University College Cork

MS4090 - Mathematical Sciences Project

Estimation of Non-Stationary, Linear Time series

Author
C. Clery - 118307903

Project Supervisor
Prof. B. Hanzon

April 26, 2022

Contents

1 Abstract 3

2 Background 4

3 Equivalence of ARMA(p,q) and AR(∞) 7

3.1 AR(∞) of an ARMA(1,1) 7

3.2 Generalisation to ARMA(p,q) 8

3.3 Conditions of AR(∞) 9

4 Estimation of AR(p*) 11

4.1 Equating Conditional Likelihood and Sum of Squares of an AR(p*) process 11

4.2 Strong Consistency of AR(p) by Sum of squares 13

4.3 Standard errors of stationary ARMA coefficient estimates 14

5 Convergence of AR(p*) coefficient estimates 15

5.1 Convergence of auto-regressive parameter estimates 16

5.1.1 Convergence of purely explosive ARMA models 16

5.1.2 Extension to ARMA(1,1) and Mixed root ARMA(2,1) 18

5.2 Convergence of moving average parameter estimates 19

5.2.1 Convergence with respect to p* 19

5.2.2 Convergence with respect to T 21

5.2.3 Variance of coefficient estimates with respect to θ^s 21

6 Using parameter estimates to create stationary time series 23

6.1 Purely Explosive auto-regressive component 23

6.1.1 Example: ARMA(2,1) 24

6.2 Mixed root autoregressive component 26

6.2.1 Example: ARMA(2,1) 27
6.3 On the AR(p*) and filtering methods 28

7 Computational errors and inaccuracies **32**
7.1 Non-Stationary Data and Numerical accuracies 32
7.2 Optimisation errors 33

8 Conclusions and Further Research **34**
8.1 Theoretical Findings 34
8.2 Applications to Forecasting 34
8.3 Generalisation to higher order ARMA processes 35

A Examples of Code **37**
A.1 Data Generating Process (DGP) 37
A.2 Estimation Criterion 38
A.3 Monte Carlo Simulation 39
A.4 Filtering of purely explosive model 41
A.5 Filtering unstable root from a Mixed Root AR component 41
A.6 Multivariate Grid Search 42

Abstract

This project tackles the maximum likelihood estimation of non-stationary, linear ARMA processes with a Gaussian, homoscedastic error process.

Theory and estimation procedures for the corresponding stationary ARMA process and non-stationary special case of ARIMA (Integrated) processes are well established and in widespread use. However, many of the the theories and methods surrounding the stationary ARMA process do not generalise to the non-stationary ARMA case.

The motivation for this paper has its roots in the work of Prof. Bernard Hanzon (University College Cork) and Prof. Wolfgang Scherrer (TU Vienna). Specifically, their finding that for non-minimum phase State-Space models, the negative log-likelihood has an infimum of negative infinity and hence, the Maximum-Likelihood Estimator does not exist! (Hanzon & Scherrer , 2019).

In this project we provide an alternative approximation to an more elementary non-stationary time series process. By expressing the ARMA(p,q) model in its equivalent AR(∞) form, we create an equivalent AR process. For some p^* large, we truncate the infinite AR process to form a process which, although not equivalent, may form a close approximation.

Support for this approximation arises from the work of Lai and Wei (1982), who proved strong consistency of coefficient estimates of an AR(p) process under Sum of Squares estimation. We draw on the equivalence of Conditional-Maximum likelihood and sum of squares estimation in relating these results.

We form in chapter 5 a number of hypotheses surrounding the convergence of the autoregressive and moving average coefficients under the Ar(p^*) method. Building upon this in chapter 6 we develop the AR(p^*) method such to remove the source of non-stationarity in the process, creating stationary data for which theory and estimation procedures are well understood.

Background

We begin by defining the object of interest and specify our definitions surrounding stationary of a scalar ARMA process.

section

Definition 1 (Auto-regressive Moving-Average process)**.** An Auto-regressive Moving-Average (ARMA(p,q)) process with gaussian, homoscedastic error process $\{u_t\}$ is defined as

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + u_t + \theta_1 u_{t-1} \theta_2 u_{t-2} + \cdots + \theta_q u_{t-q} \tag{2.1}$$

for $t \in \mathcal{Z}, t \leq T$ and $u_t \overset{\text{i.i.d}}{\sim} N(0, \sigma^2)$. p, q are positive integers and are defined as the order of the autoregressive (AR) and moving-average (MA) components respectively. Other definitions may include an intercept term μ. For the purpose of this paper, assume $\mu = 0$.

