh} | Y_n = y_n) + \frac{f'_\epsilon(\alpha)}{2} E[Z_{nh}^2 | Y_n = y_n] + o(E[Z_{nh}^2 | Y_n = y_n]).
\end{aligned} \tag{1.5}$$

Here $F_{Z_{nh}}(z)$ is the conditional distribution function of Z_{nh} given $Y_n = y_n$.

In the bootstrap world, we have

$$\begin{aligned}
& P^*(X_{n+h}^* - \hat{X}_{n+h}^* \leq \alpha | Y_n^* = y_n) \\
&= \hat{F}_\epsilon(\alpha) + \hat{f}_\epsilon(\alpha) E^*(Z_{nh}^* | Y_n^* = y_n) + \frac{\hat{f}'_\epsilon(\alpha)}{2} E^*[(Z_{nh}^*)^2 | Y_n^* = y_n] \\
&+ o_p(E^*[(Z_{nh}^*)^2 | Y_n^* = y_n])
\end{aligned} \tag{1.6}$$

Where E^* is the conditional expectation given the original observations and P^* is the conditional probability given the observations.

Theorem 1.3.3 guarantees the asymptotic validity of interval (1.3). But it is more than validity. By Theorem 1.3.1, $\sqrt{n}(\hat{\phi}^* - \hat{\phi})$ and $\sqrt{n}(\hat{\phi} - \phi)$ have the same limit. This limit is a normal distribution with mean 0 and some finite variance. cf. Kreiss(1997 Property 4.2).

Then we have $E(\sqrt{n}Z_{nh} | Y_n = y_n) = o(1)$, $E^*(\sqrt{n}Z_{nh}^* | Y_n^* = y_n) = o_p(1)$ and $E(nZ_{nh}^2 | Y_n = y_n) = O(1)$, $E^*[n(Z_{nh}^*)^2 | Y_n^* = y_n] = O_p(1)$, but $E(nZ_{nh}^2 | Y_n = y_n) - E^*[n(Z_{nh}^*)^2 | Y_n^* = y_n] = o_p(1)$.

Since $E(Z_{nh} | Y_n = y_n) - E^*(Z_{nh}^* | Y_n^* = y_n) = o_p(1/\sqrt{n})$ and $E(Z_{nh}^2 | Y_n = y_n) - E^*[(Z_{nh}^*)^2 | Y_n^* = y_n] = o_p(1/n)$, (1.5) and (1.6) will match up to the first three terms if $\hat{F}_\epsilon = F_\epsilon$, which leads to asymptotic validity and correctness.

We will discuss the sieve bootstrap and the bootstrap method introduced by Pascual, Romo and Ruiz(2004) later in Chapter 1.8.4 and show that even if $\hat{F}_\epsilon = F_\epsilon$, the discrepancy of third term of the distribution functions is $O_p(1/n)$ instead of $o_p(1/n)$. Thus, our method has higher order accuracy.

Lemma 1.3.2 and 1.3.4 ensure the same conclusion for the method with predictive residuals.

1.4 Backward Bootstrap and Asymptotic Properties

The difference of the backward bootstrap and forward bootstrap is in generating the bootstrap-pseudo data. The idea of starting from the last p observations that are given and using the backward errors to generate the bootstrap-pseudo data $\{X_{n-p}^*, \dots, X_1^*\}$ backward in time using the backward representation $\phi(B^{-1})X_t = \phi_0 + \frac{\phi(B^{-1})}{\phi(B)}\epsilon_t$ was first proposed by Thombs and Schucany(1990) and improved by Breidt, Davis and Dunsmuir(1995). Here, B is the backward shift operator: $B^k X_t = X_{t-k}$.

Let $\{w_t\}$ be the backward noises defined by

$$w_t = \frac{\phi(B^{-1})}{\phi(B)}\epsilon_t. \quad (1.7)$$

Thombs and Shucany(1990) generated the fitted backward residuals $\hat{w}_t$ as $\hat{w}_t = x_t - \hat{\phi}_0 - \hat{\phi}_1 x_{t+1} - \dots - \hat{\phi}_p x_{t+p}$, for $t = 1, 2, \dots, n-p$. Then they fixed the last p values and generated the pseudo series backwards through the following backwards recursion, $x_t^* = \hat{\phi}_0 + \hat{\phi}_1 x_{t+1}^* + \dots + \hat{\phi}_p x_{t+p}^* + w_t^*$, for $t = n-p, n-p-1, \dots, 1$. Where w_t is generated from $\hat{F}_w$, the empirical distribution of $\hat{w}_t$ s. However, as mentioned in Breidt, Davis and Dunsmuir(1995), although the backward errors w_t s are uncorrelated, they are dependent. So we cannot resample $\{w_t^*\}$ directly from $\hat{F}_w$. However, the forward noise $\{\epsilon_t\}$ are independent and we can generate ϵ_t^* directly from $\hat{F}_\epsilon$. After getting the ϵ_t^* s, we are able to generate the backward noise $\{w_t\}$ via the following recursion,

$$w_t^* = \frac{\hat{\phi}(B^{-1})}{\hat{\phi}(B)}\epsilon_t^*.$$

Our algorithm for backward bootstrap with fitted residuals is exactly the same as Breidt, Davis and Dunsmuir(1995). But we also propose the backward bootstrap with predictive residuals which works much better as shown in the simulation section.

1.4.1 Algorithm for Backward Bootstrap with Fitted Residuals

Algorithm 1.4.1. *Backward bootstrap with fitted residuals(Bf)*

1-2. same as the steps in Algorithm 1.2.1.

3. Center the fitted residuals: let $r_t = \hat{\epsilon}_t - \bar{\hat{\epsilon}}$ for $t = p + 1, \dots, n$, and $\bar{\hat{\epsilon}} = (n - p)^{-1} \sum_{p+1}^n \hat{\epsilon}_t$, 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 $\epsilon_{-M}^*, \dots, \epsilon_n^*, \epsilon_{n+1}^*, \dots$ from $\hat{F}_n$, and then generate the bootstrap backward noises $\{w_t^*, t = -M, \dots, n\}$ recursively as follows,

$$w_t^* = \begin{cases} 0, & t < -M \\ \hat{\phi}_1 w_{t-1}^* + \dots + \hat{\phi}_p w_{t-p}^* + \epsilon_t^* - \hat{\phi}_1 \epsilon_{t+1}^* - \dots - \hat{\phi}_p \epsilon_{t+p}^*, & t \geq -M \end{cases}$$

(b) Fix the last p values, i.e., $x_n^* = x_n, \dots, x_{n-p+1}^* = x_{n-p+1}$, and then generate a bootstrap realization $\{X_t^*\}$ by the backward recursion:

$$x_t^* = \begin{cases} \hat{\phi}_0 + \hat{\phi}_1 x_{t+1}^* + \dots + \hat{\phi}_p x_{t+p}^* + w_t^* & t = n - p, n - p - 1, \dots, 1 \\ x_t & t = n, n - 1, \dots, n - p + 1. \end{cases}$$

(c) Based on the pseudo-data $\{x_1^*, \dots, x_n^*\}$, re-estimate the coefficients $\underline{\phi}$ by least-squares estimators $\hat{\underline{\phi}}^* = (\hat{\phi}_0^*, \hat{\phi}_1^*, \dots, \hat{\phi}_p^*)'$ as in step 1. Then compute the future bootstrap predicted values $x_{n+1}^*, \dots, x_{n+h}^*$ via:

$$\hat{x}_{n+t}^* = \hat{\phi}_0^* + \sum_{j=1}^p \hat{\phi}_j^* \hat{x}_{n+t-j}^* \text{ for } t = 1, \dots, h.$$

Where $\hat{x}_{n+t-j}^* = x_{n+t-j}^*$ when $t \leq j$

(d) Compute the future bootstrap observations $x_{n+1}^*, x_{n+2}^*, \dots, x_{n+h}^*$ through the last p observations by the forward recursion:

$$x_{n+t}^* = \hat{\phi}_0 + \sum_{j=1}^p \hat{\phi}_j x_{n+t-j}^* + \epsilon_{n+t}^*, \text{ for } t = 1, 2, \dots, h$$

(e) Calculate a bootstrap root replicate as

$$x_{n+h}^* - \hat{x}_{n+h}^*$$

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 future values $\hat{x}_{n+1} \cdots, \hat{x}_{n+h}$ by following recursion:

$$\hat{x}_{n+t} = \hat{\phi}_0 + \sum_{j=1}^p \hat{\phi}_j \hat{x}_{n+t-j} \text{ for } t = 1, \dots, h.$$

Note that when $t \leq j$, $\hat{x}_{n+t-j} = x_{n+t-j}$

6. Construct the $(1 - \alpha)100\%$ equal-tailed predictive interval for X_{n+h} as

$$[\hat{x}_{n+h} + q(\alpha/2), \hat{x}_{n+h} + q(1 - \alpha/2)]. \quad (1.8)$$

1.4.2 Algorithm for Backward Bootstrap with Predictive Residuals

Algorithm 1.4.2. *Backward bootstrap with predictive residuals(Bp)*

1-2. Same as steps 1-2 in Algorithm 1.2.2

3-6. Change the $\hat{\epsilon}_t$ into $\hat{\epsilon}_t^{(t)}$, the predictive residual defined in Algorithm 1.2.2 step 2', the rest is the same in Algorithm 1.4.1.

1.4.3 Algorithm for Backward Studentized Bootstrap with fitted Residuals

Algorithm 1.4.3. *Backward bootstrap with predictive residuals(BSf)*

This algorithm is the same as Algorithm 1.4.1 except steps 3(e) and 6. These 2 steps are the same as steps 3(e) and 6 in Algorithm 1.2.3.

1.4.4 Algorithm for Backward Studentized Bootstrap with Predictive Residuals

Algorithm 1.4.4. *Backward bootstrap with predictive residuals(BSp)*

Change the $\hat{\epsilon}_t$ into $\hat{\epsilon}_t^{(t)}$, the predictive residual defined in Algorithm 1.2.2 step 2', the rest is the same in Algorithm 1.4.3.

Remark 1.4.1. *Both Masarotto(1990) and Pascual et al.(2004) only worked on forward bootstrap methods. Masarotto(1990) did not fix the last p values as mentioned in Remark 1.2.1 and Pascual et al.(2004) did not use the distribution of the bootstrap predictive root to approximate the distribution of the predictive root, we will discussed it in detail later in Section 1.8.*

Remark 1.4.2. *Same as the forward methods, we will see in the simulation that the Bp and BSf improve a lot compared to Bf, and technique of predictive residuals is not necessary for studentized method(BSp).*

Remark 1.4.3. *The least-squares estimator $\hat{\phi}$ is asymptotically causal but not guaranteed to be causal in finite samples while the Yule-Walker estimator is always causal. However Yule-Walker will not work with our predictive residuals as you need to delete the present x_t and its future value. So if the LS estimator $\hat{\phi}$ is not causal, then we can use Yule-Walker estimator and do the algorithm with fitted residuals only. If $\hat{\phi}$ is causal, we can use both fitted and predictive residuals but discard the bootstrap pseudo series that lead to a non-causal $\hat{\phi}^*$. In practice, for the forward bootstrap method, even though we get the non-causal estimates which cause the pseudo series to explode, since we fix the last p values to be the true observed values, if h is not large, the $\hat{\phi}^*$ is also close to $\hat{\phi}$ and the distribution of the bootstrap predictive root is still close to the predictive root. But for the backward bootstrap, the backward errors $\{w_t\}$ computed via (1.7) are going to explode when n is large, which leads to erratic behavior of $\hat{\phi}^*$ and incredibly large interval lengths.*

1.5 Joint Prediction intervals

If observe time series $\{X_1, \dots, X_n\}$, we wish to find the joint prediction interval for $\{X_{n+1}, \dots, X_{n+h}\}$. It is very hard to find this joint prediction interval if we do not use the bootstrap method. Let us look at the predictive root in the form of an expression only depending on past data and future errors as follows:

$$X_{n+h} = g(Y_n; \underline{\phi}) + \sum_{j=0}^{h-1} \psi_j \epsilon_{n+h-j} \quad (1.9)$$

$$\hat{X}_{n+h} = g(Y_n; \hat{\underline{\phi}}) \quad (1.10)$$

$$X_{n+h} - \hat{X}_{n+h} = \left(g(Y_n; \underline{\phi}) - g(Y_n; \hat{\underline{\phi}}) \right) + \sum_{j=0}^{h-1} \psi_j \epsilon_{n+h-j} \quad (1.11)$$

$g(\cdot)$ is a linear function of Y_n and the coefficients only depend on $\underline{\phi}$ where $Y_n = (X_n, \dots, X_{n-p+1})$ and $\{\psi_j\}$ are the coefficients of the power series $\Phi^{-1}(B)$. If we do not use the bootstrap method, we need to find the distribution of (1.11). The first difficulty is in capturing the distribution of the first term $g(Y_n; \underline{\phi}) - g(Y_n; \hat{\underline{\phi}})$ in (1.11). This term is quite small compared to the second term $\sum_{j=0}^{h-1} \psi_j \epsilon_{n+h-j}$, however, even if we ignore the first term, the second difficulty is that the second terms of $\{X_{n+1}, \dots, X_{n+h}\}$ are dependent! If we assume the $\{\epsilon_t\}$ are i.i.d. normal distributed errors, given $Y_n = y$, $\{X_{n+1}, \dots, X_{n+h}\}$ has multivariate normal distribution. Then we can estimate the covariance matrix and find the inverse of the estimated covariance matrix and form the joint prediction interval. However, this method ignores the variability from estimating $\underline{\phi}$ by $\hat{\underline{\phi}}$ and the assumption of normal errors is often violated in practice. Moreover, the estimated covariance matrix is not always invertible.

We can construct not only an prediction interval for any future value but also the joint prediction interval for $\{X_{n+1}, \dots, X_{n+h}\}$ because bootstrap can equally mimic the univariate distribution or multivariate distribution. In our case, we use the bootstrap to mimic the distribution of predictive roots or studentized predictive roots.

To construct the joint prediction interval using bootstrap predictive root, we can collect B bootstrap replicates of the t -step ahead bootstrap predictive roots

$X_{n+t}^* - \hat{X}_{n+t}^*$ in the form of empirical distribution whose α -quantile is denoted as $q_t(\alpha)$ for each $t \in \{1, \dots, h\}$. A possible way to construct the joint prediction interval is as follows through Bonferroni correction

$$[\hat{X}_{n+t} + q_t(\alpha/2h), \hat{X}_{n+t} + q_t(1 - \alpha/2h)] \text{ for } t = 1, \dots, h.$$

However, as mentioned above, the t -step ($t = 1, \dots, h$) ahead prediction intervals are dependent. We can not simply apply the Bonferroni correction to our case.

To find the joint $1 - \alpha$ prediction intervals, we can look at the distribution of all the Bh predictive roots we collected (h roots from each bootstrap path and in total B paths). However, the variance of the predictive roots $X_{n+h} - \hat{X}_{n+h}$ increase with h , which may lead to unbalanced joint interval. This issue will be automatically taken care of if we use the bootstrap method to mimic the distribution of studentized predictive roots instead of predictive roots. We collect h studentized bootstrap root replicates for each bootstrap step, repeat the bootstrap step B times and in total we have Bh studentized bootstrap root replicates. Let $q(\alpha)$ be the α -quantile of the empirical distribution of those Bh bootstrap replicates. Then the joint prediction interval is

$$[\hat{X}_{n+t} + \hat{\sigma}(\sum_{j=0}^{t-1} \hat{\psi}_j^2)^{1/2} q(\alpha/2), \hat{X}_{n+t} + \hat{\sigma}(\sum_{j=0}^{t-1} \hat{\psi}_j^2)^{1/2} q(1 - \alpha/2)] \text{ for } t = 1, \dots, h$$

1.6 Generalized Bootstrap prediction Intervals

Chatterjee and Bose(2005) introduced the generalized bootstrap method for estimators obtained by solving estimating equations. The least-squares estimators of the coefficients we use in this chapter is a special case. With a bootstrapped weight $(w_{n1}, \dots, w_{nn})$ in the estimating equations, we have the generalized bootstrapped estimators by solving the bootstrapped estimating equations. If we sample $(w_{n1}, \dots, w_{nn})$ from Multinomial($n, 1/n, \dots, 1/n$), we get the paired bootstrap estimator of the coefficients.

Inspired by the idea of generalized bootstrap, we apply this bootstrap method

into constructing bootstrap prediction interval and form a new bootstrap approach for bootstrap prediction intervals. The generalized bootstrap method is computationally fast because we do not need to generate the pseudo series, instead we sample the weight $(w_{n1}, \dots, w_{nn})$ from some distribution. Following is the bootstrap algorithm for generalized bootstrap with fitted residuals.

Algorithm 1.6.1. *Generalized bootstrap with fitted residuals(Gf)*

1-2 Same as the steps in algorithm 1.2.1

3 (a) Calculated the bootstrapped estimator of the coefficients

$$\hat{\phi}^* = (X'WX)^{-1}X'WY,$$

where $X = \begin{bmatrix} 1 & x_{n-1} & \cdots & x_{n-p} \\ 1 & x_{n-2} & \cdots & x_{n-p-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_p & \cdots & x_1 \end{bmatrix}$, $Y = \begin{pmatrix} x_n \\ x_{n-1} \\ \vdots \\ x_{p+1} \end{pmatrix}$ and W is a diagonal matrix whose diagonal elements $(w_1, \dots, w_{n-p})$ are sampled from a multinomial distribution $(n-p, 1/(n-p), \dots, 1/(n-p))$.

(b) Compute the future bootstrap predicted values $\hat{X}_{n+1}^*, \dots, \hat{X}_{n+h}^*$ by the recursion:

$$\hat{X}_{n+t}^* = \hat{\phi}_0^* + \sum_{j=1}^p \hat{\phi}_j^* \hat{X}_{n+t-j}^* \text{ for } t = 1, \dots, h.$$

and the future bootstrap observations $X_{n+1}^*, X_{n+2}^*, \dots, X_{n+h}^*$ by the recursion:

$$X_{n+t}^* = \hat{\phi}_0 + \sum_{j=1}^p \hat{\phi}_j X_{n+t-j}^* + \epsilon_{n+t}^*, \text{ for } t = 1, 2, \dots, h.$$

Where $\hat{X}_{n+t-j}^* = X_{n+t-j}^* = x_{n+t-j}$ when $t \leq j$ and ϵ_{n+t}^* is sampled from the centered fitted residuals. And then calculate the bootstrap predictive root replicate as $X_{n+h}^* - \hat{X}_{n+h}^*$

4-6 Same as the steps from algorithm 1.2.1

Algorithm 1.6.2. *Generalized bootstrap with predictive residuals(Gp)*

Change the step 2 of algorithm 1.6.1 into the step 2 of algorithm 1.2.2 and use the predictive residuals instead of the fitted residuals in step 3[b] of the algorithm 1.6.1, we have the algorithm for generalized bootstrap with predictive residuals.

The consistency of the conditional distribution of $\hat{\underline{\phi}}^* - \hat{\underline{\phi}}$ and the distribution of $\hat{\underline{\phi}} - \underline{\phi}$ is given in Chatterjee and Bose(2005).

Remark 1.6.1. *Chatterjee and Bose(2005) shows that the resampling conditional variance of $\sqrt{n}\sigma_n^{-1}(\hat{\underline{\phi}}^* - \hat{\underline{\phi}})$ is consistent for the asymptotic variance of $\sqrt{n}(\hat{\underline{\phi}} - \underline{\phi})$. Note that there's a σ_n^{-1} in the first expression, where σ_n^2 is the variance of w_i . If we sample the weight $(w_1, \dots, w_n)$ from a multinomial distribution and the variance is $1 - 1/(n-p)$, the difference between the conditional variance of $(\hat{\underline{\phi}}^* - \hat{\underline{\phi}})$ and $\sigma_n^{-1}(\hat{\underline{\phi}}^* - \hat{\underline{\phi}})$ is of order $O(\frac{1}{n^2})$, which is negligible.*

1.7 Monte Carlo Studies

In this section, we evaluate the performance of all the 10 methods above, 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 in the case of AR(1) model: $X_{t+1} = \phi_1 X_t + \epsilon_t$ with

- (1) $\phi_1 = 0.9$ or 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 data sets each of size $n = 50$ or 100 , and for each data set we create $B = 1000$ bootstrap pseudo series.
- (4) prediction interval with nominal coverage levels of 95% and 90%.

For each type of intervals, we assess the corresponding empirical coverage level (CVR) and average length (LEN). For each type of bootstrap method, we generate 500 data sets each with size $n = 50$ or 100 . And for each data set, we use the bootstrap method to create $B = 1000$ bootstrap sample paths(step 4) and 1000 one-step ahead future values($Y_{n+1,j}, j = 1, 2, \dots, 1000$). We construct the prediction interval(step 6) $(L_i, U_i), i = 1, 2, \dots, 500$, and the coverage level and length as

$$CVR_i = \frac{1}{1000} \sum_{j=1}^{1000} 1_{[L_i, U_i]}(Y_{n+1,j}),$$

$$LEN_i = U_i - L_i.$$

Note that for i th data set of size $n, i = 1, 2, \dots, 500$, we have the prediction interval (L_i, U_i) of X_{n+1} given $X_n = x_{ni}$, where x_{ni} is the last observation from the i th data set. The values of these x_{ni} s are different and are determined by the data sets.

Then the coverage level and length for each bootstrap method can be calculated by the average $\{CVR_i\}$ and $\{LEN_i\}$, i.e.

$$CVR = \frac{1}{500} \sum_{i=1}^{500} CVR_i,$$

$$LEN = \frac{1}{500} \sum_{i=1}^{500} LEN_i.$$

Table 1.1: Simulation Results of AR(1) with normal innovations and $\phi_1 = 0.5$

normal $\phi_1 = 0.5$	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
$n = 100$						
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

Table 1.2: Simulation Results of AR(1) with normal innovations and $\phi_1 = 0.9$

normal $\phi_1 = 0.9$	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
Ff	0.933	3.906	0.489	0.884	3.306	0.395
Fp	0.943	4.063	0.513	0.898	3.443	0.411
Bf	0.927	3.824	0.485	0.878	3.255	0.388
Bp	0.939	4.007	0.516	0.893	3.402	0.408
FSf	0.945	4.107	0.515	0.898	3.450	0.411
FSp	0.945	4.099	0.506	0.899	3.444	0.408
BSf	0.939	4.018	0.502	0.892	3.397	0.403
BSp	0.940	4.032	0.501	0.894	3.406	0.399
Gf	0.928	3.838	0.490	0.878	3.261	0.394
Gp	0.938	3.999	0.505	0.892	3.393	0.406
$n = 100$						
Ff	0.941	3.915	0.355	0.893	3.306	0.282
Fp	0.945	3.989	0.371	0.899	3.368	0.282
Bf	0.939	3.892	0.363	0.891	3.284	0.273
Bp	0.945	3.976	0.372	0.898	3.356	0.287
FSf	0.946	4.005	0.355	0.900	3.373	0.286
FSp	0.946	3.997	0.365	0.899	3.369	0.279
BSf	0.945	3.981	0.362	0.898	3.352	0.274
BSp	0.945	3.982	0.355	0.898	3.355	0.282
Gf	0.939	3.890	0.355	0.890	3.286	0.275
Gp	0.944	3.967	0.381	0.897	3.353	0.285

Table 1.3: Simulation Results of AR(1) with Laplace innovations and $\phi_1 = 0.5$

Laplace $\phi_1 = 0.5$	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
$n = 100$						
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	0.318	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

Table 1.4: Simulation Results of AR(1) with Laplace innovations and $\phi_1 = 0.9$

Laplace $\phi_1 = 0.9$	nominal coverage 95%				nominal coverage 90%		
$n = 50$	CVR	LEN	st.err		CVR	LEN	st.err
Ff	0.930	4.236	0.826		0.886	3.317	0.574
Fp	0.937	4.417	0.833		0.896	3.462	0.597
Bf	0.929	4.179	0.894		0.881	3.280	0.645
Bp	0.937	4.380	0.977		0.892	3.426	0.681
FSf	0.943	4.520	0.885		0.898	3.502	0.595
FSp	0.943	4.510	0.866		0.899	3.510	0.601
BSf	0.939	4.451	0.963		0.894	3.458	0.671
BSp	0.940	4.481	0.987		0.896	3.470	0.685
Gf	0.929	4.159	0.780		0.880	3.260	0.576
Gp	0.936	4.343	0.864		0.891	3.411	0.597
$n = 100$							
Ff	0.940	4.229	0.620		0.892	3.286	0.429
Fp	0.944	4.321	0.636		0.897	3.358	0.428
Bf	0.940	4.225	0.618		0.891	3.270	0.423
Bp	0.943	4.302	0.626		0.896	3.336	0.427
FSf	0.946	4.367	0.620		0.899	3.382	0.428
FSp	0.946	4.365	0.627		0.899	3.380	0.423
BSf	0.945	4.356	0.624		0.898	3.365	0.431
BSp	0.945	4.353	0.617		0.898	3.356	0.416
Gf	0.940	4.233	0.620		0.892	3.285	0.418
Gp	0.943	4.316	0.641		0.896	3.348	0.436

The two tables above summarize the average coverage level and the average length of the prediction intervals for normal and Laplace errors. The st.err in the table is the standard error associated with each average length. We can see some important features through the two tables above:

- When the error is i.i.d form $N(0,1)$ or Laplace distribution re-scaled to unit variance, for all methods mentioned above, the coverage levels and intervals are quite reliable.
- The coverage of the Gf intervals resemble that of Ff and Bf intervals. Similarly, the coverage of Gp intervals resemble that of Fp and Bp intervals.
- All the prediction intervals are characterized by under- coverage. However Fp,

Bp and Gp methods using predictive residuals outperform Ff, Bf and Gf using fitted residuals since they have better coverage in all cases of ϕ for both normal and Laplace errors.

- The predictive residuals are inflated as compared to the fitted residuals. i.e. the empirical distribution of the predictive residuals is very much like a re-scaled version of the empirical distribution of fitted residuals.
- The price to pay for using predictive residuals is the increased variability of the interval length. However this is a finite sample effect since asymptotically, the omission of a finite number of points from the scatter plot makes no difference. The use of predictive residuals appears to partially correct the problem of under-coverage of bootstrap predictive intervals as it is the case in linear regression; see Politis(2010,2013).
- The four studentized methods have similar performance of Fp and Bp methods.
- Using predictive residuals is not necessary for studentized method as discussed in Remark 1.2.3.

To see the convergence properties Ff, Fp,Bf,Bp methods, consider one data set of sample size $n = 100$ with normal errors, and generate 1000 bootstrap sample paths to construct the prediction interval. The following figure is the distribution of the bootstrap roots and the true root. Note that the true root, $X_{n+1} - \hat{X}_{n+1} = \hat{\epsilon}_{n+1} \xrightarrow{d} \epsilon_{n+1}$, can be approximated by the true error which has $N(0,1)$ distribution. Figure 1.1 contains the densities of the bootstrap and the true predictive roots

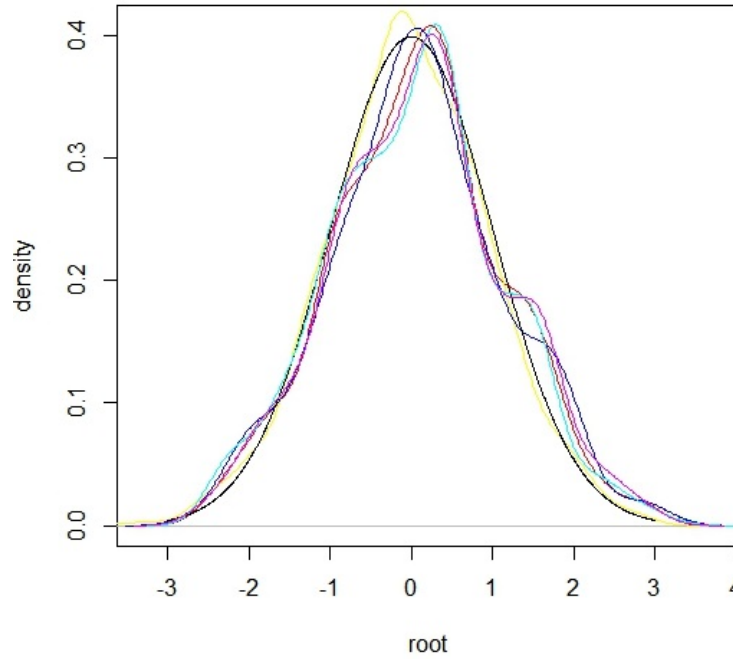


Figure 1.1: densities of bootstrap predictive roots vs. true predictive root

The black curve is the probability density of $N(0,1)$ distribution. The yellow curve is the density of the predictive root: $X_{n+1} - \hat{X}_{n+1}$, where X_{n+1} is generated by $X_{n+1} = \phi X_n + \epsilon_{n+1}$, $\phi = 0.9$ and ϵ_{n+1} is generated from $N(0,1)$, $\hat{X}_{n+1} = \hat{\phi} X_n$. The red curve is the density of Ff bootstrap predictive root, the blue curve is Fp predictive bootstrap root, the green curve is Bf bootstrap predictive root, the purple curve is Bp bootstrap predictive root. From the figure, we see that the densities of all the bootstrap roots are very close to the $N(0,1)$ distribution.

1.8 Alternative Approaches to Bootstrap Prediction Intervals for Linear Autoregression Model

1.8.1 Bootstrap Prediction Intervals Based on Studentized Predictive Root

Box and Jenkins(1976) proposed a widely used prediction interval for an AR(p) model as

$$[\hat{x}_{n+h} + z_{\alpha/2}\hat{\sigma}(\sum_{j=0}^{h-1}\hat{\psi}_j^2)^{1/2}, \hat{x}_{n+h} + z_{1-\alpha/2}\hat{\sigma}(\sum_{j=0}^{h-1}\hat{\psi}_j^2)^{1/2}], \quad (1.12)$$

where $\hat{\psi}_j$, $j = 0, 1, \dots$ are the coefficients of the power series $\hat{\psi}(B) = \hat{\phi}^{-1}(B)$, z_α is the α th quantile of a standard normal variate. $\hat{\sigma}$ is an estimate of σ , the standard deviation of the innovations $\{\epsilon_t\}$. This prediction interval only takes into account the variability from the errors but does not account for the variability from the estimation of the model, i.e parameter estimates. Moreover, to use this prediction interval, we need to assume the errors are Gaussian. Although this interval is asymptotically valid, it is not correct and is expected to be characterized by under-coverage in finite samples.

