Stockholms universitet

Bootstrap Methods in Time Series Analysis

Fanny Bergström

Kandidatuppsats i matematisk statistik
Bachelor Thesis in Mathematical Statistics

Kandidatuppsats 2018:4
Matematisk statistik
Juni 2018

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Mathematical Statistics
Stockholm University
Bachelor Thesis **2018:4**
http://www.math.su.se

Bootstrap Methods in Time Series Analysis

Fanny Bergström*

June 2018

Abstract

Bootstrap methods can be used to non-parametrically make inference about a sample estimates e.g. through its bias, standard error and confidence intervals. In this thesis we consider four different block bootstrap methods, namely the non-overlapping block bootstrap (NBB), moving block bootstrap (MBB), circular block bootstrap (CBB) and stationary bootstrap (SB) which are commonly used for data correlated in time. The purpose of the thesis is to quantify and compare the efficiency of the block bootstrap methods. We do this by simulations of linear time series models such as the AR and MA model and evaluate the bootstrap methods in estimating the sample mean as well as refit the model parameters and autocorrelation when varying sample size and block length.

We find that the methods preform different when sample size and block length varies, but common for all methods is that the dependence is underestimated compared to the true underlying model. The conclusion is that methods using overlapping blocks are to be preferred over non-overlapping blocks and that random block lengths leads to a larger variance of the parameter estimates than for the other methods when block length is fixed.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: fabe4028@su.se. Supervisor: Mathias Lindholm, Filip Lindskog.

Preface

This thesis of 15 ECTS will lead to a Bachelors Degree in Mathematical Statistics at the Department of Mathematics at Stockholm University.

I would like to thank my supervisors Mathias Lindholm and Filip Lindskog, Department of Mathematical Statistics, for all their help and guidance through this thesis. I would also like to thank my friends and family for their support.

Contents

Some of the notation

X - random variable
$\{x_t\}$ - sample
x_t - sample value at time t
"*" - bootstrap variable, $\mathbf{X}^*$ is a bootstrap data set generated from $\mathbf{X}$
θ - or other Greek letters are parameters
F - probability distribution
f - probability density function
ℓ - lag between two time series obesvations
ρ - correlation coefficient
γ - covariance
BM - bench mark, optimal value
CBB- circular block bootstrap
MBB- moving block bootstrap
NBB- none overlapping block bootstrap
SB - stationary bootstrap

1 Introduction

In statistics a common problem is forecasting future events or explain how events happen over time. We try to forecast e.g. the closing price of a stock or when the next earthquake will hit. When historical data are available we can develop models that fit the data to make accurate forecasts. Some classical time series models used for forecasting are the autoregressive (AR) models and moving average (MA) models. These models describes future behavior based on past behavior and are used when there is some correlation between the values in data. The AR model is a linear regression of the current value against past values of data while in the MA model the present value can be seen as a weighted sum of the past values.

When we have developed a model based on historical data, what can we say about the accuracy of it? Resampling methods such as the bootstrap is a way of answering this questions without the use of any parametric assumptions. Efron first introduced the bootstrap method with the main idea to replicate the original data using sampling with replacement, see [Efron, 1979]. New estimators of parameters can be made from the pseudo data generated this way and when repeating this process we can derive confidence intervals and standard errors of the parameter estimate which can be used to make inference about the model parameter.

Efron's initial bootstrap method of resampling individual observations works well for independent data but when there are dependence in the data simple sampling with replacement has been proved, among others by [Carlstein, 1986], to perform poorly. The simple bootstrap will not be able to replicate the original data well as it will not capture the dependence structure. Several methods have been proposed for bootstrapping dependent data in attempts to reproduce different aspects of the dependence. Some of these methods include resampling of blocks instead of individual observations, so called block bootstrap methods. In this paper we consider some of the most popular block bootstrap methods and try to quantify their effecencies in estimating parameters and their standard errors and confidence intervals when fitting time series models. We are interested in how sample size and block length affect the bootstrap method's effiencies in preserving the dependence structure from the original data sets. The methods considered are:

1. Non overlapping block bootstrap (NBB) by [Carlstein, 1986].
2. Moving block bootstrap (MBB), proposed by both [Künch, 1989] and [Liu and Singh, 1992].
3. Circular block bootstrap (CBB), proposed by [Politis and Romano, 1992].
4. Stationary bootstrap (SB), also proposed by [Politis and Romano, 1994].

The first three methods resample blocks of observations with a nonrandom block length while the last method, the SB method, uses a random block

length. A description of these methods are presented in Section 3.3. We evaluate the performance of the bootstrap methods through a simulation study found in Section 4. Data is simulated from AR and MA processes and based on pseudo data generated by the different bootstrap methods we investigate besides the dependence also how well we can refit the models from which data is simulated. We evaluate the accuracy of the estimated parameters and their deviations through e.g. their confidence intervals.

The rest of this thesis is organized as follows. Section 2 and 3 is the theoretical framework of the simulation. Section 2 includes theory about time series analysis, e.g. properies of time series and time series models. Bootstrap methods are described in Section 3 and Section 5 provides concluding comments.

2 Time Series Analysis

Treating observations as a collection of random variables over a specified time period corresponds to a *time series*. Example of a time series can be the closing price of a stock or the temperature outside for each hour. If the observations have been collected at equally-spaced time points we can use the notation x_t, $(t = ..., -1, 0, 1, 2, ...)$ so that the observations are indexed with the time t at which they were collected. A *time series process* is a stochastic process of random variables x_t indexed in time. We denote this process $\{x_t, t \in \mathcal{T}\}$, or simply $\{x_t\}$.

Time series analysis is the framework of analyzing the structure and behavior of the time series process. It is common that statistical models assume that observations are independent random variables, in contrast in time series analysis concerned with describing the dependence among observations in data. In this section, we describe some properties of time series, including stationarity, the autocorrelation function and two time series models. The models are autoregressive (AR) models and moving average (MA) models. All theory (unless stated otherwise) used in this section is found in [Tsay, 2005], Chapter 2.

2.1 Stationarity

Time series models are often based on the assumption of *stationarity*. A stationary time series process is a process whose behaviour does not depend on when we start to observe it, which implies that the series will look roughly the same at different intervals of the same length. In application the stationarity of a time series makes it possible to make inference about the future based on previous observations.

A time series process is stationary if it for all integers *r, s* and *t* the following three properties hold:

(i) $\mathrm{E}(x_t) = \mu$, a constant

(ii) $\mathrm{Var}(x_t) \leq \infty$

(iii) $\mathrm{Cov}(x_t, x_{t+s}) = \gamma_s$,

where the covariance γ_s is a function of s and independent t as it only depends on the length of the interval between the time points. Time series with these properties are called *weakly stationary*. For a time series to be *strictly stationary* the joint distribution of $(x_{t_1}, ..., x_{t_k})$ is identical to $(x_{t_{1+s}}, ..., x_{t_{k-s}})$ for all s. This condition is very strong and hard to verify empirically which is why the weaker condition of stationarity is often considered. As we will only consider weak stationarity in this thesis we will simply denote it stationarity further on.

2.2 Autocorrelation function

A first step in analyzing a time series process is to look at the correlation pattern at different time points. This can be made by plotting the sample *autocorrelation function* (ACF). The ACF for a stationary time series does not depend on the time in the time series but on the length between the observations in $\{x_t\}$, we denote this length the lag ℓ. The ACF between x_t and $x_{t-\ell}$ is denoted ρ_ℓ and defined as

$$\rho_\ell = \frac{\mathrm{Cov}(x_t, x_{t-\ell})}{\sqrt{\mathrm{Var}(x_t)\mathrm{Var}(x_{t-\ell})}} = \frac{\mathrm{Cov}(x_t, x_{t-\ell})}{\mathrm{Var}(x_t)} = \frac{\gamma_\ell}{\gamma_0}, \tag{1}$$

where we use the property $\mathrm{Var}(x_t)$=$\mathrm{Var}(x_{t-\ell})$ of a weakly stationary series. Eq. (1) is the regular definition of a correlation coefficient. From this definition we have that $\rho_0 = 1$, $\rho_\ell = \rho_{-\ell}$ and -1 $\leq \rho_\ell \leq$ 1. A weakly stationary series is not serially correlated if and only if $\rho_\ell = 0$ for all $\ell > 0$. For a sample of observations $\{x_t, t = 1, ..., T\}$, let $\bar{x} = \sum_{t=1}^{T} x_t/T$ be the sample mean, the lag-ℓ sample autocorrelation is defined as

$$\hat{\rho}_\ell = \frac{\sum_{t=\ell+1}^{T}(x_t - \bar{x})(x_{t-\ell} - \bar{x})}{\sum_{t=\ell+1}^{T}(x_t - \bar{x})^2}, \qquad 0 \leq \ell < T - 1. \tag{2}$$

The statistics $\hat{\rho}_1$, $\hat{\rho}_2$,... defined in Eq. (2) is the *sample autocorrelation function* of $\{x_t\}$. The ACF is important in time series analysis as a time series model can be characterized by its ACF since it describes the correlation pattern of observations in the data.