Definition 2 (Characteristic polynomial of the AR and MA components)**.** The ARMA(p,q) process as in definition 2.1 can equivalently be denoted as

$$\phi(L) y_t = \theta(L) u_t \tag{2.2}$$

where $\phi(L), \theta(L)$ are the characteristic polynomials of the AR and MA components and L is the Lag operator.

$$\begin{aligned} L y_t &= y_{t-1} \\ \phi(L) &= 1 - \phi_1 L - \phi_2 L^2 - \cdots - \phi_p L^p \\ \theta(L) &= 1 + \theta_1 L + \theta_2 L^2 + \cdots + \theta_q L^q \end{aligned}$$

Definition 3 (Stationary Process)**.** The time series is said to be stationary if its first and second moments are time invariant and the autocovariance function depends only on the difference in time h and not t. Formally, a stochastic process y_t is *stationary* if

$$E[y_t] = \mu \ \forall \ t \tag{2.3}$$

and

$$E[(y_t - \mu)(y_{t-h} - \mu)] = \gamma_h = \gamma_{-h} \text{ for all } t, h \in \mathcal{Z} \tag{2.4}$$

A Non-Stationary process is one which violates one, or both of the conditions 2.3,2.4. (Lütkepohl, 2019 , p.24)

It is worth noting at this point that Stationarity (or non-Stationarity) is a characteristic of the time series **data**. From the notion of stationarity arise stablility of an AR process.

Definition 4 (Stability)**.** An ARMA process as defined in 2.2 is said to be *stable* if the characteristic polynomial of the AR process as roots which are strictly outside the unit disk of the complex plane.

$$\phi(L) = 1 - \phi_1 L - \phi_2 L^2 - \cdots - \phi_p L^p \neq 0 \text{ for } |L| \leq 1 \tag{2.5}$$

An unstable AR process generates a series which is non-stationary. (Lütkepohl, 2019 , p.24)

We will throughout refer to *purely explosive* ARMA process. This can be understood to be a process for which the roots of the AR characteristic polynomial *strictly inside* the complex unit circle. Similarly, a *mixed root* process can be understood to be a process for which *at least one but not all* of the roots of the AR characteristic polynomial inside the complex unit circle.

Definition 5 (Invertible)**.** An ARMA process as defined in 2.1 is said to be *invertible* if

$$\theta(L) = 1 + \theta_1 L + \theta_2 L^2 + \cdots + \theta_p L^p \neq 0 \text{ for } |L| \leq 1 \tag{2.6}$$

(Lütkepohl, 2019 , p.421)

An invertible ARMA (or MA) process is called named this way as, if the condition 2.6 is satisfied the MA process $\theta(L)$ can be expressed as an equivalent AR process:

$$\theta(L)^{-1} y_t = u_t$$

In the MA(1) case this clearly be understood by expressing the process as an AR(∞).

$$y_t = u_t + \sum_{i=1}^{\infty} \theta y_{t-i}$$

The AR characteristic polynomial of this process is therefore

$$\phi(L) = \sum_{i=1}^{\infty} (\theta L)^i \tag{2.7}$$

Equation 2.7 is an infinite geometric sum and so is convergent if and only if $|\theta| < 1$. If this condition is satisfied we have that

$$\phi(L) = \sum_{i=1}^{\infty} (\theta L)^i = \frac{1}{1 - \theta L} = \theta(L)^{-1}$$

and hence, the MA process is invertible. If $\theta \geq 1$ the series is not cauchy and there is no such $\theta(L)^{-1}$. ie. The MA process is not invertible.

The *spectral factorisation problem* has been studied extensively. Papers on the topic such as that of Hallin (1984) and following work prove the non-uniqueness of autocovariance functions resulting from an MA process. From studies of the spectral factorisation problem we have the following result.

Theorem 1 (Spectral Factorisation of Moving Average Process)**.** For each non-invertible MA(q) process $\theta(L)u_t$, there exists a unique invertible MA(q) process $\theta^s(L)u_t^s$ such that the autocovariance functions of each process are equal and the processes are equivalent under this condition.