To relax the Gaussian assumption and consider the variability from both the errors and the model estimation, Masarotto(1990) proposed a bootstrap method to construct the prediction interval as follows:

for each pseudo series $x_1^*, \dots, x_n^*, \dots, x_{n+h}^*$, we can generate the studentized bootstrap predictive root

$$r^* = \frac{x_{n+h}^* - \hat{x}_{n+h}^*}{\hat{\sigma}^*(\sum_{j=0}^{h-1}\hat{\psi}_j^{*2})^{1/2}}, \quad (1.13)$$

where $\hat{\sigma}^*$ and $\hat{\psi}_j^{*2}$ are obtained from the pseudo series in the same way $\hat{\sigma}$ and $\hat{\psi}_j$ obtained from true series. Suppose we generate B values of r^* , and order them as $(r_1^*, \dots, r_B^*)$, then $k = \lfloor B\alpha \rfloor$. Then the $(1 - \alpha)100\%$ prediction interval is

$$[\hat{x}_{n+h} + r_k^*\hat{\sigma}(\sum_{j=0}^{h-1}\hat{\psi}_j^2)^{1/2}, \hat{x}_{n+h} + r_{B-k}^*\hat{\sigma}(\sum_{j=0}^{h-1}\hat{\psi}_j^2)^{1/2}].$$

In his paper, he also mentioned the procedure to estimate the order p by $\hat{p}$ if p is unknown. This idea is also used in Sieve bootstrap when we want to approximate an invertible ARMA model by an AR(p) model for some finite order p , cf. Alonso, Peña, Romo(2002).

One difference of this approach to the prediction intervals from our approach is that we fix the last p pseudo values to obtain the bootstrap predicted value $\hat{x}_{n+h}^*$ in Algorithm 1.2.1 step 3(c) and bootstrap future value x_{n+h}^* in Algorithm 1.2.1 step 3(d). So our bootstrap predictive root is conditioned on the last p observations. But Masarotto's method generate the bootstrap fitted value and bootstrap future value without fixing the last p pseudo values to be the true observations.

1.8.2 Grigoletto's Method

A computationally more efficient method based on Masarotto's method is proposed by Grigoletto(1998). Taking advantage of the independence of the future errors and the data containing all past and current information, Grigoletto(1998) generated B^2 standardized bootstrap prediction root out of B pseudo series. Take one-step ahead studentized bootstrap predictive root for example:

$$\begin{aligned} r^* &= \frac{x_{n+1}^* - \hat{x}_{n+1}^*}{\sigma^*} \\ &= \frac{\epsilon_{n+1}^* + \begin{pmatrix} 1 & Y_n' \end{pmatrix} (\underline{\hat{\phi}} - \underline{\hat{\phi}}^*)}{\sigma^*} \\ &= \frac{a + b}{c} \end{aligned}$$

In the real world, the term a is ϵ_{n+1} which is independent of b and c which are $\begin{pmatrix} 1 & Y_n' \end{pmatrix} (\underline{\hat{\phi}} - \underline{\hat{\phi}})$ and $\hat{\sigma}$. We repeat the bootstrap procedure to generate B pseudo standardized bootstrap root $r^{*(1)}, \dots, r^{*(B)}$, and $r^{*(i)} = \frac{a_i + b_i}{c_i}$. Then we can obtain B^2 standardized bootstrap roots as

$$r_{i,j} = \frac{a_i + b_j}{c_j}, \text{ for } i = 1, \dots, B, j = 1, \dots, B.$$

We can also apply Grigoletto's idea into our procedure and generate many

more (B^2) bootstrap roots using the fact that term a and b are independent. This approach allows us to get the predictive interval much faster.

1.8.3 A computationally faster method

Another approach to save computer time in constructing the prediction interval is proposed by Cao *et al.*(1997). Conditional on the last p observations, they only generate the future bootstrap observations instead of generating the whole pseudo series up to x_{n+h}^* . i.e. they define $x_s^* = x_s$, for $s = n-p+1, \dots, n$, and then compute the future observations by the following recursion:

$$x_t^* = \hat{\phi}_0 + \hat{\phi}_1 x_{t-1}^* + \dots + \hat{\phi}_p x_{t-p}^* + \hat{\epsilon}_t^*, \quad t = n+1, \dots, n+h \quad (1.14)$$

$\hat{\epsilon}_s^*$ are drawn from $\hat{F}_\epsilon$, the estimate of F_ϵ . Finally, they estimate the distribution of X_{n+h} by the empirical distribution of X_{n+h}^* . So the prediction interval is constructed as

$$[Q(\alpha/2), Q(1 - \alpha/2)],$$

$Q(\alpha)$ is the α -quantile of the empirical distribution of X_{n+h}^* .

The prediction interval they constructed is computed much faster. However, this prediction interval does not take the variability from $\underline{\hat{\phi}} = (\hat{\phi}_0, \dots, \hat{\phi}_p)$ into consideration and introduce more bias. To see these disadvantages, let us take $h = 1$ for example. We estimate the distribution of future value

$$X_{n+1} = \begin{pmatrix} 1 & Y_n' \end{pmatrix} \underline{\phi} + \epsilon_{n+1}$$

by the distribution of the bootstrap future value

$$X_{n+1}^* = \begin{pmatrix} 1 & Y_n' \end{pmatrix} \underline{\hat{\phi}} + \epsilon_{n+1}^*.$$

Given a data set $\{x_1, \dots, x_n\}$, $\underline{\hat{\phi}}$ is a constant vector. Using this prediction interval, there will be a discrepancy of $O_p(1/\sqrt{n})$ from estimating $\underline{\phi}$ by $\underline{\hat{\phi}}$, and the variability of $\underline{\hat{\phi}}$ is not captured. In our approach, we split X_{n+1} as $X_{n+1} - \hat{X}_{n+1} + \hat{X}_{n+1}$, and

then estimate the distribution of the predictive root $X_{n+1} - \hat{X}_{n+1}$ by the distribution of the bootstrap predictive root $X_{n+1}^* - \hat{X}_{n+1}^*$. Note that

$$\begin{aligned} X_{n+1} - \hat{X}_{n+1} &= \begin{pmatrix} 1 & Y_n' \end{pmatrix} (\underline{\phi} - \hat{\underline{\phi}}) + \epsilon_{n+1}, \\ X_{n+1}^* - \hat{X}_{n+1}^* &= \begin{pmatrix} 1 & Y_n' \end{pmatrix} (\hat{\underline{\phi}} - \hat{\underline{\phi}}^*) + \epsilon_{n+1}^*. \end{aligned}$$

And the variability from $\underline{\phi} - \hat{\underline{\phi}}$ is captured by $\hat{\underline{\phi}} - \hat{\underline{\phi}}^*$ by the asymptotic property stated in Theorem 1.3.1 part(a).

1.8.4 Sieve/PRR Bootstrap

Alonso, Peña, Romo(2002) use the autoregression sieve bootstrap in order to approximate the distribution of X_{n+h} by the distribution of X_{n+h}^* . But the way they generate the future bootstrap value is different from all other methods mentioned previously. They generate the future pseudo value through this recursion:

$$x_s^* = \hat{\phi}_0^* + \hat{\phi}_1^* x_{s-1}^* + \cdots + \hat{\phi}_p^* x_{s-p}^* + \hat{\epsilon}_s^*. \quad (1.15)$$

Same as before, we let $x_s^* = x_s$, for $s = n - p + 1, \dots, n$. The difference of the recursion (1.14) and (1.15) is that the sieve bootstrap method generates the future pseudo values using $\hat{\underline{\phi}}^*$ instead of $\hat{\underline{\phi}}$. So the sieve bootstrap prediction interval does consider the variability from the model estimation.

The bootstrap method proposed by Pascual, Romo and Ruiz(2004) is same as the sieve bootstrap in autoregression case when the order p is known. From now on, let us call their bootstrap method as PRR method. The Sieve/PRR bootstrap method is a forward method using the fitted residuals, the differences between our forward fitted(Ff) approach and the sieve bootstrap/ PRR method for AR(p) model is as follows,

- Sieve/PRR bootstrap methods construct the prediction interval by using the bootstrap distribution of X_{n+h}^* to estimate the distribution of the future value X_{n+h} while we construct the prediction interval using the distribution of bootstrap predictive root to estimate the true predictive root.

- In the recursion to get X_{n+h}^* , Sieve/PRR bootstrap methods use $\hat{\underline{\phi}}^*$ instead of $\hat{\underline{\phi}}$.

So for Sieve/PRR bootstrap, the difference of the limit conditional distribution of X_{n+h}^* and the distribution of X_{n+h} is due to two parts, one is the difference in distribution between the bootstrap future errors ϵ_t^* and true error ϵ_t , the second part is the discrepancy between $\hat{\underline{\phi}}^*$ to $\underline{\phi}$, which is of $O_p(1/\sqrt{n})$. In our approach, the difference between the conditional distribution of X_{n+h}^* and the distribution of X_{n+h} is the difference between the bootstrap predictive root and the true predictive root, which also can be split into two parts. The first part is the same as that of Sieve/PRR bootstrap, but the second part comes from the difference between the distribution of $\hat{\underline{\phi}} - \hat{\underline{\phi}}^*$ and the distribution of $\underline{\phi} - \hat{\underline{\phi}}$, which agree up to order of $O_p(1/\sqrt{n})$. So our approach is more accurate and can be called asymptotically correct, the Sieve/PRR bootstrap are asymptotically valid but only partially correct.

We take $h = 1$, then $X_{n+1}^* = \begin{pmatrix} 1 & Y_n' \end{pmatrix} \hat{\underline{\phi}}^* + \epsilon_{n+1}^*$ while $X_{n+1} = \begin{pmatrix} 1 & Y_n' \end{pmatrix} \underline{\phi} + \epsilon_{n+1}$. We use the same technique as in (1.5) and (1.6), let $Z_n = \begin{pmatrix} 1 & Y_n' \end{pmatrix} \underline{\phi}$ and $Z_n^* = \begin{pmatrix} 1 & Y_n' \end{pmatrix} \hat{\underline{\phi}}^*$, even if $\hat{F}_\epsilon = F_\epsilon$, we have

$$\begin{aligned} P(X_{n+1} \leq \alpha | Y_n = y_n) \\ = F_\epsilon(\alpha) + f_\epsilon(\alpha)E(Z_n | Y_n = y_n) + f'_\epsilon(\alpha)E(Z_n^2 | Y_n = y_n) + o(E(Z_n^2 | Y_n = y_n)) \end{aligned} \quad (1.16)$$

and

$$\begin{aligned} P^*(X_{n+1}^* \leq \alpha | Y_n^* = y_n) \\ = F_\epsilon(\alpha) + f_\epsilon(\alpha)E^*(Z_n^* | Y_n^* = y_n) + f'_\epsilon(\alpha)E^*(Z_n^{*2} | Y_n^* = y_n) + o(E^*(Z_n^{*2} | Y_n^* = y_n)) \end{aligned} \quad (1.17)$$

The difference of the third term of (1.16) and (1.17) is $O_p(1/n)$, which is bigger than $o_p(1/n)$, the different of the third term of (1.5) and (1.6). So our forward fitted method is more accurate than the Sieve/PRR method.

Mukhopadhyay and Samaranayake(2010) made two improvements on sieve bootstrap. First, they considered the variability from estimating the mean of the

time series by adjusting X_{n+h}^* into $X_{n+h}^* + \bar{X}^* - \bar{X}$. The model (1.1) I use has a constant term ϕ_0 and this variability has already been captured in ϕ_0^* . Secondly, they also noticed the deflation of underlying variability, they thus inflated ϵ by multiplying $[(n-p)/(n-2p)]^{1/2}$ and inflated ϵ^* by multiplying $[(n-p)/(n-p-1)]^{1/2}$. So this modified Sieve bootstrap approach is expected to have larger interval length and better coverage rate.

1.8.5 More Monte Carlo Studies

In the following two tables, we compare the simulation results of all the methods we discussed in the chapter for an AR(1) model with $\phi_1 = 0.5$. The following two tables contain all the results from table 1.1 and 1.3, plus the simulation results of B&J, Cao, Sieve/PRR and Masarotto methods.

Table 1.5: Simulation Results of AR(1) with normal innovations and $\phi_1 = 0.5$

normal $\phi_1 = 0.5$	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
BJ	0.934	3.832	0.402	0.880	3.216	0.338
Cao	0.917	3.720	0.532	0.871	3.199	0.417
Sieve/PRR	0.930	3.858	0.498	0.880	3.268	0.390
M	0.946	4.510	0.599	0.898	3.792	0.493
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
$n = 100$						
BJ	0.943	3.887	0.275	0.892	3.262	0.231
Cao	0.936	3.853	0.392	0.888	3.262	0.291
Sieve/PRR	0.939	3.893	0.368	0.891	3.283	0.283
M	0.948	4.514	0.430	0.898	3.793	0.348
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

Table 1.6: Simulation Results of AR(1) with Laplace innovations and $\phi_1 = 0.5$

Laplace $\phi_1 = 0.5$	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
BJ	0.923	3.812	0.603	0.885	3.199	0.506
Cao	0.921	4.065	0.863	0.873	3.197	0.605
Sieve/PRR	0.930	4.211	0.832	0.882	3.279	0.573
M	0.942	4.827	0.960	0.897	3.817	0.692
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
$n = 100$						
BJ	0.931	3.877	0.456	0.894	3.254	0.383
Cao	0.938	4.198	0.650	0.888	3.245	0.452
Sieve/PRR	0.940	4.226	0.628	0.892	3.282	0.434
M	0.946	4.802	0.668	0.897	3.789	0.479
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	0.318	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

Following table shows the simulation results based on an AR(2) model:

$$X_t = 1.55X_{t-1} - 0.6X_{t-2} + \epsilon_t$$

Same as before, we use both normal and Laplace innovations, for each type of innovation we generate 500 data sets each of size $n = 50$ or 100 . And for each data set, we generate $B = 1000$ pseudo data sets.

Table 1.7: Simulation Results of AR(2) model $X_t = 1.55X_{t-1} - 0.6X_{t-2} + \epsilon_t$ with
normal innovations

normal errors	nominal coverage 95%				nominal coverage 90%		
$n = 50$	CVR	LEN	st.err		CVR	LEN	st.err
BJ	0.926	3.771	0.402		0.868	3.165	0.337
Cao	0.905	3.640	0.560		0.859	3.150	0.424
Sieve/PRR	0.926	3.931	0.523		0.875	3.324	0.410
M	0.942	4.098	0.515		0.894	3.455	0.413
Ff	0.931	3.933	0.521		0.883	3.328	0.417
Fp	0.946	4.171	0.543		0.902	3.527	0.434
Bf	0.926	3.850	0.499		0.874	3.254	0.394
Bp	0.942	4.100	0.527		0.897	3.471	0.424
FSf	0.946	4.185	0.547		0.901	3.521	0.441
FSp	0.945	4.159	0.537		0.899	3.496	0.431
BSf	0.941	4.082	0.517		0.893	3.429	0.408
BSp	0.941	4.085	0.513		0.894	3.443	0.409
Gf	0.926	3.858	0.524		0.875	3.270	0.416
Gp	0.941	4.098	0.541		0.895	3.470	0.430
$n = 100$	CVR	LEN	st.err		CVR	LEN	st.err
BJ	0.939	3.862	0.274		0.885	3.241	0.230
Cao	0.931	3.816	0.393		0.879	3.218	0.286
Sieve/PRR	0.937	3.904	0.361		0.887	3.296	0.282
M	0.945	4.002	0.361		0.897	3.370	0.281
Ff	0.939	3.908	0.358		0.889	3.295	0.280
Fp	0.945	4.026	0.366		0.899	3.399	0.283
Bf	0.938	3.893	0.367		0.889	3.294	0.277
Bp	0.946	4.026	0.377		0.898	3.384	0.286
FSf	0.946	3.020	0.362		0.898	3.384	0.286
FSp	0.945	4.014	0.360		0.897	3.381	0.278
BSf	0.945	4.003	0.368		0.897	3.379	0.286
BSp	0.945	4.016	0.368		0.897	3.371	0.279
Ff	0.938	3.895	0.369		0.888	3.284	0.280
Fp	0.945	4.007	0.369		0.899	3.389	0.284

Table 1.8: Simulation Results of AR(2) model $X_t = 1.55X_{t-1} - 0.6X_{t-2} + \epsilon_t$ with
Laplace innovations

Laplace errors	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
BJ	0.918	3.755	0.580	0.877	3.151	0.487
Cao	0.915	3.994	0.841	0.864	3.151	0.599
Sieve/PRR	0.929	4.235	0.800	0.880	3.335	0.596
M	0.940	4.488	0.839	0.894	3.489	0.576
Ff	0.931	4.252	0.828	0.883	3.322	0.582
Fp	0.942	4.529	0.873	0.898	3.537	0.613
Bf	0.929	4.200	0.814	0.878	3.266	0.570
Bp	0.939	4.448	0.859	0.893	3.469	0.601
FSf	0.943	4.569	0.858	0.899	3.563	0.611
FSp	0.943	4.563	0.871	0.899	3.564	0.620
BSf	0.940	4.507	0.881	0.894	3.493	0.596
BSp	0.940	4.483	0.832	0.894	3.479	0.586
Gf	0.928	4.201	0.825	0.878	3.279	0.592
Gp	0.939	4.456	0.874	0.893	3.486	0.606
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
BJ	0.929	3.842	0.440	0.890	3.224	0.369
Cao	0.934	4.151	0.644	0.883	3.213	0.443
Sieve/PRR	0.938	4.219	0.615	0.887	3.256	0.419
M	0.944	4.358	0.596	0.896	3.371	0.420
Ff	0.939	4.206	0.566	0.889	3.271	0.410
Fp	0.944	4.357	0.645	0.897	3.387	0.430
Bf	0.937	4.194	0.600	0.888	3.259	0.421
Bp	0.943	4.343	0.653	0.896	3.367	0.438
FSf	0.945	4.374	0.591	0.897	3.388	0.421
FSp	0.944	4.377	0.620	0.898	3.393	0.424
BSf	0.944	4.363	0.623	0.896	3.372	0.429
BSp	0.944	4.369	0.637	0.897	3.377	0.432
Gf	0.938	4.199	0.604	0.888	3.258	0.417
Gp	0.944	4.344	0.630	0.896	3.366	0.427

From the previous four tables, we see that

- The BJ(Box and Jenkins) method has similar coverage rates as Sieve/PRR and our Forward fitted method when the error is normal. However, when the errors have Laplace distribution, the BJ method is not good, especially, when $n = 100$, BJ method is worse than all the other methods.

- Our forward and backward methods with fitted residuals(Ff and Bf) outperform Cao(Cao *et al.*), Sieve/PRR methods. This conclusion is expected and consistent with the discussion we have in the previous sections. And the methods with predictive residuals (Fp and Bp) and the studentized method (FSf,FSp,BSf,BSp) are even better.
- The M(Masarotto) method has similar performance as our FSf method as discussed in Remark 1.2.1. M method works better than our Ff and Bf methods.

The following tables compare the Masarotto method with our four forward methods. As discussed in Section 1.8.1, Masarotto method is a forward method, the different between Masarotto method with our four forward method is that Masarotto do not fix the last p values to obtain the bootstrap future values and bootstrap predicted values. Following two compares M, Ff, Fp, FSf, FSp methods for AR(1) model with $\phi_1 = 0.5$ when $X_n = 0, 1, 2, 3$ and $n = 100$. Similar as previous simulations, we have 500 dataset and each dataset is a series of size n . But this time, we generate each dataset through backward bootstrap having X_n fixed.

Table 1.9: Simulation Results of AR(1) model $X_t = 0.5X_{t-1} + \epsilon_t$ with normal innovations when $n = 100$

normal errors	nominal coverage 95%				nominal coverage 90%		
$X_n = 3$	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
$X_n = 2$	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
$X_n = 1$	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
$X_n = 0$	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 1.10: Simulation Results of AR(1) model $X_t = 0.5X_{t-1} + \epsilon_t$ with Laplace innovations when $n = 100$

Laplace errors	nominal coverage 95%			nominal coverage 90%		
$X_n = 3$	CVR	LEN	st.err	CVR	LEN	st.err
M	0.946	4.475	0.594	0.897	3.473	0.422
FSf	0.948	4.562	0.572	0.901	3.557	0.401
FSp	0.948	4.568	0.581	0.902	3.563	0.410
Ff	0.944	4.424	0.558	0.895	3.465	0.398
Fp	0.947	4.521	0.578	0.900	3.537	0.410
$X_n = 2$	CVR	LEN	st.err	CVR	LEN	st.err
M	0.944	4.366	0.602	0.897	3.392	0.412
FSf	0.946	4.407	0.592	0.899	3.426	0.404
FSp	0.946	4.403	0.597	0.900	3.434	0.407
Ff	0.940	4.265	0.579	0.893	3.338	0.397
Fp	0.944	4.359	0.593	0.898	3.410	0.407
$X_n = 1$	CVR	LEN	st.err	CVR	LEN	st.err
M	0.943	4.326	0.614	0.896	3.343	0.426
FSf	0.944	4.329	0.613	0.896	3.339	0.424
FSp	0.944	4.333	0.607	0.897	3.352	0.422
Ff	0.938	4.188	0.601	0.889	3.259	0.420
Fp	0.942	4.286	0.610	0.894	3.331	0.427
$X_n = 0$	CVR	LEN	st.err	CVR	LEN	st.err
M	0.944	4.324	0.613	0.896	3.336	0.429
FSf	0.943	4.316	0.622	0.895	3.321	0.429
FSp	0.944	4.310	0.606	0.896	3.334	0.431
Ff	0.938	4.172	0.608	0.889	3.238	0.432
Fp	0.941	4.267	0.609	0.894	2.316	0.439

The distribution of the bootstrap prediction root depends on $X_n = x$ since

$$X_{n+1}^* - \hat{X}_{n+1}^* = (\hat{\phi} - \hat{\phi}^*)x + \epsilon_{n+1}^*. \quad (1.18)$$

So the first term depends on x , however this term is too small compared to the error term. The situation is the same for studentized bootstrap root since the first term of the numerator contains a term including x .

When $x=1,2,3$ in (1.18), the distribution of $\hat{\phi} - \hat{\phi}^*$ is amplified by 1,2, or 3 for Ff,Fp,FSf,FSp. For Masarotto method, X_n is random(with mean 0) instead of a fixed value x , so $(\hat{\phi} - \hat{\phi}^*)X_n$ has a smaller spread and thus lower coverage level. When

$x=0$, the first term of the prediction root in (1.18) is exactly 0 for Ff,Fp,FSf,FSp. For Masarotto, the first term of the prediction root: $(\hat{\phi} - \hat{\phi}^*)X_n$ is a random variable which does not depend on x . So the distribution of Masarotto bootstrap root has a larger spread and higher coverage level.

For linear autoregression, we see that the effect of fixing the last p or not is negligible. However in nonlinear case, the bootstrap error term will also depend on x , then the effect of x is not negligible and the Masarotto idea will have problems for large x .

Now let's look at the performance of all the methods for a large p . The following two tables show the simulation results for an AR(20) model:

$$X_t = 0.7X_{t-10} - 0.1X_{t-20} + \epsilon_t.$$

We compare the simulation results for 500 datasets with sample size $n = 50, 100$ or 200 and $\{\epsilon_t\}$ s are standard normal errors or Laplace errors rescaled to unit variance. For each dataset, we create $B = 1000$ bootstrap pseudo series.

Table 1.11: Simulation Results of AR(20) model $X_t = 0.7X_{t-10} - 0.1X_{t-20} + \epsilon_t$ with
normal innovations

normal errors	nominal coverage 95%				nominal coverage 90%		
$n = 50$	CVR	LEN	st.err		CVR	LEN	st.err
BJ	0.645	2.458	0.515		0.564	2.063	0.432
Cao	0.649	2.538	0.595		0.555	2.045	0.453
Sieve/PRR	0.914	5.348	2.448		0.848	4.313	1.787
M	0.993	8.288	1.788		0.977	6.670	1.425
Ff	0.908	5.342	2.461		0.844	4.312	1.787
Fp	0.997	11.357	4.168		0.993	9.360	2.989
Bf	0.895	4.742	0.950		0.825	3.918	0.776
Bp	0.987	9.267	2.192		0.972	7.859	1.748
FSf	0.995	10.516	5.001		0.980	8.140	3.482
FSp	0.949	6.210	2.324		0.900	4.935	1.631
BSf	0.981	8.102	1.615		0.953	6.504	1.297
BSp	0.976	8.519	2.157		0.950	6.976	1.742
$n = 100$	CVR	LEN	st.err		CVR	LEN	st.err
BJ	0.861	3.422	0.326		0.786	2.872	0.273
Cao	0.851	3.404	0.413		0.779	2.865	0.329
Sieve/PRR	0.912	3.983	0.418		0.850	3.348	0.345
M	0.950	4.620	0.465		0.901	3.865	0.386
Ff	0.906	3.994	0.423		0.844	3.355	0.353
Fp	0.965	5.098	0.540		0.927	4.298	0.447
Bf	0.894	3.844	0.405		0.830	3.236	0.334
Bp	0.957	4.971	0.512		0.916	4.195	0.430
FSf	0.949	4.699	0.499		0.900	3.931	0.414
FSp	0.934	4.420	0.464		0.882	3.708	0.379
BSf	0.939	4.504	0.468		0.888	3.779	0.383
BSp	0.939	4.571	0.471		0.889	3.842	0.394
$n = 200$	CVR	LEN	st.err		CVR	LEN	st.err
BJ	0.917	3.720	0.207		0.855	3.122	0.173
Cao	0.911	3.691	0.284		0.853	3.116	0.231
Sieve/PRR	0.932	3.938	0.278		0.877	3.312	0.223
M	0.948	4.208	0.293		0.900	3.536	0.235
Ff	0.931	3.954	0.279		0.875	3.321	0.227
Fp	0.956	4.416	0.312		0.912	3.711	0.247
Bf	0.928	3.913	0.280		0.872	3.293	0.228
Bp	0.955	4.387	0.316		0.911	3.692	0.259
FSf	0.947	4.230	0.300		0.898	3.546	0.245
FSp	0.944	4.172	0.289		0.893	3.502	0.232
BSf	0.944	4.184	0.297		0.894	3.513	0.239
BSp	0.945	4.192	0.304		0.895	3.521	0.243

Table 1.12: Simulation Results of AR(20) model $X_t = 0.7X_{t-10} - 0.1X_{t-20} + \epsilon_t$ with
Laplace innovations

Laplace errors	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
BJ	0.672	2.397	0.684	0.595	2.012	0.574
Cao	0.684	2.622	0.952	0.581	1.987	0.580
Sieve/PRR	0.915	5.149	2.611	0.860	4.138	1.880
M	0.986	8.139	2.488	0.968	6.529	1.931
Ff	0.912	5.154	2.626	0.852	4.136	1.877
Fp	0.995	11.329	5.228	0.989	9.302	3.790
Bf	0.902	4.669	1.169	0.842	3.852	0.951
Bp	0.986	9.372	3.147	0.971	7.866	2.382
FSf	0.992	10.104	5.165	0.976	7.814	3.614
FSp	0.945	6.047	2.877	0.899	4.786	2.012
BSf	0.977	7.963	2.146	0.949	6.401	1.701
BSp	0.974	8.547	3.073	0.949	6.936	2.364
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
BJ	0.867	3.397	0.442	0.809	2.851	0.371
Cao	0.871	3.541	0.582	0.795	2.788	0.423
Sieve/PRR	0.911	4.043	0.588	0.855	3.281	0.442
M	0.942	4.710	0.669	0.899	3.814	0.509
Ff	0.909	4.026	0.579	0.852	3.272	0.455
Fp	0.957	5.225	0.775	0.921	4.225	0.573
Bf	0.900	3.916	0.546	0.837	3.165	0.419
Bp	0.950	5.102	0.730	0.909	4.116	0.532
FSf	0.944	4.779	0.715	0.899	3.865	0.547
FSp	0.934	4.559	0.692	0.886	3.676	0.511
BSf	0.934	4.608	0.659	0.884	3.708	0.486
BSp	0.936	4.730	0.696	0.886	3.795	0.508
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
BJ	0.913	3.674	0.313	0.869	3.084	0.262
Cao	0.920	3.871	0.476	0.861	3.021	0.323
Sieve/PRR	0.932	4.074	0.426	0.879	3.214	0.300
M	0.944	4.380	0.460	0.897	3.449	0.318
Ff	0.931	4.090	0.430	0.879	3.229	0.299
Fp	0.949	4.562	0.486	0.906	3.602	0.347
Bf	0.929	4.047	0.419	0.876	3.198	0.303
Bp	0.949	4.549	0.494	0.904	3.583	0.343
FSf	0.944	4.406	0.461	0.897	3.466	0.322
FSp	0.941	4.333	0.456	0.893	3.416	0.323
BSf	0.941	4.346	0.446	0.893	3.421	0.318
BSp	0.943	4.374	0.485	0.894	3.436	0.327

From the previous two tables, we see that all the methods proposed in this chapter work well for the AR(20) model when $n = 200$. The non-studentized forward and backward method with predictive residuals show a little over-coverage when $n = 100$. However, when $n = 50$, which is not large compared to $p = 20$, BJ, Cao, Sieve/PRR Ff and Bf are under-covered while M, Fp, Bp, FSf, BSf, BSp are over-covered. FSp performs best but the mean of the interval length are still too large compared to the nominal length of standard normal or Laplace distribution and the standard deviation of the interval length is large.

Chapter 1, in full, is currently being prepared for the following paper prepared for future submission for publication: Bootstrap prediction intervals for linear, non-linear and nonparametric autoregressions. Pan,Li; Politis, Dimitris. The dissertation author was the primary investigator and author of this material.

Chapter 2

Bootstrap Prediction Intervals for ARCH model

In Chapter 1, we have discussed the bootstrap prediction intervals for a simple parametric autoregression model: linear model. In this chapter we will continue study the parametric autoregression but in a nonlinear case: Autoregressive Conditional Heteroskedasticity model (ARCH). Engle(1982) first introduced the ARCH model in which the volatility of current state is a linear function of the squares of past observations. Baillie and Bollerslev(1992) constructed the Gaussian prediction error intervals using the estimated time-varying variance. Weiss(1986), Bollerslev and Wooldridge (1992) studied the properties of the quasi-maximum likelihood estimator (QMLE), which is the most commonly used method in estimating the parameters from an ARCH model. The bootstrap methods for constructing prediction intervals introduced by Reeves(2000) and Rubio(1999) are both based on QMLE.