2.3 White noise

Let a_t be a time series, then

$$a_t \sim WhiteNoise(0, \sigma_a^2)$$

if and only if $\gamma_0 = \sigma_a^2 \in \mathbb{R}$ and $\gamma_k = 0$ for all lags $k > 0$. That is, if a time series process has zero mean and zero covariance between its values it is said to be a white noise process.

2.4 Linear time series models

A time series $\{x_t\}$ is considered *linear* if it can be written as

$$x_t = \mu + \sum_{i=0}^{\infty} \psi_i a_{t-i},$$

where μ is the series mean, the parameter $\psi_0 = 1$ and $\{a_t\}$ is a white noise series (i.e. a sequence of i.i.d. random variables with finite mean and variance). We let a_t denote the new information of the time series at time t, also called the *chock* or *innovation* at time t. Not all times series are linear, however in this paper we will not consider nonlinear time series. Two common linear time series models are the AR models and MA models.

The AR model specifies that the output variable depends linearly on its previous values and a stochastic term, hence it can be seen as a multiple linear regression on past values. The notation AR(p) indicates an autoregressive model of order p which means that the output value x_t can be expressed in terms of p past values and the stochastic term. An AR(p) model is defined

$$x_t = c + \sum_{i=1}^{p} \varphi_i x_{t-i} + \varepsilon_t, \tag{3}$$

where $\varphi_1, ..., \varphi_p$ are the parameters of the model, c is a constant and ε_t is a sequence of uncorrelated error terms (a white noise process). The ε_t are often assumed to be normally distributed, $N(0, \sigma_\varepsilon^2)$, but is valid given that ε_t is i.i.d.

The MA model specifies that the output variable depends linearly on the current and past values of a stochastic variable. The notation MA(q) refers to a moving average model of order q. We define the MA(q) model

$$x_t = c + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \tag{4}$$

where c is the series mean and $\theta_1, ..., \theta_q$ the parameters of the model and $\varepsilon_t, \varepsilon_{t-1}, ..., \varepsilon_{t-q}$ are white noise error terms with zero mean and variance σ_ε^2.

MA-models are always stationary as they are finite linear combinations of a white noise series for which the first two moments are time invariant.

2.5 Properties of AR/MA models

In this section we will look at some properties of the time series models described in Section 2.4, including their ACF. Detailed results are given for models of order 1 and 2 and more general results for higher orders of p/q. For simplicity we assume $c = 0$ for the AR/MA models. We also assume that the time series are stationary.

For the AR model the stationary condition means that $\mathrm{E}(x_t) = \mu$, $\mathrm{Var}(x_t) = \gamma_0$ and $\mathrm{Cov}(x_t, x_{t-s}) = \gamma_s$, where c and γ_0 are constant and γ_s is a function of s and doeas not depend of t. The expectation for an AR(1) model is

$$\mathrm{E}(x_t) = \varphi_0 + \varphi_1 \mathrm{E}(x_{t-1}) = \mu,$$

which follows by taken the expectation of (3) and because $\mathrm{E}(\varepsilon_t)$=0. We use the stationary condition, $\mathrm{E}(x_t) = \mathrm{E}(x_{t-1}) = \mu$, to get following result

$$\mathrm{E}(x_t) = \frac{\varphi_0}{1 - \varphi_1}.$$

The variance of the AR(1) model is

$$\mathrm{Var}(x_t) = \varphi_1^2 \mathrm{Var}(x_{t-1}) + \sigma_\varepsilon^2,$$

again using the stationary condition we get

$$\mathrm{Var}(x_t) = \gamma_0 = \frac{\sigma_\varepsilon^2}{1 - \varphi_1^2} \quad \text{and} \quad \gamma_\ell = \varphi_1 \gamma_{\ell-1}, \quad \text{for} \quad \ell > 0.$$

That is

$$\gamma_\ell = \begin{cases} \varphi_1 \gamma_1 + \sigma_\varepsilon^2 & \text{if } \ell = 0 \\ \varphi_1 \gamma_{\ell-1} & \text{if } \ell > 0, \end{cases}$$

where we use that $\gamma_\ell = \gamma_{-\ell}$. From this we can see that the ACF of x_t is

$$\rho_\ell = \varphi_1 \rho_{\ell-1}, \text{ for } \ell > 0.$$

Since $\rho_0 = 1$ the ACF of an AR(1) process can be written as $\rho_\ell = \varphi_1^\ell$. This results implies that the ACF of a stationary AR(1) series has a starting value $\rho_0 = 1$ and declines exponentially with rate φ_1. For an AR(2) process the theoretical ACF for the first two lags are

$$\rho_1 = \frac{\varphi_1}{1 - \varphi_2} \text{ and } \rho_2 = \frac{\varphi_1^2}{1 - \varphi_2} + \varphi_2, \tag{5}$$

and will after the second lag decline exponentially. The ACF for AR(p) processes are derived in the same way according to Eq. (1) and will decline

exponentially after lag p. The sample ACF for a simulated AR(2) process is shown in Fig. 8, where one can observe the exponentially declining ACF after lag 2.

Next we will look at properties for an MA process. Taking the expectation of Eq. (4), we have

$$\mathrm{E}(x_t) = c,$$

which is time invariant. The variance of the same equation will give us

$$\mathrm{Var}(x_t) = (1 + \theta_1^2 + \theta_2^2 + \cdots + \theta_q^2)\sigma_\varepsilon^2.$$

This implies that if we look at the simple case of an MA(1) process the covariance is

$$\gamma_1 = -\theta_1\sigma_a^2 \quad \text{and} \quad \gamma_\ell = 0, \quad \text{for} \quad \ell > 0.$$

This will give us the followong ACF for the MA(1) model

$$\rho_0 = 1, \quad \rho_1 = \frac{-\theta_1}{1+\theta_1^2}, \quad \rho_\ell = 0, \quad \text{for} \quad \ell \geq 2.$$

For an MA(2) process the theoretical ACF is

$$\rho_1 = \frac{\theta_1 + \theta_1\theta_2}{1+\theta_1^2+\theta_2^2}, \quad \rho_2 = \frac{\theta_2}{1+\theta_1^2+\theta_2^2} \quad \text{and} \quad \rho_\ell = 0 \quad \text{for} \quad \ell \geq 3. \qquad (6)$$

As shown above the ACF for an MA(1) model will cut off after lag 1 the ACF for an MA(2) will cut off after lag two. It follows that the ACF for a MA(q) model that the ACF will cut off after lag q, that is $\rho_\ell = 0$ for $\ell > q$. This can be observed for an MA(2) model in Fig. 10 in the Appendix. This implies that the MA(q) time series is only linearly related to the first q lag values.

2.6 Order determination and parameter estimation

In application the order p and q of an AR/MA model is unknown and must be specified empirically. One way of determine the order is by model selection techniques such as the Akaike and Bayesian information criterion (AIC and BIC) methods which are based on maximum likelihood theory. We will in this subsection look at these techniques and discuss when they might not be appropriate.

We want to estimate the model parameter θ from a sample $\{x_t\}$, $(t = 1, ..., T)$. If $x_t \sim f(x;\theta)$ the likelihood function of θ, denoted $L(\theta)$, is given by

$$L(\theta) = f(x_1, ...x_T; \theta) = \prod_{t=1}^{T} f(x_t;\theta),$$

where $f(x_1, ...x_T; \theta)$ is the joint density of $\{x_t\}$. The log likelihood function is defined similarly

$$\ln\, L(\theta) = \sum_{t=1}^{T} \ln\, f(x_t; \theta).$$

The maximum likelihood estimate $\hat{\theta}_{ML}$ is the value of θ that maximizes the log likelihood. We define the estimate

$$\hat{\theta}_{ML} = \arg\max_{\theta} \ln\, L(\theta).$$

For time series samples we assume that the sample values are not i.i.d., hence the conditional density is no longer the product of the marginal densities of x_t. We recall the definition of conditional density that given that a random variable Y takes the value y the conditional density of the random variable X is

$$f_{X|Y}(x; \theta) = \frac{f_{X,Y}(x, y; \theta)}{f_Y(y; \theta)},$$

or equivalent

$$f_{X,Y}(x, y; \theta) = f_Y(y; \theta) f_{X|Y}(x; \theta).$$

When applied to a time series the joint density of $\{x_t\}$ is

$$f(x_1, x_2, ..., x_T; \theta) = f(x_1; \theta) f(x_2|x_1; \theta) ... f(x_T|x_1, x_2, ..., x_{T-1}; \theta).$$