In the MA(1) case, it can be shown that for any non-invertible process $\theta(L)u_t$, the equivalent invertible process is given by

$$\theta^s = \frac{1}{\theta} \qquad u_t^s \sim \mathcal{N}\left(0, \theta^2\sigma^2\right)$$

MA processes differ from AR in that invertiblility of an MA process, unlike stability of an AR process, is not required for stationarity. All MA processes are stationary. Further, each MA(q) process can be described and estimated as an invertible MA(q) process. Throughout this paper the notation θ^s or $\theta^s(L)$ is used to denote the invertible MA process under estimation, often in the case where the data generating process is the equivalent non-invertible process.

Equivalence of ARMA(p,q) and AR(∞)

As previously outlined, the estimation procedure we are detailing in this project involves estimating an ARMA(p,q) process in it's equivalent AR(∞) form. The AR(∞) can be expressed as an infinite sum of the ARMA(p,q) coefficients and the infinite set observations $\{y_{t-i}\}_{i=1}^{\infty}$. The existence, assumptions and implications of such a model are not well defined when the ARMA process is non-stationary. We will discuss this in more detail to conclude this chapter. We begin, however, by deriving the AR(∞) form of an ARMA(1,1) process and provide a procedure for a generalised procedure for performing the derivation for an ARMA(p,q) process. These derivations are, of course, central to the AR(p^*) approximation method.

3.1 AR(∞) of an ARMA(1,1)

Take an ARMA(1,1) described by:

$$y_t = \phi y_{t-1} + u_t + \theta u_{t-1} \tag{3.1}$$

By isolating u_t we get the equation:

$$u_t = y_t - \phi y_{t-1} - \theta u_{t-1}$$

Applying the lag (Backshift) operator to both sides of the equation:

$$u_{t-1} = y_{t-1} - \phi y_{t-2} - \theta u_{t-2}$$

subbing this into 3.1:

$$\begin{aligned} y_t &= \phi y_{t-1} + u_t + \theta(y_{t-1} - \phi y_{t-2} - \theta u_{t-2}) \\ &= (\phi + \theta) y_{t-1} - \phi\theta y_{t-2} + u_t - \theta^2 u_{t-2} \end{aligned}$$

Similarly, we can find an expression for u_{t-2}.

$$u_{t-2} = y_{t-2} - \phi y_{t-3} - \theta u_{t-3}$$

Further repeating this process we form an infinite sum

$$\begin{aligned}
y_t &= (\phi+\theta)y_{t-1} - \phi\theta y_{t-2} + u_t - \theta^2(y_{t-2} - \phi y_{t-3} - \theta u_{t-3}) \\
&= (\phi+\theta)y_{t-1} - (\phi\theta+\theta^2)y_{t-2} + \phi\theta^2 y_{t-3} + u_t + \theta^3 u_{t-3} \\
&= (\phi+\theta)y_{t-1} - (\phi\theta+\theta^2)y_{t-2} + \phi\theta^2 y_{t-3} + u_t + \theta^3(y_{t-3} - \phi y_{t-4} - \theta u_{t-4}) \\
&= \dots
\end{aligned}$$

$$= u_t + \sum_{i=1}^{\infty}(\phi+\theta)(-\theta)^{i-1}y_{t-i} \tag{3.2}$$

Which is an AR(∞).

3.2 Generalisation to ARMA(p,q)

For ARMA(p,q) models with order greater than 1, deriving the equivalent AR(∞) becomes complicated and time consuming. For this reason, it is more logical to transform the models to State Space form and perform the derivation via matrix vector algebra.

Define the state space representation of the ARMA(p,q) process as

$$\begin{cases} \chi_t = A\chi_{t-1} + bu_{t-1} \\ y_t = c'\chi_t + u_t \end{cases} \tag{3.3}$$

where $y_t, u_t \in \mathrm{R}\,\forall\, t \in \mathrm{N}$ and

$$\chi_t = \begin{pmatrix} v_{t-k} \\ v_{t-(k-1)} \\ \vdots \\ v_{t-1} \\ v_t - u_t \end{pmatrix} \in \mathrm{R}^{k+1} \qquad A = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & \phi_k & \phi_{k-1} & \dots & \phi_2 & \phi_1 \end{pmatrix} \in \mathrm{R}^{k+1,k+1}$$

$$c' = \begin{pmatrix} \theta_p & \theta_{k-1} & \dots & \theta_1 & 1 \end{pmatrix} \in \mathrm{R}^{k+1} \qquad b = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \phi_1 \end{pmatrix} \in \mathrm{R}^{k+1}$$

v_t is an auxilary variable defined such that

$$\begin{aligned} \phi(L)v_t &= u_t \\ y_t &= \theta(L)v_t \end{aligned}$$

$k = \max\{p,q\}$. Note that for q < p , $\theta_i = 0\,\forall\, i = q+1, q+2, ...p$. Similarly for q > p, $\phi_i = 0\,\forall\, i = p+1, p+2, ...q$. (Aoki, 1987 , pp. 40-46)