However, the QMLE requires nonlinear optimization and does not have a closed form expression. Bose and Mukherjee(2003) proposed an linear estimator(LE) of the ARCH parameter by solving two sets of linear equations. This method does not involve nonlinear optimization and gives a closed form expression, so it is computationally easier to obtain the estimator compared to QMLE. Simulation results in Bose and Mukherjee(2003) also show that the proposed estimator performs better

than the QMLE even for small sample size such as $n = 30$ for normal, exponential or t-distributed errors.

Bose and Mukherjee(2009) further proposed weighted linear estimator(WLE) to estimate the positive parameters and the bootstrap weighted linear estimator(BWLE) whose asymptotically valid is showed via proof and simulations.

Inspired by Bose and Mukherjee(2009), we will propose a new bootstrap method to construct the predicion intervals for ARCH model based on BWLE in this chapter.

2.1 Bootstrap Algorithm and Asymptotic Properties

We consider the stationary and ergodic process $\{X_t\}$ from the following ARCH model of order p :

$$X_i = \sigma_{i-1}(\beta)\epsilon_i \quad (2.1)$$

$\beta = (\beta_0, \beta_1, \dots, \beta_p)'$ is the unknown parameter to estimated, assume $\beta_0 > 0$ and $\beta_j \geq 0$ for $j = 1, 2, \dots, p$. $\sigma_{i-1}^2(\beta) = \beta_0 + \beta_1 X_{i-1}^2 + \dots + \beta_p X_{i-p}^2$. $\{\epsilon_t\}$ are independent and identically distributed with zero mean and unit variance. ϵ_t is independent of the previous $\{X_i, i < t\}$ and $E(\epsilon_t^4) < \infty$.

2.1.1 Bootstrap Algorithm Based on BWLE with Fitted Residuals

Let $\{x_1, \dots, x_n\}$ be the observations from model (2.1), $y_i = x_i^2$,

$$z_i = (1, y_i, y_{i-1}, \dots, y_{i-p+1})',$$

$$Z = \begin{pmatrix} z_p' \\ z_{p+1}' \\ \vdots \\ z_{n-1}' \end{pmatrix} \text{ and } Y = \begin{pmatrix} y_{p+1} \\ y_{p+2} \\ \vdots \\ y_n \end{pmatrix}.$$

Following is algorithm for constructing the bootstrap prediction interval for ARCH(p) model based on BWLE with fitted residuals(BWLEf).

Algorithm 2.1.1. *Bootstrap algorithm based on BWLE with fitted residuals(BWLEf)*

- (1) Compute the preliminary weighted least squares estimator(PWLS) as $\hat{\beta}_{pr} = (Z'UZ)^{-1}Z'UY$, where U is a $(n-p) \times (n-p)$ diagonal matrix whose i th diagonal term is

$$u_i = \frac{1}{[(1+y_i) \cdots (1+y_{i+p-1})]}.$$

And then compute the wieghted linear estimator(WLE) of β as

$$\hat{\beta}_n = \left\{ \sum_{i=p}^{n-1} v_{i-p+1} [z_i z'_i / (z'_i \hat{\beta}_{pr})^2] \right\}^{-1} \left\{ \sum_{i=p}^{n-1} v_{i-p+1} [z_i y_{i+1} / (z'_i \hat{\beta}_{pr})^2] \right\},$$

where $v_i = u_i$ for $i = 1, 2, \dots, n-p$.

- (2) Compute the residuals as

$$\hat{\epsilon}_t = \frac{x_t}{\sqrt{\hat{\beta}'_n z_{t-1}}}.$$

Then center the residuals: $\hat{r}_t = \hat{\epsilon}_t - \frac{1}{n-p} \sum_{i=p+1}^n \hat{\epsilon}_i$ for $t = p+1, \dots, n$. Denote the empirical distribution of $\hat{r}_t$ as $\hat{F}_\epsilon$.

- (3) (a) Generate a $(n-p) \times (n-p)$ diagonal matrix W whose diagonal elements $(w_1, \dots, w_{n-p})$ are from a multinomial $(n-p, \frac{1}{n-p}, \frac{1}{n-p}, \dots, \frac{1}{n-p})$ distribution.

- (b) Compute the bootstrapped preliminary weighted least squares estimator (BPWLS) as $\hat{\beta}_{pr}^* = (Z'WUZ)^{-1}Z'WUY$ and the bootstrapped weighted linear estimator (BWLE) $\hat{\beta}_n^*$ as

$$\left\{ \sum_{i=p}^{n-1} w_{i-p+1} v_{i-p+1} [z_i z'_i / (z'_i \hat{\beta}_{pr}^*)^2] \right\}^{-1} \left\{ \sum_{i=p}^{n-1} w_{i-p+1} v_{i-p+1} [z_i y_{i+1} / (z'_i \hat{\beta}_{pr}^*)^2] \right\}$$

- (c) Compute $H_n^2 = \text{var}(w_i) = 1 - \frac{1}{n-p}$ and

$$x_{n+1}^* = \sigma_n(\hat{\beta}_n + \frac{\hat{\beta}_n - \hat{\beta}_n^*}{H_n}) \epsilon_{n+1}^*,$$

where $\sigma_n(\beta) = \sqrt{\beta_0 + \beta_1 X_n^2 + \cdots + \beta_p X_{n-p+1}^2}$, and ϵ_{n+1}^* is generated from $\hat{F}_\epsilon$ in step(2).

- (4) Repeat step(3) B times and collect $x_{n+1,1}^*, \dots, x_{n+1,B}^*$ in the form of an empirical distribution whose α -quantile is denoted as $q(\alpha)$. Then the $(1 - \alpha)100\%$ equal-tailed predictive interval for X_{n+1} is given by

$$[q(\alpha), q(1 - \alpha/2)] \quad (2.2)$$

Remark 2.1.1. There are other choices for preliminary weight U , bootstrapped preliminary weight W and the bootstrap weight V as long as they satisfy assumption(W1) and equations (11),(12),(17)-(19),(21) from Bose and Mukherjee(2009). See Chatterjee and Bose(2005) for more examples of the bootstrapped preliminary weight W .

2.1.2 Bootstrap Algorithm Based on BWLE with Predictive Residuals

An advantage of using linear estimator instead of QMLE is that we can obtain the predictive residuals through deleting one pair of data. To get the predictive residual ϵ_t , $t = p + 1, \dots, n$, we can exclude the pair (z_{t-1}, y_t) from the scatter plot of y_k vs. z_{k-1} , $k = p + 1, \dots, n$.

Algorithm 2.1.2. Bootstrap algorithm based on BWLE with predictive residuals (BWLEp)

To get the bootstrap algorithm based on BWLE with predictive residuals we only need to substitute $\{\hat{\epsilon}_t^{(t)}, t = p + 1, \dots, n\}$ for $\{\hat{\epsilon}_t, t = p + 1, \dots, n\}$ in step(2) of Algorithm 2.1.1.

Following steps describe how to get the predictive residuals $\{\epsilon_t, t = p+1, \dots, n\}$ in detail.

- 1 Let $Z^{(t)}$ be the matrix Z excluding the row z'_{t-1} , $Y^{(t)}$ be the vector Y excluding y_t , $U^{(t)}$ be the diagonal matrix excluding the diagonal element u_{t-p} .

2 Compute $\hat{\beta}_{pr}^{(t)} = (Z'^{(t)}U^{(t)}Z^{(t)})^{-1}Z'^{(t)}U^{(t)}Y^{(t)}$, and

$$\hat{\beta}_n^{(t)} = \left\{ \sum_{i=p, i \neq t-1}^{n-1} u_{i-p+1} [z_i z'_i / (z'_i \hat{\beta}_{pr}^{(t)})^2] \right\}^{-1} \left\{ \sum_{i=p}^{n-1} u_{i-p+1} [z_i y_{i+1} / (z'_i \hat{\beta}_{pr}^{(t)})^2] \right\}.$$

3 compute the predictive residual $\hat{\epsilon}_t^{(t)} = \frac{x_t}{\sqrt{\hat{\beta}_n^{(t)} z_{t-1}}}$.

2.1.3 Bootstrap Algorithm Based on QMLE

The bootstrap method for bootstrap prediction intervals for ARCH models based on QMLE are discussed by Reeves(2000) and Rubio(1999). You can also find the algorithms in their papers. To compare the two methods introduced above, we will give the more detailed algorithm based on QMLE as follows.

Algorithm 2.1.3. *Bootstrap algorithm based on QMLE Let $\{x_1, \dots, x_n\}$ be the observations from model (2.1).*

- 1 Fit an ARCH(p) model to the data. Let $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p)'$ be the quasi-maximum likelihood estimator.
- 2 Calculate the residuals: $\hat{\epsilon}_t = x_t / \sigma_{t-1}(\hat{\beta})$, for $t = p+1, \dots, n$. And then center the residuals: $r_t = \hat{\epsilon}_t - \bar{\epsilon}$ for $t = p+1, \dots, n$, where $\bar{\epsilon} = (n-p)^{-1} \sum_{t=p+1}^n \hat{\epsilon}_t$.
- 3 (a) Use $\hat{\beta}$ and residuals $\{\hat{\epsilon}_t\}$, along with initial conditions $u_1^* = x_1, \dots, u_p^* = x_p$ to generate $\{u_t^*, t \geq p+1\}$ by recursion:

$$u_t^* = \sqrt{\hat{\beta}_0 + \hat{\beta}_1 u_{t-1}^{*2} + \dots + \hat{\beta}_p u_{t-p}^{*2}} \epsilon_t^*,$$

where ϵ_t^* is a random draw from the pool of centered residuals $\{r_t, t = p+1, \dots, n\}$. To ensure stationarity of the pseudo series, we generate $n+m$ pseudo data for some large positive m , i.e. $\{u_1^*, \dots, u_n^*, u_{n+1}^*, \dots, u_{n+m}^*\}$, and discard the first m data.

- (b) Fit an ARCH(p) model to pseudo data $\{x_t^* = u_{t+m}^*, t = 1, 2, \dots, n\}$ and re-estimate the QMLE $\hat{\beta}^* = (\hat{\beta}_0^*, \hat{\beta}_1^*, \dots, \hat{\beta}_p^*)'$.

(c) Fix the last p pseudo data as the true data: $x_{n-p+1}^* = x_{n-p+1}$, $x_{n-p+2}^* = x_{n-p+2}, \dots, x_n^* = x_n$ and generate the future bootstrap value $\{x_{n+t}^*, t \geq 1\}$ by the following recursion,

$$x_{n+t}^* = \sqrt{\hat{\beta}_0^* + \hat{\beta}_1^* x_{n+t-1}^{*2} + \dots + \hat{\beta}_p^* x_{n+t-p}^{*2}} \epsilon_{n+t}^*,$$

where ϵ_{n+t}^* is a random draw from the centered residuals.

4 Repeat steps 3(a)-(c) B times and collect B bootstrap h -step ahead future values in the form of empirical distribution whose α -quantile is denoted $q(\alpha)$. Construct the $(1 - \alpha)100\%$ equal-tailed predictive intervals for X_{n+h} as

$$[q(\alpha/2), q(1 - \alpha/2)].$$

Remark 2.1.2. Our bootstrap Algorithm BWLEf and BWLEp are computationally easier compared to the bootstrap method based on QMLE because of the following two reasons. First, BWLEf and BLWEp have closed form expressions of the solutions of the two involved linear equations while QMLE requires nonlinear approximation and does not have a close form. Secondly, there's no need to create pseudo series through recursion in Algorithms for BWLEf and BWLEp.

2.1.4 Asymptotic Properties

We assume that the process $\{X_t, t \geq 1\}$ is stationary and ergodic and satisfies (2.1) and $E(\epsilon_t^4) < \infty$. We further assume that the assumptions for all the weight V , U and W mentioned in Remark 2.1.1 are satisfied and the assumptions of Lemma 2 from Bose and Mukherjee(2009) hold, the following theorem is proved by Bose and Mukherjee(2009).

Theorem 2.1.1 (Bose and Mukherjee(2009) Theorem 2).

$$\sup\{|F_n^*(x) - F_n(x)|; x \in \mathbb{R}^{p+1}\} = o_p(1) \quad (2.3)$$

where $F_n(x) = P[\sqrt{n}(\hat{\beta}_n - \beta) \leq x]$ and $F_n^*(x) = P^*[H_n^{-1}\sqrt{n}(\hat{\beta}_n^* - \hat{\beta}_n) \leq x]$.

In Algorithm 2.1.1 step (3)(c), we use the conclusion from the previous theorem and approximate the distribution of $\beta - \hat{\beta}_n$ by the distribution of $\frac{\hat{\beta}_n - \hat{\beta}_n^*}{H_n}$.

2.2 Monte Carlo Studies

In this section, we use Monte Carlo simulations to assess the performance of our two methods based on BWLE and the traditional bootstrap method based on QMLE as suggested by Reeve(2000) and Rubio(1999). We create 500 data sets for each of the following scenarios: sample size $n = 50, 100$ or 200 ; innovations are from standard normal or Laplace (rescaled to unit variance) distribution; series are generated from Model 1 or Model 2 listed below.

- Model 1: ARCH(1) $X_t = \sqrt{0.5 + 0.25X_{t-1}^2}\epsilon_t$
- Model 2: ARCH(2) $X_t = \sqrt{0.1 + 0.2X_{t-2}^2}\epsilon_t$

We compare the coverage levels(CVR) and length(Len) of all the intervals constructed by the three methods metioned above with nominal coverage levels of 95% and 90%. Following four tables show the simulation results.

Table 2.1: Simulation Results of ARCH(1) model $X_t = \sqrt{0.5 + 0.25X_{t-1}^2}\epsilon_t$ with normal innovations

normal errors	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
BWLEf	0.927	3.040	0.762	0.877	2.610	0.608
BWLEp	0.941	3.269	0.883	0.890	2.675	0.654
QMLE	0.924	2.964	0.657	0.873	3.507	0.523
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
BWLEf	0.940	3.098	0.607	0.890	2.607	0.491
BWLEp	0.946	3.203	0.631	0.896	2.655	0.496
QMLE	0.937	3.055	0.562	0.888	2.582	0.448
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
BWLEf	0.945	3.190	0.720	0.896	2.680	0.587
BWLEp	0.948	3.230	0.716	0.898	2.700	0.590
QMLE	0.943	3.152	0.650	0.894	2.661	0.545

Table 2.2: Simulation Results of ARCH(1) model $X_t = \sqrt{0.5 + 0.25X_{t-1}^2}\epsilon_t$ with
Laplace innovations

Laplace errors	nominal coverage 95%				nominal coverage 90%		
$n = 50$	CVR	LEN	st.err		CVR	LEN	st.err
BWLEf	0.931	3.428	1.457		0.883	2.645	1.080
BWLEp	0.943	3.811	1.735		0.892	2.773	1.128
QMLE	0.928	3.279	1.211		0.877	2.542	0.892
$n = 100$	CVR	LEN	st.err		CVR	LEN	st.err
BWLEf	0.941	3.466	1.503		0.894	2.662	1.086
BWLEp	0.947	3.615	1.500		0.897	2.717	1.101
QMLE	0.937	3.334	1.228		0.887	2.575	0.919
$n = 200$	CVR	LEN	st.err		CVR	LEN	st.err
BWLEf	0.941	3.320	0.882		0.892	2.559	0.644
BWLEp	0.944	3.378	0.865		0.893	2.577	0.631
QMLE	0.940	3.272	0.778		0.889	2.531	0.569

Table 2.3: Simulation Results of ARCH(2) model $X_t = \sqrt{0.1 + 0.2X_{t-2}^2}\epsilon_t$ with
normal innovations

normal errors	nominal coverage 95%				nominal coverage 90%		
$n = 50$	CVR	LEN	st.err		CVR	LEN	st.err
BWLEf	0.926	1.339	0.337		0.875	1.124	0.275
BWLEp	0.943	1.477	0.407		0.894	1.193	0.295
QMLE	0.926	1.304	0.225		0.875	1.099	0.179
$n = 100$	CVR	LEN	st.err		CVR	LEN	st.err
BWLEf	0.936	1.351	0.264		0.886	1.135	0.212
BWLEp	0.946	1.414	0.277		0.895	1.169	0.220
QMLE	0.936	1.332	0.189		0.887	1.125	0.151
$n = 200$	CVR	LEN	st.err		CVR	LEN	st.err
BWLEf	0.942	1.374	0.237		0.892	1.154	0.198
BWLEp	0.948	1.407	0.252		0.898	1.173	0.202
QMLE	0.943	1.366	0.183		0.892	1.148	0.147

Table 2.4: Simulation Results of ARCH(2) model $X_t = \sqrt{0.1 + 0.2X_{t-2}^2}\epsilon_t$ with
Laplace innovations

Laplace errors	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
BWLEf	0.930	1.599	1.485	0.883	1.242	1.105
BWLEp	0.944	1.852	2.051	0.896	1.324	1.130
QMLE	0.928	1.537	2.224	0.879	1.134	0.582
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
BWLEf	0.941	1.454	0.357	0.891	1.116	0.257
BWLEp	0.948	1.549	0.405	0.898	1.156	0.271
QMLE	0.939	1.434	0.419	0.890	1.103	0.245
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
BWLEf	0.939	1.486	0.509	0.890	1.145	0.365
BWLEp	0.943	1.523	0.500	0.893	1.165	0.380
QMLE	0.941	1.455	0.291	0.890	1.122	0.212

The previous four tables show that BWLEf and QMLE have similar coverage level, but the QMLE has smaller variance of interval length. Our BWLEp method outperforms other with respect to the coverage level but the variability of the interval length is increased as a price to pay for using predictive intervals. When $n = 50$, we see that the standard error related to the interval length is fairly large, this is due the heavy tail of the distribution of arch errors.

Chapter 2, in full, is currently being prepared for the following paper prepared for future submission for publication: Bootstrap prediction intervals for linear, non-linear and nonparametric autoregressions. Pan,Li; Politis, Dimitris. The dissertation author was the primary investigator and author of this material.

Chapter 3

Bootstrap Prediction Intervals for Nonparametric Autoregression

In this chapter, we construct the prediction interval in a general nonparametric autoregression model fitted via kernel smoothing. We introduce the forward bootstrap methods with fitted and predictive residuals as those in chapter 1, and show that they properly estimate the distribution of the future value capturing both the variability of the kernel estimator and the variability of the innovations from the autoregression model. Monte Carlo simulations are given in Section 3.4 to show that our method achieves near nominal coverage.

A nonparametric approach for fitting nonparametric autoregression is studied in Franke, Kreiss and Neumann(2002Franke et al. (2002b)Franke et al. (2002a)) who used the kernel smoothing estimator and constructed asymptotically valid confidence bands by bootstrap method. We complement the Franke *et al.* results by constructing prediction intervals as opposed to confidence intervals. In addition, we propose the alternative of resampling the predictive residuals.

In Section 3.1, we give the bootstrap procedure for a nonlinear nonparametric autoregression with independent and identically distributed errors. In Section 3.2, we extend the model to heteroscedastic errors. These two models are given below.

- **with same variance**

$$X_t = m(X_{t-1}, \dots, X_{t-p}) + \epsilon_t, \quad (3.1)$$

- **with heteroscedastic variance**

$$X_t = m(X_{t-1}, \dots, X_{t-p}) + \sigma(X_{t-1}, \dots, X_{t-p})\epsilon_t. \quad (3.2)$$

In the above, $m(\cdot)$ and $\sigma(\cdot)$ are functions that are unknown but assumed smooth and $\{\epsilon_t\}$ s are independent, identically distributed random variables with mean 0 and variance σ^2 . In (3.2), we further assume $\sigma^2 = 1$.

Simulations and results are given in Section 3.4.

3.1 Nonparametric Autoregression with i.i.d Innovations

In this section, we consider a stationary and geometrically ergodic process of the form (3.1). We will first give the algorithms using fitted residuals and predictive residuals and then discuss their asymptotic validity.

3.1.1 Bootstrap Algorithm with Fitted Residuals

Given a sample $\{x_1, x_2, \dots, x_n\}$, let $y_t = (x_t, x_{t-1}, \dots, x_{t-p+1})'$. The resampling algorithm for the predictive distribution of future value X_{n+1} is as follows:

Algorithm 3.1.1.

- (1) Construct the Nadaraya-Watson Kernel estimator $\{\hat{m}(y)\}$, for $i = 1, \dots, n-2$, where $y \in \mathbb{R}^p$ and

$$\hat{m}(y) = \frac{\sum_{t=p}^{n-1} K\left(\frac{\|y - y_t\|}{h}\right) x_{t+1}}{\sum_{t=p}^{n-1} K\left(\frac{\|y - y_t\|}{h}\right)},$$

where $\|\cdot\|$ is a norm in $\mathbb{R}^p$. $K(\cdot)$ is compactly supported, symmetric density function on $\mathbb{R}$ with bounded derivative and satisfying $\int K(v)dv = 1$. The

bandwidth satisfies $h \rightarrow 0$ and $nh^5 \rightarrow c > 0$. In practice, we choose h through cross validation.

(2) Compute the fitted residuals: $\hat{\epsilon}_i = x_i - \hat{m}(y_{i-1})$, for $i = p+1, \dots, n$

(3) Center the residuals: $\hat{r}_i = \hat{\epsilon}_i - (n-p)^{-1} \sum_{t=p+1}^n \hat{\epsilon}_t$, for $i = p+1, \dots, n$.

(a) Sample randomly (with replacement) from the values $\hat{r}_{p+1}, \dots, \hat{r}_n$ to create bootstrap pseudo errors ϵ_i^* , $i = -M+p, \dots, n+1$ for some large positive M .

(b) Set $(x_{-M}^*, x_{-M+1}^*, \dots, x_{-M+p-1}^*)$ equal to p consecutive values drawn from $\{x_1, \dots, x_n\}$. Then generate x_i^* , by the recursion: $x_i^* = \hat{m}(y_{i-1}^*) + \epsilon_i^*$ for $i = -M+p, \dots, n$.

(c) Drop the first M ‘burn in’ observations to make sure that the starting values have an insignificant effect. Then construct the kernel estimator $\hat{m}^*$ using bootstrap series $\{x_1^*, \dots, x_n^*\}$.

$$\hat{m}^*(y) = \frac{\sum_{i=p}^{n-1} K\left(\frac{\|y - y_i^*\|}{h}\right) x_{i+1}^*}{\sum_{i=p}^{n-1} K\left(\frac{\|y - y_i^*\|}{h}\right)},$$

where $y_t^* = (x_t^*, x_{t-1}^*, \dots, x_{t-p+1}^*)'$.

(d) Now we fix the last p pseudo values to be the true observations, i.e., $y_n^* = y_n$, and then calculate the bootstrap predictor

$$\hat{X}_{n+1}^* = \hat{m}^*(y_n^*) = \hat{m}^*(y_n)$$

and the future bootstrap observation x_{n+1}^* as

$$X_{n+1}^* = \hat{m}(y_n^*) + \epsilon_{n+1}^* = \hat{m}(y_n) + \epsilon_{n+1}^*.$$

(e) Calculate the bootstrap predictive root replicate as $X_{n+1}^* - \hat{X}_{n+1}^*$.

(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 X_{n+1} is given by

$$[\hat{m}(y_n) + q(\alpha/2), \hat{m}(y_n) + q(1 - \alpha/2)] \quad (3.3)$$

3.1.2 Bootstrap Algorithm with Predictive Residuals

Algorithm 3.1.2.

(1) same as step(1) in Algorithm 3.1.1.

(2) Use the delete- x_t dataset as described in Section 1.2.2 to compute the delete- x_t kernel smoothing estimator

$$\hat{m}^{(t)}(y) = \frac{\sum_{i=p+1, i \neq t}^n K\left(\frac{\|y - y_{i-1}\|}{h}\right) x_i}{\sum_{i=p+1, i \neq t}^n K\left(\frac{\|y - y_{i-1}\|}{h}\right)} \text{ for } t = p+1, \dots, n, \text{ and } y \in \mathbb{R}^p$$

Then calculate the predictive residuals: $\hat{\epsilon}_t^{(t)} = x_t - \hat{m}_t^{(t)}(y_{t-1})$, $t = p+1, \dots, n$.

(3)-(5) Replace $\hat{\epsilon}_t$ into $\hat{\epsilon}_t^{(t)}$, the rest is the same as in Algorithm 3.1.1.

3.1.3 Asymptotic Properties

In this subsection, we focus on a stationary and geometrically ergodic process of order $p = 1$; the general case of order p is similar. Then the nonparametric autoregression model which we study is tailored as the following form,

$$X_t = m(X_{t-1}) + \epsilon_t \quad (3.4)$$

where the innovations $\{\epsilon_t\}$ are independent, identically distributed random variables with mean 0 and variance σ^2 .

We assume that the ϵ_t have a density of f_ϵ ; then the following assumption(A1) implies that $\{X_t\}$ is geometrically ergodic(Franke,Kreiss and Neumann(2002)Franke et al. (2002b)), while assumption(A2) ensures the independence of X_t and ϵ_m for $m \geq t$.

(A1) $\{X_t\}$ obeys (3.4), where

1. $|m(x)| \leq C_1 + C_2|x|$ for all x and some $C_1 < \infty$, $C_2 < 1$,
2. $E|\epsilon_t| < \infty$,
3. $f_\epsilon(x) \geq C_3 > 0$
for all $x \in [-C_4 - \sup_{x \in K}\{m(x) - x\}, C_4 - \inf_{x \in K}\{m(x) - x\}]$ and some $C_4 > 0$, where $K = [-C_5, C_5]$, $C_5 > (C_1 + \mathbb{E}|\epsilon_t|)/(1 - C_2)$

(A2) ϵ_t is independent of $\{X_i, \text{ for } i < t\}$

We suppose that the data generating process obeys (3.4) , and take Nadaraya-Watson Kernel estimator to estimate $m(x)$.

$$\hat{m}(x) = \frac{\sum_{t=1}^{n-1} K\left(\frac{x-X_t}{h}\right)X_{t+1}}{\sum_{t=1}^{n-1} K\left(\frac{x-X_t}{h}\right)} \quad (3.5)$$

let $K(\cdot)$ be an appropriately chosen compactly supported, symmetric and nonnegative function on $\mathbb{R}$ with bounded derivative and satisfying $\int K(v)dv = 1$. The bandwidth satisfies $h \rightarrow 0$ and $nh^5 \rightarrow c > 0$.

Franke, Kreiss and Neumann(2002)Franke et al. (2002a) have proved the consistency of $\hat{m}(x)$ as follows. All the following theorems and lemmas in the section are under assumptions (AB1)-(AB10)(Franke, Kreiss and Neumann 2002Franke et al. (2002a)).

Theorem 3.1.1. [Franke, Kreiss and Neumann(2002)Franke et al. (2002a)]

$d_0(\mathcal{L}^*(\sqrt{nh}\{\hat{m}(x_n) - \hat{m}^*(x_n)\}), (\mathcal{L}(\sqrt{nh}\{m(x_n) - \hat{m}(x_n)\}))) \rightarrow 0$ in probability, where d_0 denotes the Kolmogorov distance, $\mathcal{L}^*$ is the distribution given $X_1 = x_1, \dots, X_n = x_n$.

Lemma 3.1.2. $\hat{\epsilon}_t - \hat{\epsilon}_t^{(t)} = O_p(\frac{1}{n})$, when $n \rightarrow \infty$.

Proof.

$$\begin{aligned}\hat{m}(X_{t-1}) &= \frac{\sum_{i=1}^{n-1} K(\frac{X_{t-1}-X_i}{h})X_{i+1}}{\sum_{i=1}^{n-1} K(\frac{X_{t-1}-X_i}{h})} \\ \hat{m}^{(t)}(X_{t-1}) &= \frac{\sum_{i=1}^{n-1} K(\frac{X_{t-1}-X_i}{h})X_{i+1} - K(\frac{X_{t-1}-X_{t-1}}{h})X_t}{\sum_{i=1}^{n-1} K(\frac{X_{t-1}-X_i}{h}) - K(\frac{X_{t-1}-X_{t-1}}{h})} \\ &= \frac{\sum_{i=1}^{n-1} K(\frac{X_{t-1}-X_i}{h})X_{i+1} - K(0)X_t}{\sum_{i=1}^{n-1} K(\frac{X_{t-1}-X_i}{h}) - K(0)}\end{aligned}$$

let $a_n = \sum_{i=1}^{n-1} K(\frac{X_{t-1}-X_i}{h})X_{i+1}$, $\delta_a = K(0)X_t$; $b_n = \sum_{i=1}^{n-1} K(\frac{X_{t-1}-X_i}{h})$, $\delta_b = K(0)$.
then

$$\begin{aligned}\hat{m}^{(t)}(X_{t-1}) &= \frac{a_n - \delta_a}{b_n - \delta_b} \\ &= \frac{a_n}{b_n} \left(\frac{1 - \delta_a/a_n}{1 - \delta_b/b_n} \right) \\ &= \frac{a_n}{b_n} (1 - \delta_a/a_n) (1 + \delta_b/b_n + (\delta_b/b_n)^2 + o(\delta_b/b_n)^2) \\ &= \frac{a_n}{b_n} + O_p(1/n) \\ &= \hat{m}(X_{t-1}) + O_p(1/n)\end{aligned}$$

It follows that $\hat{\epsilon}_t - \hat{\epsilon}_t^{(t)} = X_t - \hat{m}(X_{t-1}) - (X_t - \hat{m}^{(t)}(X_{t-1})) = O_p(1/n)$ □

We know that the predictive root of the one-step ahead future value is

$$X_{n+1} - \hat{X}_{n+1} = m(X_n) - \hat{m}(X_n) + \epsilon_{n+1}, \quad (3.6)$$

and the bootstrap predictive root conditioned on $x_1, \dots, x_n$ is

$$X_{n+1}^* - \hat{X}_{n+1}^* = \hat{m}(X_n) - \hat{m}^*(X_n) + \epsilon_{n+1}^*. \quad (3.7)$$

Theorem 3.1.1 showed that given $X_n = x_n$, $m(X_n) - \hat{m}(X_n)$ and $\hat{m}(X_n) - \hat{m}^*(X_n)$ are negligible compared to ϵ_{n+1} and ϵ_{n+1}^* . ϵ_{n+1} is sampled from the ditribution of F_ϵ , $\hat{\epsilon}_{n+1}$ is sampled from the the empirical distribution of the fitted residuals $\hat{F}_n$. Let F_n be the empirical distribution of errors, suppose the density of the error f_ϵ is Hölder continuous with exponent ξ greater than $1/3$. Furthermore, strengthen the assumption of the error ϵ in assumption (AB4) to having a finite moment of

order great than 8/3. Müller, Schick and Wefelmeyer(2009, Theorem 1) showed that $\sup_{\alpha} |\hat{F}_n(\alpha) - F_n(\alpha)| = O_p(\frac{1}{n-1} \sum_{j=2}^n \epsilon_j) = O_p(1/\sqrt{n})$. Same as the arguments in Section 1.3, apply the property of empirical distribution, we get $\sup_{\alpha} |\hat{F}_n(\alpha) - F_{\epsilon}(\alpha)| = O_p(1/\sqrt{n})$. Immediately we have

$$\sup_a |P(X_{n+1} - \hat{X}_{n+1} \leq a | X_n = x_n) - P^*(X_{n+1}^* - \hat{X}_{n+1}^* \leq a | X_n = x_n)| = O_p(1/\sqrt{n}).$$

The prediction interval (3.3) is asymptotic valid.