This can also be written as

$$f(x_1, x_2, ..., x_T; \theta) = f(x_1; \theta) \prod_{t=2}^{T} f(x_t|x_1, ..., x_{t-1}; \theta),$$

where $f(x_1; \theta)$ is the marginal density of the very first observation and $f(x_t|x_1, ..., x_{t-1}; \theta)$ is the conditional density of x_t given previous observations. E.g. for an AR(1) process where $x_t = \varphi x_{t-1} + \varepsilon_t$, $(t = 1, ..., T)$ and $\varepsilon_t \sim$ i.i.d. $N(0, \sigma^2)$ we want to estimate $\theta = \theta(\varphi, \sigma^2)$. The conditional density the AR(1) process is

$$f(x_t|x_1, ..., x_{t-1}; \theta) = f(x_t|x_{t-1}; \theta).$$

We also know that $x_t|x_{t-1} \sim N(\mathrm{E}(x_t|x_{t-1}), \mathrm{Var}(x_t|x_{t-1}))$, where $\mathrm{E}(x_t|x_{t-1}) = \varphi x_{t-1}$ and $\mathrm{Var}(x_t|x_{t-1}) = \sigma^2$. It follows that the conditional density is

$$f(x_t|x_{t-1}; \theta) = \frac{1}{\sigma\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(x_t - \varphi x_{t-1})^2}{2\sigma^2} \right\}. \tag{7}$$

The conditional likelihood for an AR(1) process is therefore

$$L(\theta) = f(x_1; \theta) \prod_{t=2}^{T} f(x_t|x_{t-1}; \theta),$$

where $f(x_t|x_{t-1};\theta)$ is given of Eq. (7). The model parameters are estimated so that the likelihood function is maximized but can also be used when comparing two models. If one model have a higher likelihood than another model, then the former fits better to data. An information criterion is a function of the likelihood value and the number of parameters. The two most common information criterions are the AIC and BIC which are defined as

$$\text{AIC} = 2p - 2\hat{\ell} \quad \text{and} \quad \text{BIC} = -2\hat{\ell} + p\ln(T),$$

where p is the number of estimated model parameters, $\hat{\ell}$ the log-likelihood value and T is the sample size. Both information criterions rewards models with a high likelihood but penalizes a model for each added parameter.

According to [Neusser, 2016] another way to determine the order of an AR/MA process, preferred when data can not be assumed e.g. normal and/or sample size is small, is by testing the significance of the ACF . Since the ACF is the correlation between current values of the time series and values at different time points, as described in section 2.1, meaning that is describes the dependence of observations as a function of the time lag between them. In the case of an AR(p) model, the ACF will decline exponentially after lag p and for an MA(q) model the ACF cuts off after lag q as was described in Section 2.5. This can be observed in the ACF for an AR and MA model of order two in Fig. 8 and Fig. 10 which are found in the Appendix. For an AR process it might be more appropriate to look at the *partial autocorrelation function* (PACF), this is due to the fact that for an AR(p) process the PACF cuts off at lag p. In this paper we will limit us to the ACF but the interested reader can find more about the PACF in [Tsay, 2005] Chapter 2.

3 Bootstrap Methods

Two of the most important problems in applied statistics are determination of an estimator and its standard error and the determination of confidence intervals. Without any assumptions made about the asymptotic distribution of the parameter, these problems can be answered with the use of resampling methods as the bootstrap.

The bootstrap method was first published in [Efron, 1979], based on earlier work on the jackknife in [Quenouille, 1949]. The jackknife is a a resampling technique mainly used for variance and bias estimation. The jackknife estimator of a parameter is derived by systematically leaving out each observation from a data set and calculating the estimate and then finding the average of these calculations.

In this section the bootstrap will be presented more in detail followed by description of the four block bootstrap methods considered in this paper.

Lastly we look closer on how to choose the optimal block length in block bootstrap and one way of constructing bootstrap confidence intervals, the *percentile method*.

3.1 Simple bootstrap

In this subsection we consider bootstrap methods that are applicable to samples of data with i.i.d. values. We let X be a sample with sample values $x_1, x_2, ..., x_n$ from an *unknown* population with probability distribution F. The sample is used to make inference about a population characteristic which can be expressed as a parameter θ estimated by the sample denoted $\hat{\theta}$. To each sample value x_j we put equal probabilities n^{-1}, we call this the *empirical distribution* denoted $\hat{F}$. The bootstrap can then be used to answer questions about the probability distribution of θ, like its variance or quantiles, with the use of $\hat{\theta}$.

The basic idea underlying the bootstrap is to recreate the original population with resampling with replacement from the sample. The bootstrap method of [Efron, 1979] suggest that when the sample is small we can estimate the properties we require from simulated data sets with the same size as the original sample, denoted $x_1^*, ..., x_n^*$ and where x_j^* is independently sampled from $\hat{F}$. From a simulated data set we can calculate the parameter of interest, denoted $\hat{\theta}^*$ and repeat this k times. Properties of θ can now be estimated from $\hat{\theta}_1^*, ..., \hat{\theta}_k^*$. The procedure of this method is

1. Generate a sample of size n, same size of the original data set with replacement from the empirical distribution.
2. Compute $\hat{\theta}^*$, the value of $\hat{\theta}$ obtained by using the bootstrap sample in place of the original sample.
3. Repeat steps 1 and 2 k times.

For example, if we let $\bar{\theta}^* = k^{-1} \sum_{i=1}^{k} \hat{\theta}_i^*$, the standard error of $\hat{\theta}$ is

$$\mathrm{sd}(\hat{\theta}) = \sqrt{\frac{1}{k-1} \sum_{i=1}^{k} (\hat{\theta}_i^* - \bar{\theta}^*)^2}. \tag{8}$$

Other similar moments can be derived in the same way. According to [Davidson and Hinkley, 1997] page 16, these approximations of Eq. (8) are justified by the law of large numbers. For $X_n = x_1 + ... + x_n$, a sample of size n with finite mean μ and variance σ^2 the law of large numbers states that

$$P(|\frac{X_n}{n} - \mu| > \epsilon) \to 0 \quad \text{as} \quad n \to \infty, \quad \text{for all} \quad > 0,$$

i.e.,

$$\frac{X_n}{n} \xrightarrow{p} \mu \quad \text{as} \quad n \to \infty,$$

which is shown e.g. in Example 1.3. in [Gut, 1995].

3.2 Bootstrap methods for time series

When there is dependence present in data, which is highly common in the case of a time series, the simple resampling procedure will fail as it is not able to replicate the dependence structure. Since the introduction of the bootstrap, several extensions of the method has been made for it to be applicable for dependent data. One of these extensions are the block bootstap, first introduced in [Carlstein, 1986], other methods are the *residual bootstrap* (a type of model-based bootstrap) and the *autoregressive-sieve bootstrap*. This paper is focused on block resampling but one can read more about bootstrap methods for time series in [Kreiss and Lahiri, 2012] Chapter 1. Block bootstrap tries to create psuedo data and replicate the dependence in data by resampling blocks of data instead of single observations. It has mainly been used when data are dependent in time (time series) but can also be used on data dependent in space or in groups (so called cluster data).

For a stationary series, observations removed sufficiently far in time are uncorrelated. One of the main ideas in block resampling is that, for stationary series, individual blocks of observations that are separated far enough in time will be approximately uncorrelated and can be treated as exchangable. Depending on the method, you select blocks of the time series, either overlapping or not and of fixed or random length, which will generate stationarity in the samples. The resampling is then made with sampling with replacement from the blocks to create a pseudo times series on which you compute your statistic of interest. The key idea in this resampling approach is that if the blocks are sufficiently long the dependence structure of the original data will present also in the pseudo data.

Since the introduction of block resampling several different block bootstrap methods have been developed. The non-overlapping block bootstrap, the moving block bootstrap, the circular block bootstrap and the stationary bootstrap methods are four of the most commonly used methods, a detailed description of these can be found in Section 3.3.

3.3 Block bootstrap methods

Common block bootstrap methods are the non-overlapping block bootstrap, the moving block bootstrap, the circular block bootstrap and the stationary bootstrap methods. These four methods will be described in detail below.

NBB. The non-overlapping block bootstrap was the initial method based on the work by [Carlstein, 1986]. The NBB divides the data set X_n of size n into b non-overlapping blocks of length l, where we suppose $1 < l < n$ and $n = bl$. We define the non-overlapping blocks $B_1, ..., B_b$ of length l contained in X_n as

$B_1 = (x_1, ..., x_l)$,
$B_2 = \qquad\qquad (x_{l+1}, ..., x_{2l})$,
...
$B_b = \qquad\qquad\qquad\qquad (x_{(b-1)l+1}, ..., x_n)$.
The NBB samples are then generated by selecting b blocks at random from the collection $B_1, ..., B_b$. Stitching these blocks together in the order they were picked will give the bootstrapped sample.