From the state space form we can find that the inverse system is

$$\begin{cases} \chi_{t+1} = A\chi_t + b(-c'\chi_t + y_t) \\ u_t = y_t - c'\chi_t \end{cases} \iff \begin{cases} \chi_{t+1} = (A - bc')\chi_t + by_t \\ u_t = y_t - c'\chi_t \end{cases}$$

set $\tilde{A} = (A - bc')$, such that

$$\begin{aligned}
\chi_{t+1} &= \tilde{A}\chi_t + by_t \\
&= by_t + \tilde{A}(\tilde{A}\chi_{t-1} + by_{t-1}) \\
&= by_t + \tilde{A}by_{t-1} + \tilde{A}^2(\tilde{A}\chi_{t-2} + by_{t-2}) \\
&\vdots \\
&= by_t + \tilde{A}by_{t-1} + \tilde{A}^2by_{t-2} + \tilde{A}^3y_{t-3} + \tilde{A}^4y_{t-4} + \dots \\
&= \sum_{i=0}^{\infty} \tilde{A}^i by_{t-i}
\end{aligned}$$

which can be substituted into the expression for y_t in equation 3.3 deriving

$$\begin{aligned}
u_t &= y_t - c'\chi_t \\
&= y_t - c'\sum_{i=1}^{\infty} \tilde{A}^{i-1}by_{t-i}
\end{aligned}$$

$$= y_t - \sum_{i=1}^{\infty} c'\tilde{A}^{i-1}by_{t-i} \tag{3.4}$$

which is an AR(∞) process.

3.3 Conditions of AR(∞)

The derivation of an AR(∞) is rather simple mathematically. However, there are a number of difficulties which are implicit in the estimation of non-stationary ARMA processes which are exposed in deriving the corresponding AR(∞).

It is clear by the AR(∞) expression that the process is dependent on the existence of an infinite set of observations $\{y_{t-i}\}_{i=1}^{\infty}$. This is true of both the ARMA(p,q) and AR(∞) processes and in both the stationary and non-stationary cases. From here, arises the need for an initial condition.

In the stationary case, we can derive and initial condition by expressing the ARMA process in its MA(∞) form. In the ARMA(1,1) case, this is

$$y_t = u_t + \sum_{t=1}^{\infty} (\theta + \phi)\phi^{i-1}u_{t-i} \tag{3.5}$$

The u_t are assumed independently, identically, normally distributed with zero mean and variance σ^2 as per definition 1. Given that the ARMA process is stationary, we require $|\phi| < 1$ and so equation 3.5 is a geometric sum of independently normally distributed random variables and converges to a normal random variable with finite variance. This is true for all $t \in \mathcal{Z}$ and aligns with our definition of stationarity and stationarity conditions 2.3, 2.4.

In the non-stationary case however, we have that $|\phi| \geq 1$ and equation 3.5 is a divergent sum. The resulting normal distribution will have infinite variance and hence, does not exist! As a result, an initial condition cannot be defined in this way and applied in maximum likelihood estimation. This is the case for all non-stationary ARMA(p,q) processes.

In the absence of an initial condition for the infinite set of observations, we take two key measures in establishing the AR(p^*) method. The first is to truncate the expression for the AR(∞) process at some lag y_{t-p^*}, $p^* \in \mathbb{N}$, forming instead an AR(p^*) process. The second is to condition the likelihood on the initial p^* observations of the finite set of observations $\{y_t\}_{t=1}^{T}$.

These two measures result in the AR(p^*) method providing an approximation to the ARMA(p,q) model under estimation as opposed to an exact estimate.

Remark. From 3.5 we can also show for all non-stationary processes that the autocovariance function $Cov(y_t, y_{t+h}) = \infty$ for all $h \in \mathcal{Z}$. Equivalence of two ARMA processes is defined in the spectral factorisation problem as two processes which return the same autocovariance function. This definition clearly holds only in the stationary case. Equivalence of an ARMA(p,q) process and an AR(∞) process is considered as the AR(∞) which derives from the ARMA(p,q) process.