Let $Z_n = \hat{m}(X_n) - m(X_n)$, $Z_n^* = \hat{m}^*(X_n) - m^*(X_n)$. We smooth $\hat{F}_n$ to get $\hat{F}_{\epsilon}$. Assume F_{ϵ} and $\hat{F}_{\epsilon}$ are twice differentiable and the derivatives are f_{ϵ} and $\hat{f}_{\epsilon}$. Then

$$\begin{aligned} & P(X_{n+1} - \hat{X}_{n+1} \leq \alpha | X_n = x_n) \\ &= F_{\epsilon}(\alpha) + f_{\epsilon}(\alpha) E(Z_n | X_n = x_n) + \frac{f'_{\epsilon}(\alpha)}{2} E(Z_n^2 | X_n = x_n) + o(E[Z_n^2 | X_n = x_n]) \end{aligned} \quad (3.8)$$

and

$$\begin{aligned} & P^*(X_{n+1}^* - \hat{X}_{n+1}^* \leq \alpha | X_n^* = x_n) \\ &= \hat{F}_{\epsilon}(\alpha) + \hat{f}_{\epsilon}(\alpha) E^*(Z_n^* | X_n^* = x_n) + \frac{\hat{f}'_{\epsilon}(\alpha)}{2} E^*(Z_n^{*2} | X_n^* = x_n) + o(E^*[Z_n^{*2} | X_n^* = x_n]) \end{aligned} \quad (3.9)$$

Here P^* and E^* are the conditional probability and expectation given all the observations.

By Theorem 3.1.1, we know that $\sqrt{nh}\{\hat{m}(x_n) - m(x_n)\}$ and $\sqrt{nh}\{\hat{m}^*(x_n) - m^*(x_n)\}$ have same limit distribution with variance and mean given in Frank, Kreiss and Mammen(2002)Franke et al. (2002a). Unlike the zero bias of $\sqrt{n}\hat{\phi}$ and $\sqrt{n}\underline{\phi}$ or the zero bias of $\sqrt{n}\hat{\phi}^*$ and $\sqrt{n}\underline{\phi}$ in linear autoregression case, the limit bias in nonparametric autoregression is nonzero, see Frank, Kreiss and Mammen(2002)Franke et al. (2002a), eq.(41)-(44)).

So if we estimate (3.8) by $\hat{F}_{\epsilon}(\alpha)$ only, the bias of $E(Z_n | X_n = x_n)$ in the second term from (3.8) is of order $O_p(1/\sqrt{nh})$ and is not captured. In addition, $E(Z_n^2 | X_n = x_n)$ from the third term is of order $O_p(1/nh)$, which is not captured by $\hat{F}_{\epsilon}(\alpha)$ too. even if we know the distribution of error F_{ϵ} , i.e., $\hat{F}_{\epsilon} = F_{\epsilon}$, the discrepancy from estimating (3.8) by $\hat{F}_{\epsilon}(\alpha)$ is $O_p(1/\sqrt{nh})$.

If we use (3.9) to approximate (3.8), we have

$$\sqrt{nh}(E(Z_n|X_n = x_n) - E^*(Z_n^*|X_n^* = x_n)) = o(1),$$

$$nh(E(Z_n^2|X_n = x_n) - E^*(Z_n^{*2}|X_n^* = x_n)) = o(1).$$

So

$$\begin{aligned} & |P(X_{n+1} - \hat{X}_{n+1} \leq \alpha | X_n = x_n) - P^*(X_{n+1}^* - \hat{X}_{n+1}^* \leq \alpha | X_n^* = x_n)| \\ &= |F_\epsilon(\alpha) - \hat{F}_\epsilon(\alpha)| + o_p(1/\sqrt{nh}) \end{aligned}$$

If we assume $\hat{F}_\epsilon = F_\epsilon$, the discrepancy from estimating (3.8) by (3.9) is of $o_p(1/\sqrt{nh})$.

We conclude that the variability of (3.8) is fully captured by (3.9) and the prediction interval (3.3) is asymptotically valid and correct.

Remark 3.1.1. *The prediction root (3.6) contains two parts. The first part $-Z_n$ is unknown but can be estimated by the distribution of $-Z_n^*$ collected in step (3)(e) in Algorithm 3.1.1 and 3.1.2. The distribution of the second part ϵ_{n+1} can be approximate by ϵ_{n+1}^* , which is resampled from the fitted or predictive residuals. We approximate the distribution of X_{n+1} through shifting the bootstrap root (3.7) by $\hat{m}(x_n)$,*

3.2 Nonparametric Autoregression Bootstrap with Heteroscedastic Innovations

Now we consider the nonparametric autoregression model (3.2). Similar as Section 3.1, we use Nadaraya-Watson estimators to estimate the unknown function m and σ based on observations $\{x_1, \dots, x_n\}$ as follows,

$$\hat{m}(y) = \frac{\sum_{t=1}^{n-1} K(\frac{\|y - y_t\|}{h}) x_{t+1}}{\sum_{t=1}^{n-1} K(\frac{\|y - y_t\|}{h})}, \quad (3.10)$$

$$\hat{\sigma}^2(y) = \frac{\sum_{t=1}^{n-1} K(\frac{\|y - y_t\|}{h}) (x_{t+1} - \hat{m}(y_t))^2}{\sum_{t=1}^{n-1} K(\frac{\|y - y_t\|}{h})}, \quad (3.11)$$

where $y \in \mathbb{R}^p$, $y_t = (x_t, \dots, x_{t-p+1})'$.

Remark 3.2.1. Sometimes, people use an alternative estimator for σ as follows

$$\tilde{\sigma}^2(y) = \frac{\sum_{t=1}^{n-1} K\left(\frac{\|y-y_t\|}{h}\right) x_{t+1}^2}{\sum_{t=1}^{n-1} K\left(\frac{\|y-y_t\|}{h}\right)} - (\hat{m}(y))^2, \quad (3.12)$$

In practice, we prefer $\hat{\sigma}$ as in (3.11) because $\hat{\sigma}$ is guaranteed to be nonnegative while $\tilde{\sigma}$ in (3.12) can be negative.

We generate the bootstrap pseudo series based on the following process:

$$x_t^* = \hat{m}_g(y_{t-1}^*) + \hat{\sigma}_g(y_{t-1}^*)\epsilon_t^*,$$

where $\hat{m}_g$ and $\hat{\sigma}_g$ are computed in the same way as $\hat{m}$ and $\hat{\sigma}$ respectively but using an over-smoothed bandwidth $g \gg h$.

The asymptotic properties of these estimates will be given after the algorithms.

3.2.1 Bootstrap Algorithm with Fitted Residuals

Given a stationary sample $\{x_1, x_2, \dots, x_n\}$, let $y_t = (x_t, \dots, x_{t-p+1})'$. The resampling algorithm for the predictive distribution of future value X_{n+1} is as follows:

Algorithm 3.2.1.

(1) Construct the estimates $\hat{m}(\cdot)$ and $\hat{\sigma}^2(\cdot)$ by formula (3.10) and (3.11).

(2) Compute the residuals:

$$\hat{\epsilon}_i = \frac{x_i - \hat{m}(y_{i-1})}{\hat{\sigma}(y_{i-1})} \quad (3.13)$$

for $i = p+1, \dots, n$

(3) Center the residuals: $\hat{r}_i = \hat{\epsilon}_i - (n-p)^{-1} \sum_{t=p+1}^n \hat{\epsilon}_t$, for $i = p+1, \dots, n$.

(a) Sample randomly (with replacement) from the values $r_{p+1}, \dots, r_n$ to create bootstrap pseudo errors ϵ_i^* , $i = -M+p, \dots, 1, 2, \dots, n+1$. M is some large positive integer.

- (b) First set $(x_{-M}^*, x_{-M+1}^*, \dots, x_{-M+p-1}^*)$ equal to p consecutive values drawn from $\{x_1, \dots, x_n\}$. And then generate x_i^* by the recursion:

$$x_i^* = \hat{m}_g(y_{i-1}^*) + \hat{\sigma}_g(y_{i-1}^*)\epsilon_i^*.$$

for $i = -M + p, \dots, n$.

- (c) Drop the first M 'burn in' observations to make sure that the starting values have an insignificant effect. And then construct the kernel estimator $\hat{m}^*$ by bootstrap series $\{x_1^*, \dots, x_n^*\}$. Let $y_i^* = (x_i^*, x_{i-1}^*, \dots, x_{i-p+1}^*)'$.

$$\hat{m}^*(y) = \frac{\sum_{i=1}^{n-1} K\left(\frac{\|y - y_i^*\|}{h}\right) x_{i+1}^*}{\sum_{i=1}^{n-1} K\left(\frac{\|y - y_i^*\|}{h}\right)}, y \in \mathbb{R}^p.$$

- (d) Fix the last p pseudo values as: $X_n^* = x_n, \dots, X_{n-p+1}^* = x_{n-p+1}$, i.e., $y_n^* = y_n$ and compute the bootstrap root replicate as

$$X_{n+1}^* - \hat{X}_{n+1}^*,$$

where $\hat{X}_{n+1}^* = \hat{m}^*(y_n^*) = \hat{m}^*(y_n)$, and

$$X_{n+1}^* = \hat{m}_g(y_n^*) + \hat{\sigma}_g(y_n^*)\epsilon_{n+1}^* = \hat{m}_g(y_n) + \hat{\sigma}_g(y_n)\epsilon_{n+1}^*.$$

- (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 X_{n+1} is given by

$$[\hat{m}(y_n) + q(\alpha/2), \hat{m}(y_n) + q(1 - \alpha/2)] \quad (3.14)$$

3.2.2 Bootstrap Algorithm with Predictive Residuals

Algorithm 3.2.2.

- (1) same as step(1) in Algorithm 3.2.1

(2) Use the delete- x_t data set to compute the delete- t th-observation kernel smoothing estimator $\hat{m}^{(t)}$ and $\hat{\sigma}^{(t)}$ using the similar formula as (3.10) and (3.11),

$$\hat{m}^{(t)}(y) = \frac{\sum_{i=p+1, i \neq t}^n K\left(\frac{\|y - y_{i-1}\|}{h}\right) x_i}{\sum_{i=p+1, i \neq t}^n K\left(\frac{\|y - y_{i-1}\|}{h}\right)}, \quad (3.15)$$

$$\hat{\sigma}^{(t)}(y) = \frac{\sum_{i=p+1, i \neq t}^n K\left(\frac{\|y - y_{i-1}\|}{h}\right) (x_i - \hat{m}^{(t)}(y_{i-1}))^2}{\sum_{i=p+1, i \neq t}^n K\left(\frac{\|y - y_{i-1}\|}{h}\right)}, \quad (3.16)$$

Then calculate the predictive residuals:

$$\hat{\epsilon}_t^{(t)} = \frac{x_t - \hat{m}^{(t)}(y_{t-1})}{\hat{\sigma}^{(t)}(y_{t-1})}, t = p+1, \dots, n. \quad (3.17)$$

(3)-(5) Replace $\hat{\epsilon}_t$ into $\hat{\epsilon}_t^{(t)}$, the rest is the same in 3.2.1.

Remark 3.2.2. The kernel smoothing estimator $\hat{m}(y)$ has poor performance if y is located in sparse area. In order to get a reliable estimator of X_{n+1} , $Y_n = y_n$ can NOT be an extreme value.

Remark 3.2.3. In practice, especially when n is not large, $\sum_{i=p+1, i \neq t}^n K\left(\frac{\|y_{t-1} - y_{i-1}\|}{h}\right)$ in the denominator of (3.15) and (3.16) could be 0 if y_{t-1} is far from $\{y_i, i \neq t-1\}$. Then these estimates in (3.15) and (3.16) are not defined.

If $K\left(\frac{\|y_{t-1} - y_{i-1}\|}{h}\right)$ is close to 0 for all $i \neq t, j$, and $K\left(\frac{\|y_{t-1} - y_{j-1}\|}{h}\right) \neq 0$. then $\hat{m}^{(t)}(y_{t-1}) \approx x_j$ and $\hat{\sigma}^{(t)}(y_{t-1}) \approx x_j - \hat{m}^{(t)}(y_{j-1}) \approx 0$. So the denominator of (3.17) is close to 0 if $\hat{m}^{(t)}(y_{j-1})$ is close to x_j . In this case, $\hat{\epsilon}_t^{(t)}$ will be extremely large.

To avoid these situations, we delete all $\hat{\epsilon}_t^{(t)}$ that is not defined or extremely large.

3.2.3 Asymptotic Properties

For simplicity, we again consider a stationary and geometrically ergodic process or order 1. Then the nonparametric autoregression model with heteroscedastic innovations is tailored as the following form,

$$X_t = m(X_{t-1}) + \sigma(X_{t-1})\epsilon_t \quad (3.18)$$

where the innovations $\{\epsilon_t\}$ are independent, with mean 0 and variance 1.

We estimate the conditional mean $m(x) = E[X_{t+1}|X_t = x]$ and the conditional variance $\sigma^2(x) = E[(X_{t+1} - m(x))^2|X_t = x]$ by the following kernel estimates $\hat{m}(x), \hat{\sigma}^2(x)$ defined as

$$\hat{m}(x) = \frac{\sum_{t=1}^{n-1} K\left(\frac{x-X_t}{h}\right) X_{t+1}}{\sum_{t=1}^{n-1} K\left(\frac{x-X_t}{h}\right)} \quad (3.19)$$

$$\hat{\sigma}^2(x) = \frac{\sum_{t=1}^{n-1} K\left(\frac{x-X_t}{h}\right) (X_{t+1} - \hat{m}(X_t))^2}{\sum_{t=1}^{n-1} K\left(\frac{x-X_t}{h}\right)} \quad (3.20)$$

Franke, Kreiss and Neumann(2002)Franke et al. (2002a)) have proved the consistency of the bootstrap in estimating the distribution of the estimators of $m(x)$ and $\sigma(x)$. From now on all the following theorems and lemmas in this section are under assumptions (AB1)-(AB12)(Franke, Kreiss and Neumann 2002Franke et al. (2002a)).

Theorem 3.2.1. [Franke, Kreiss and Neumann(2002)Franke et al. (2002a)]

$d_0(\mathcal{L}^*(\sqrt{nh}\{\hat{m}(x_n) - \hat{m}^*(x_n)\}), (\mathcal{L}(\sqrt{nh}\{m(x_n) - \hat{m}(x_n)\}))) \rightarrow 0$ in probability, and $d_0(\mathcal{L}^*(\sqrt{nh}\{\hat{\sigma}(x_n) - \hat{\sigma}^*(x_n)\}), (\mathcal{L}(\sqrt{nh}\{\sigma(x_n) - \hat{\sigma}(x_n)\}))) \rightarrow 0$ in probability, where d_0 denotes the kolmogorov distance, $\mathcal{L}^*$ is the distribution given the original observation.

Lemma 3.2.2. $\hat{\epsilon}_t - \hat{\epsilon}_t^{(t)} = O_p(\frac{1}{n})$, when $n \rightarrow \infty$.

Proof. We can prove in a similar way as lemma 3.1.2 that $\hat{m} - \hat{m}^{(t)} = O_p(1/n)$ and $\hat{\sigma} - \hat{\sigma}^{(t)} = O_p(1/n)$. Then

$$\begin{aligned} \hat{\epsilon}_t - \hat{\epsilon}_t^{(t)} &= \frac{X_t - \hat{m}(X_{t-1})}{\hat{\sigma}(X_{t-1})} - \frac{X_t - \hat{m}^{(t)}(X_{t-1})}{\hat{\sigma}^{(t)}(X_{t-1})} \\ &= \frac{X_t - \hat{m}(X_{t-1}) - (X_t - \hat{m}^{(t)}(X_{t-1}))}{\hat{\sigma}(X_{t-1})} (1 + O_p(1/n)) \\ &= O_p(1/n)(1 + O_p(1/n)) = O_p(1/n) \end{aligned}$$

□

Following the Definition 1.1.1, the predictive root is:

$$\begin{aligned} X_{n+1} - \hat{X}_{n+1} &= m(X_n) + \sigma(X_n)\epsilon_{n+1} - \hat{m}(X_n) \\ &= (m(X_n) - \hat{m}(X_n)) + \sigma(X_n)\epsilon_{n+1} \end{aligned} \quad (3.21)$$

And the bootstrap predictive root is as follows,

$$X_{n+1}^* - \hat{X}_{n+1}^* = \hat{m}(X_n) - \hat{m}^*(X_n) + \hat{\sigma}(X_n)\epsilon_{n+1}^* \quad (3.22)$$

As mentioned in 3.1.3, the distribution of $(m(X_n) - \hat{m}(X_n))$ can be approximated by the distribution of $\hat{m}(X_n) - \hat{m}^*(X_n)$ (Theorem 3.1.1) and distribution of ϵ_{n+1} can be approximated by the distribution of ϵ_{n+1}^* , which is re-sampled from the fitted or predictive residuals. Theorem 3.2.1 further shows that we can use the distribution of $\hat{\sigma}(X_n) - \hat{\sigma}^*(X_n)$ to approximate distribution of $\sigma(X_n) - \hat{\sigma}(X_n)$.

Follows from Politis(2013), we have

$$\begin{aligned} &P(X_{n+1} - \hat{X}_{n+1} \leq \alpha | X_n = x_n) \\ &= F_\sigma(\alpha) + f_\sigma(\alpha)E(\hat{m}(x_n) - m(x_n)) + f'_\sigma(\alpha)E(\hat{m}(x_n) - m(x_n))^2 + o(1/\sqrt{nh}) \end{aligned} \quad (3.23)$$

and

$$\begin{aligned} &P^*(X_{n+1}^* - \hat{X}_{n+1}^* \leq \alpha | X_n = x_n) \\ &= F_{n,\hat{\sigma}}(\alpha) + f_{n,\hat{\sigma}}(\alpha)E^*(\hat{m}^*(x_n) - m^*(x_n)) + f'_{n,\hat{\sigma}}(\alpha)E^*(\hat{m}^*(x_n) - m^*(x_n))^2 \\ &\quad + o_p(1/\sqrt{nh}) \end{aligned} \quad (3.24)$$

Here, F_σ and f_σ are the distribution and density function of $\sigma(x_n)\epsilon_{n+1}$, $F_{n,\hat{\sigma}}$ and $f_{n,\hat{\sigma}}$ are the distribution and density function of $\hat{\sigma}(x_n)\epsilon_{n+1}^*$. We further assume $F_{n,\hat{\sigma}}$ and F_σ are differentiable. When $n \rightarrow \infty$, both (3.23) and (3.24) converge to $F_\sigma(\alpha)$. However, when investigating the high order accuracy, even if the distribution of ϵ is given, i.e., $\hat{F}_\epsilon = F_\epsilon$, the difference of the first term of (3.23) and (3.24) is of order $O_p(1/\sqrt{nh})$ since $\hat{\sigma}(x_n) = \sigma(x_n) + O_p(1/\sqrt{nh})$ while the first term in (3.8) and (3.9) in Section 3.1.3 are exactly same in this case. So (3.24) does not fully capture the variability of (3.23).

Remark 3.2.4. Although we know that given $X_n = x_n$, the the distribution of the first term of (3.21) and the distribution of the first term in (3.22) agree up to $O_p(1/\sqrt{nh})$ as in i.i.d. case we discussed in 3.1.3. However, the second term of the expansion (3.22) will differ by $O_p(1/\sqrt{nh})$ from the second term of (3.21) due to the difference of

$\hat{\sigma}$ and σ . We can analyze $m(X_n) - \hat{m}(X_n)$ and $\sigma(X_n)\epsilon_{n+1}$ in (3.21) separately because given $X_n = x_n$, $(m(X_n) - \hat{m}(X_n)|X_n = x_n) = m(x_n) - \hat{m}(x_n)$ and $(\sigma(X_n)\epsilon_{n+1}|X_n = x_n) = \sigma(x_n)\epsilon_{n+1}$ are independent.

Due to the bias of $O_p(1/\sqrt{nh})$ from estimating $\sigma(X_n)$ by $\hat{\sigma}(X_n)$ in the second term of (3.21) and (3.22), we expect to see under-coverage phenomenon of the prediction interval in the simulation section. To fix this problem, we use an over-smoothed bandwidth in creating the pseudo data in the bootstrap world. i.e, $x_{t+1}^* = \hat{m}_g(x_t^*) + \hat{\sigma}_g(x_t^*)\epsilon_t^*$. $\hat{m}_g$ and $\hat{\sigma}_g$ are computed in the same way as $\hat{m}$ and $\hat{\sigma}$ but using a bandwidth g , where $g \gg h$. This idea was originally developed for confidence interval by Hardle and Marron(1991).

3.3 Partially Studentized Method

In Chapter one, we define the studentized predictive root as $\frac{X_{n+h} - \hat{X}_{n+h}}{\hat{V}_n}$, where $\hat{V}_n^2$ is an estimate of the error variance in h-step ahead prediction. For a linear autoregression model, we discussed that

$$X_{n+h} - \hat{X}_{n+h} = \left(g(Y_n; \underline{\phi}) - g(Y_n; \underline{\hat{\phi}})\right) + \sum_{j=0}^{h-1} \psi_j \epsilon_{n+h-j},$$

where $g(Y_n; \underline{\hat{\phi}})$ is a linear function of the last p observations and the coefficients only depend on $\underline{\hat{\phi}}$. Take AR(1) for example, if $h = 2$, then $g(X_n, \phi) = \phi^2 X_n$ and $\sum_{j=0}^{h-1} \psi_j \epsilon_{n+h-j} = \epsilon_{n+2} + \phi \epsilon_{n+1}$.

We estimate the h-step ahead error variance by the variance of $\sum_{j=0}^{h-1} \psi_j \epsilon_{n+h-j}$, which can be estimated by $\hat{V}_n^2 = (\sum_{j=0}^{h-1} \hat{\psi}_j^2) \hat{\sigma}^2$, where $\hat{\sigma}^2$ is the variance of the fitted or predictive residuals.

For a nonparametric model, it is not easy to estimate the error variance of $X_{n+h} - \hat{X}_{n+h}$. We can not capture the prediction error caused by noises in an explicit expression as we do for linear autoregression. However, we can partially capture the error variance in h-step ahead prediction by the variance of residuals. Then we can

define the partially studentized predictive root and partially studentized bootstrap predictive root as follows,

Definition 3.3.1.

(i) partially studentized predictive root

The partially studentized predictive root is

$$\frac{X_{n+h} - \hat{X}_{n+h}}{\hat{S}_n},$$

Where $\hat{X}_{n+h}$ is the predictor of X_{n+h} , $\hat{S}_n$ is an estimate of the residual variance, $\hat{X}_{n+h}$ and $\hat{S}_n$ are both functions of the data, let $\hat{X}_{n+h} = \Pi(X_1, \dots, X_n)$ and $\hat{S}_n = S(X_1, \dots, X_n)$.

(ii) partially studentized bootstrap predictive root

Consider a bootstrap pseudo series $X_1^, \dots, X_n^*$, the partially studentized bootstrap predictive root is defined as:*

$$\frac{X_{n+h}^* - \hat{X}_{n+h}^*}{\hat{S}_n^*}.$$

Here $\hat{X}_{n+h}^* = \Pi(X_1^*, \dots, X_n^*)$ and $\hat{S}_n^* = V(X_1^*, \dots, X_n^*)$ are functions of pseudo data, where Π and S are the same functions as above.

If we collect B partially studentized bootstrap predictive root replicates in the form of an empirical distribution whose α -quantile is denoted as $q(\alpha)$, then the $(1 - \alpha)100\%$ equal-tailed predictive intervals for X_{n+h} is given by

$$[\hat{X}_{n+h} + \hat{S}_n q(\alpha/2), \hat{X}_{n+h} + \hat{S}_n q(1 - \alpha/2)],$$

where $\hat{S}_n^2$ is the variance of the fitted or predictive residuals calculated in step (2) of Algorithm 3.1.1, Algorithm 3.1.2, Algorithm 3.2.1 and Algorithm 3.2.2. Note that for nonparametric model with i.i.d. errors, when $h = 1$, $\frac{X_{n+h} - \hat{X}_{n+h}}{\hat{S}_n}$ is also the studentized predictive root.

3.4 Monte Carlo Studies

3.4.1 Simulation Results for Nonparametric Autoregression with i.i.d. Errors

In this subsection, we still use Monte Carlo simulations to assess the performance of the bootstrap methods with fitted and predictive residuals through average coverage level(CVR) and length(LEN) as defined in Section 1.7.

To evaluate the performance of the bootstrap methods for nonparametric non-linear auto-regression with i.i.d. innovations, we use the following models of order $p = 1$.

- Model 1: $X_{t+1} = \sin(X_t) + \epsilon_{t+1}$
- Model 2: $X_{t+1} = 0.8 \log(3X_t^2 + 1) + \epsilon_{t+1}$
- Model 3: $X_{t+1} = -0.5 \exp(-50X_t^2)X_t + \epsilon_{t+1}$

with following features:

- (1) $\{\epsilon_t\}$ are i.i.d. from $N(0,1)$ or two-sided exponential(Laplace) distribution rescaled to unit variance,
- (2) 500 data sets each of size $n = 100$ and $n = 200$,
- (3) prediction interval with nominal coverage levels of 95% and 90%,
- (4) 1-step ahead and 3-step ahead prediction intervals
- (5) choose $K(\cdot)$ to be the normal kernel and select h by cross validation method.

Table 2.1 -2.3 summarize the simulation results of each model.

Table 3.1: nonparametric autoregression with i.i.d innovations-model 1

normal innovation	nominal coverage 95%			nominal coverage 90%		
$h = 1, n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
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
nonpara-Sf	0.938	4.020	0.403	0.887	3.387	0.314
nonpara-Sp	0.939	4.030	0.405	0.888	3.390	0.313
$h = 1, n = 200$						
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
nonpara-Sf	0.945	3.966	0.280	0.894	3.339	0.222
nonpara-Sp	0.945	3.970	0.282	0.895	3.344	0.228
$h = 3, n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
nonpara-f	0.924	4.534	0.396	0.867	3.829	0.339
nonpara-p	0.941	4.801	0.444	0.889	3.048	0.378
nonpara-Sf	0.936	4.734	0.450	0.881	3.981	0.379
nonpara-Sp	0.937	4.719	0.423	0.882	3.969	0.357
$h = 3, n = 200$						
nonpara-f	0.933	4.589	0.271	0.878	3.873	0.221
nonpara-p	0.941	4.733	0.305	0.889	3.995	0.253
nonpara-Sf	0.939	4.704	0.298	0.886	3.967	0.239
nonpara-Sp	0.938	4.674	0.291	0.885	3.945	0.242
Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$h = 1, n = 100$						
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
nonpara-Sf	0.942	4.388	0.675	0.892	3.386	0.466
nonpara-Sp	0.942	4.364	0.641	0.892	3.386	0.465
$h = 1, n = 200$						
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
nonpara-Sf	0.943	4.250	0.473	0.893	3.293	0.333
nonpara-Sp	0.941	4.234	0.447	0.893	3.299	0.327
$h = 3, n = 100$						
nonpara-f	0.931	4.462	0.564	0.872	3.702	0.433
nonpare-p	0.943	4.930	0.656	0.891	3.940	0.497
nonpara-Sf	0.941	4.913	0.632	0.887	3.906	0.490
nonpara-Sp	0.940	4.866	0.631	0.886	3.875	0.466
$h = 3, n = 200$						
Laplace innovations	CVR	LEN	st.err	CVR	LEN	st.err
nonpara-f	0.934	4.622	0.396	0.877	3.692	0.298
nonpara-p	0.941	4.781	0.419	0.888	3.821	0.305
nonpara-Sf	0.940	4.768	0.418	0.885	3.800	0.314
nonpara-Sp	0.940	4.749	0.423	0.884	3.782	0.303

Table 3.2: nonparametric autoregression with i.i.d innovations-model 2

normal innovations	nominal coverage 95%			nominal coverage 90%		
$h = 1, n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
nonpara-f	0.928	3.870	0.388	0.875	3.260	0.308
nonpara-p	0.946	4.143	0.393	0.900	3.495	0.305
nonpara-Sf	0.940	4.051	0.393	0.890	3.407	0.309
nonpara-Sp	0.941	4.049	0.387	0.891	3.407	0.306
$h = 1, n = 200$						
nonpara-f	0.936	3.877	0.297	0.883	3.270	0.238
nonpara-p	0.948	4.053	0.295	0.899	3.418	0.238
nonpara-Sf	0.943	3.992	0.302	0.893	3.359	0.241
nonpara-Sp	0.944	3.999	0.292	0.894	3.364	0.234
$h = 3, n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
nonpara-f	0.922	4.908	0.479	0.863	4.136	0.411
nonpara-p	0.941	5.234	0.504	0.889	4.412	0.432
nonpara-Sf	0.935	5.149	0.517	0.880	4.325	0.447
nonpara-Sp	0.935	5.120	0.485	0.880	4.304	0.419
$h = 3, n = 200$						
nonpara-f	0.931	4.989	0.311	0.874	4.214	0.266
nonpara-p	0.941	5.177	0.335	0.889	4.369	0.290
nonpara-Sf	0.938	5.135	0.333	0.884	4.330	0.282
nonpara-Sp	0.937	5.099	0.327	0.883	4.300	0.281
Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$h = 1, n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
nonpara-f	0.932	4.175	0.639	0.880	3.237	0.449
nonpara-p	0.944	4.458	0.649	0.898	3.472	0.464
nonpara-Sf	0.942	4.414	0.667	0.893	3.415	0.466
nonpara-Sp	0.941	4.380	0.631	0.893	3.406	0.452
$h = 1, n = 200$						
nonpara-f	0.937	4.110	0.455	0.884	3.192	0.322
nonpara-p	0.944	4.284	0.452	0.895	3.347	0.336
nonpara-Sf	0.942	4.247	0.467	0.892	3.296	0.328
nonpara-Sp	0.942	4.234	0.453	0.892	3.297	0.327
$h = 3, n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
nonpara-f	0.925	4.992	0.653	0.867	4.044	0.529
nonpara-p	0.940	5.318	0.715	0.889	4.312	0.563
nonpara-Sf	0.938	5.302	0.744	0.884	4.277	0.599
nonpara-Sp	0.937	5.250	0.709	0.883	4.234	0.538
$h = 3, n = 200$						
nonpara-f	0.931	4.971	0.463	0.872	4.028	0.373
nonpara-p	0.940	5.174	0.479	0.886	4.184	0.377
nonpara-Sf	0.938	5.139	0.494	0.883	4.162	0.403
nonpara-Sp	0.937	5.115	0.486	0.882	4.136	0.378

Table 3.3: nonparametric autoregression with i.i.d innovations-model 3

normal innovations	nominal coverage 95%			nominal coverage 90%		
$h = 1, n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
nonpara-f	0.932	3.832	0.369	0.881	3.236	0.282
nonpara-p	0.940	3.946	0.380	0.892	3.332	0.296
nonpara-Sf	0.940	3.945	0.372	0.891	3.323	0.280
nonpara-Sp	0.940	3.944	0.376	0.891	3.322	0.288
$h = 1, n = 200$						
nonpara-f	0.941	3.862	0.265	0.890	3.257	0.210
nonpara-p	0.945	3.926	0.274	0.896	3.312	0.214
nonpara-Sf	0.944	3.921	0.263	0.895	3.302	0.206
nonpara-Sp	0.945	3.929	0.271	0.896	3.309	0.210
$h = 3, n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
nonpara-f	0.937	3.858	0.362	0.886	3.256	0.274
nonpara-p	0.944	3.987	0.389	0.898	3.370	0.297
nonpara-Sf	0.944	3.976	0.376	0.896	3.344	0.289
nonpara-Sp	0.945	3.985	0.374	0.897	3.355	0.286
$h = 3, n = 200$						
nonpara-f	0.943	3.894	0.276	0.894	3.284	0.212
nonpara-p	0.947	3.949	0.285	0.899	3.312	0.218
nonpara-Sf	0.947	3.951	0.272	0.899	3.330	0.212
nonpara-Sp	0.947	3.953	0.276	0.899	3.327	0.214
Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$h = 1, n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
nonpara-f	0.934	4.192	0.636	0.882	3.228	0.447
nonpara-p	0.938	4.295	0.640	0.890	3.322	0.455
nonpara-Sf	0.941	4.361	0.647	0.891	3.350	0.454
nonpara-Sp	0.940	4.317	0.622	0.891	3.335	0.443
$h = 1, n = 200$						
nonpara-f	0.939	4.143	0.478	0.889	3.205	0.327
nonpara-p	0.941	4.190	0.467	0.893	3.269	0.337
nonpara-Sf	0.943	4.226	0.471	0.894	3.268	0.324
nonpara-Sp	0.942	4.206	0.460	0.894	3.273	0.329
$h = 3, n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
nonpara-f	0.936	4.178	0.612	0.886	3.232	0.431
nonpara-p	0.943	4.329	0.677	0.895	3.349	0.448
nonpara-Sf	0.944	4.352	0.634	0.895	3.356	0.448
nonpara-Sp	0.944	4.357	0.655	0.896	3.365	0.445
$h = 3, n = 200$						
nonpara-f	0.940	4.130	0.457	0.891	3.226	0.332
nonpara-p	0.943	4.212	0.470	0.894	3.269	0.319
nonpara-Sf	0.943	4.219	0.457	0.896	3.285	0.326
nonpara-Sp	0.944	4.228	0.464	0.895	3.281	0.313

From Table 2.1-2.3, we see that

- when $n = 100$ the p, Sf and Sp methods have significant improvement on the coverage rate compared to the nonstudentized method with fitted residuals for both $h = 1$ and $h = 3$. The coverage levels of all but the f method are close to the true coverage level in both normal and Laplace case.
- When $n = 200$, the coverage levels of the nonstudentized method with fitted residuals improve a lot. The coverage levels for the other methods are already very good when $n = 100$, so the coverage level almost stay same for other methods when changing to a larger sample size.
- The standard deviation of the interval length become smaller when $n = 200$ for both normal and Laplace innovations. Larger sample size leads to the decreased variability of the interval length.
- The technique of using predictive residuals is not necessary for studentized method.