MBB. Proposed by [Künch, 1989] and [Liu and Singh, 1992] is the moving block bootstrap method which allows the blocks to overlap. This allows for more blocks than if they are not allowed to overlap. For a data set of n observations the MBB split the data into $N = n - l + 1$ overlapping blocks of length l where we for simplicity suppose that l divides n. Then we can define the MBB blocks from X_n as
$B_1 = (x_1, ..., x_l)$
$B_2 = \quad (x_2, ..., x_{l+1})$
...
$B_N = \qquad (x_{n-l+1}, ..., x_n)$.
From these N blocks, $b = n/l$ blocks will be drawn at random with replacement. The bootstrap sample is given by stitching the b blocks together. An advantage of this method compared to the NBB method is that by allowing overlapping blocks we have a wider range of blocks to sample from. This is especially useful when the sample is small.

CBB. The circular block bootstrap of [Politis and Romano, 1992] has the main purpose to remove the edge effect of uneven weighting of the observations at the beginning and at the end in the MBB method. For example, in the MBB the first sample value will only appear in the first block while the 5^{th} value will appear in five blocks if $b \geq 5$. The CBB blocks are defined as in the MBB method, i.e. overlapping, but uses an end-to-start wrap around of the data around a circle to make additional blocks. This method is also useful when data can be considered circular, e.g. the outside temperature over a year.

SB. The stationary bootstrap, by [Politis and Romano, 1994], differs from the other three earlier mentioned methods in the sense that block length is not fixed but a *random* variable geometrically distributed with expected value l. Because of the random block length, the number of blocks is also random. Similar to CBB, the SB method was introduced with the purpose to remove the uneven weighting of the observations at the beginning and at the end of the sample.

Some other variants of the block bootstrap is the matched block bootstrap (MaBB) in [Carlstein et al, 1998], the tapered block bootstrap (TBB)

[Paparoditis and Politis, 2001], and the generalized seasonal block bootstrap (GSBB) [Dudek et al, 2013]. Both the MaBB and TBB is used to reduce an edge effect among the blocks, MaBB by using a stochastic mechanism to match independent blocks from the MBB method while TBB adjust the boundary values of the blocks towards a common value (like the series mean). The GSBB method is used for periodically correlated time series as the main idea of this method is that it preserves the periodic structure of the time series.

3.4 Choosing block length

To be able to implement any of the block bootstrap methods we need to choose the (expected) block length. The optimal block length depends on the sample size and its correlation structure and differs for different bootstrap methods. According to [Politis, 2003] there are two main approaches in deciding block length, either by cross-validation or plug-in methods. In the cross-validation method we choose a criterion (e.g. the mean squared error) and minimize the estimated criterion to get an estimate of the optimal block size. The plug-in method involve deriving an expression of the optimal value and to plug in all unknown parameters in the expression. The cross-validation method usually involves less analytic work but requires more computation.

In this thesis we will not look further into the issue of selecting the optimal block length but more generally look at the performance of the block bootstrap methods given different block length considered in the simulation study. For more information about selecting the optimal block length the interested reader can read about it in [Lahiri, 2003] Chapter 7.

3.5 Confidence intervals

A commonly used method for interval constructing for resampling methods is the simple percentile interval method in [Efron, 1987]. Since the bootstrap produces a large sample from the sampling distribution of a statistic, a way to generate confidence intervals for this statistic is to take the quantiles from this sample. If we let $\hat{\theta}^*_{(\alpha/2)}$ be the $100\alpha/2^{\text{th}}$ percentile of the parameter estimate of interest from B resampling replications. The percentile interval with coverage 1-α is obtained by the percentiles $[\hat{\theta}^*_{(\alpha/2)}, \hat{\theta}^*_{(1-\alpha/2)}]$. E.g. using a bootstrap method we create a replication sample to compute our chosen bootstrap parameter estimate $\hat{\theta}$ k times. If we choose α=0.05 and k=1000 the 25^{th} and 975^{th} value of the ordered estimates will form a confidence interval for the estimate $\hat{\theta}$ of confidence level 95 %. Different from standard asymptotic confidence intervals, the bootstrap percentile intervals will not be symmetric around the parameter estimate, this is useful for cases when the true sampling distribution is not symmetric.

4 The simulation study

We carried out a simulation study to evaluate and compare the performance of the MBB, the NBB, the CBB, and the SB methods considered in Section 3.3 under some linear time series models. The following sections contains the simulation design and the simulation results that provide a comparison of the four block bootstrap methods in terms of their accuracy in model parameter estimation, the estimates standard deviation and confidence intervals.

4.1 Simulation design

Data were simulated from a set of moving average (MA(2)) and autoregressive (AR(2)) models using the arima.sim function in R. AR series were generated from AR models $x_t = \varphi_1 x_{t1} + \varphi_2 x_{t2} + \varepsilon_t$ with AR parameters φ_1 and φ_2. Similarly, MA series were generated from MA models $x_t = \theta_1 \varepsilon_{t1} + \theta_2 \varepsilon_{t2} + \varepsilon_t$ with MA parameters θ_1 and θ_2. The parameters of both models were chosen at random but so that the series were stationary and that in one of each model there was a negative parameter. All models have a mean of zero and the innovations ε_t are white noise series with observations independently distributed $N(0, 1)$. The models are found in Table 1. The model autocorrelations are the theoretical ACFs, defined in Eq. (5) for the AR(2) model and Eq. (6) for the MA(2) model. Figure 7-10 in the Appendix illustrate simulated time series of one AR model and one MA model, namely M1 and M3 with their respective sample ACF, computed with the acf function in R.

Table 1. Parameter values used in the simulation

Models		AR parameters		Autocorrelation	
		φ_1	φ_2	ρ_1	ρ_2
AR(2)	M1	0.2	0.4	0.333	0.467
	M2	0.5	-0.3	0.384	-0.108
		MA parameters		Autocorrelation	
		θ_1	θ_2	ρ_1	ρ_2
MA(2)	M3	0.2	0.3	0.230	0.265
	M4	-0.3	0.4	-0.336	0.320

Two influential factors of the general performance of resampling methods were considered, length of the time series and block length. The replicated time series, generated by the bootstrap methods, have the same length n as the simulated time series. The model parameter of interest θ have been estimated from $k = 1000$ bootstrap time series to obtain the empirical distribution. The standard error of the parameter estimate is computed from the k estimates $\hat{\theta}^*$ using Eq. (8). Confidence intervals are made with the percentile method considered in Section 3.5 with chosen confidence level α.

1000 time series (as many as the bootstrap estimates) from the chosen models M1-M4 are simulated from which we derive parameter estimates which are used as a bench mark (BM). We use the bench mark as a point of reference as no better estimate can be made than the ones derived from the true distribution, hence the bench mark parameters are the indicator of the true model parameters.

The *bias* of an estimator is the difference between the expected value of the estimator and the true value of the parameter being estimated and can be used to mesure the accuracy of an estimator. We define the bias

$$\text{Bias}[\hat{\theta}^*] = E[\hat{\theta}^*] - \theta,$$

where in this simulation study $\hat{\theta}^*$ is the mean value of the bootstrap estimates and θ is the mean value of the bench mark estimates.

In the first part of the simulation study we will use a p-value based on

$$\hat{p}(\theta) = \frac{1}{T}\sum_{i=1}^{T} \mathbf{I}(\theta^* \leq \theta), \tag{9}$$

where I$(\cdot)$ is the *indicator function*, with value 1 when its argument is true and 0 if not true.

First we compared the behavior of the block bootstrap methods of estimating the sample mean. According to [Lahiri, 2003] page 115, even though the simulation treats the simple case of sample mean, it provides a representative picture of the properties of the four methods also in more general problems. We looked on the performances using different block lengths for a fixed set of observations as well as by fix the block length but alter the number of observations. Next we looked at the accuracy of estimating more complex parameters as the p/q values and ACF-functions of the AR/MA models simulated.