Remark. The above discussion does not mean necessarily that deriving an initial condition for the non-stationary case is not possible although the problem has not been solved. Doing so is far beyond the scope of this project and hence, we take the measures discussed above.

Estimation of AR(p*)

Having established the required AR(∞) form of the ARMA process, we can now define the AR(p^*) method.

Definition 6 (AR(p^*) approximation method)**.** Given a set of observations $\{y_t\}_{t=1}^{\infty}$ and an ARMA(p,q) process with gaussian error process $u_t \overset{i.i.d}{\sim} \mathcal{N}(0,\sigma^2)$, the resulting parameter estimates $\hat{\phi}_1, \hat{\phi}_2, \ldots, \hat{\phi}_p, \hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_q, \hat{\sigma}^2$ are those which maximise the conditional maximum likelihood of the AR(p^*) model

$$y_t = u_t + \sum_{i=1}^{p^*} c' \tilde{A}^{i-1} b y_{t-i}$$

where $\tilde{A} = (A - bc')$ and A, b, c' are defined as in equation 3.3.

Remark. Given that the error process u_t is independently gaussian distributed for each t, maximising the conditional likelihood is equivalent to minimising the sum of squares. For the purposes of computational efficiency, the estimation procedure has been implemented by minimising the sum of the squared residuals throughout this project. $\hat{\sigma}^2$ is equal the mean squared error $(\sum_{t=p^*+1}^{T} u_t^2)/(T - p^*)$

Remark. This method is defined under the assumption that the ARMA process being approximated is non-stationary. In the case where stationary ARMA is being approximated, the full likelihood, as opposed to the conditional likelihood, can be taken.

4.1 Equating Conditional Likelihood and Sum of Squares of an AR(p*) process

Theorem 2. Given an AR(p^*) corresponding to some ARMA(p,q) process, the the parameter estimates under Conditional Maximum Likelihood Estimation and Least Squares estimation, are equivalent.

Proof. Consider an AR(∞) process defined as 6. The error process u_t is Gaussian distributed and so we have that

$$y_t \mid y_{t-1}, y_{t-2}, \ldots, y_{t-p^*} \sim N(0, \sigma^2)$$

and

$$P(y_t \mid y_{t-1}, y_{t-2}, \ldots, y_{t-p^*}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}\left(y_t - \hat{y}_t\right)^2\right)$$

where

$$\hat{y}_t = \sum_{i=1}^{p^*} c' \tilde{A}^{i-1} b y_{t-i}$$

The likelihood function can be defined conditional on $y_1, y_2 \ldots y_p^*$ as

$$\begin{aligned}
P(y_T, y_{T-1}, \ldots, y_1) &= P(y_T \mid y_{T-1}, \ldots, y_1) P(y_{T-1}, \ldots, y_1) \\
&= (y_T \mid y_{T-1}, \ldots, y_1) P(y_{T-1} \mid y_{T-2}, , \ldots, y_1) P(y_{T-3}, \ldots, y_1) \\
&\vdots \\
&= (\prod_{t=p^*+1}^{T} P(y_t \mid y_{t-1}, \ldots, y_1)) P(y_{p^*}, ..., y_1)
\end{aligned}$$

From which

$$\begin{aligned}
L(\sigma, \phi_1, \ldots, \phi_p^* \mid y_p^* \ldots y_1) &= P(y_T, y_{T-1}, \ldots, y_{p^*+1} \mid y_p^*, \ldots y_1) \\
&= \prod_{t=p^*+1}^{T} P(y_t \mid y_{t-1}, \ldots, y_1) \\
&= \prod_{t=p^*+1}^{T} \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{2\sigma^2}(y_t - \hat{y}_t)^2) \\
&= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{T-p} \exp\left(-\frac{1}{2\sigma^2} \sum_{t=p+1}^{T} (y_t - \hat{y}_t)^2\right)
\end{aligned}$$

Now that we have defined the conditional maximum likelihood, we can derive a Maximum likelihood for σ.

$$-\ln L(\sigma, \phi \mid y_p^* \ldots y_1) = (T - p^*)(\ln \sqrt{2\pi} + \ln \sigma) + \frac{1}{2\sigma^2} \sum_{t=p+1}^{T} (y_t - \hat{y}_t)^2$$

$$\frac{\partial}{\partial \sigma} - \ln L(\sigma, \phi \mid y_p^* \ldots y_1) = \frac{(T - p^*)}{\sigma} - \frac{1}{\sigma^3} \sum_{t=p+1}^{T} (y_t - \hat{y}_t)^2$$