3.4.2 Simulation Results for Nonparametric Autoregression with Heteroscedastic Errors

To evaluate the performance of the bootstrap methods for nonparametric non-linear auto-regression with heteroscedastic innovations, we use the following model,

- Model 4: $X_t = \sin(X_{t-1}) + \sqrt{0.5 + 0.25X_{t-1}^2}\epsilon_t$
- Model 5: $X_{t+1} = 0.75X_t + 0.15X_t\epsilon_{t+1} + \epsilon_{t+1}$

with the features same as (1)-(3) and (5) of model 1-3.

Following two tables summarize the simulation results of model 4 and model 5 without over-smoothed resampling bandwidth.

Table 3.4: heteroscedastic model 4 without over-smoothed bandwidth

heteroscedastic normal innovations	nominal coverage 95%				nominal coverage 90%		
	CVR	LEN	st.err		CVR	LEN	st.err
$n = 100$							
nonpara-f	0.838	2.630	1.093		0.776	2.238	0.930
nonpara-p	0.871	2.889	1.193		0.804	2.391	0.989
$n = 200$							
nonpara-f	0.864	2.733	1.007		0.801	2.317	0.845
nonpara-p	0.885	2.888	1.024		0.817	2.405	0.846
Laplace innovations	CVR	LEN	st.err		CVR	LEN	st.err
$n = 100$							
nonpara-f	0.847	2.719	1.349		0.785	2.147	1.000
nonpara-p	0.878	3.094	1.571		0.812	2.330	1.063
$n = 200$							
nonpara-f	0.871	2.687	1.098		0.810	2.128	0.873
nonpara-p	0.890	2.914	1.207		0.824	2.233	0.921

Table 3.5: heteroscedastic model 5 without over-smoothed bandwidth

heteroscedastic normal innovations	nominal coverage 95%				nominal coverage 90%		
	CVR	LEN	st.err		CVR	LEN	st.err
$n = 100$							
nonpara-f	0.884	3.619	1.575		0.831	3.079	1.317
nonpara-p	0.911	3.969	1.732		0.856	3.290	1.414
$n = 200$							
nonpara-f	0.904	3.706	1.490		0.854	3.146	1.255
nonpara-p	0.919	3.916	1.602		0.866	3.264	1.323
Laplace innovations	CVR	LEN	st.err		CVR	LEN	st.err
$n = 100$							
nonpara-f	0.880	3.886	2.351		0.829	3.085	1.841
nonpara-p	0.906	4.452	3.852		0.850	3.354	2.066
$n = 200$							
nonpara-f	0.900	3.803	1.787		0.846	3.005	1.405
nonpara-p	0.915	4.129	1.962		0.860	3.172	1.485

From above two tables, we see that there is significant under coverage for the models with heteroscedastic errors in both normal and Laplace case. As mentioned in the Section 3.2, the nonparametric autoregression bootstrap discussed in this section doesn't work well for heteroscedastic models without using over-smoothed resampling

bandwidth.

As a rule of thumb, the resampling bandwidth g are often recommended to be $1.5h$ or $2h$. So we re-do the simulation for model 4 and model 5 using $g = 1.5h$ and $g = 2h$.

Table 3.6: heteroscedastic model 4 with $g = 1.5h$

$g = 1.5h$	nominal coverage 95%			nominal coverage 90%		
normal innovations	CVR	LEN	st.err	CVR	LEN	st.err
$n = 100$						
nonpara-f	0.870	2.809	0.968	0.813	2.391	0.824
nonpara-p	0.901	3.090	1.058	0.839	2.556	0.870
$n = 200$						
nonpara-f	0.885	2.840	0.857	0.825	2.408	0.719
nonpara-p	0.904	3.003	0.874	0.841	2.502	0.726
Laplace innovations	CVR	LEN	st.err	CVR	LEN	st.err
$n = 100$						
nonpara-f	0.874	2.951	1.240	0.819	2.329	0.894
nonpara-p	0.904	3.371	1.464	0.843	2.526	0.955
$n = 200$						
nonpara-f	0.889	2.835	0.973	0.831	2.245	0.763
nonpara-p	0.907	3.076	1.063	0.846	2.357	0.806

Table 3.7: heteroscedastic model 4 with $g = 2h$

$g = 2h$	nominal coverage 95%			nominal coverage 90%		
normal innovations	CVR	LEN	st.err	CVR	LEN	st.err
$n = 100$						
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
nonpara-f	0.894	3.018	0.934	0.843	2.569	0.790
nonpara-p	0.923	3.337	1.017	0.869	2.761	0.839
$n = 200$						
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
nonpara-Sf	0.903	2.986	0.779	0.847	2.534	0.652
nonpara-Sp	0.921	3.168	0.796	0.863	2.638	0.657
Laplace innovations	CVR	LEN	st.err	CVR	LEN	st.err
$n = 100$						
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
nonpara-Sf	0.894	3.200	1.300	0.843	2.523	0.930
nonpara-Sp	0.922	3.691	1.553	0.866	2.762	0.989
$n = 200$						
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
nonpara-Sf	0.904	3.026	0.972	0.850	2.392	0.757
nonpara-Sp	0.921	3.294	1.041	0.864	2.520	0.783

Table 3.8: heteroscedastic model 5 with $g = 1.5h$

$g = 1.5h$	nominal coverage 95%			nominal coverage 90%		
normal innovations	CVR	LEN	st.err	CVR	LEN	st.err
$n = 100$						
nonpara-f	0.911	3.816	1.494	0.862	3.250	1.253
nonpara-p	0.934	4.197	1.655	0.885	3.475	1.342
$n = 200$						
nonpara-f	0.925	3.848	1.384	0.878	3.267	1.169
nonpara-p	0.938	4.065	1.483	0.889	3.390	1.229
Laplace innovations	CVR	LEN	st.err	CVR	LEN	st.err
$n = 100$						
nonpara-f	0.906	4.145	2.292	0.860	3.290	1.796
nonpara-p	0.929	4.760	2.782	0.880	3.596	2.029
$n = 200$						
Nonpara-f	0.920	4.014	1.592	0.871	3.174	1.240
Nonpara-p	0.934	4.360	1.758	0.884	3.347	1.343

Table 3.9: heteroscedastic model 5 with $g = 2h$

$g = 2h$	nominal coverage 95%			nominal coverage 90%		
normal innovations	CVR	LEN	st.err	CVR	LEN	st.err
$n = 100$						
nonpara-f	0.931	4.063	1.453	0.888	3.465	1.228
nonpara-p	0.950	4.672	1.621	0.908	3.704	1.321
nonpara-Sf	0.930	4.054	1.462	0.887	3.457	1.232
nonpara-Sp	0.950	4.481	1.639	0.908	3.711	1.332
$n = 200$						
nonpara-f	0.941	4.034	1.322	0.898	3.427	1.113
nonpara-p	0.952	4.262	1.404	0.909	3.556	1.173
nonpara-Sf	0.940	4.029	1.326	0.898	3.422	1.117
nonpara-Sp	0.953	4.262	1.411	0.910	3.557	1.178
Laplace innovations	CVR	LEN	st.err	CVR	LEN	st.err
$n = 100$						
nonpara-f	0.923	4.414	2.283	0.881	3.512	1.811
nonpara-p	0.943	5.081	2.865	0.899	3.827	2.001
nonpara-Sf	0.922	4.384	2.256	0.880	3.490	1.784
nonpara-Sp	0.943	5.103	2.951	0.900	3.844	2.076
$n = 200$						
nonpara-f	0.933	4.222	1.484	0.888	3.338	1.151
nonpara-p	0.945	4.594	1.656	0.900	3.525	1.249
nonpara-Sf	0.932	4.204	1.475	0.887	3.323	1.149
nonpara-Sp	0.945	4.597	1.661	0.900	3.530	1.255

From the previous four tables, we see that

- When we choose a resampling bandwidth g as $1.5h$ and $2h$ the predictive interval for model 4 improved compared to those without using over-smoothed bandwidth in creating the pseudo series. Based on the simulation result, it is proper to choose a g at least as big as $2h$ to make the prediction intervals closer to the true ones. And the predictive residuals works better than the fitted residuals. For model 5, the coverage levels of the predictive intervals using predictive residuals are close to the nominal coverage level when $g = 2h$. So for model 5, we can choose a $g = 2h$.
- For both model 4 and model 5, we observe that when $n = 200$, the variability of the interval length decreased significantly. However the standard deviations

corresponding to the interval lengths are still quite large with respect to the length.

- The undercoverage of heteroscedastic model is mostly due to the bias of estimating $\sigma(\cdot)$ by $\hat{\sigma}(\cdot)$, and partially studentized approaches do not help for heteroscedastic models.

3.5 Backward Bootstrap for Nonparametric Autoregression

We use the backward scheme in chapter 1 for simple linear autoregression. The backward errors are not i.i.d., however we can construct the backward errors explicitly through the autoregression coefficients and the forward i.i.d. errors as in formula (1.7). When we try to apply this idea to the totally nonparametric procedure, the problem arises because it's hard to find the explicit distribution for the backward errors. Let us take the model of order 1 with i.i.d. errors for example, $X_t = m(X_{t-1}) + \epsilon_t$, and the backward presentation is $X_t = m^{-1}(X_{t+1} - \epsilon_{t+1}) = v(X_{t+1}) + w_t$ both the distribution of backward error $\{w_t\}$ and the backward function $v(\cdot)$ are hard to find explicitly through $m(\cdot)$ and $\{\epsilon_t\}$.

However, we can use the bootstrap based on the estimation of the transition density or the local bootstrap method discussed in next chapter to find the distribution of $X_t|X_{t+1}$. So we can use the bootstrap based methods introduced in next chapter to generate the pseudo bootstrap series and still use the nonparametric kernel estimator derived in this chapter to get $\hat{m}$ and $\hat{m}^*$ and construct the prediction root.

We will revisit the backward method for nonparametric autoregression in the context of resampling Markov processes in next chapter.

Chapter 3, in full, is currently being prepared for the following paper prepared for future submission for publication: Bootstrap prediction intervals for linear, non-linear and nonparametric autoregressions. Pan, Li; Politis, Dimitris. The dissertation

author was the primary investigator and author of this material.

Chapter 4

Bootstrap Prediction Intervals for Markov Processes Based on Conditional Distributions

In chapter 1 and 2 we constructed the bootstrap prediction intervals for linear and nonlinear autoregression, and in chapter 3 we addressed the nonparametric autoregression model. In this chapter, we will extend to a more general autoregressive structure: strictly stationary and real-valued stochastic processes $\{X_t, t = 1, 2, \dots\}$ that follow a general finite order Markovian structure. All the models discussed in previous chapters are included in the Markovian structure.

Rajarshi(1990) developed a bootstrap method based on kernel estimates of the transition density of the Markov processes. We will use this method to construct the prediction interval in section 4.1. In the procedure of generating pseudo series $X_1^*, X_2^*, \dots, X_n^*$, we use both forward and backward schemes. We will assess the performance of the forward and backward bootstrap method in Monte Carlo simulation.

Paparoditis and Politis(2002) proposed another bootstrap method for Markov processes based on conditional distributions, the so called local bootstrap. The idea of backward local bootstrap for prediction was first proposed and used in an autoregressive process with ARCH errors by Paparoditis and Politis(1998). We will also use

the local bootstrap method to construct prediction intervals using both forward and backward schemes in section 4.2.

In Section 4.3, we will apply the backward bootstrap methods introduced in Section 4.1 and 4.2 to the nonparametric autoregression model discussed in last chapter. Simulation results will be given in the end of this section.

4.1 Bootstrap Based on estimates of Transition Density

In this section, we start to consider a strictly stationary Markov sequence $X = \{X_1, X_2, \dots\}$ of order p . Motivated by Rajarshi(1990), we introduce a forward bootstrap method based on transition density estimates to construct the prediction intervals in Section 4.1.2. After showing the reverse of a Markov chain is also a Markov chain in Section 4.1.3, we give the algorithm of backward bootstrap based on backward transition density. The asymptotic properties and Monte Carlo Studies will be discussed in Section 4.1.4 and 4.1.5.

4.1.1 Notation and Definition

Let $x \in \mathbb{R}, y \in \mathbb{R}^p, \{X_t, t = 1, 2, \dots\}$ is a order p strictly stationary Markov chain with

$$\begin{aligned} F(y) &= P[Y_p \leq y], \\ F(x, y) &= P[X_{p+1} \leq x, Y_p \leq y], \\ F(x|y) &= P[X_{p+1} \leq x | Y_p = y]. \end{aligned} \tag{4.1}$$

Here $Y_t = (X_t, X_{t-1}, \dots, X_{t-p+1})'$. $Y = \{Y_p, Y_{p+1}, \dots\}$ forms an aperiodic, strictly stationary and geometrically ergodic Markov chain on $(\mathbb{R}^p, \mathcal{B}_p)$. $\mathcal{B}_p$ is the Borel σ -algebra over $\mathbb{R}^p$. Let $f(y), f(x, y), f(x|y)$ be the corresponding densities of (4.1).

Let $x_1, x_2, \dots, x_n$ be the observed sample path from the Markov chain X , $y_n = (x_n, \dots, x_{n-p+1})'$.

Definition 4.1.1. *The predictive root is the error in prediction:*

$$X_{n+1} - \hat{X}_{n+1},$$

Given $Y_n = y_n$, X_{n+1} is sampled from the transition density $f(\cdot|y_n)$ and $\hat{X}_{n+1} = \int x \hat{f}_n(x|y_n) dx$ is an estimate of conditional expectation of X_{n+1} given $X_1, \dots, X_n$. Since $\hat{f}_n(x|y_n)$ is a function of data and x, y_n , let $\hat{f}_n(x|y_n) = \Pi(x, y_n, X_1, \dots, X_n)$.

Consider a pseudo series $X_1^*, \dots, X_n^*$, the bootstrap predictive root is defined as:

$$X_{n+1}^* - \hat{X}_{n+1}^*,$$

Given $Y_n^* = y_n$, X_{n+1}^* is sampled from $\hat{f}_n(\cdot|y_n)$ and $\hat{X}_{n+1}^* = \int x \hat{f}_n^*(x|y_n) dx$ is an estimate of conditional expectation of X_{n+1}^* given $X_1^*, \dots, X_n^*$, where $\hat{f}_n^*(x|y_n) = \Pi(x, y_n, X_1^*, \dots, X_n^*)$. Π is the same function as above.

4.1.2 Forward Bootstrap Based on Transition Density

Let $x_1, x_2, \dots, x_n$ be the observed sample path from the Markov chain X , $y_n = (x_n, \dots, x_{n-p+1})'$. The construction of the bootstrap prediction interval can be performed as follows,

Algorithm 4.1.1.

- (1) Choose a proper kernel on $\mathbb{R}^2$ and bandwidth h_1, h_2 to construct the estimator of the transition density $\hat{f}_n(x|y)$, $x \in \mathbb{R}$, $y \in \mathbb{R}^p$, as follows,

$$\hat{f}_n(x, y) = \frac{1}{(n-p)h_1h_2} \sum_{i=p+1}^n K\left(\frac{x-x_i}{h_1}, \frac{\|y-y_{i-1}\|}{h_2}\right) \quad (4.2)$$

$$\hat{f}_n(y) = \int \hat{f}_n(x, y) dx \quad (4.3)$$

$$\hat{f}_n(x|y) = \frac{\hat{f}_n(x, y)}{\hat{f}_n(y)}. \quad (4.4)$$

Here $\|\cdot\|$ is a norm in $\mathbb{R}^p$.

- (2) Calculate the predictor of future value $\hat{x}_{n+1}$ by an estimate of conditional expectation $\int x \hat{f}_n(x|y_n) dx$.

- (3) (a) Generate $y_p^* = (x_p^*, \dots, x_1^*)$ with probability density function $\hat{f}_n(\cdot)$ given by (4.3).
- (b) Generate x_{p+1}^* with probability density function $\hat{f}_n(\cdot|y_p^*)$ given by (4.4).
- (c) Repeat (b) to generate $x_{t+1}^* \sim \hat{f}_n(\cdot|y_t^*)$, $t = p, \dots, n-1$, where

$$y_t^* = (x_t^*, \dots, x_{t-p+1}^*)'$$
- (d) Construct $\hat{f}_n^*(x|y)$ is constructed in the similar way as (4.4) but based on the pseudo data $x_1^*, x_2^*, \dots, x_n^*$ instead of the original observed data.
- (e) Calculate the bootstrap predictor of future value $\hat{x}_{n+1}^*$ by $\int x \hat{f}_n^*(x|y_n) dx$
- (f) Generate the bootstrap future value $x_{n+1}^* \sim \hat{f}_n^*(\cdot|y_n)$.
- (g) Calculate the bootstrap root replicate as $x_{n+1}^* - \hat{x}_{n+1}^*$
- (4) Repeat (3) 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 predictive interval for X_{n+1} is given by

$$[\hat{x}_{n+1} + q(\alpha/2), \hat{x}_{n+1} + q(1 - \alpha/2)]. \quad (4.5)$$

Remark 4.1.1. *Rajarshi(1990) used the bandwidth $0.9An^{-1/6}$, where $A = \min(\hat{\sigma}, \frac{IQR}{1.34})$ and $\hat{\sigma}$ is the estimated standard deviation of the data, IQR is the interquartile range($Q_3 - Q_1$). When we use this bandwidth selection rule, we have significantly over-coverage prediction interval. As suggested in assumption (B3): $\sum_{n=1}^{\infty} (nh^4)^{k+1} < \infty$, for some $k \geq 3$. we need h of order $n^{-\frac{1}{4}(1+\frac{1+a}{1+k})}$ for some $k \geq 3$ and $a > 0$. For infinitely large a and k , the order of h is $n^{-\frac{1}{4}}$. So the order of h should be no more than $O_p(n^{-1/4})$. We adjust h to be $0.9An^{-1/4}$. We do not recommend the cross validation method for bandwidth selection since the order of the optimal h selected by cross validation is of $O_p(n^{-1/5})$*

4.1.3 Backward Bootstrap Based on Transition Density

In chapter 1, we used both the forward bootstrap and backward bootstrap methods to create the bootstrap pseudo data. Inspired by the idea of generating pseudo data backwards, we introduce the backward bootstrap based on transition density for Markov process. Before applying the backward bootstrap for Markov process, however we need to show that the reverse Markov process is also a Markov process. Here we study the properties of Markov processes using the copulas. Following Sklar(1959,1973), an m -dimensional joint distribution function F with marginal distribution functions $F_1, F_2, \dots, F_m$ can be written as

$$F_{1,\dots,m}(x_1, \dots, x_m) = C(F_1(x_1), \dots, F_m(x_m)),$$

for some function $C : [0, 1]^m \mapsto [0, 1]$ called an m -copula. For an m copula $C(\alpha_1, \dots, \alpha_m)$, $0 \leq \alpha_j \leq 1$, $j = 1, 2, \dots, m$. Let $C_{,j}$ denote the partial derivative $\partial C / \partial \alpha_j$.

Darsow, Nguyen and Olsen(1992) have proved the following theorem

Theorem 4.1.1 (Darsow, Nguyen and Olsen(1992)). *A real valued stochastic process X_t , $t \in T$ (T is some set of real numbers) is a Markov process if and only if for all positive integers n and for all $t_1, \dots, t_n \in T$ satisfying $t_k < t_{k+1}$, $k = 1, \dots, n-1$,*

$$C_{t_1 \dots t_n} = C_{t_1 t_2} \star C_{t_2 t_3} \star \dots \star C_{t_{n-1} t_n},$$

where $C_{t_1 \dots t_n}$ is the copula associated with the joint distribution of $X_{t_1}, \dots, X_{t_n}$.

The definition of the $\star$ is as follows:

Let A be an m -copula and B be an n -copula. $A \star B$ is an $m+n-1$ copula given by

$$A \star B(\alpha_1, \dots, \alpha_{m+n-1}) = \int_0^{\alpha_m} A_{,m}(\alpha_1, \dots, \alpha_{m-1}, \xi) B_{,1}(\xi, \alpha_{m+1}, \dots, \alpha_{m+n-1}) d\xi.$$

A useful property for 2-copula and $\star$ operation is that

$$C_{st} = C_{su} \star C_{ut}, \tag{4.6}$$

for $s, u, t \in T$ satisfying $s < u < t$. Here C_{st} is the copula of X_s and X_t . cf. Darsow, Nguyen and Olsen(1992, Theorem 3.2).

Let $X_t, t \in T$ be a Markov process, then we have the following 2 claims.

Claim 4.1.2. $C_{t_3t_1} = C_{t_3t_2} \star C_{t_2t_1}$ for any $t_1, t_2, t_3 \in T$ and $t_1 < t_2 < t_3$.

Proof.

$$\begin{aligned}
C_{t_3t_1}(\alpha_{t_3}, \alpha_{t_1}) &= C_{t_1t_3}(\alpha_{t_1}, \alpha_{t_3}) && \text{by the definition of copula } C \\
&= C_{t_1t_2} \star C_{t_2t_3} && \text{by (4.6)} \\
&= \int_0^1 C_{t_1t_2,2}(\alpha_{t_1}, \xi) C_{t_2t_3,1}(\xi, \alpha_{t_3}) d\xi && \text{by the definition of } \star \text{ operation} \\
&= \int_0^1 C_{t_2t_1,1}(\xi, \alpha_{t_1}) C_{t_3t_2,2}(\alpha_{t_3}, \xi) d\xi && \text{by the definition of } C, C_{,1} \text{ and } C_{,2} \\
&= \int_0^1 C_{t_3t_2,2}(\alpha_{t_3}, \xi) C_{t_2t_1,1}(\xi, \alpha_{t_1}) d\xi \\
&= C_{t_3t_2} \star C_{t_2t_1}(\alpha_{t_3}, \alpha_{t_1})
\end{aligned}$$

□

Claim 4.1.3. $C_{t_n \dots t_1} = C_{t_n t_{n-1}} \star C_{t_{n-1} t_{n-2}} \star \dots \star C_{t_2 t_1}$ for all positive integers n and for all $t_1, \dots, t_n \in T$ satisfying $t_k < t_{k+1}, k = 1, \dots, n-1$.

Proof. We prove the claim by induction.

we know that $C_{t_3t_1} = C_{t_3t_2} \star C_{t_2t_1}$ (one $\star$ operation, $n = 3$).

If $C_{t_k \dots t_1} = C_{t_k t_{k-1}} \star C_{t_{k-1} t_{k-2}} \star \dots \star C_{t_2 t_1}$ (($k-2$) $\star$ operations, $n = k$),

to show: $C_{t_{k+1}t_k \dots t_1} = C_{t_{k+1}t_k} \star C_{t_k t_{k-1}} \star \dots \star C_{t_2 t_1}$. Note that

$$\begin{aligned}
& C_{t_{k+1}t_k \dots t_1}(\alpha_{t_{k+1}}, \alpha_{t_k}, \dots, \alpha_{t_1}) \\
&= C_{t_1 \dots t_k t_{k+1}}(\alpha_{t_1}, \dots, \alpha_{t_k}, \alpha_{t_{k+1}}) && \text{by Copula definition} \\
&= (C_{t_1 t_2} \star C_{t_2 t_3} \star \dots \star C_{t_{k-1} t_k}) \star C_{t_k t_{k+1}} && \text{by Theorem 4.1.1} \\
&= C_{t_1 \dots t_k} \star C_{t_k t_{k+1}} && \text{by Theorem 4.1.1} \\
&= \int_0^{\alpha_{t_k}} C_{t_1 \dots t_k, k}(\alpha_{t_1}, \dots, \alpha_{t_{k-1}}, \xi) C_{t_k t_{k+1}, 1}(\xi, \alpha_{t_{k+1}}) d\xi && \text{by the definition of } \star \text{ operation} \\
&= \int_0^{\alpha_{t_k}} C_{t_{k+1} t_k, 2}(\alpha_{t_{k+1}}, \xi) C_{t_k \dots t_1, 1}(\xi, \alpha_{t_{k-1}}, \dots, \alpha_{t_1}) d\xi && \text{by the definition of Copula} \\
&= C_{t_{k+1} t_k} \star C_{t_k \dots t_1} && \text{by the definition of } \star \text{ operation} \\
&= C_{t_{k+1} t_k} \star C_{t_k t_{k-1}} \star \dots \star C_{t_2 t_1} && \text{by the assumption for } n = k
\end{aligned}$$

□

Let $X = \{X_t, t \in T\}$ be a Markov process, the reversed process $\tilde{X}$ is just those of X with t reversed in time. Reversing the process $\{X_t, t \in T\}$ from a random time U , we get a process $\{\tilde{X}_t = X_{U-t}\}$.

Lemma 4.1.4. *The reverse of a Markov process is also a Markov process.*

Proof. The proof is immediate by Claim 4.1.2 and 4.1.3 in connection with Theorem 4.1.1. □

Follows from the above results, we can generate the bootstrap pseudo series backwards. If we have observed $x_1, x_2, \dots, x_n$, a sample path of a Markov chain. Then $x_n, x_{n-1}, \dots, x_1$ is a sample path of a Markov chain with densities $f_{bn}(x, y), f_{bn}(x)$ and $f_{bn}(y|x)$. And the estimators of these densities are as follows,

$$\hat{f}_{bn}(x, y) = \frac{1}{(n-p)h_1 h_2} \sum_{i=p+1}^n K\left(\frac{x - x_{i-p}}{h_1}, \frac{\|y - y_i\|}{h_2}\right) \quad (4.7)$$

$$\hat{f}_{bn}(y) = \int \hat{f}_{bn}(x, y) dx \quad (4.8)$$

$$\hat{f}_{bn}(x|y) = \frac{\hat{f}_{bn}(x, y)}{\hat{f}_{bn}(y)} \quad (4.9)$$

The algorithm of backward bootstrap based on transition density is very similar to that of forward bootstrap based on transition density. The only difference is in step(3). We generate the pseudo series $(x_1^*, \dots, x_n^*)$ backwards. The backward bootstrap algorithm can be performed as follows,

Algorithm 4.1.2.