4.2 Simulation results

We initially compare the performance of the MBB, the NBB, the CBB, and the SB methods in terms of making a confidence interval including the true value of the sample mean. The true value of the sample mean is zero for all simulated time series. For each bootstrap estimator 1000 replicated time series were used to create a confidence interval. P-values are constructed by the indicator function in Eq. (9) of 500 confidence intervals to which the value 1 is assigned to each confidence interval that contains the true value and 0 in the other case. From the simulated time series we create a confidence interval of chosen level of significance $\alpha = 0.05$. If the true parameter

θ is in the interval $[\hat{\theta}^*_{(\alpha/2)}, \hat{\theta}^*_{(1-\alpha/2)}]$ 400 times of the 500 simulation runs the p-value is 0.80. We fix the number of observations at $n = 1000$ and vary the block length between 1-20. Table 2 show the results for model M1 with 4 different block lengths. In the case of the SB method it is the expected block length we have considered.

Table 2. Fixed number of 1000 observations for M1.

Block length	NBB	MBB	CBB	SB
1	0.778	0.768	0.770	0.809
4	0.887	0.897	0.901	0.895
10	0.920	0.940	0.928	0.908
20	0.909	0.936	0.915	0.886

Since $\alpha = 0.05$ the coverage ratio is 95 % if the bootstrap methods are preforming well. When the block length is one it is the simple bootstrap where single elements are sampled with replacements, this means that it will not take into account the dependence structure of the time series. As we can see from Table 2 the p-values of this block length is low and we can conclude that the simple bootstrap preforms poorly in estimating the sample mean. We can from these results also see that for this time series of size 1000 the block length $l = 10$ is the best choice of the tested block lengths as it for any of the methods shows the highest p-value. Overall the MBB and CBB methods preforms with a higher accuracy then the two other methods.

The next question we want to answer is how sample size affects the coverage of probability. In this simulation the same underlying model M1 were used but the block length was fixed at $l = 10$ and the sample size n of the time series varied. The results are shown in Table 3. We observe that a

Table 3. Fixed block length of 10 for M1.

Observations	NBB	MBB	CBB	SB
50	0.780	0.818	0.824	0.760
100	0.850	0.868	0.854	0.856
500	0.906	0.916	0.914	0.928
1000	0.920	0.940	0.928	0.908
2000	0.930	0.912	0.918	0.916

larger sample not always leads to a higher accuracy of the prediction of the sample mean in the cases considered. From Section 3.4 we know that the optimal block length depends on the sample size so it is possible that we would have been able to achieve a higher p-value with a sample size of n=2000 using a larger block length. However we can see that the overall performance of the MBB and CBB is higher than for the NBB and SB methods in the

cases considered. The same observations were made with simulated data from M2, M3 and M4.

In the following simulations we will investigate the ability of the bootstrap methods to estimate the ACF and the model parameters. We compare the methods parameter estimates by their confidence interval, standard error and bias. We use the acf and arima functions in R to get the parameter estimates for each replicated time series. A sample size of $n = 1000$ and block length $l = 10$ has been used in these simulations. Figure 1 and 2 show the results of the ACF estimates for lag 0-5 of M1 and M2 with 95 % confidence intervals. The performance under the MA models can be seen in

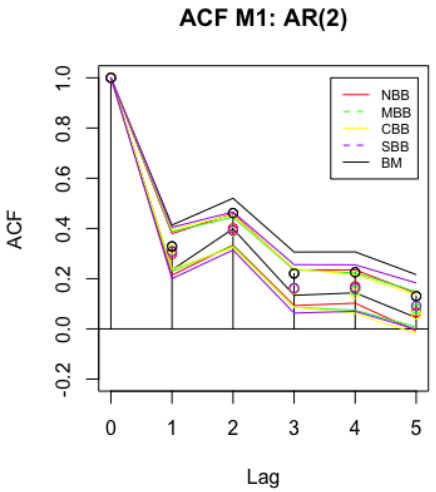


Figure 1: Comparison of ACF estimates for Model 1.

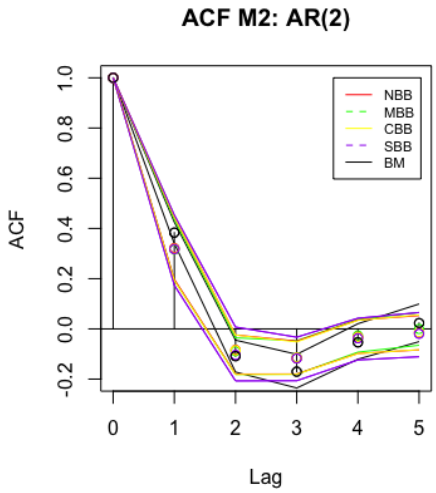


Figure 2: Comparison of ACF estimates for Model 2.

Fig. 11 and 12 in the Appendix. The simulation indicates that for any given bootstrap method there will be a weaker correlation among the observations than in the bench mark values. That is if the autocorrelation is positive we well see a less positive autocorrelation and if the autocorrelation is negative it will be less negative. There are methods that would possibly strengthen the dependence of the replicated time series, such as other bootstrap methods as the MaBB and TBB mentioned in Section 3.3 and post-blackening (more about this is found in [Srinivas and Srinivasan, 2000]). However we have not considered any of these methods in this thesis as we just focus on comparing the four methods, not finding the best method to replicate a time series. Further we can observe that the SB method appears to result in the largest confidence interval. In terms of the standard deviation of the bootstrap estimator we find the result for each lag and method for Model 1

Table 4. Standard error of ACF estimates in Model 1.

Method/Lag	1	2	3	4	5
NBB	0.0428	0.0316	0.0373	0.0369	0.0383
MBB	0.0400	0.0302	0.0365	0.0368	0.0356
CBB	0.0411	0.0319	0.0384	0.0362	0.0398
SB	0.0451	0.0379	0.0408	0.0396	0.0413

in Table 4, where we can see that the MBB method provides the estimate with the overall lowest standard error and the SB method the highest. Similar observations were made for Model 2-4. The bias of the ACF estimates of Model 1 are found in Table 5. We observe that for all lags there is a underestimation of the autocorrelation as the bias for all ACF estimates are negative. The bias tend to vary and it is not possible to see that any method have a smaller or larger bias than any other. The MBB and CBB methods appear to have similar pattern in terms of the estimate bias.

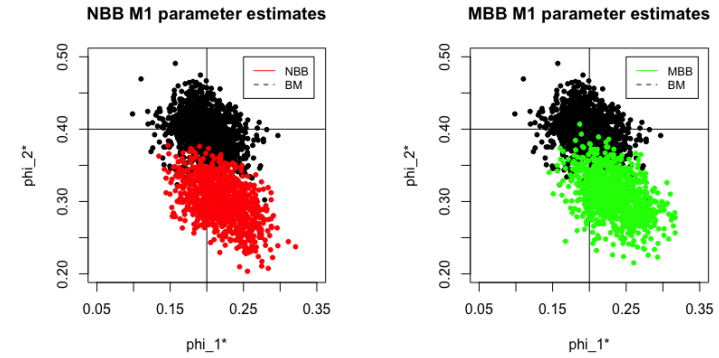
Table 5. Bias of ACF estimates in Model 1.

Method/Lag	1	2	3	4	5
NBB	-0.0307	-0.0606	-0.0608	-0.0757	-0.0646
MBB	-0.0136	-0.0712	-0.0581	-0.0720	-0.0501
CBB	-0.0128	-0.0724	-0.0611	-0.0761	-0.0728
SB	-0.0204	-0.0698	-0.0580	-0.0628	-0.0387

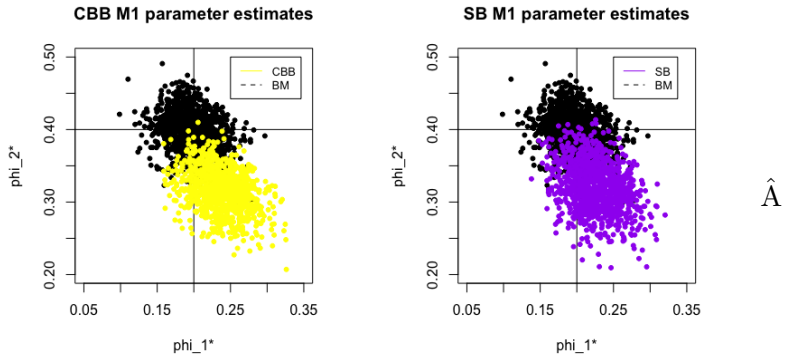
Next we will see the performance of the bootstrap methods in estimating the parameters of the AR/MA models. We try to refit the pseudo time series generated from the bootstrap methods to the models they were simulated from. Assuming that the order p/q of the models are known to be 2 we fit the models by maximum likelihood estimation using the arima function in R. Fig. 3 show the results of the four different block bootstrap methods estimating the two parameters of Model 1. Results for the other models are found in the Appendix, Fig. 13-15. The black dots represent the bench mark parameter estimates and the black lines are the parameter values from which the models are simulated from. We can conclude that the process of fitting the models using the arima function in R is accurate as the parameters derived from the BM time series are centered around the true parameters from which the they were simulated from.