Equating this to 0 we find that

$$\hat{\sigma}^2 = \frac{\sum_{t=p+1}^{T} (y_t - \hat{y}_t)^2}{T - p^*} \tag{4.1}$$

This MLE for sigma can be substituted back into our conditional likelihood function as follows:

$$
\begin{aligned}
L(\sigma,\phi \mid y_p^* \dots y_1) &= \left(\frac{1}{\sqrt{2\pi}\hat{\sigma}}\right)^{T-p} \exp\left(-\frac{1}{2\hat{\sigma}^2}\sum_{t=p+1}^{T}(y_t-\hat{y}_t)^2\right) \\
&= \left(\frac{T-p^*}{2\pi\sum_{t=p^*+1}^{T}(y_t-\hat{y}_t)^2}\right)^{\frac{T-p^*}{2}} \exp\left(-\frac{T-p^*}{2\sum_{t=p^*+1}^{T}(y_t-\hat{y}_t)^2}\sum_{t=p^*+1}^{T}(y_t-\hat{y}_t)^2\right) \\
&= \left(\frac{T-p^*}{2\pi\sum_{t=p^*+1}^{T}(y_t-\hat{y}_t)^2}\right)^{\frac{T-p^*}{2}} \exp\left(-\frac{T-p^*}{2}\right)
\end{aligned}
$$

Maximising the conditional Maximum Likelihood function is therefore equivalent to minimising the sum of squared residuals $\sum_{t=p^*+1}^{T}(y_t-\hat{y}_t)^2$ $\square$

4.2 Strong Consistency of AR(p) by Sum of squares

Lai & Wei (1982) proved the Strong Consistency of Least Squares estimation of general autoregressive models. The property holds for stable models, unstable models and models with mixed roots.

These findings paired with the equivalence of Conditional Maximum likelihood and Sum of Squares estimation, provided the reasoning for the AR(p^*) method. Given strong consistency of an AR(p^*) model regardless of stationarity, we would expect that the AR(p^*) process,derived from an ARMA process, would provide sensible parameter estimates for the ARMA process.

The findings of Lai and Wei are defined with the condition that the underlying data generating process (DGP) is truly an AR(p) process. This condition is not satisfied in our case, where the DGP is an ARMA(p,q) process. For this reason we have not directly applied these findings to the estimation procedure.

The questions we are asking regarding the AR(p^*) method are:

1. Do least squares estimates converge to the true values of the original ARMA process?
2. If the parameter estimates do converge to the true values, do they show strong/weak consistency or convergence in distribution?
3. What are the effects of T and p^* on the estimation?
4. Does the level of non-stationarity affects the estimation?

4.3 Standard errors of stationary ARMA coefficient estimates

Under the assumption of stationary, we have that the coefficient estimates under maximum likelihood estimation are consistent and asymptotically normal

$$\sqrt{T}(\hat{\nu} - \nu) \xrightarrow{\mathrm{d}} \mathcal{N}(0, \Sigma) \tag{4.2}$$

where $\nu = \{\phi_1, \phi_2, \ldots, \phi_p, \theta_1, \theta_2, \ldots, \theta_q\}$ and $\hat{\nu}$ is the corresponding vector of coefficient estimates.

Σ is the variance-covariance matrix and corresponds to the inverse of the asymptotic Information matrix $\mathcal{I}(\nu)$. Methods for calculating the asymptotic Information matrix are specified in detail in Hanzon & Peeters (1999).

As an example consider an MA(1) process. We have that $n = 0, m = 1, \mathcal{I}(\nu) = \Delta_{q,q}$ and $A_q = -\theta$. $\Delta_{q,q}$ is such that,

$$\Delta_{q,q} - A_q \Delta_{q,q} {A_q}^T = 1$$
$$\Delta_{q,q} = \frac{1}{1 - \theta^2}$$

Finally, we can calculate the Information matrix and classify the distribution of coefficient estimates.

$$\Sigma = \mathcal{I}^{-1}(\nu) = \left(\frac{1}{1 - \theta^2}\right)^{-1} = 1 - \theta^2$$
$$\hat{\theta} \sim \mathcal{N}\left(\theta, \frac{1 - \theta^2}{T}\right)$$