(1)-(2) *Same as the steps in Algorithm 4.1.1.*

(3) (a) *Let $y_n^* = y_n$.*

(b) *Generate $x_{n-p}^* \sim \hat{f}_{bn}(\cdot | y_n^* = y_n)$*

(c) *Repeat (b) to generate $x_t^* \sim \hat{f}_{bn}(\cdot | y_{t+p}^*)$, $t = n - p, \dots, 1$.*

(d) *Generate bootstrap future value $x_{n+1}^* \sim \hat{f}_n(\cdot | y_n)$.*

(e) *Based on $x_1^*, \dots, x_n^*$, compute $\hat{f}_n^*(\cdot | y_n)$ as in (4.4). Then calculate $\hat{x}_{n+1}^*$ as $\int x \hat{f}_n^*(x | y_n) dx$.*

(f) *Calculate the bootstrap root replicate as $x_{n+1}^* - \hat{x}_{n+1}^*$*

(4)-(5) *Same as the steps in Algorithm 4.1.1.*

4.1.4 Asymptotic Properties

For simplicity, in this section we consider a strictly stationary Markov sequence X of order one. Suppose

(A1) $X = \{X_1, X_2, \dots\}$ forms an aperiodic, strictly stationary and geometrically ergodic Markov chain on $(\mathbb{R}, \mathcal{B})$. $\mathcal{B}$ is a Borel- σ algebra over $\mathbb{R}$.

The existence and uniqueness of the following functions are ensured by assumption (A1),

$$\begin{aligned} F(y) &= P[X_1 \leq y], \\ F(x, y) &= P[X_1 \leq y, X_2 \leq x], \\ F(x|y) &= P[X_2 \leq x | X_1 = y]. \end{aligned} \tag{4.10}$$

(A2) $F(y)$, $F(x, y)$ and $F(x|y)$ defined in (4.10) are absolutely continuous, and have uniformly continuous and bounded densities $f(y)$, $f(x, y)$ and $f(x|y)$ respectively. Assume a compact subset $S \subset \mathbb{R}$ exists such that $X_t \in S$ a.s. and $f(y) \geq \delta > 0$ for each $y \in S$.

(A3) The sequence $X = \{X_1, X_2, \dots\}$ is ϕ -mixing.

Let $k(x, y)$ be an appropriately chosen kernel on $\mathbb{R}^2$. The conditions of the kernel $k(x, y)$ would be specified later in assumption (B1-B3). Then the estimator of $f(x, y)$, $f(y)$ and $f(x|y)$ are given by

$$\hat{f}_n(x, y) = \frac{1}{(n-1)h^2} \sum_{i=1}^{n-1} k\left(\frac{x - X_{i+1}}{h}, \frac{y - X_i}{h}\right) \quad (4.11)$$

$$\hat{f}_n(y) = \int \hat{f}_n(x, y) dx \quad (4.12)$$

$$\hat{f}_n(x|y) = \frac{\hat{f}_n(x, y)}{\hat{f}_n(y)} \quad (4.13)$$

By comparison to eq.(4.2), here we let $h_1 = h_2 = h$ since the role of the two arguments of $k(\cdot, \cdot)$ is interchangeable.

Following are the assumptions for the choice of the kernel and the bandwidth.

(B1) $k(x, y) \rightarrow 0$ as $\|(x, y)\| \rightarrow \infty$, uniformly continuous in (x, y) . Here $\|\cdot\|$ is the Euclidean norm.

(B2) $k(x, y)$ is of bounded variation on $S \times S$

(B3) $h = h(n) \rightarrow 0$, $nh \rightarrow \infty$ and $\sum_1^\infty (nh^4)^{k+1} < \infty$, for some $k \geq 3$.

Under assumption (A1)-(A3) and (B1)-(B3), the following results are proved by Rajarsh(1990, Lemma 2.2, Corollary 2.1 and 2.2)

$$\sup_{x, y} |\hat{f}_n(x, y) - f(x, y)| \rightarrow 0 \quad \text{a.s.} \quad (4.14)$$

$$\sup_x |\hat{f}_n(x) - f(x)| \rightarrow 0 \quad \text{a.s.} \quad (4.15)$$

$$\sup_{x, y} |\hat{f}_n(y|x) - f(y|x)| \rightarrow 0 \quad \text{a.s.} \quad (4.16)$$

Now suppose we have observations $x_1, x_2, \dots, x_n$ then $\hat{f}_n(x, y), \hat{f}_n(x)$ and $\hat{f}_n(y|x)$ are determined. Then we generate the Markov-sequence $X^* = (X_1^*, X_2^*, \dots)$ with densities $\hat{f}_n(x)$ and $\hat{f}_n(y|x)$ as described in Algorithm 4.1.1. Based on $X_1^*, X_2^*, \dots, X_n^*$, we re-estimate the density functions as

$$\hat{f}_n^*(x, y) = \frac{1}{(n-1)h^2} \sum_{i=1}^{n-1} k\left(\frac{x - X_{i+1}^*}{h}, \frac{y - X_i^*}{h}\right) \quad (4.17)$$

$$\hat{f}_n^*(y) = \int \hat{f}_n^*(x, y) dx \quad (4.18)$$

$$\hat{f}_n^*(x|y) = \frac{\hat{f}_n^*(x, y)}{\hat{f}_n^*(y)} \quad (4.19)$$

Theorem 4.1.5. *Under assumptions (A1)-(A3) and (B1)-(B3), given observations $x_1, x_2, \dots, x_n$, we have*

$$\sup_{x,y} |\hat{f}_n^*(x, y) - \hat{f}_n(x, y)| \rightarrow 0 \text{ a.s.} \quad (4.20)$$

Proof. From Lemma 2.3 of Rajarshi(1990), it follows that X^* is also ϕ -mixing. Following the same argument from the proof of Theorem 1 part(B) in Rüschemdorf(1977), given observations $x_1, x_2, \dots, x_n$, we have

$$\sup_{x,y} |\hat{f}_n^*(x, y) - E\hat{f}_n^*(x, y)| \rightarrow 0 \text{ a.s.}$$

By eq. (7) of Nadaraya (1965), we have

$$\begin{aligned} & \sup_{x,y} |E\hat{f}_n^*(x, y) - \hat{f}_n(x, y)| \\ & \leq \sup_{x,y} \sup_{\|(x_1, y_1)\| < \eta} |\hat{f}_n(x - x_1, y - y_1) - \hat{f}_n(x, y)| + 2M \int_{\|(x_1, y_1)\| \geq \eta/h} K(x_1, y_1) dx_1 dx_2 \end{aligned} \quad (4.21)$$

where $M = \max_x \hat{f}_n(x)$. Note that

$$\begin{aligned} & \sup_{x,y} \sup_{\|(x_1, y_1)\| < \eta} |\hat{f}_n(x - x_1, y - y_1) - \hat{f}_n(x, y)| \\ & = \sup_{x,y} \sup_{\|(x_1, y_1)\| < \eta} |(\hat{f}_n(x - x_1, y - y_1) - f(x - x_1, y - y_1)) \\ & \quad + (f(x, y) - \hat{f}_n(x, y)) + (f(x - x_1, y - y_1) - f(x, y))| \\ & \leq \sup_{x,y} \sup_{\|(x_1, y_1)\| < \eta} |\hat{f}_n(x - x_1, y - y_1) - f(x - x_1, y - y_1)| \\ & \quad + \sup_{x,y} |f(x, y) - \hat{f}_n(x, y)| + \sup_{x,y} \sup_{\|(x_1, y_1)\| < \eta} |f(x - x_1, y - y_1) - f(x, y)| \end{aligned}$$

By (4.14) and the uniform continuity of $f(x, y)$, for any $\epsilon > 0$, choose η sufficiently small, we can make the first term on the right hand side of (4.21) $< \epsilon/2$. Having so chosen η , choose n so large that $\eta/h \rightarrow \infty$; then the second term of the right hand side of (4.21) is also $< \epsilon/2$. $\square$

Corollary 4.1.6. *Under the same assumptions of above theorem, given observations $x_1, x_2, \dots, x_n$ we have*

$$\sup_x |\hat{f}_n^*(x) - \hat{f}_n(x)| \rightarrow 0, \text{ a.s.} \quad (4.22)$$

$$\sup_{x,y} |\hat{f}_n^*(y|x) - \hat{f}_n(y|x)| \rightarrow 0, \text{ a.s.} \quad (4.23)$$

In the following theorems of this section, we let $X_{n+1} \sim f(\cdot|x_n)$, and $X_{n+1}^*|\{X_1, \dots, X_n\} \sim \hat{f}_n(\cdot|x_n)$, i.e.,

$$f(X_{n+1} = x) = f(x|x_n),$$

$$f(X_{n+1}^* = x|X_1, \dots, X_n) = \hat{f}_n(x|x_n),$$

and $\hat{f}_n(x|x_n)$ is given in (4.13).

We denote P^* as the probability given $X_1, \dots, X_n$ and E^* as the expectation given $X_1, \dots, X_n$.

Remark 4.1.2. *When we use P^* and E^* , we would not write the conditional random variables $X_1, \dots, X_n$ explicitly. For example, $P^*(X_{n+1}^* \leq x)$ in the following theorem means $P(X_{n+1}^* \leq x|X_1, \dots, X_n)$.*

Theorem 4.1.7. *Suppose the assumptions of Theorem 4.1.5 hold, then*

$$\sup_x |P(X_{n+1} \leq x) - P^*(X_{n+1}^* \leq x)| \rightarrow 0 \text{ a.s.}$$

Proof. Let $(x_1, \dots, x_n)$ be any realization of $X_1, \dots, X_n$. Given such a realization, we have

$$\begin{aligned}
& |P(X_{n+1} \leq x) - P^*(X_{n+1}^* \leq x)| \\
&= \left| \int_{-\infty}^x (f(u|x_n) - \hat{f}_n(u|x_n)) du \right| \\
&= \left| \int_x^\infty (f(u|x_n) - \hat{f}_n(u|x_n)) du \right| \quad \left(\text{since } \int f(u|x_n) du = \int \hat{f}_n(u|x_n) du = 1 \right) \\
&= \frac{1}{2} \left| \int_{-\infty}^x (f(u|x_n) - \hat{f}_n(u|x_n)) du \right| + \frac{1}{2} \left| \int_x^\infty (f(u|x_n) - \hat{f}_n(u|x_n)) du \right| \\
&\leq \int \frac{1}{2} |f(u|x_n) - \hat{f}_n(u|x_n)| du \\
&= \int (f(u|x_n) - \hat{f}_n(u|x_n))^+ du \\
&\quad \left(\text{since } \int (f(u|x_n) - \hat{f}_n(u|x_n))^+ du = \int (f(u|x_n) - \hat{f}_n(u|x_n))^- du, \right. \\
&\quad \left. \text{here } f^+ = \max(f, 0), f^- = -\min(f, 0) \right) \\
&\rightarrow \int \lim_{n \rightarrow \infty} (f(u|x_n) - \hat{f}_n(u|x_n))^+ du = 0
\end{aligned}$$

since $(f(u|x_n) - \hat{f}_n(u|x_n))^+ \leq f(u|x_n)$ and the last step follows by lemma 4.1.6 and the dominated convergence theorem. So $\sup_x |P(X_{n+1} \leq x) - P^*(X_{n+1}^* \leq x)| \rightarrow 0$ holds for any realization of $X_1, \dots, X_n$. $\square$

Let the predictor of future value be

$$\hat{X}_{n+1} = \int x \hat{f}_n(x|x_n) dx$$

and the bootstrap predictor be

$$\hat{X}_{n+1}^* = \int x \hat{f}_n^*(x|x_n) dx.$$

Note that $\hat{X}_{n+1}$ is a function of $X_1, \dots, X_n$ and x_n while $\hat{X}_{n+1}^*$ is a function of $X_1^*, \dots, X_n^*$ and x_n .

Theorem 4.1.8. *Suppose the assumptions of Theorem 4.1.7 are met,*

$$E\hat{X}_{n+1} - E^*\hat{X}_{n+1}^* \rightarrow 0 \text{ a.s.}$$

Proof. Let $(x_1, \dots, x_n)$ be any realization of $X_1, \dots, X_n$. Under each realization $(x_1, \dots, x_n)$, $\hat{f}_n(\cdot|\cdot)$ is a non-random function and $E\hat{X}_{n+1} = \int x\hat{f}_n(x|x_n)dx$. We generate $x_1^*, \dots, x_n^*$, where x_t^* is sampled from $\hat{f}_n(\cdot|x_{t-1}^*)$. Then we have non-random function $\hat{f}_n^*(u|x_n)$ based on $x_1^*, \dots, x_n^*$ and $E^*\hat{X}_{n+1}^* = \int x\hat{f}_n^*(x|x_n)dx$.

Let $\hat{F}_n(y|x_n) = \int_{-\infty}^y \hat{f}_n(u|x_n)du$ and $\hat{F}_n^*(y|x_n) = \int_{-\infty}^y \hat{f}_n^*(u|x_n)du$, follow the argument of theorem 4.1.7,

$$\begin{aligned} & |\hat{F}_n(y|x_n) - \hat{F}_n^*(y|x_n)| \\ & \leq \int (\hat{f}_n(u|x_n) - \hat{f}_n^*(u|x_n))^+ du \\ & \rightarrow \int \lim_{n \rightarrow \infty} (\hat{f}_n(u|x_n) - \hat{f}_n^*(u|x_n))^+ = 0 \end{aligned}$$

Note that $(\hat{f}_n(u|x_n) - \hat{f}_n^*(u|x_n))^+ \leq \hat{f}_n(u|x_n) \rightarrow f(u|x_n)$. Apply Fatou to both $\hat{f}_n(u|x_n) - (\hat{f}_n(u|x_n) - \hat{f}_n^*(u|x_n))^+$ and $\hat{f}_n(u|x_n) + (\hat{f}_n(u|x_n) - \hat{f}_n^*(u|x_n))^+$, then apply dominant convergence theorem to $\hat{f}_n(u|x_n)$, we have

$\lim_{n \rightarrow \infty} \int (\hat{f}_n(u|x_n) - \hat{f}_n^*(u|x_n))^+ du = \int \lim_{n \rightarrow \infty} (\hat{f}_n(u|x_n) - \hat{f}_n^*(u|x_n))^+ du$. And the last step follows by Corollary 4.1.6.

Given any realization of the data $(x_1, \dots, x_n)$ and $(x_1^*, \dots, x_n^*)$, let U_n be a random variable with density function $\hat{f}_n(\cdot|x_n)$ and V_n be a random variable with density $\hat{f}_n^*(\cdot|x_n)$. Because $X_t \in S$ a.s, S is compact and the assumption (B1)-(B2) of kernel, U_n and V_n have same limit distribution when n is large, which implies $EU_n - EV_n \rightarrow 0$. So $E\hat{X}_{n+1} - E^*\hat{X}_{n+1}^* \rightarrow 0$ holds for any realization of $(X_1, \dots, X_n)$. $\square$

Theorem 4.1.9. *Under assumption (A1)-(A3) and (B1)-(B3), The prediction interval (4.5) from Algorithm 4.1.1 is asymptotic valid.*

Proof.

$$\begin{aligned} P(X_{n+1} - \hat{X}_{n+1} \leq x) &= \int P(X_{n+1} \leq x + z) d_{F_{\hat{X}_{n+1}}}(z) \\ &= P(X_{n+1} \leq x) + f(x|x_n)E\hat{X}_{n+1} + o(E\hat{X}_{n+1}) \end{aligned}$$

Similarly

$$P^*(X_{n+1}^* - \hat{X}_{n+1}^* \leq x) = P^*(X_{n+1}^* \leq x) + \hat{f}_n(x|x_n)E^*\hat{X}_{n+1}^* + o(E^*\hat{X}_{n+1}^*)$$

The rest of the proof follows by Theorem 4.1.7, Theorem 4.1.8 and Corollary 4.1.6 $\square$

Remark 4.1.3. *The asymptotic validity for the backward method follows similarly since the reverse of a Markov process is also a Markov process.*

4.1.5 Simulation Results

In the Monte Carlo study, we simulate 500 samples each of size 50 or 100 from the following Markov models.

- Model 1: $X_{t+1} = \sin(X_t) + \epsilon_{t+1}$,
- Model 2: $X_{t+1} = 0.8 \log(3X_t^2 + 1) + \epsilon_{t+1}$,
- Model 3: $X_{t+1} = -0.5 \exp(-50X_t^2)X_t + \epsilon_{t+1}$,
- Model 4: $X_{t+1} = \sin(X_t) + \sqrt{0.5 + 0.25X_t^2}\epsilon_{t+1}$,
- Model 5: Bilinear model $X_{t+1} = 0.75X_t + 0.15X_t\epsilon_{t+1} + \epsilon_{t+1}$,

where $\{\epsilon_t\}$ is a sequence of independent $N(0, 1)$ variables.

In the bootstrap procedure, we choose $B=250$. We check the performance of the bootstrap methods based on coverage level and interval length defined in section 1.5. And $k(x, y) = k_1(x)k_1(y)$, where $k_1(x)$ is the standard normal density. As suggested in remark 4.1.1 We choose h to be $0.9An^{-1/4}$.

Table 4.1: bootstrap based on transition density with i.i.d. errors-model 1

normal innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.931	4.126	0.760	0.886	3.544	0.667
Backward	0.931	4.130	0.757	0.887	3.555	0.677
$n = 100$						
Forward	0.942	4.137	0.627	0.901	3.535	0.519
Backward	0.942	4.143	0.621	0.900	3.531	0.519
$n = 200$						
Forward	0.944	4.061	0.501	0.902	3.472	0.415
Backward	0.944	4.058	0.507	0.902	3.470	0.424
Laplace innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.913	4.072	1.033	0.873	3.429	0.894
Backward	0.913	4.081	1.021	0.873	3.441	0.897
$n = 100$						
Forward	0.925	4.236	1.027	0.885	3.424	0.763
Backward	0.926	4.239	1.024	0.885	3.437	0.764
$n = 200$						
Forward	0.928	4.184	0.914	0.884	3.307	0.619
Backward	0.929	4.202	0.904	0.883	3.299	0.619

Table 4.2: bootstrap based on transition density with i.i.d. errors-model 2

normal innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.936	4.270	0.798	0.893	3.658	0.720
Backward	0.937	4.278	0.812	0.893	3.667	0.723
$n = 100$						
Forward	0.948	4.219	0.608	0.906	3.600	0.526
Backward	0.948	4.225	0.612	0.905	3.594	0.525
$n = 200$						
Forward	0.949	4.151	0.512	0.907	3.533	0.448
Backward	0.948	4.147	0.529	0.907	3.530	0.446
Laplace innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.919	4.252	1.166	0.880	3.565	0.991
Backward	0.919	4.250	1.153	0.879	3.571	1.014
$n = 100$						
Forward	0.932	4.372	1.071	0.893	3.549	0.797
Backward	0.933	4.396	1.080	0.895	3.573	0.799
$n = 200$						
Forward	0.935	4.336	0.943	0.893	3.454	0.664
Backward	0.935	4.337	0.963	0.893	3.454	0.674

Table 4.3: bootstrap based on transition density with i.i.d. errors-model 3

normal innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.910	3.821	0.716	0.862	3.310	0.638
Backward	0.911	3.840	0.706	0.862	3.316	0.628
$n = 100$						
Forward	0.929	3.933	0.588	0.883	3.373	0.506
Backward	0.928	3.934	0.589	0.882	3.366	0.505
$n = 200$						
Forward	0.934	3.938	0.528	0.888	3.345	0.442
Backward	0.934	3.938	0.537	0.887	3.339	0.443
Laplace innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.881	3.791	1.207	0.835	3.186	1.032
Backward	0.884	3.827	1.225	0.838	3.208	1.033
$n = 100$						
Forward	0.911	4.102	1.114	0.867	3.336	0.864
Backward	0.912	4.109	1.110	0.867	3.341	0.865
$n = 200$						
Forward	0.916	4.060	1.038	0.870	3.224	0.739
Backward	0.917	4.069	1.040	0.871	3.227	0.726

Table 4.4: bootstrap based on transition density with i.i.d. errors-model 4

normal innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.927	3.477	0.802	0.882	2.983	0.716
Backward	0.928	3.495	0.832	0.882	2.992	0.722
$n = 100$						
Forward	0.943	3.421	0.629	0.901	2.928	0.561
Backward	0.943	3.439	0.648	0.901	2.930	0.573
$n = 200$						
Forward	0.943	3.428	0.627	0.901	2.921	0.548
Backward	0.943	3.430	0.633	0.901	2.921	0.552
Laplace innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.910	3.359	0.964	0.870	2.828	0.847
Backward	0.911	3.369	0.953	0.871	2.839	0.841
$n = 100$						
Forward	0.926	3.518	0.952	0.887	2.867	0.779
Backward	0.926	3.526	0.963	0.886	2.872	0.789
$n = 200$						
Forward	0.932	3.492	0.915	0.890	2.783	0.714
Backward	0.932	3.494	0.920	0.890	2.780	0.721

Table 4.5: bootstrap based on transition density with i.i.d. errors-model 5

normal innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.941	4.286	0.987	0.902	3.670	0.870
Backward	0.940	4.284	0.993	0.901	3.666	0.870
$n = 100$						
Forward	0.950	4.203	0.885	0.913	3.584	0.747
Backward	0.949	4.198	0.900	0.912	3.586	0.769
$n = 200$						
Forward	0.954	4.109	0.815	0.916	3.502	0.689
Backward	0.954	4.101	0.810	0.916	3.505	0.685
Laplace innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.926	4.329	1.368	0.890	3.613	1.171
Backward	0.925	4.304	1.353	0.889	3.599	1.164
$n = 100$						
Forward	0.936	4.332	1.307	0.899	3.533	1.049
Backward	0.936	4.330	1.258	0.899	3.531	1.026
$n = 200$						
Forward	0.935	4.191	1.053	0.895	3.375	0.809
Backward	0.935	4.180	1.047	0.895	3.375	0.808

From the above tables, we see that

- The coverage level improved when we increase the sample size in all the cases.
- The standard deviation of the interval length decreased when $n = 100$ except for the 95% prediction interval of model 4 with Laplace error using backward method. However, after further investigation of this case, we notice that the inter quantile range decreased when $n = 100$ and the large standard deviation is due to an outlier in the 500 backward interval lengths we collected.
- The forward and backward methods have similar performances.

4.2 Local Bootstrap

Paparoditis and Politis(2002) proposed the local bootstrap for Markov processes. Let $X = \{X_1, X_2, \dots\}$ be a Markov sequence of order p . Suppose assumptions (A1)-(A3) and (B1)-(B2) of Paparoditis and Politis(2002) are satisfied. All the assumptions mentioned in this section are referred to the assumptions in Paparoditis and Politis(2002). Let $x_1, \dots, x_n$ be the observed sample path or a Markov chain of order p . Let $Y_t = (X_t, X_{t-1}, \dots, X_{t-p+1})'$, $y_t = (x_t, x_{t-1}, \dots, x_{t-p+1})'$. Following the idea of Paparoditis and Politis(2002), we have the estimator of the one-step ahead transition probability given by

$$\hat{P}(X_{t+1} = x_{j+1} | Y_t = y_t) = \frac{W_b(y_t - y_j)}{\sum_{m=p}^{n-1} W_b(y_t - y_m)}, \quad (4.24)$$

$j \in \{p, \dots, n-1\}$, where $W_b(\cdot) = 1/bW(\cdot/b)$ and $W(\cdot)$ is a bounded, Lipschitz continuous and symmetric probability density kernel in $\mathbb{R}^p$ satisfying assumption (B1); as in Paparoditis and Politis(2002), the bandwidth satisfies the assumption (B2).

Then the estimator of the transition distribution is as follows,

$$\begin{aligned} \hat{F}_n(x|y) &= \sum_{j=p}^{n-1} 1_{(-\infty, x]}(x_{j+1}) \hat{P}(X_{t+1} = x_{j+1} | Y_t = y) \\ &= \frac{\sum_{j=p}^{n-1} 1_{(-\infty, x]}(x_{j+1}) W_b(y - y_j)}{\sum_{m=p}^{n-1} W_b(y - y_m)}. \end{aligned} \quad (4.25)$$

The process of generating the bootstrap pseudo data is based on the conditional distribution as above.

4.2.1 Forward Local Bootstrap

To construct the bootstrap prediction interval, we estimate the distribution of the future value conditioned on the last p observations. Applying the same technique as before, we fix the last p pseudo values to be the true observations in generating the bootstrap replicates of bootstrap predictive root. The approach for the local bootstrap can be described by the following steps:

Algorithm 4.2.1.

(1) Choose a resampling kernel W and select bandwidth b through cross validation.

Then we calculate the predictor $\hat{x}_{n+1}$ as

$$\frac{\sum_{j=p}^{n-1} W_b(y_n - y_j) x_{j+1}}{\sum_{m=p}^{n-1} W_b(y_n - y_m)}.$$

(2) (a) Set start value y_p^* to any consecutive p values from $\{x_1, \dots, x_n\}$.

(b) Suppose $x_1^*, \dots, x_t^*$ for $t \geq p$ have already been generated. Let J be a discrete random variable taking its values in the set $\{p, \dots, n-1\}$ with probability mass function given by

$$P(J = s) = \frac{W_b(y_t^* - y_s)}{\sum_{m=p}^{n-1} W_b(y_t^* - y_m)},$$

Then let $x_{t+1}^* = x_{J+1}$.

(c) Repeat (b) to generate $x_{p+1}^*, \dots, x_n^*$.

(d) Calculate the bootstrap predictor $\hat{x}_{n+1}^*$ as

$$\frac{\sum_{j=p}^{n-1} W_b(y_n - y_j^*) x_{j+1}^*}{\sum_{m=p}^{n-1} W_b(y_n - y_m^*)},$$

where $y_t^* = (x_t^*, \dots, x_{t-p+1}^*)'$.

(e) Redefine $y_n^* = y_n$, and then generate $x_{n+1}^* = x_{J+1}$ as in step (b), where J is a discrete random variable taking its values in the set $\{p, \dots, n-1\}$ with probability mass function given by

$$P(J = s) = \frac{W_b(y_n - y_s)}{\sum_{m=p}^{n-1} W_b(y_n - y_m)},$$

(f) Calculate the bootstrap prediction root replicate as $x_{n+1}^* - \hat{x}_{n+1}^*$

(3) Repeat step (2) B times and 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 predictive interval for X_{n+1} is given by

$$[\hat{x}_{n+1} + q(\alpha/2), \hat{x}_{n+1} + q(1 - \alpha/2)].$$

4.2.2 Backward Local Bootstrap

As mentioned in Lemma 4.1.4, the reverse of a Markov process is also a Markov process. So in this section we will discuss the backward local bootstrap using backward transition kernel given by

$$\hat{P}_b(X_t = x_j | Y_{t+p} = y_{t+p}) = \frac{W_b(y_{t+p} - y_{j+p})}{\sum_{m=1}^{n-p} W_b(y_{t+p} - y_{m+p})}, \quad (4.26)$$

$j \in \{1, 2, \dots, n-p\}$, and the backward conditional distribution is

$$\begin{aligned} \hat{F}_{bn}(x|y) &= \sum_{j=1}^{n-p} 1_{(-\infty, x]}(x_j) \hat{P}_b(X_j = x_j | Y_{j+p} = y) \\ &= \frac{\sum_{j=1}^{n-p} 1_{(-\infty, x]}(x_j) W_b(y - y_{j+p})}{\sum_{m=1}^{n-p} W_b(y - y_{m+p})}. \end{aligned} \quad (4.27)$$

This backward local bootstrap resampling scheme has been used by Paparoditis and Politis(1998) in a first order autoregressive process with ARCH errors as follows,

$$X_t = \phi X_{t_1} + \sqrt{\alpha_0 + \alpha_1 X_{t-1}^2} \epsilon_t,$$

where $\{\epsilon_t\}$ are i.i.d with mean 0 and variance 1.

In the Markov case, the algorithm for backward local bootstrap can be described as following steps:

Algorithm 4.2.2.

- (1) Same as the forward local bootstrap in Algorithm 4.2.1.
- (2) (a) Set start value $y_n^* = y_n$.
 (b) Suppose y_{t+p}^* have already been generated. Let J be a discrete random variable taking its values in the set $\{1, 2, \dots, n-p\}$ with probability mass function given by

$$P(J = s) = \frac{W_{b'}(y_{t+p}^* - y_{s+p})}{\sum_{m=1}^{n-p} W_{b'}(y_{t+p}^* - y_{m+p})},$$

where b' is the backward bandwidth. Then let $x_t^* = x_J$

(c) Repeat (b) to generate $x_{n-p}^*, \dots, x_2^*, x_1^*$ backwards in time.

(d) Let J be a discrete random variable taking its values in the set $\{p, p+1, \dots, n-1\}$ with probability mass function given by

$$P(J = s) = \frac{W_b(y_n - y_s)}{\sum_{m=p}^{n-1} W_b(y_n - y_m)},$$

Then let $x_{n+1}^* = x_{J+1}$

(e) Calculate the bootstrap predictor $\hat{x}_{n+1}^*$ by

$$\frac{\sum_{j=p}^{n-1} W_b(y_n - y_j^*) x_{j+1}^*}{\sum_{m=p}^{n-1} W_b(y_n - y_m^*)}$$

(f) Calculate the bootstrap prediction root replicate as $x_{n+1}^* - \hat{x}_{n+1}^*$

(3)-(4) Same as the steps of forward local bootstrap in Algorithm 4.2.1.

4.2.3 Asymptotic Properties

Paparoditis and Politis(2002) proved the following theorem of the estimator of the transition distribution $\tilde{F}(y|x)$ defined in (4.25).

Theorem 4.2.1. [Paparoditis and Politis(2002)] Under assumptions (A1)-(A3) and (B1)-(B2) we have as $n \rightarrow \infty$

$$\sup_{x,y} |\hat{F}_n(y|x) - F(y|x)| \rightarrow 0 \text{ a.s.}$$

Theorem 4.2.2. Under the same assumptions as in Theorem 4.2.1, we have as $n \rightarrow \infty$

$$\sup_{x,y} |\hat{F}_{bn}(y|x) - F(y|x)| \rightarrow 0 \text{ a.s.}$$

Proof. Proof follows from the result showed in Lemma 4.1.4 and Theorem 4.2.1. $\square$

4.2.4 Simulation Results

In this section, some simulated sample paths generated by the following models are used to demonstrate the performance of the bootstrap prediction interval.

- Model 1: $X_{t+1} = \sin(X_t) + \epsilon_{t+1}$
- Model 2: $X_{t+1} = 0.8 \log(3X_t^2 + 1) + \epsilon_{t+1}$
- Model 3: $X_{t+1} = -0.5 \exp(-50X_t^2)X_t + \epsilon_{t+1}$
- Model 4: $X_{t+1} = \sin(X_t) + \sqrt{0.5 + 0.25X_t^2}\epsilon_{t+1}$
- Model 5: $X_{t+1} = 0.75X_t + 0.15X_t\epsilon_{t+1} + \epsilon_{t+1}$

Where the errors $\{\epsilon_t\}$ are chosen from i.i.d. $N(0,1)$ distribution or i.i.d. Laplace distribution rescaled to variance 1. Above 5 models are the same as those in Section 4.1. We choose sample size $n=50, 100$ and 200 , construct the 95% and 90% prediction interval, and repeat the experiment 500 times. The kernel W is the normal kernel. And the forward and backward bandwidth b and b' are chosen by cross validation method.