The result from the simulations indicate that the bootstrap estimates for all methods and models are misspecified to some extent as they are not centered around the true values. A summary of the results of Model 1 in terms of the parameter estimators confidence interval, standard error and bias is found in Table 6. The NBB and CBB methods appears to preform similarly and have the smallest standard error and the SB method the highest standard error. In terms of the estimators bias we can not draw any

conclusions. For Model 1 we can observe an overestimation of parameter one and underestimation of parameter 2, however this observation can not be made for the other parameter estimates of Model 2-4 .



(a) NBB estimates. (b) MBB estimates.



(c) CBB estimates. (d) SB estimates.

Figure 3: Parameter estimates of Model 1.

Table 6. Coefficient estimates for M1.

Method	NBB	MBB	CBB	SB
φ_1*	0.195 (0.134, 0.254)	0.213 (0.158, 0.272)	0.213 (0.153, 0.273)	0.209 (0.148, 0.278)
φ_2*	0.3449 (0.274, 0.401)	0.3225 (0.262,0.378)	0.3226 (0.258, 0.381)	0.3255 (0.256, 0.393)
sd(φ_1^*)	0.0308	0.0302	0.0304	0.0330
sd(φ_2^*)	0.0323	0.0297	0.0305	0.0349
Bias(φ_1^*)	0.00341	0.00515	0.00641	0.00409
Bias(φ_2^*)	-0.0707	-0.0148	-0.0148	-0.0109

To capture the simultaneous effect of both parameter estimates we can look at the autocorrelation. From the parameter estimates we can derive the theoretical autocorrelation for lag one and two for the AR and MA models. We use the estimates from the previous simulation to compute ρ_1^* and ρ_2^* according to Eq. (5)-(6). We see the density plot of ρ_1^* and ρ_2^* of Model 1 in Fig. 4 and 5. The density plots of the two first autocorrelations of Model 2-4 is found in the Appendix, Fig. 16-21. In the figures the black vertical line is the theoretical value found in Table 1. For Model 1 we can see that we get a slightly lower autocorrelation estimate of lag 1 than the theoretical value, and even more so for the second lag. This can also be observed in Fig. 2. This observation can not be made in the other models M2-M4, however we can conclude that there is an underestimation of the autocorrelation. That is when the autocorrelation is positive the bootstrap estimates are less positive and when negative they are less negative.

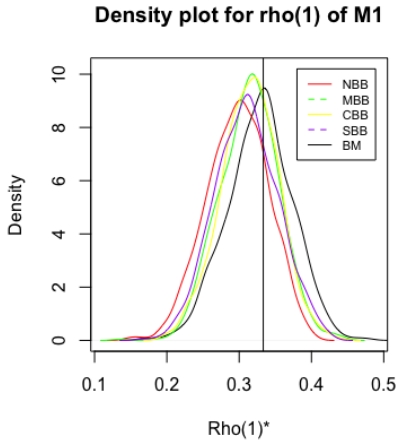


Figure 4: Estimation of autocorrelation of lag 1.

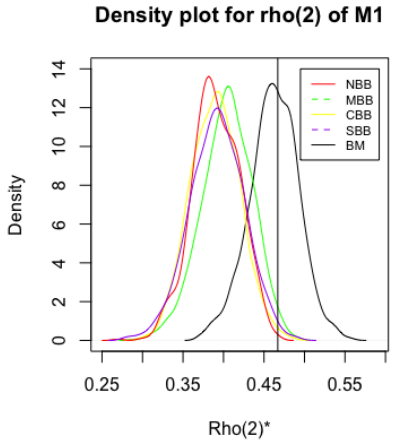


Figure 5: Estimation of autocorrelation of lag 2.

Since the MBB method appears to be preforming well we will look at the performance of the method under different block lengths to illustrate how different block length manage to capture the autocorrelation of lag 1. We considered five different block lengths between 1-50. The result is illustrated in Fig. 6. We can observe that when block length is 1, meaning the simple bootstrap, the autocorrelation of lag 1 is estimated to be zero. This implies that the correlation structure in the sample is completely lost in the pseudo samples which now appears to be white noise. When block length is 5 we can observe some correlation and after block length 10 there appear to be some convergence in how well the bootstrap method is able to capture the autocorrelation of lag 1.

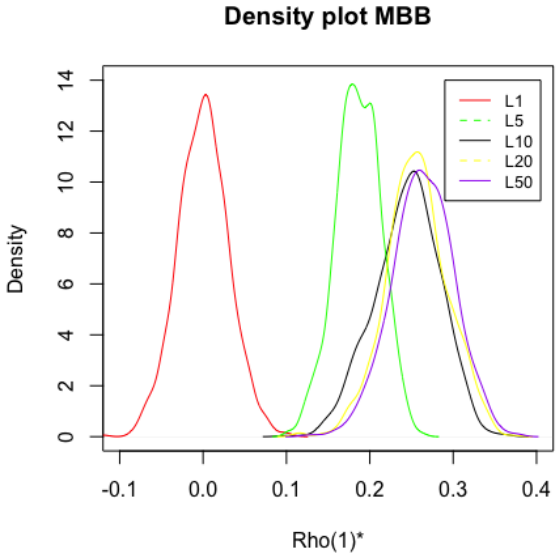


Figure 6: Autocorrelation estimate of lag 1 over various block lengths.

5 Conclusion

Resampling methods are commonly used to make inference about model parameters. There are several resampling methods that can be used when data is correlated in time such as block bootstrap methods. This thesis has focused on four different block bootstrap methods, the non-overlapping block bootstrap, overlapping block bootstrap, circular block bootstrap and stationary bootstrap. The main idea of these resampling methods are that by retaining the neighboring observations in blocks the dependence structure of the original sample is preserved in the bootstrap pseudo time series.

Through a simulation study we found that the the pseudo data generated from the bootstrap methods always showed a weaker dependence among the observations than the time series they were sampled from, hence we can draw the conclusion that even by resampling blocks instead of single observations we will loose some of the structural from of the original sample. Changing the block length and sample size affects overall results, but at the same time it does not make a different decision about the comparison among bootstrap relative performances under the cases considered in this study. We find that overall the MBB and the CBB methods give the bootstrap estimators with the smallest standard error and the SB method the largest. In terms of the bias of the bootstrap estimators there is no method outperforming the other.

We have in this thesis looked at some applications of the bootstrap meth-

ods and ways to illustrate and compare their efficiency. However, there is a comparison problem of measuring the efficiency since there is different optimal block lengths for each sample size and method. In this paper we did not look further into the optimal block lengths for different samples and we limited us to a few sample sizes. Nor did we consider the performance of the block bootstrap methods under non linear time series, which is a possible topic for further studies in this matter.

The conclusion of this thesis is that bootstrap procedure selection is not an easy task and should be made with caution. Sample size and dependence structure plays a role in selecting the optimal bootstrap procedure. We can also conclude that for any block bootstrap method and sample size considered in this paper the pseudo data showed a lost or misspecified dependence among the observations than the data it was sampled from.

References

[Carlstein, 1986] Carlstein, E. (1986). The use of subseries methods for estimating the variance of a general statistic from a stationary time series. *Ann. Statist.* Vol 14, 1171-1179.

[Carlstein et al, 1998] Carlstein, E., Do, K-A., Hall, P. Hesterberg, T. and Künch, H. R. (1998). Matched-Block Bootstrap for Dependent Data. *Bernoulli.* Vol 4, No. 3. (Sep., 1998), 305-328.

[Chernick, 1979] Chernick, M. R. (1999). *Bootstrap Methods: A Practioner's giude.* John Wiley and Sons, Inc.

[Davidson and Hinkley, 1997] Davidson, A. C. and Hinkley, D.V. (1997). *Bootstrap methods and their application.* Cambrige University Press.

[Dudek et al, 2013] Dudek, A., Jacek LeÅkow, J., Paparoditis, E. and Politis, D. (2014). A generalized block bootstrap for seasonal time series. *Journal of Time Series Analysis.* Vol 35, Issue 2, 89-114.

[Efron, 1979] Efron, B. (1979). *Bootstrap methods: another look at the jackknife.* Ann. Statist. Vol 7, No. 1.

[Efron, 1987] Efron, B. (1987). Better bootstrap confidence intervals. *Amer. Statist. Assoc.* Vol 82, 171-185.

[Efron and Tibshirani, 1994] Efron, R. and Tibshirani, J. (1994). *An Introduction to the Bootstrap.* New York : Chapman Hall.

[Gut, 1995] Gut, A. (1995). *An Intermediate Course in Probability.* Springer.

[Kreiss and Lahiri, 2012] Kreiss, J-P. and Lahiri, S. N. (2012) *Time Series Analysis: Methods and Applications.* Vol 30, North Holland.