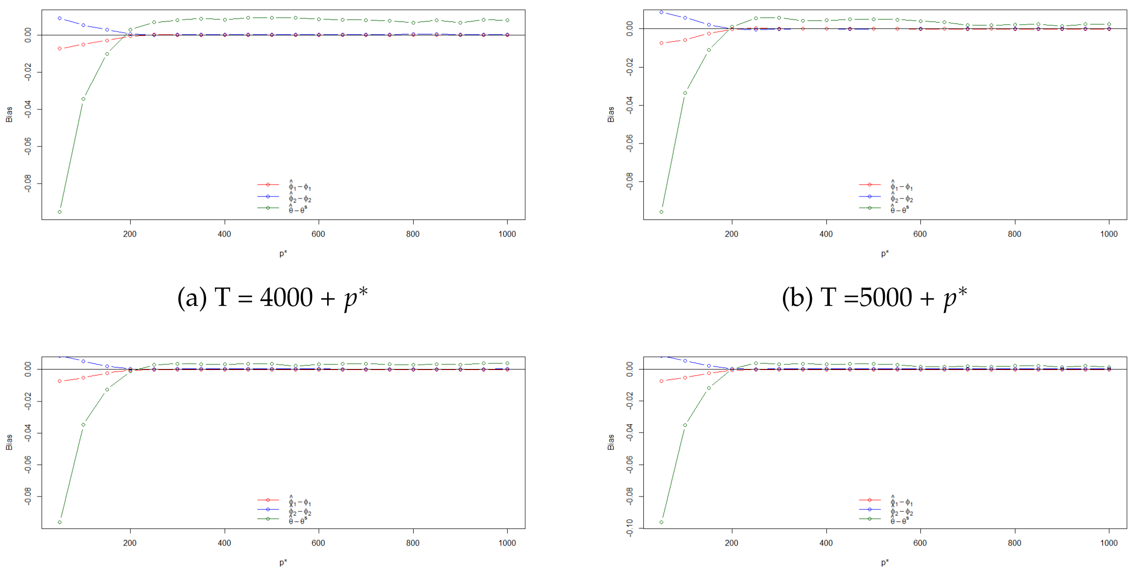
Calculations of higher order processes should be completed via computer algebra as discussed by Hanzon Peeters (1999).

This result is formed under the assumption that the ARMA process is stationary and hence, cannot be directly applied in the non-stationary case under discussion in this paper. In the absence of an equivalent result for the non-stationary case this provides an important reference for the consistency and variance estimates under AR(p^*) method.

Convergence of AR(p*) coefficient estimates

Initial tests will show that the AR(p^*) method approximate reasonably well the true AR coefficients and that the estimation is consistent with spectral factorisation. That is to say that the MA coefficient estimates converge to the invertible process $\theta^s(L)u_t^s$ if the true data process is one of $\theta^s(L)u_t^s$ and $\theta(L)u_t$.

Figure 5.1 displays the bias of an ARMA(2,1) process generating a series $\{y_t\}_{t=1}^{T}$ and estimated as an AR($p*$).



(a) T = 4000 + p^*

(b) T =5000 + p^*

(c) T = 6000 + p^*

(d) T =7000 + p^*

Figure 5.1: Bias of ARMA(2,1) coefficient estimates

Generally speaking, the bias of the coefficient estimates appear to be inversely proportional to p^*, T.

More detail is required to make insights into the behaviour of the parameter estimates under the AR(p^*) method. Our approach towards observing the behaviour of estimates has been to test by Monte Carlo simulation. We have taken for each of the cases in this section, $N = 150$ as the number of Monte Carlo simulations in each distribution. That is to say, for a simulation investigating convergence with respect to T, we generate

N simulations for each value of T included in the estimation. Many of the following simulations have been computationally intensive and so generating more samples than this can become very computationally intensive.

5.1 Convergence of auto-regressive parameter estimates

5.1.1 Convergence of purely explosive ARMA models

Defining the strong error of a sequence of estimators ν_i as $\mathbb{E}\left[|\nu_i - \nu|\right]$ we can generate strong error plots for the coefficient estimates under the AR(p^*) method. Figure 5.2 displays such an process for varying p^*. In each case the length of the series $T = 6000 + p^*$.

We have that the strong error decreases in a systematic manner for both ϕ_1, ϕ_2 coefficient estimates, leading us to the conclusion that the strong error of autoregressive coefficient estimates under the AR(p^*) method are inversely proportional to p^*.