Table 4.6: local bootstrap with i.i.d. errors-model 1

normal innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.910	3.885	0.778	0.862	3.337	0.685
Backward	0.911	3.920	0.795	0.863	3.355	0.676
$n = 100$						
Forward	0.930	3.980	0.625	0.886	3.409	0.511
Backward	0.932	4.001	0.605	0.886	3.411	0.508
$n = 200$						
Forward	0.937	3.968	0.530	0.891	3.369	0.439
Backward	0.937	3.979	0.551	0.893	3.383	0.448
Laplace innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.902	4.036	1.138	0.856	3.313	0.935
Backward	0.905	4.046	1.103	0.861	3.324	0.847
$n = 100$						
Forward	0.923	4.153	0.935	0.878	3.323	0.714
Backward	0.923	4.189	0.986	0.879	3.356	0.724
$n = 200$						
Forward	0.928	4.140	0.838	0.883	3.274	0.586
Backward	0.929	4.142	0.850	0.884	3.298	0.610

Table 4.7: local bootstrap with i.i.d. errors-model 2

normal innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.908	3.982	0.869	0.862	3.426	0.761
Backward	0.912	4.016	0.863	0.865	3.446	0.747
$n = 100$						
Forward	0.932	4.016	0.642	0.887	3.437	0.557
Backward	0.932	4.031	0.648	0.888	3.445	0.574
$n = 200$						
Forward	0.936	4.008	0.585	0.890	3.387	0.473
Backward	0.936	3.993	0.559	0.891	3.387	0.470
Laplace innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.895	4.021	1.263	0.854	3.360	1.021
Backward	0.900	4.004	1.188	0.856	3.339	0.982
$n = 100$						
Forward	0.920	4.179	1.041	0.877	3.375	0.813
Backward	0.922	4.239	1.081	0.880	3.417	0.830
$n = 200$						
Forward	0.931	4.259	0.947	0.886	3.385	0.740
Backward	0.930	4.228	0.929	0.883	3.346	0.663

Table 4.8: local bootstrap with i.i.d. errors-model 3

normal innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.912	3.744	0.648	0.862	3.200	0.534
Backward	0.913	3.771	0.649	0.863	3.220	0.525
$n = 100$						
Forward	0.927	3.818	0.499	0.879	3.248	0.399
Backward	0.928	3.828	0.487	0.879	3.256	0.391
$n = 200$						
Forward	0.935	3.846	0.404	0.885	3.246	0.304
Backward	0.934	3.832	0.386	0.885	3.253	0.308
Laplace innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.910	4.090	1.178	0.858	3.182	0.788
Backward	0.910	4.035	1.103	0.858	3.172	0.766
$n = 100$						
Forward	0.929	4.162	0.819	0.878	3.229	0.597
Backward	0.928	4.156	0.822	0.881	3.261	0.597
$n = 200$						
Forward	0.931	4.055	0.658	0.881	3.161	0.449
Backward	0.932	4.081	0.647	0.883	3.203	0.467

Table 4.9: local bootstrap with heteroscedastic errors-model 4

normal innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.914	3.424	0.833	0.867	2.893	0.717
Backward	0.914	3.451	0.803	0.867	2.924	0.692
$n = 100$						
Forward	0.938	3.425	0.628	0.895	2.908	0.553
Backward	0.937	3.410	0.616	0.894	2.903	0.549
$n = 200$						
Forward	0.942	3.425	0.578	0.898	2.894	0.483
Backward	0.941	3.406	0.562	0.895	2.858	0.462
Laplace innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.904	3.423	1.029	0.864	2.809	0.818
Backward	0.907	3.433	0.983	0.864	2.798	0.766
$n = 100$						
Forward	0.924	3.482	0.839	0.877	2.762	0.654
Backward	0.924	3.531	0.871	0.881	2.810	0.676
$n = 200$						
Forward	0.932	3.425	0.717	0.888	2.724	0.580
Backward	0.933	3.477	0.735	0.888	2.751	0.592

Table 4.10: local bootstrap with heteroscedastic errors-model 5

normal innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.914	3.967	1.039	0.870	3.407	0.907
Backward	0.916	6.993	0.991	0.871	3.406	0.879
$n = 100$						
Forward	0.936	3.996	0.910	0.893	3.405	0.766
Backward	0.937	4.021	0.944	0.896	3.442	0.787
$n = 200$						
Forward	0.945	3.980	0.832	0.903	3.395	0.697
Backward	0.944	3.969	0.813	0.901	3.377	0.694
Laplace innovation	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	std.err	CVR	LEN	std.err
Forward	0.905	4.144	1.410	0.864	3.429	1.168
Backward	0.910	4.177	1.371	0.869	3.454	1.178
$n = 100$						
Forward	0.926	4.196	1.269	0.883	3.364	1.014
Backward	0.926	4.173	1.240	0.885	3.385	1.045
$n = 200$						
Forward	0.930	4.073	0.967	0.887	3.272	0.765
Backward	0.930	4.098	1.080	0.886	3.292	0.839

The above 5 tables show the simulation results of the corresponding models. We see that the backward and forward methods have similar performances in both normal and Laplace cases. Moreover, when n is increased, the coverage level improved and the standard deviation of the interval length decreased.

Compare these simulation results with those from the bootstrap based on transition density, we observe that the bootstrap based on transition density tend to have better coverage level in most models (except for model 3) while the standard deviation corresponding to the interval lengths of both bootstrap methods are about the same.

Compare all the simulation results above from Chapter 4 to those of Chapter 3, for models with i.i.d errors, the methods in Chapter 3, especially the nonparametric methods with predictive residuals, have better performance than those methods in Chapter 4 since the nonparametric methods have significantly smaller standard de-

viation corresponding to the interval length. For models with heteroscedastic errors, the methods in Chapter 4 have better coverage level and smaller variability compared to those in Chapter 3.

Remark 4.2.1. *The local bootstrap is computationally faster compared to the bootstrap based on transition density because the point mass density function (4.24) we used in local bootstrap is a discrete function while the transition density function (4.4) is a continuous function.*

4.3 Hybrid Backward Markov Bootstrap for Nonparametric Autoregression

In this section, we will apply the backward bootstrap based on the transition density and the backward local bootstrap to the nonparametric autoregression model. In chapter 1, we applied a backward bootstrap method for a linear autoregression model based on a backward representation:

$$\phi(B^{-1})X_t = \phi_0 + \frac{\phi(B^{-1})}{\phi(B)}\epsilon_t.$$

However, as mentioned in the last section of chapter 2, it is hard to find the explicit backward representation for a nonparametric model. Actually, the explicit calculation of the backward conditional distribution $X_{t+1}|X_t$ is difficult even it is a parametric model. For example, a first order autoregression process with ARCH(1) error: $X_t = \phi X_{t-1} + \sqrt{\alpha_0 + \alpha_1 X_{t-1}^2} \epsilon_t$, cf. Paparoditis and Politis(1998). The methods introduced in this chapter give us a bootstrap-based method to estimate the backward conditional distribution. So we will use these methods to generate the pseudo data and form the backward bootstrap based on transition density and backward local bootstrap for nonparametric autoregression with i.i.d innovations.

4.3.1 Algorithms

We consider Markov Processes of the form (3.1) and (3.2) given in Chapter 2. Given a sample $\{x_1, x_2, \dots, x_n\}$, we give the algorithms of hybrid backward Markov bootstrap based on transition density for nonparametric model with i.i.d errors and heteroscedastic errors in Algorithm 4.3.1 and 4.3.2 respectively, and the corresponding algorithms based on local bootstrap in Algorithm 4.3.3 and 4.3.4.

Algorithm 4.3.1. *Hybrid backward Backward Markov bootstrap based on transition density for nonparametric model with i.i.d errors*

- (1) Select the bandwidth h and construct the estimates $\{\hat{m}(y_i)\}$, for $i = p, \dots, n-1$, where

$$\hat{m}(y_i) = \frac{\sum_{t=p}^{n-1} K(\frac{\|y_i - y_t\|}{h}) x_{t+1}}{\sum_{t=p}^{n-1} K(\frac{\|y_i - y_t\|}{h})},$$

and $y_t = (x_t, x_{t-1}, \dots, x_{t-p+1})'$, $\|\cdot\|$ is a norm in $\mathbb{R}^p$.

- (2) Compute the residuals: $\hat{\epsilon}_i = x_i - \hat{m}(y_{i-1})$, for $i = p+1, \dots, n$.
- (3) Center the residuals: $\hat{r}_i = \hat{\epsilon}_i - (n-p)^{-1} \sum_{t=p+1}^n \hat{\epsilon}_t$, for $i = p+1, \dots, n$; let the empirical distribution of $\hat{r}_t$ denoted by $\hat{F}_\epsilon$.

- (a) Select a resampling kernel and the proper bandwidth and Construct the backward transition density $\hat{f}_b$ as follows,

$$\begin{aligned} \hat{f}_b(x, y) &= \frac{1}{(n-1)b_1b_2} \sum_{i=p}^{n-1} k\left(\frac{x - x_{i+1}}{b_1}, \frac{\|y - y_i\|}{b_2}\right) \\ \hat{f}_b(y) &= \int \hat{f}_b(x, y) dx \\ \hat{f}_b(x|y) &= \frac{\hat{f}_b(x, y)}{\hat{f}_b(y)} \end{aligned}$$

- (b) Let $y_n^* = y_n$.

- (c) Generate $x_{n-p}^* \sim \hat{f}_b(\cdot | y_n^*)$. Repeat it to generate $x_t^* \sim \hat{f}_b(\cdot | y_{t+p}^*)$, $t = n-p, \dots, 1$.

(d) Compute the future bootstrap observation X_{n+1}^* by formula

$$X_{n+1}^* = \hat{m}(y_n^*) + \epsilon_{n+1}^* = \hat{m}(y_n) + \epsilon_{n+1}^*,$$

where ϵ_{n+1}^* is generated from $\hat{F}_\epsilon$. Then re-estimate $m(\cdot)$ based on the pseudo data, i.e.,

$$\hat{m}^*(y) = \frac{\sum_{i=p}^{n-1} K\left(\frac{\|y-y_i^*\|}{h}\right) x_{i+1}^*}{\sum_{i=p}^{n-1} K\left(\frac{\|y-y_i^*\|}{h}\right)}.$$

Then we let $\hat{X}_{n+1}^* = \hat{m}^*(y_n^*) = \hat{m}^*(y_n)$

(e) Calculate the bootstrap root replicate as $X_{n+1}^* - \hat{X}_{n+1}^*$.

(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 X_{n+1} is given by

$$[\hat{m}(x_n) + q(\alpha/2), \hat{m}(x_n) + q(1 - \alpha/2)]$$

Algorithm 4.3.2. Hybrid backward Markov bootstrap based on transition density for nonparametric model with heteroscedastic errors

(1) Select the bandwidth h and construct the estimates $\{\hat{m}(y_i), \hat{\sigma}(y_i)\}$, for $i = p, \dots, n-1$, where

$$\hat{m}(y_i) = \frac{\sum_{t=p}^{n-1} K\left(\frac{\|y_i - y_t\|}{h}\right) x_{t+1}}{\sum_{t=p}^{n-1} K\left(\frac{\|y_i - y_t\|}{h}\right)},$$

$$\hat{\sigma}^2(y_i) = \frac{\sum_{t=1}^{n-1} K\left(\frac{\|y_i - y_t\|}{h}\right) (x_{t+1} - \hat{m}(y_t))^2}{\sum_{t=1}^{n-1} K\left(\frac{\|y_i - y_t\|}{h}\right)},$$

and $y_t = (x_t, x_{t-1}, \dots, x_{t-p+1})'$, $\|\cdot\|$ is a norm in $\mathbb{R}^p$.

(2) Compute the residuals:

$$\hat{\epsilon}_i = \frac{x_i - \hat{m}(y_{i-1})}{\hat{\sigma}(y_{i-1})},$$

for $i = p+1, \dots, n$.

(3) This step is similar as step (3) in Algorithm *mkvback*, the only difference is in step (d) when computing the future bootstrap observation X_{n+1}^* :

$$X_{n+1}^* = \hat{m}_g(y_n^*) + \hat{\sigma}_g(y_n^*)\epsilon_{n+1}^* = \hat{m}_g(y_n) + \hat{\sigma}_g(y_n)\epsilon_{n+1}^*.$$

Here $\hat{m}_g$ and $\hat{\sigma}_g$ are over-smoothed estimates of m and g computed in the same way as $\hat{m}$ and $\hat{\sigma}$ but using bandwidth g much bigger than h .

(4)-(5) Same as those in Algorithm 4.3.1.

Given a sample $\{x_1, x_2, \dots, x_n\}$, the algorithms for hybrid backward local bootstrap with fitted residuals for nonparametric models are similar to the algorithms for the hybrid Markov bootstrap based on the backward transition density as given in Algorithm 4.3.1 and 4.3.2. The only difference is in generating the pseudo data $x_n^*, \dots, x_1^*$ backwards in time.

Algorithm 4.3.3. *Hybrid backward local bootstrap for nonparametric model with i.i.d errors*

By changing the step 3 (a) to (c) of Algorithm 4.3.1 into the following steps, we have the algorithm for backward local bootstrap with fitted residuals for nonparametric models.

(a)' Select a resampling bandwidth b and kernel W .

(b)' Let $y_n^* = y_n$ Suppose y_{t+p}^* have already been generated. Let J be a discrete random variable taking its values in the set $\{1, 2, \dots, n-p\}$ with probability mass function given by

$$P(J = s) = \frac{W_b(y_{t+p}^* - y_{s+p})}{\sum_{m=1}^{n-p} W_b(y_{t+p}^* - y_{m+p})},$$

Then $x_t^* = x_J$

(c)' Repeat (b) to generate $x_{n-p}^*, \dots, x_2^*, x_1^*$

Algorithm 4.3.4. *Hybrid backward local bootstrap for nonparametric model with heteroscedastic errors*

(1)-(2) Same as the corresponding steps in Algorithm 4.3.2

(3)

(a)-(c) Same as (a)' to (c)' in Algorithm 4.3.3

(d)-(e) Same as the corresponding steps in Algorithm 4.3.2

(4)-(5) Same as the corresponding steps in Algorithm 4.3.2

Remark 4.3.1. The hybrid algorithms for the backward bootstrap based on transition density and the backward local bootstrap with predictive residuals for nonparametric models are similar as the above algorithms. We just need to change the fitted residuals $\{\hat{\epsilon}_i\}$ in step (2) of Algorithm 4.3.1 to 4.3.4 into the predictive residuals $\{\hat{\epsilon}_t^{(t)}\}$ as follows: Use the delete- $\{x_t\}$ data set, i.e., the dataset excluding the pair (y_{t-1}, x_t) , to compute the delete- t -th-observation kernel smoothing estimator $\hat{m}^{(t)}$ as described in step(2) of Algorithm 3.1.2. Then calculate the predictive residuals: $\hat{\epsilon}_t^{(t)} = x_t - \hat{m}_t^{(t)}(y_{t-1})$, $t = p + 1, \dots, n$.

4.3.2 Simulation

In this section, we do the Monte Carlo studies on 3 nonparametric models with i.i.d errors and 2 models with heteroscedastic error. The five models we choose are the same as those in Chapter 2 and in Section 4.1 and 4.2.

The estimation bandwidth h for nonparametric bootstrap and the resampling bandwidth b for local bootstrap are all selected by cross validation based on corresponding regression kernel estimators. Same as in Section 4.1, the resample bandwidth b for the backward bootstrap based on the transition density is given by $0.9An^{-1/4}$, where $A = \min(\hat{\sigma}, \frac{IQR}{1.34})$, $\hat{\sigma}$ is the estimated standard deviation of the data and IQR is the interquartile range. The over-smoothing bandwidth we use is $g = 2h$.

The following five tables show the simulation results based on 500 dataset using hybrid backward bootstrap based on transition density.

Table 4.11: Hybrid bootstrap based on transition density for nonlinear
autoregression-model 1

normal innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.908	3.822	0.522	0.852	3.230	0.432
predictive residuals	0.935	4.181	0.583	0.889	3.553	0.470
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.921	3.822	0.412	0.868	3.241	0.335
predictive residuals	0.936	4.045	0.430	0.889	3.441	0.341
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.932	3.838	0.359	0.880	3.238	0.290
predictive residuals	0.942	3.977	0.360	0.893	3.358	0.281
Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.905	4.070	0.925	0.848	3.174	0.633
predictive residuals	0.926	4.425	0.931	0.878	3.507	0.635
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.925	4.056	0.702	0.872	3.174	0.495
predictive residuals	0.939	4.370	0.748	0.891	3.418	0.513
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.931	4.041	0.509	0.880	3.172	0.388
predictive residuals	0.939	4.221	0.544	0.891	3.307	0.385

Table 4.12: Hybrid bootstrap based on transition density for nonlinear
autoregression-model 2

normal innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.901	3.862	0.557	0.845	3.265	0.464
predictive residuals	0.928	4.255	0.625	0.879	3.600	0.507
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.921	3.828	0.437	0.867	3.238	0.352
predictive residuals	0.940	4.089	0.435	0.894	3.476	0.345
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.931	3.841	0.354	0.880	3.253	0.290
predictive residuals	0.943	4.022	0.360	0.895	3.399	0.289
Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.909	4.068	0.878	0.848	3.201	0.635
predictive residuals	0.932	4.555	0.907	0.884	3.599	0.633
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.925	4.101	0.685	0.872	3.198	0.510
predictive residuals	0.938	4.410	0.731	0.891	3.430	0.515
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.930	4.021	0.522	0.878	3.158	0.387
predictive residuals	0.938	4.186	0.526	0.890	3.315	0.397

Table 4.13: Hybrid bootstrap based on transition density for nonlinear
autoregression-model 3

normal innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.912	3.749	0.538	0.860	3.169	0.415
predictive residuals	0.926	3.918	0.521	0.880	3.350	0.430
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.925	3.769	0.415	0.874	3.194	0.324
predictive residuals	0.934	3.882	0.421	0.886	3.300	0.331
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.936	3.818	0.321	0.886	3.238	0.263
predictive residuals	0.940	3.879	0.342	0.892	3.293	0.265
Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.917	4.058	0.904	0.862	3.156	0.661
predictive residuals	0.930	4.312	1.040	0.881	3.367	0.643
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.925	4.050	0.682	0.875	3.176	0.508
predictive residuals	0.932	4.231	0.760	0.883	3.294	0.519
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.935	4.081	0.539	0.885	3.171	0.372
predictive residuals	0.936	4.100	0.580	0.888	3.224	0.397

Table 4.14: Hybrid bootstrap based on transition density for nonlinear
autoregression-model 4

normal innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.865	3.169	1.212	0.812	2.721	1.050
predictive residuals	0.897	3.639	1.494	0.840	2.978	1.148
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.888	2.997	0.873	0.833	2.564	0.747
predictive residuals	0.914	3.301	0.980	0.855	3.726	0.792
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.892	2.991	0.789	0.836	2.541	0.652
predictive residuals	0.908	3.147	0.816	0.849	2.631	0.663
Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.866	3.209	1.562	0.814	2.595	1.195
predictive residuals	0.899	3.848	1.980	0.841	2.895	1.340
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.885	3.149	1.232	0.830	2.499	0.890
predictive residuals	0.910	3.576	1.468	0.849	2.707	0.990
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.894	2.998	1.006	0.837	2.386	0.768
predictive residuals	0.910	3.241	1.040	0.850	2.494	0.768

Table 4.15: Hybrid bootstrap based on transition density for nonlinear
autoregression-model 5

normal innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.904	4.121	1.467	0.860	3.552	1.289
predictive residuals	0.931	4.683	1.720	0.887	3.891	1.392
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.925	4.049	1.399	0.881	3.464	1.163
predictive residuals	0.944	4.416	1.496	0.900	3.686	1.237
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.935	4.019	1.309	0.889	3.423	1.115
predictive residuals	0.946	4.244	1.407	0.900	3.540	1.163
Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.905	4.441	2.343	0.864	3.603	1.798
predictive residuals	0.931	5.352	2.875	0.889	4.099	2.040
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.920	4.351	2.149	0.876	3.501	1.700
predictive residuals	0.938	5.020	3.051	0.893	3.805	1.988
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.926	4.154	1.485	0.880	3.322	1.165
predictive residuals	0.938	4.497	1.650	0.890	3.475	1.216

From the previous five tables, we observe that

- The method with predictive residuals outperforms the one with fitted residuals. Especially when sample size is not large enough, the method with predictive residuals significantly improve the coverage level.
- When we increase the sample size, all models have increased coverage levels and decreased variability of the interval lengths.

The following five tables show the simulation results from the Hybrid backward local bootstrap methods:

Table 4.16: Hybrid backward local bootstrap for nonlinear autoregression-model 1

normal innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.914	3.782	0.525	0.860	3.199	0.433
predictive residuals	0.938	4.136	0.592	0.892	3.496	0.463
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.923	3.815	0.430	0.869	3.226	0.343
predictive residuals	0.937	4.018	0.433	0.890	3.414	0.338
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.932	3.798	0.336	0.882	3.228	0.272
predictive residuals	0.942	3.958	0.338	0.895	3.356	0.265
Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.913	4.081	0.934	0.857	3.184	0.623
predictive residuals	0.929	4.447	0.973	0.882	3.498	0.655
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.927	4.094	0.687	0.876	3.202	0.493
predictive residuals	0.938	4.310	0.731	0.891	3.400	0.512
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.932	4.041	0.560	0.881	3.170	0.405
predictive residuals	0.940	4.204	0.552	0.892	3.302	0.395

Table 4.17: Hybrid backward local bootstrap for nonlinear autoregression-model 2

normal innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.916	3.836	0.537	0.861	3.231	0.448
predictive residuals	0.940	4.231	0.622	0.893	3.562	0.486
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.926	3.824	0.412	0.874	3.242	0.334
predictive residuals	0.943	4.073	0.439	0.895	3.438	0.327
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.930	3.815	0.342	0.879	3.237	0.276
predictive residuals	0.942	3.997	0.374	0.895	3.390	0.296
Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.913	4.045	0.923	0.859	3.175	0.617
predictive residuals	0.934	4.489	0.960	0.887	3.524	0.648
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.929	4.092	0.683	0.877	3.205	0.497
predictive residuals	0.940	4.359	0.718	0.895	3.426	0.526
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.931	4.016	0.564	0.880	3.151	0.393
predictive residuals	0.940	4.209	0.550	0.893	3.310	0.378

Table 4.18: Hybrid backward local bootstrap for nonlinear autoregression-model 3

normal innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.913	3.721	0.524	0.863	3.172	0.418
predictive residuals	0.929	3.935	0.531	0.882	3.334	0.424
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.928	3.798	0.419	0.877	3.217	0.324
predictive residuals	0.935	3.880	0.412	0.887	3.294	0.314
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.934	3.790	0.332	0.885	3.226	0.268
predictive residuals	0.940	3.882	0.337	0.892	3.294	0.267
Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.917	4.041	0.922	0.866	3.143	0.605
predictive residuals	0.290	4.263	0.971	0.880	3.298	0.615
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.928	4.090	0.677	0.877	3.189	0.502
predictive residuals	0.934	4.215	0.706	0.885	3.281	0.498
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.934	4.057	0.559	0.883	3.168	0.413
predictive residuals	0.937	4.128	0.559	0.888	3.231	0.401

Table 4.19: Hybrid backward local bootstrap for nonlinear autoregression-model 4

normal innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.862	3.146	1.256	0.808	2.686	1.070
predictive residuals	0.895	3.639	1.637	0.840	2.956	1.180
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.888	2.988	0.877	0.834	2.553	0.748
predictive residuals	0.916	3.301	0.996	0.858	2.727	0.796
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.892	2.953	0.763	0.837	2.520	0.643
predictive residuals	0.911	3.148	0.810	0.853	2.628	0.663
Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.866	3.194	1.674	0.814	2.572	1.245
predictive residuals	0.901	3.875	2.093	0.846	2.894	1.391
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.888	3.153	1.201	0.835	2.514	0.906
predictive residuals	0.913	3.580	1.470	0.854	2.699	1.026
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.897	3.006	0.961	0.841	2.397	0.765
predictive residuals	0.911	3.229	1.036	0.852	2.481	0.752

Table 4.20: Hybrid backward local bootstrap for nonlinear autoregression-model 5

normal innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.901	4.079	1.516	0.857	3.508	1.306
predictive residuals	0.929	4.636	1.756	0.884	3.832	1.403
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.924	4.005	1.393	0.881	3.437	1.186
predictive residuals	0.945	4.418	1.589	0.901	3.668	1.282
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.932	3.971	1.293	0.886	3.382	1.098
predictive residuals	0.944	4.214	1.378	0.900	3.534	1.155
Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.905	4.426	2.469	0.864	3.583	1.866
predictive residuals	0.932	5.423	3.217	0.889	4.073	2.223
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.920	4.338	2.091	0.877	3.470	1.612
predictive residuals	0.938	4.976	2.835	0.893	3.759	1.941
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
fitted residuals	0.927	4.177	1.531	0.881	3.320	1.203
predictive residuals	0.938	4.496	1.629	0.893	3.486	1.228

The previous five tables show the simulation results using hybrid backward local bootstrap, we observe similar results as those from the backward bootstrap with transition density. The backward bootstrap using predictive residuals outperforms the backward bootstrap using fitted residuals and the standard deviation corresponding to the interval length drops when the sample size increases.

From Table 4.14, 4.15, 4.19 and 4.20, we see that the standard deviation of the interval lengths are incredibly large for the models with heteroscedastic error. We observed the same phenomenon in nonparametric autoregression for models with heteroscedastic errors.

Chapter 4, in full, is currently being prepared for the following paper prepared for future submission for publication: Bootstrap prediction intervals for Markov Processes. Pan,Li; Politis, Dimitris. The dissertation author was the primary investigator and author of this material.

Chapter 5

Bootstrap Prediction Intervals for Markov Processes Based on the Model-free Method

In this chapter we will discuss a fully model-free approach for predictive inference. The idea of model-free prediction is first proposed by Politis(2010). The principle of model-free prediction is to transform the dependent data into i.i.d data in a model-free environment. This transformation makes it easy to work with the i.i.d data, for example, we can resample the transformed i.i.d. data.

Instead of generating one-step ahead pseudo data by some estimated conditional distribution, for example the transition density given in (4.4) and the probability mass function given in (4.24), the model free method first resamples a value from a transformed i.i.d. data set and then transforms it back to one-step ahead pseudo data x_{t+1} based on the inverse transformation and x_n .

5.1 Construct the Transformation

Let $X = \{X_1, X_2, \dots\}$ is a stationary time series from a Markov process of order p , $Y_{t-1} = (X_{t-1}, \dots, X_{t-p})'$. Given $Y_{t-1} = y$, denote the conditional distribution

of X_t as

$$D_y(x) = P(X_t \leq x | Y_{t-1} = y) \quad (5.1)$$

and the unconditional distribution of Y_t as $D(y) = P(Y_t \leq y)$. We assume that $D(\cdot)$ and $D_y(\cdot)$ are continuous and invertible over the support of X .

Let $\eta_t = D_{Y_{t-1}}(X_t)$, for $t = p+1, \dots, n$ and $\eta_p = D(Y_p)$ Then

$$\begin{aligned} P(\eta_t \leq z | Y_{t-1} = y) &= P(D_y(X_t) \leq z | Y_{t-1} = y) \\ &= P(X_t \leq D_y^{-1}(z) | Y_{t-1} = y) \\ &= D_y(D_y^{-1}(z)) \\ &= z \text{ does not depend on } y, \text{ for } t = p+1, \dots, n \end{aligned}$$

and

$$P(\eta_p \leq z) = P(D(Y_p) \leq z) = P(Y_p \leq D^{-1}(z)) = D(D^{-1}(z)) = z$$

So $\eta_p \sim U(0, 1)$ and $P(\eta_p \leq z_p, \eta_{p+1} \leq z_{p+1}) = z_p \cdot z_{p+1}$ which is the density of two i.i.d. Uniform(0,1). Prove by induction, we have $P(\eta_p \leq z_p, \eta_{p+1} \leq z_{p+1}, \dots, \eta_n \leq z_n) = z_p z_{p+1} \dots z_n$ which is the density of $n - p + 1$ i.i.d. uniform distributions.

Given observation $x_1, \dots, x_n$, we estimate $D_y(x)$ by following kernel estimator $\hat{D}_y(x)$,

$$\hat{D}_y(x) = \frac{\sum_{i=p+1}^n 1_{\{x_i \leq x\}} K\left(\frac{\|y - y_{i-1}\|}{h}\right)}{\sum_{k=p+1}^n K\left(\frac{\|y - y_{k-1}\|}{h}\right)}. \quad (5.2)$$

The $\hat{D}$ in (5.2) is a step function, we use linear interpolation to smooth the function $\hat{D}$ in (5.2) to $\tilde{D}$. See Politis(2010 Section 4.1) for the detail of linear interpolation. Therefore, the estimated transformation amount can be calculated as follows and we claim they are approximately i.i.d. Uniform(0,1).

$$u_t = \tilde{D}_{y_{t-1}}(x_t), \text{ for } t = p+1, \dots, n. \quad (5.3)$$

Let us take $p = 1$ and $x, y \in \mathbb{R}$, intuitively, the model free estimator

$$\hat{D}_x(y) = \frac{\sum_{i=1}^{n-1} 1_{\{x_{i+1} \leq y\}} K\left(\frac{x - x_i}{h}\right)}{\sum_{k=1}^{n-1} K\left(\frac{x - x_k}{h}\right)} \quad (5.4)$$

works because that

$$\mathbb{E}(1_{X_{t+1} \leq y} | X_t = x) = P(X_{t+1} \leq y | X_t = x) = F(y|x). \quad (5.5)$$

Here $F(y|x)$ is the transition distribution. Li and Racine(2007 Chapter 6.1) studied how to estimate the conditional distribution function similar as (5.4) in regression.

5.2 Algorithm for Model-free Prediction Interval:MF

Given observations $x_1, \dots, x_n$, the model-free algorithm for constructing the predictive interval can be described as follows,

Algorithm 5.2.1.