[Künch, 1989] Künsch, H. R. (1989) The jackknife and the bootstrap for general stationary observations. *Ann. Statist.* Vol 17, 1217-1261.

[Lahiri, 2003] Lahiri, S. N. (2003). *Resampling Methods for Dependent Data.* Springer.

[Liu and Singh, 1992] Liu, R. Y. and Singh, K. (1992). *Moving blocks jackknife and bootstrap capture weak dependence.* In *Exploring the Limits of Bootstrap* (R. Lepage and L. Billard, eds.) 225-248. Wiley, New York.

[Neusser, 2016] Neusser, K. (2016). *Time Series Econometrics.* Springer.

[Paparoditis and Politis, 2001] Paparoditis, E. and Politis, D. (2001). Tapered Block Bootstrap. *Biometrika*. Vol 88, no 4 (Dec., 2001), 1105-1119.

[Politis and Romano, 1992] Politis, D. and Romano, J. P. (1992) *A circular block resampling procedure for stationary data.* In *Exploring the Limits of Bootstrap* (R. Lepage and L. Billard, eds.) 263-270. Wiley, New York.

[Politis and Romano, 1994] Politis, D. and Romano, J. P. (1994) The stationary bootstrap. *Amer. Statist. Assoc.* Vol 89, 1303-1313.

[Politis, 2003] Politis, D. (2003) The Impact of Bootstrap Methods on Time Series Analysis. *Statistical Science*. Vol 18, no 2, 219-230.

[Srinivas and Srinivasan, 2000] Srinivas, V. and Srinivasan, K. (2000). Post-blackening approach for modeling dependent annual streamflows. *Journal of Hydrology*. Vol 230 (April, 2000), 86-126.

[Tsay, 2005] Tsay, R. S. (2005). *Analysis of financial time series*, Second edition. John Wiley and Sons, Inc.

[Quenouille, 1949] Quenouille, M. (1949). Approximate tests of correlation in time series. *Journal of the Royal Statistical Society*. Ser. B, B11, 18-84.

Appendix

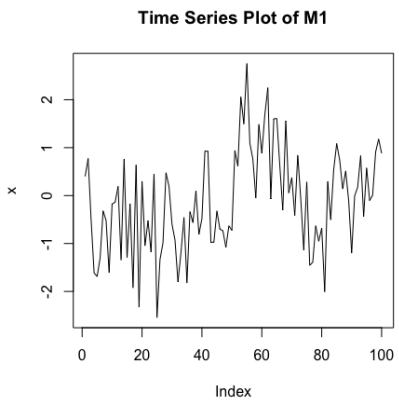


Figure 7: Simulated AR(2) model with parameters $\varphi_1 = 0.2$ and $\varphi_2 = 0.4$.

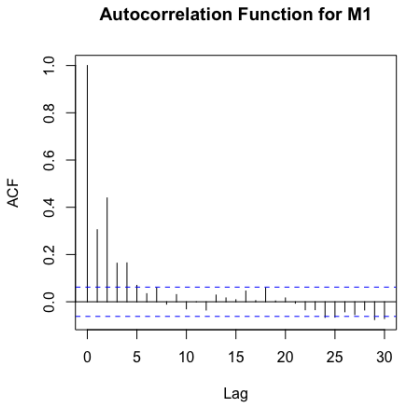


Figure 8: The ACF of an AR(2) model with parameters $\varphi_1 = 0.2$ and $\varphi_2 = 0.4$.

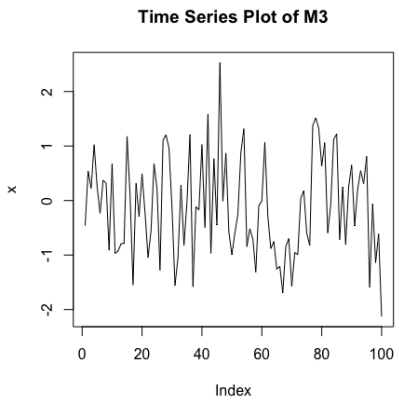


Figure 9: Simulated MA(2) model with parameters $\theta_1 = 0.2$ and $\theta_2 = 0.3$.

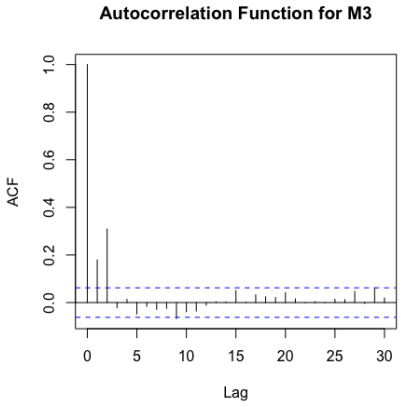


Figure 10: The ACF of a MA(2) model with parameters $\theta_1 = 0.2$ and $\theta_2 = 0.3$.

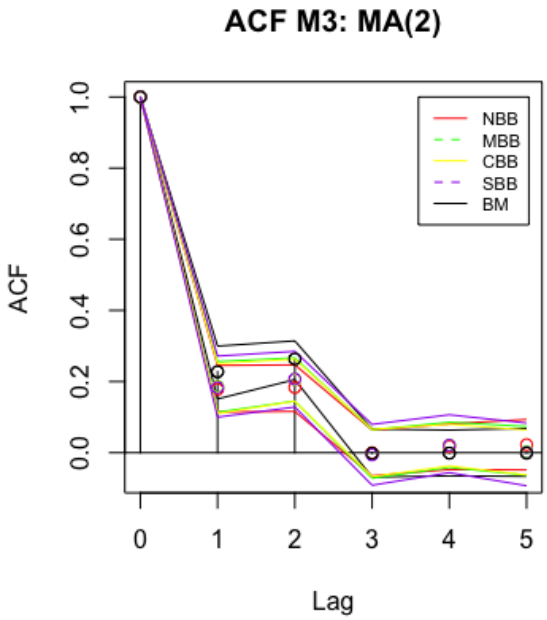


Figure 11: ACF estimates of Model 3.

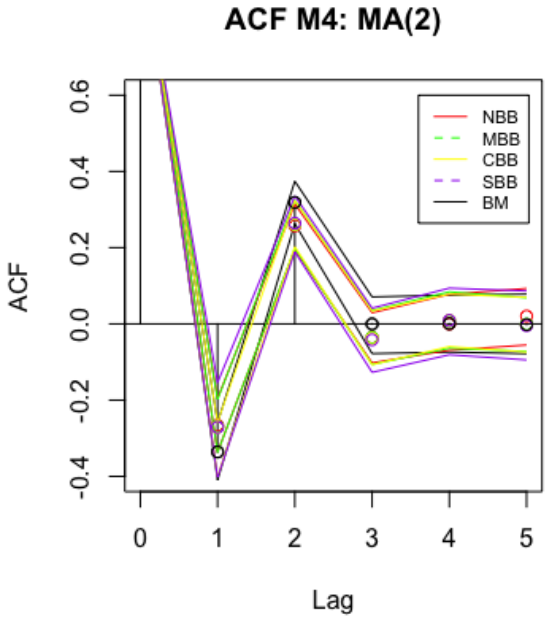
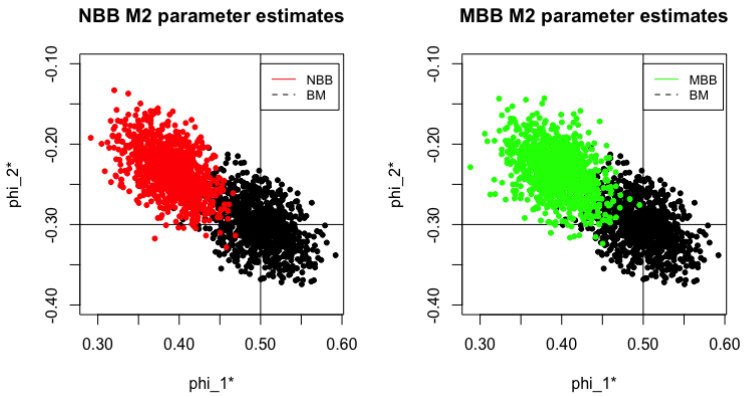
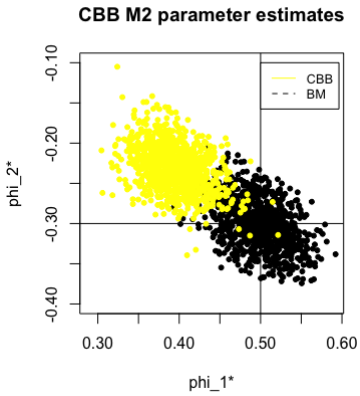


Figure 12: ACF estimates of Model 4.