(1) Use (5.3) to obtain the transformed data $u_{p+1}, \dots, u_n$.

(2) Calculate $\hat{x}_{n+1}$, the predictor of x_{n+1} , by

$$\hat{x}_{n+1} = \frac{1}{n-p} \sum_{t=p+1}^n \tilde{D}_{y_n}^{-1}(u_t)$$

(3) (a) Resample randomly(with replacement) the transformed data $u_{p+1}, \dots, u_n$ to create the pseudo data $u_{-M}^*, u_{-M+1}^*, \dots, u_0^*, u_1^*, \dots, u_{n-1}^*, u_n^*$ and u_{n+1}^* for some large positive integer M .

(b) Draw $x_{-M}^*, \dots, x_{-M+p-1}^*$ from any consecutive p values of $(x_1, \dots, x_n)$.

(c) Use $\tilde{D}_{y_{t-1}}^{-1}$ to generate $x_t^* = \tilde{D}_{y_{t-1}}^{-1}(u_t^*)$ for $t = -M + p, \dots, n$.

(d) Calculate the bootstrap future value $x_{n+1}^* = \tilde{D}_{y_n}^{-1}(u_{n+1}^*)$.

(e) Calculate $\hat{x}_{n+1}^* = \frac{1}{n-p} \sum_{t=p+1}^n \tilde{D}_{y_n}^{*-1}(u_t^*)$, where $\tilde{D}^*$ is a smoothed version of the following step function through linear interpolation

$$\hat{D}_y^*(x) = \frac{\sum_{i=p+1}^n 1_{\{x_i^* \leq x\}} K(\frac{\|y - y_{i-1}^*\|}{h})}{\sum_{k=p+1}^n K(\frac{\|y - y_{k-1}^*\|}{h})}.$$

(f) Calculate the bootstrap root $x_{n+1}^* - \hat{x}_{n+1}^*$.

(4) Repeat (3) 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 predictive interval for X_{n+1} is given by

$$[\hat{x}_{n+1} + q(\alpha/2), \hat{x}_{n+1} + q(1 - \alpha/2)].$$

5.3 Better Model-free Prediction Intervals: Predictive Model-free Method

In chapter 1 and 2, we use the predictive residuals instead of the fitted residuals to improve the performance of the predictive interval. From (5.2), we see that

$$\hat{D}_{y_n}(x) = \frac{\sum_{i=p+1}^n 1_{\{x_i \leq x\}} K\left(\frac{\|y_n - y_{i-1}\|}{h}\right)}{\sum_{k=p+1}^n K\left(\frac{\|x_n - x_{k-1}\|}{h}\right)}.$$

So the above estimated conditional distribution for x_{n+1} takes pair (y_n, x_{n+1}) as an "outside" pair. To mimic this situation, we can use the same trick mentioned in chapter 1 and 2 to calculate $\hat{D}_{y_{t-1}}^{(t)}$ using the data set excluding (y_{t-1}, x_t) data set for $t = p + 1, \dots, n$:

$$\hat{D}_{y_{t-1}}^{(t)}(x_t) = \frac{\sum_{i=p+1, i \neq t}^n 1_{\{x_i \leq x_t\}} K\left(\frac{\|y_{t-1} - y_{i-1}\|}{h}\right)}{\sum_{k=p+1, k \neq t}^n K\left(\frac{\|y_{t-1} - y_{k-1}\|}{h}\right)}, t = p + 1, \dots, n$$

After smoothing $\hat{D}_y(x)$ to $\tilde{D}_y(x)$, we get

$$u_t^{(t)} = \tilde{D}_{y_{t-1}}^{(t)}(x_t) \tag{5.6}$$

here, the $u_t^{(t)}$ serves as the analogs of the predictive residuals in chapter 1 and chapter 2. For the algorithm using the predictive transformed data, we just need to substitute $u_{p+1}^{(p+1)}, \dots, u_n^{(n)}$ for $u_{p+1}, \dots, u_n$ in Algorithm 5.2.1.

5.4 Smoothed Model Free: SMF

In section 4.1, we estimated the transition distribution $D_y(x) = P(X_t \leq x | Y_{t-1} = y)$ by $\hat{D}_y(x)$ as defined in (5.2). In this section, we discuss an alternative estimator of the conditional transition density that smooths the variable X_t . $1_{\{x_i \leq x\}}$ in (5.2) is the cumulative distribution of a point mass distribution such that $P(X_i = x) = 1$. To smooth this point mass distribution, we substitute $1_{\{x_i \leq x\}}$ with a continuous function denoted as $G(\frac{x-x_i}{h_0})$, where $G(\cdot)$ is a continuous cumulative distribution function. The new estimator $\hat{S}_y(x)$ is defined by

$$\hat{S}_y(x) = \frac{\sum_{i=p+1}^n G(\frac{x-x_i}{h_0}) K(\frac{\|y-y_{i-1}\|}{h})}{\sum_{k=p+1}^n K(\frac{\|y-y_{k-1}\|}{h})}, \quad (5.7)$$

and the estimated transformation $\{v_t, t = p+1, \dots, n\}$ can be calculated by

$$v_t = \hat{S}_{y_{t-1}}(x_t) \quad (5.8)$$

h_0 is the smoothing bandwidth associated with $X_1, \dots, X_n$ and $h_0 \rightarrow 0$.

Substituting $\hat{S}_y(x)$ for $\tilde{D}_y(x)$ and $\{v_t\}$ for $\{u_t\}$ in Algorithm 5.2.1, we have the smoothed model free method(SMF) as follows:

Algorithm 5.4.1.

(1) Use (5.8) to obtain the transformed data $v_{p+1}, \dots, v_n$.

(2) Calculate $\hat{x}_{n+1}$, the predictor of x_{n+1} , by

$$\hat{x}_{n+1} = \frac{1}{n-p} \sum_{t=p+1}^n \hat{S}_{x_n}^{-1}(v_t)$$

(3) (a) Resample randomly(with replacement) the transformed data $v_{p+1}, \dots, v_n$ to create the pseudo data $v_{-M}^*, v_{-M+1}^*, \dots, v_0^*, v_1^*, \dots, v_{n-1}^*, v_n^*$ and v_{n+1}^* for some large positive integer M .

(b) Draw $x_{-M}^*, \dots, x_{-M+p-1}^*$ from any consecutive p values of $(x_1, \dots, x_n)$.

(c) Use $\hat{S}_{y_{t-1}}^{-1}$ to generate $x_t^* = \hat{S}_{y_{t-1}}^{-1}(u_t^*)$ for $t = -M+p, \dots, n$.

(d) Calculate the bootstrap future value $x_{n+1}^* = \hat{S}_{y_n}^{-1}(v_{n+1}^*)$.

(e) Calculate $\hat{x}_{n+1}^* = \frac{1}{n-p} \sum_{t=p+1}^n \hat{S}_{y_n}^{*-1}(v_t^*)$, where $\hat{S}_y^*(x)$ is

$$\hat{S}_y^*(x) = \frac{\sum_{i=p+1}^n G(\frac{x-x_i^*}{h_0}) K(\frac{\|y-y_{i-1}^*\|}{h})}{\sum_{k=p+1}^n K(\frac{\|y-y_{k-1}^*\|}{h})}.$$

(f) Calculate the bootstrap root $x_{n+1}^* - \hat{x}_{n+1}^*$.

(4) Repeat (3) 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 predictive interval for X_{n+1} is given by

$$[\hat{x}_{n+1} + q(\alpha/2), \hat{x}_{n+1} + q(1 - \alpha/2)].$$

Remark 5.4.1. As suggested by Li and Racine(2007 Chapter 6.2), the optimal smoothing for weighted MSE requires that $h_0 = O_p(n^{-2/5})$ and $h = O_p(n^{-1/5})$, we choose h through cross validation and then let $h_0 = h^2$.

We can also use the delete- x_t estimator

$$\hat{S}_y^{(t)}(x_t) = \frac{\sum_{i=p+1, i \neq t}^n G(\frac{x_t - x_i}{h_0}) K(\frac{\|y_t - y_{i-1}\|}{h})}{\sum_{k=p+1, k \neq t}^n K(\frac{\|y_t - y_{k-1}\|}{h})}, t = p+1, \dots, n$$

to get the predictive transformed i.i.d. data:

$$v_t^{(t)} = \hat{S}_{y_{t-1}}^{(t)}(x_t), \text{ for } t = p+1, \dots, n.$$

To construct the algorithm for predictive smoothed model free method, we just need to substitute $v_{p+1}^{(p+1)}, \dots, v_n^{(n)}$ for $v_{p+1}, \dots, v_n$ in Algorithm 5.4.1.

Let us take $p = 1$ and $x, y \in \mathbb{R}$, intuitively, the smoothed model free estimator

$$\hat{S}_x(y) = \frac{\sum_{i=1}^{n-1} G(\frac{y-x_{i+1}}{h_0}) K(\frac{x-x_i}{h})}{\sum_{k=1}^{n-1} K(\frac{x-x_k}{h})} \quad (5.9)$$

works because if $F(y|x)$ is twice continuously differentiable, $k(v) = dG(v)/dv$ is bounded symmetric, and compactly supported, and that $F^{(2)}(x)$ is hölder-continuous, $0 \leq h_0 \leq n^{-\epsilon}$ for some $0 \leq \epsilon \leq \frac{1}{8}$, then as $n \rightarrow \infty$,

$$\mathbb{E}(G(\frac{y - X_{t+1}}{h_0}) | X_t = x) = F(y|x) + O(h^2). \quad (5.10)$$

See Li and Racine(2007 Theorem 1.2) for the detail.

Under regression case, $\{(x_i, y_i)\}$ are i.i.d. data. Substitute the x_{i+1} in (5.4) and (5.9) with y_i , we have

$$\hat{F}_{1x}(y) = \frac{\sum_{i=1}^n 1_{\{y_i \leq y\}} K\left(\frac{x-x_i}{h}\right)}{\sum_{k=1}^n K\left(\frac{x-x_k}{h}\right)}$$

$$\hat{F}_{2x}(y) = \frac{\sum_{i=1}^n G\left(\frac{y-y_i}{h_0}\right) K\left(\frac{x-x_i}{h}\right)}{\sum_{k=1}^n K\left(\frac{x-x_k}{h}\right)}$$

If the density distribution $\mu(x)$ for x and the conditional distribution $F(y|x)$ are larger than 0 and twice continuously differentiable, $K(\cdot)$ is a symmetric, bounded and compactly supported PDF, and $h \rightarrow 0$, $nh \rightarrow \infty$, it can be proved that

$$MSE[\hat{F}_{1x}(y)] - MSE[\hat{F}_{2x}(y)] = \frac{2h_0(\int G(v)K(v)vdv)(\int K(s)^2dsF_y(y|x)/\mu(x))}{nh} > 0. \quad (5.11)$$

See Li and Racine(2007) theorem 6.1 and 6.2.

We can linear interpolate the step function $\hat{F}_{1x}(y)$ to get $\tilde{F}_{1x}(y)$ as we get $\tilde{D}_x(y)$ from $\hat{D}_x(y)$. Note that $\tilde{F}_{1x}(y) = \hat{F}_{1x}(y) + O_p(\frac{1}{nh})$, so $MSE[\hat{F}_{1x}(y)] = MSE[\tilde{F}_{1x}(y)] + O_p(\frac{1}{n^2h^2})$. Since the optimal smoothing for weighted MSE requires that $h_0 = O_p(n^{-2/5})$ and $h = O_p(n^{-1/5})$, we have $\frac{1}{n^2h^2} \ll \frac{h_0}{nh}$, the smoothed method improve the mean square error by order of $O_p(\frac{h_0}{nh})$. We conclude that in regression the smoothed method has higher order accuracy.

In autoregression, we expect the smoothed model free method has smaller mean square error too. We will see that the smoothed model free(SMF) method indeed has better performance than model free(MF) method.

5.5 Simulation

Same as before, we conduct monte-carlo studies based on 3 models with i.i.d innovations and 2 models with heteroscedastic errors as follows,

- Model 1: $X_{t+1} = \sin(X_t) + \epsilon_{t+1}$

- Model2: $X_{t+1} = 0.8 \log(3X_t^2 + 1) + \epsilon_{t+1}$
- Model3: $X_{t+1} = -0.5 \exp(-50X_t^2)X_t + \epsilon_{t+1}$
- Model4: $X_{t+1} = \sin(X_t) + \sqrt{0.5 + 0.25X_t^2}\epsilon_{t+1}$
- Model5: $X_{t+1} = 0.75X_t + 0.15X_t\epsilon_{t+1} + \epsilon_{t+1}$

For each model, we do the simulations for $n = 50, 100$ or 200 with normal and Laplace errors and record the coverage rate, length of interval and corresponding standard deviation. For the bandwidth h , we select it through cross validation, We use standard normal CDF restricted on $[-2, 2]$ as $G(\cdot)$. We choose bandwidth h for kernel $K(\cdot)$ through cross validation and choose the smoothing bandwidth h_0 to be $h_0 = h^2$ as mentioned in remark 5.4.1.

Table 5.1: Model Free Method-model 1

normal innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.893	3.647	0.740	0.844	3.150	0.645
MF-predictive	0.926	4.108	0.854	0.880	3.482	0.721
SMF-fitted	0.892	3.627	0.717	0.843	3.131	0.619
SMF-predictive	0.939	4.293	0.828	0.893	3.614	0.709
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.917	3.771	0.602	0.869	3.238	0.497
MF-predictive	0.941	4.152	0.680	0.897	3.511	0.550
SMF-fitted	0.916	3.731	0.551	0.869	3.221	0.489
SMF-predictive	0.946	4.231	0.647	0.902	3.471	0.530
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.924	3.750	0.481	0.878	3.222	0.403
MF-predictive	0.942	4.065	0.558	0.894	3.395	0.455
SMF-fitted	0.924	3.731	0.464	0.877	3.208	0.387
SMF-predictive	0.946	4.123	0.570	0.899	3.439	0.444
Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.889	3.708	0.986	0.844	3.086	0.803
MF-predictive	0.922	4.278	1.129	0.876	3.494	0.993
SMF-fitted	0.891	3.715	0.963	0.846	3.084	0.764
SMF-predictive	0.930	4.442	1.126	0.887	3.620	0.972
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.911	3.884	0.895	0.864	3.124	0.643
MF-predictive	0.935	4.426	1.001	0.891	3.510	0.765
SMF-fitted	0.910	3.846	0.856	0.864	3.106	0.623
SMF-predictive	0.941	4.544	0.965	0.896	3.542	0.738
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.922	3.926	0.711	0.873	3.119	0.514
MF-predictive	0.938	4.387	0.900	0.891	3.386	0.639
SMF-fitted	0.921	3.895	0.684	0.873	3.103	0.491
SMF-predictive	0.942	4.447	0.868	0.894	3.407	0.629

Table 5.2: Model Free Method-model 2

normal innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.894	3.763	0.850	0.843	3.228	0.750
MF-predictive	0.932	4.269	0.927	0.886	3.614	0.794
SMF-fitted	0.893	3.713	0.805	0.844	3.211	0.713
SMF-predictive	0.943	4.440	0.885	0.902	3.755	0.751
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.918	3.790	0.620	0.868	3.247	0.542
MF-predictive	0.945	4.218	0.706	0.901	3.560	0.569
SMF-fitted	0.915	3.733	0.597	0.865	3.202	0.516
SMF-predictive	0.950	4.315	0.695	0.906	3.600	0.550
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.924	3.773	0.524	0.875	3.212	0.431
MF-predictive	0.944	4.126	0.609	0.895	3.419	0.472
SMF-fitted	0.921	3.725	0.495	0.873	3.195	0.417
SMF-predictive	0.947	4.163	0.605	0.899	3.447	0.469
Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.886	3.774	1.094	0.840	3.145	0.918
MF-predictive	0.924	4.412	1.222	0.881	3.587	1.035
SMF-fitted	0.886	3.744	1.047	0.841	3.117	0.855
SMF-predictive	0.933	4.599	1.209	0.890	3.707	1.022
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.908	3.904	0.961	0.863	3.162	0.744
MF-predictive	0.937	4.581	1.100	0.893	3.589	0.881
SMF-fitted	0.909	3.848	0.882	0.863	3.128	0.683
SMF-predictive	0.941	4.637	1.060	0.898	3.630	0.822
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.922	3.974	0.793	0.872	3.145	0.560
MF-predictive	0.939	4.464	1.000	0.893	3.444	0.701
SMF-fitted	0.920	3.941	0.749	0.872	3.123	0.540
SMF-predictive	0.941	4.497	0.990	0.895	3.471	0.689

Table 5.3: Model Free Method-model 3

normal innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.893	3.570	0.651	0.844	3.100	0.580
MF-predictive	0.922	3.892	0.697	0.874	3.338	0.588
SMF-fitted	0.893	3.566	0.643	0.843	3.084	0.567
SMF-predictive	0.935	4.057	0.676	0.889	3.477	0.589
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.919	3.736	0.506	0.870	3.195	0.410
MF-predictive	0.937	3.996	0.571	0.888	3.360	0.443
SMF-fitted	0.920	3.711	0.465	0.871	3.184	0.394
SMF-predictive	0.944	4.092	0.551	0.898	3.439	0.433
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.930	3.783	0.415	0.882	3.224	0.326
MF-predictive	0.940	3.946	0.470	0.893	3.329	0.358
SMF-fitted	0.931	3.779	0.391	0.882	3.218	0.311
SMF-predictive	0.945	4.019	0.451	0.898	3.377	0.349
Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.888	3.665	0.933	0.840	3.011	0.754
MF-predictive	0.916	4.146	1.082	0.874	3.401	0.864
SMF-fitted	0.892	3.686	0.902	0.844	3.021	0.715
SMF-predictive	0.926	4.270	1.015	0.883	3.482	0.847
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.920	3.961	0.746	0.871	3.149	0.559
MF-predictive	0.938	4.321	0.936	0.893	3.450	0.657
SMF-fitted	0.921	3.949	0.711	0.872	3.130	0.514
SMF-predictive	0.943	4.499	0.857	0.897	3.501	0.626
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.927	3.991	0.701	0.879	3.153	0.504
MF-predictive	0.938	4.331	0.894	0.890	3.318	0.568
SMF-fitted	0.927	3.962	0.651	0.879	3.139	0.476
SMF-predictive	0.942	4.403	0.863	0.892	3.340	0.551

Table 5.4: Model Free Method-model 4

normal innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.891	3.084	0.742	0.841	2.660	0.670
MF-predictive	0.926	3.549	0.849	0.880	2.989	0.725
SMF-fitted	0.890	3.073	0.712	0.842	2.653	0.630
SMF-predictive	0.931	3.687	0.836	0.891	3.078	0.699
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.921	3.134	0.578	0.874	2.685	0.504
MF-predictive	0.945	3.495	0.636	0.900	2.916	0.533
SMF-fitted	0.921	3.119	0.551	0.874	2.679	0.476
SMF-predictive	0.951	3.587	0.620	0.908	2.964	0.513
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.927	3.192	0.539	0.880	2.716	0.439
MF-predictive	0.944	3.458	0.636	0.897	2.866	0.476
SMF-fitted	0.926	3.167	0.504	0.879	2.707	0.420
SMF-predictive	0.947	3.481	0.574	0.900	2.890	0.473
Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.891	3.102	0.932	0.846	2.554	0.764
MF-predictive	0.920	3.616	1.064	0.879	2.916	0.855
SMF-fitted	0.893	3.093	0.871	0.846	2.539	0.707
SMF-predictive	0.930	3.774	1.094	0.888	3.017	0.857
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.912	3.200	0.740	0.865	2.578	0.581
MF-predictive	0.938	3.755	0.905	0.894	2.918	0.663
SMF-fitted	0.912	3.177	0.724	0.864	2.559	0.555
SMF-predictive	0.943	3.880	0.892	0.899	2.950	0.605
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.926	3.261	0.695	0.878	2.572	0.502
MF-predictive	0.942	3.648	0.862	0.895	2.774	0.664
SMF-fitted	0.926	3.224	0.648	0.878	2.559	0.463
SMF-predictive	0.945	3.716	0.846	0.898	2.812	0.562

Table 5.5: Model Free Method-model 5

normal innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.898	3.724	0.980	0.850	3.204	0.855
MF-predictive	0.941	4.310	1.121	0.898	3.635	0.919
SMF-fitted	0.898	3.710	0.937	0.852	3.206	0.812
SMF-predictive	0.949	4.509	1.071	0.910	3.772	0.883
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.920	3.715	0.851	0.874	3.189	0.714
MF-predictive	0.946	4.155	0.933	0.901	3.464	0.775
SMF-fitted	0.920	3.689	0.794	0.874	3.177	0.667
SMF-predictive	0.952	4.264	0.915	0.909	3.528	0.733
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.929	3.724	0.781	0.882	3.171	0.652
MF-predictive	0.946	4.015	0.866	0.901	3.357	0.712
SMF-fitted	0.928	3.700	0.731	0.882	3.165	0.624
SMF-predictive	0.950	4.062	0.836	0.904	3.381	0.685
Laplace innovations	nominal coverage 95%			nominal coverage 90%		
$n = 50$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.892	3.837	1.327	0.846	3.166	1.138
MF-predictive	0.926	4.435	1.482	0.882	3.575	1.224
SMF-fitted	0.894	3.813	1.235	0.849	3.166	1.053
SMF-predictive	0.937	4.748	1.497	0.898	3.757	1.149
$n = 100$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.911	3.822	1.167	0.865	3.085	0.898
MF-predictive	0.941	4.564	1.436	0.896	3.507	1.054
SMF-fitted	0.911	3.779	1.066	0.866	3.072	0.840
SMF-predictive	0.946	4.703	1.401	0.903	3.587	1.002
$n = 200$	CVR	LEN	st.err	CVR	LEN	st.err
MF-fitted	0.920	3.830	0.991	0.871	3.047	0.728
MF-predictive	0.942	4.369	1.074	0.893	3.344	0.817
SMF-fitted	0.920	3.791	0.904	0.871	3.028	0.682
SMF-predictive	0.945	4.412	1.076	0.895	3.350	0.783

From the simulation results, we conclude that

- The predictive methods improve the coverage level with the cost of high variability. However, this is a finite sample effect. For large sample size, the effect of deleting one pair of data in the scatter plot is negligible.
- The smoothed model free methods tend to have smaller standard deviation

and better coverage level.

- When the sample size is increased to 100, the standard deviation of the interval lengths decreased in all cases .
- When the sample size is increased to 100, the coverage levels are improved compared to the corresponding coverage levels when $n = 100$

Remark 5.5.1. *Due to the complexity of the dependence structure of time series, it's not easy to prove that the mean square error of the smoothed model free estimator (5.7) is smaller than that of the estimator (5.2). But at least we know it is true for regression as showed in (5.11). So the observation result that by smoothing the the indicator function for X_{t+1} with a CDF $G(\cdot)$ the standard deviation corresponding to the interval length is reduced is not unexpected.*

Remark 5.5.2. *In chapter 3, chapter 4 and chapter 5, we use the same 5 models (3 models with i.i.d. errors, 2 models with heteroscedastic errors). For the models with i.i.d. errors, the nonparametric kernel smoothing method in chapter 3, the bootstrap based on transition density, the local bootstrap and the Hybrid bootstrap in chapter 4, the model-free methods in chapter 5 all work well. Among all the methods, the nonparametric methods are more stable than others due to the lowest standard deviation associated with the interval lengths.*

However, for models with heteroscedastic errors, the nonparametric kernel smoothing methods do not work well even though we use over-smoothed resampling bandwidth in generating the pseudo series. But the bootstrap based on transition density, the local bootstrap in Chapter 4 and all the model-free methods in Chapter 5 do a good job in dealing with models with heteroscedastic errors. We observe that nonparametric approach has significantly larger variability compared to other Markov approaches discussed in chapter 4 and 5. The predictive smoothed model free method in Chapter 5 is the best among all the methods for Markov processes in Chapter 4 and 5. However, the smoothed model free method demands powerful computation power due to smoothing the conditional distribution function.

Chapter 5, in full, is currently being prepared for the following paper prepared for future submission for publication: Bootstrap prediction intervals for Markov Processes. Pan,Li; Politis, Dimitris. The dissertation author was the primary investigator and author of this material.

References

- Alonso, A. M., Peña, D., and Romo, J., 2002: Forecasting time series with sieve bootstrap. *Journal of Statistical Planning and Inference*, **100**(1), 1–11.
- Baillie, R. T., and Bollerslev, T., 1992: Prediction in dynamic models with time-dependent conditional variances. *Journal of Econometrics*, **52**(1), 91–113.
- Berg, A., and Politis, D., 2009: Cdf and survival function estimation with infinite-order kernels. *Electronic Journal of Statistics*, **3**, 1436–1454.
- Boldin, M. V., 1982: The estimate of the distribution of noise in autoregressive scheme. *Teoriya Veroyatnostei i ee Primeneniya*, **27**(4), 805–810.
- Bollerslev, T., and Wooldridge, J. M., 1992: Quasi-maximum likelihood estimation and inference in dynamic models with time-varying covariances. *Econometric reviews*, **11**(2), 143–172.
- Bose, A., and Mukherjee, K., 2003: Estimating the arch parameters by solving linear equations. *Journal of Time Series Analysis*, **24**(2), 127–136.
- Bose, A., and Mukherjee, K., 2009: Bootstrapping a weighted linear estimator of the arch parameters. *Journal of Time Series Analysis*, **30**(3), 315–331.
- Box, G. E., and Jenkins, G. M., 1976: Time series analysis, control, and forecasting.
- Breidt, F. J., Davis, R. A., and Dunsmuir, W., 1995: Improved bootstrap prediction intervals for autoregressions. *Journal of Time Series Analysis*, **16**(2), 177–200.
- Cao, R., Febreroabande, M., González-Manteiga, W., Prada-Sánchez, J., and García-Jurado, I., 1997: Saving computer time in constructing consistent bootstrap prediction intervals for autoregressive processes. *Communications in Statistics-Simulation and Computation*, **26**(3), 961–978.
- Chatterjee, S., and Bose, A., 2005: Generalized bootstrap for estimating equations. *Annals of statistics*, 414–436.
- Darsow, W. F., Nguyen, B., and Olsen, E. T., 1992: Copulas and markov processes. *Illinois Journal of Mathematics*, **36**(4), 600–642.

- Elharfaoui, E., and Harel, M., 2008: Central limit theorem of the smoothed empirical distribution functions for asymptotically stationary absolutely regular stochastic processes. *International Journal of Stochastic Analysis*, **2008**.
- Franke, J., Kreiss, J.-P., and Mammen, E., 2002a: Bootstrap of kernel smoothing in nonlinear time series. *Bernoulli*, **8**(1), 1–37.
- Franke, J., Kreiss, J.-P., Mammen, E., and Neumann, M. H., 2002b: Properties of the nonparametric autoregressive bootstrap. *Journal of Time Series Analysis*, **23**(5), 555–585.
- Freedman, D., 1984: On bootstrapping two-stage least-squares estimates in stationary linear models. *The Annals of Statistics*, 827–842.
- Grigoletto, M., 1998: Bootstrap prediction intervals for autoregressions: some alternatives. *International Journal of Forecasting*, **14**(4), 447–456.
- Hardle, W., and Marron, J., 1991: Bootstrap simultaneous error bars for nonparametric regression. *The Annals of Statistics*, 778–796.
- Kreiss, J.-P., 1997: *Asymptotical properties of residual bootstrap for autoregressions*. Citeseer.
- Li, Q., and Racine, J. S., 2007: *Nonparametric econometrics: Theory and practice*. Princeton University Press.
- Masarotto, G., 1990: Bootstrap prediction intervals for autoregressions. *International Journal of Forecasting*, **6**(2), 229–239.
- Mukhopadhyay, P., and Samaranayake, V., 2010: Prediction intervals for time series: a modified sieve bootstrap approach. *Communications in Statistics—Simulation and Computation*, **39**(3), 517–538.
- Müller, U. U., Schick, A., and Wefelmeyer, W., 2009: Estimating the innovation distribution in nonparametric autoregression. *Probability theory and related fields*, **144**(1-2), 53–77.
- Nadaraya, E., 1965: On non-parametric estimates of density functions and regression curves. *Theory of Probability & Its Applications*, **10**(1), 186–190.
- Paparoditis, E., and Politis, D. N., 1998: The backward local bootstrap for conditional predictive inference in nonlinear time series. *4th Hellenic-European Conference on Computer Mathematics and its Applications (HERCMA '98)*, E. A. Lipitakis (Ed.), Lea Publishing, Athens, Greece, 467–470.
- Paparoditis, E., and Politis, D. N., 2002: The local bootstrap for markov processes. *Journal of Statistical Planning and Inference*, **108**(1), 301–328.

- Pascual, L., Romo, J., and Ruiz, E., 2004: Bootstrap predictive inference for arima processes. *Journal of Time Series Analysis*, **25**(4), 449–465.
- Politis, D. N., 2010: Model-free model-fitting and predictive distributions. *Test*, 1–39.
- Politis, D. N., 2013: Model-free model-fitting and predictive distributions rejoinder. *Invited Discussion paper in journal TEST*.
- Rajarshi, M., 1990: Bootstrap in markov-sequences based on estimates of transition density. *Annals of the Institute of Statistical Mathematics*, **42**, 253–268. ISSN 0020-3157. doi:10.1007/BF00050835.
- Reeves, J. J., 2000: Bootstrap prediction intervals for arch models. *International Journal of Forecasting*, **21**, 237–248.
- Rubio, P. O., 1999: Forecast intervals in arch models: bootstrap versus parametric methods. *Applied Economics Letters*, **6**, 323–327.
- Rüschendorf, L., 1977: Consistency of estimators for multivariate density functions and for the mode. *Sankhyā: The Indian Journal of Statistics, Series A*, 243–250.
- Shephard, N., 1996: Statistical aspects of arch and stochastic volatility. *Monographs on Statistics and Applied Probability*, **65**, 1–68.
- Sklar, A., 1973: Random variables, joint distribution functions, and copulas. *Kybernetika*, **9**(6), 449–460.
- Sklar, M., 1959: *Fonctions de répartition à n dimensions et leurs marges*. Université Paris 8.
- Stine, R. A., 1987: Estimating properties of autoregressive forecasts. *Journal of the American statistical association*, **82**(400), 1072–1078.
- Thombs, L. A., and Schucany, W. R., 1990: Bootstrap prediction intervals for autoregression. *Journal of the American Statistical Association*, **85**(410), 486–492.
- Weiss, A. A., 1986: Asymptotic theory for arch models: Estimation and testing. *Econometric Theory*, **2**(1), pp. 107–131. ISSN 02664666.