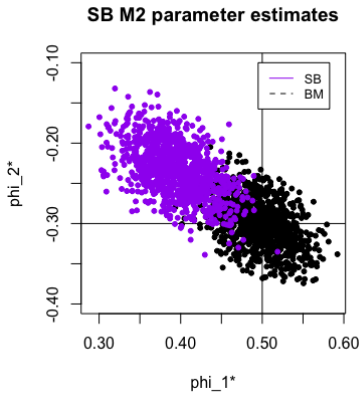


(a) NBB method.

(b) MBB method.

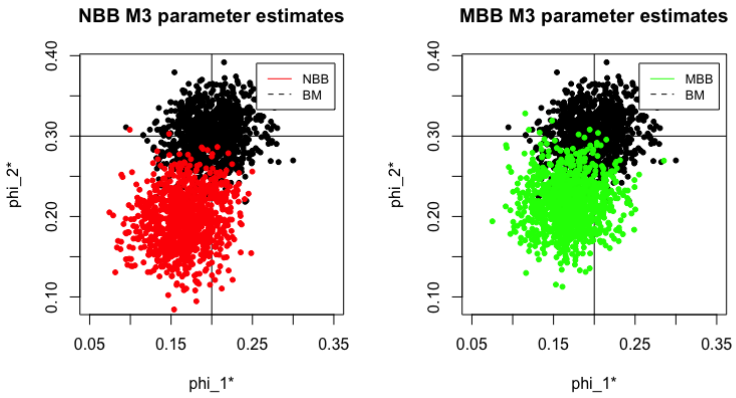


(c) CBB method.



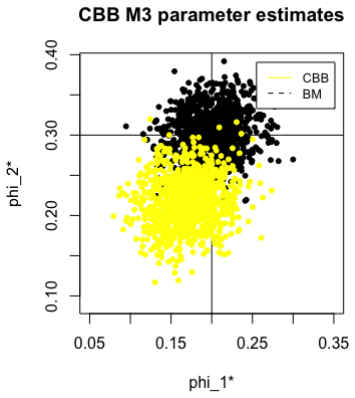
(d) SB method.

Figure 13: Estimation of parameters in an AR(2) model.

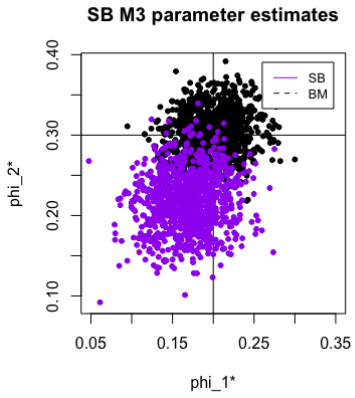


(a) NBB method.

(b) MBB method.

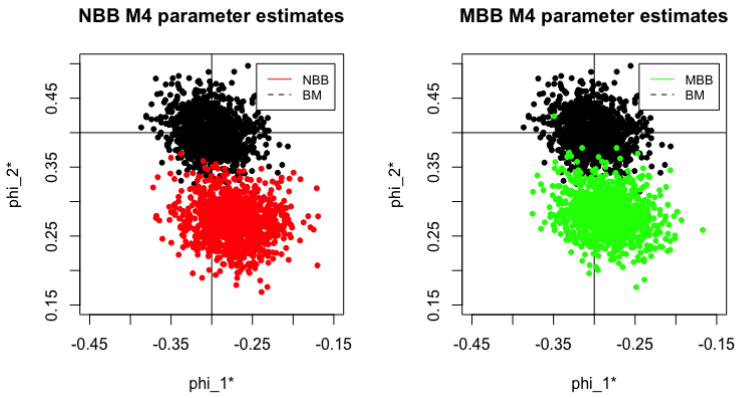


(c) CBB method.



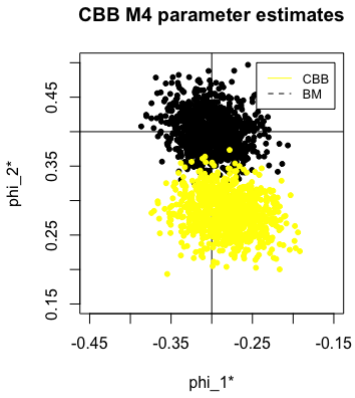
(d) SB method.

Figure 14: Estimation of parameters in an MA(2) model.

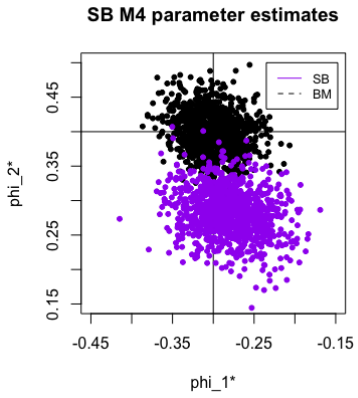


(a) NBB method.

(b) MBB method.



(c) CBB method.



(d) SB method.

Figure 15: Estimation of parameters in an MA(2) model.

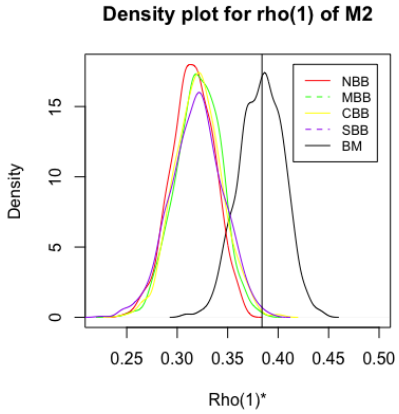


Figure 16: Estimation of autocorrelation of lag 1.

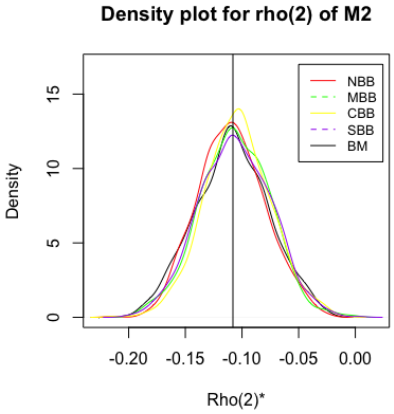


Figure 17: Estimation of autocorrelation of lag 2.

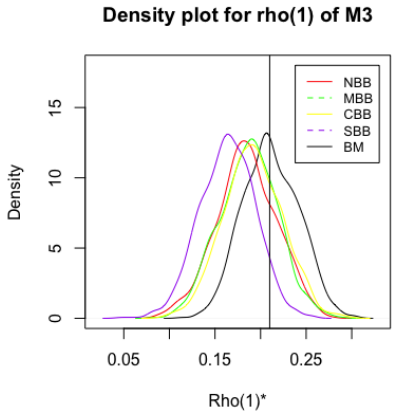


Figure 18: Estimation of autocorrelation of lag 1.

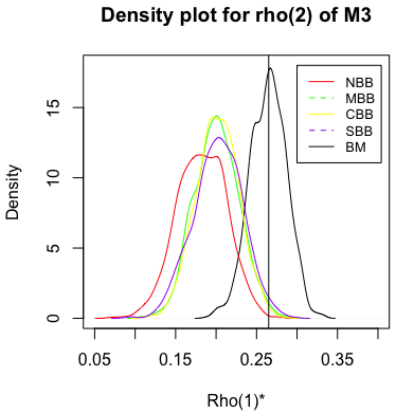


Figure 19: Estimation of autocorrelation of lag 2.

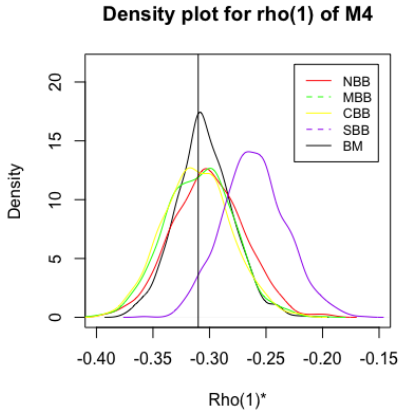


Figure 20: Estimation of autocorrelation of lag 1.

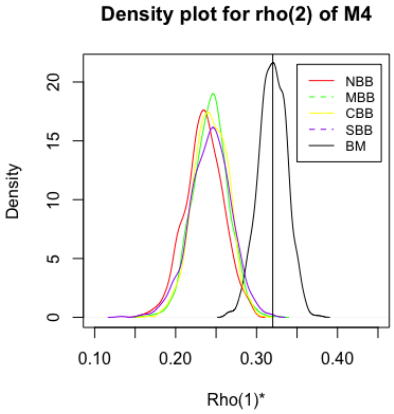


Figure 21: Estimation of autocorrelation of lag 2.