Further consider figure 5.3 where the process has been repeated for a range of values T and $p^* = 600$ fixed. Similarly we have that that the strong error decreases systematically and similarly we can conclude that the strong error of autoregressive coefficient estimates under the AR(p^*) method are inversely proportional to T. Results such as these have lead to the following hypothesis.

Hypothesis 1. The autoregressive coefficient estimates $\phi(\hat{L})$ of an non-stationary ARMA(2,q) process with purely explosive AR component obtained under the AR(p^*) method converge strongly as $p^*, T \to \infty$ such that

$$\lim_{T, p* \to \infty} \phi(\hat{L}) \to \phi(L) \quad \text{a.s}$$

This hypothesis is supported further by table 5.1, which displays strong error figures obtained from Monte Carlo simulation as in figure 5.2.

Further, the red reference line in figure 5.2 and Figure 5.3 are given by

$$\Delta = C_{p^*} p^{*-12} \qquad \Delta = C_T \exp\left(-12T\right)$$

where Δ is the strong error in each case and C_{p^*}, C_{p^*} are real-valued constants.

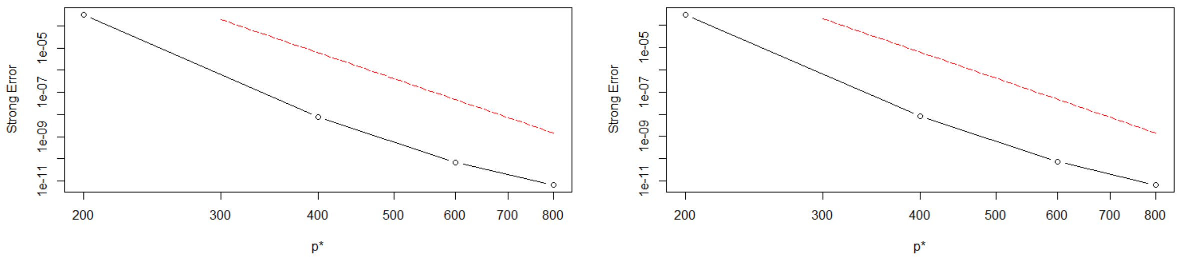
The strong error gained from the monte carlo simulations appears to have approximately equal rate of change as these function. Further, they appear to be bounded above by these functions. Based on this, we can form the following hypotheses regarding the *order of convergence* of AR(p^*) method.

Hypothesis 2. For T sufficiently large, the strong order of convergence of the AR(p^*) coefficient estimates of an ARMA($2, q$) with purely explosive AR component under the AR(p^*) method with respect to p^* is

$$\mathcal{O}\left(p^{*-12}\right)$$

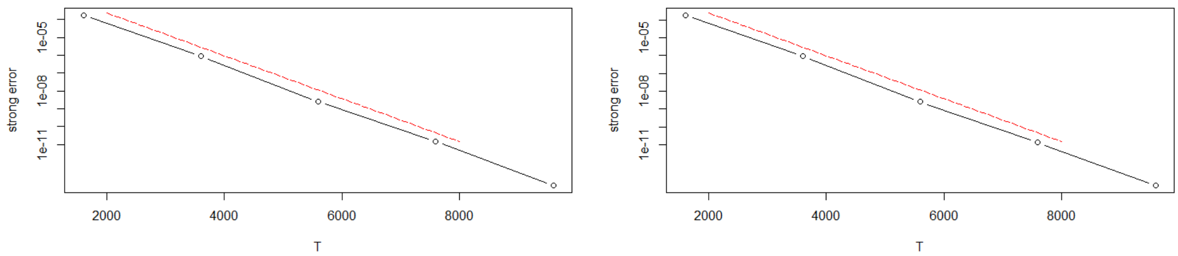
Hypothesis 3. For p^* sufficiently large, the strong order of convergence of the AR(p^*) coefficient estimates of an ARMA($2,q$) process with purely explosive AR component under the AR(p^*) method with respect to T is

$$\mathbb{O}\left(\exp\left(-12T\right)\right)$$



(a) $\hat{\phi}_1$ (b) $\hat{\phi}_2$

Figure 5.2: Strong error of ARMA(2,1) auto-regressive parameter estimates with respect to p^* ($T = 6000 + p^*$)



(a) ϕ_1 (b) ϕ_2

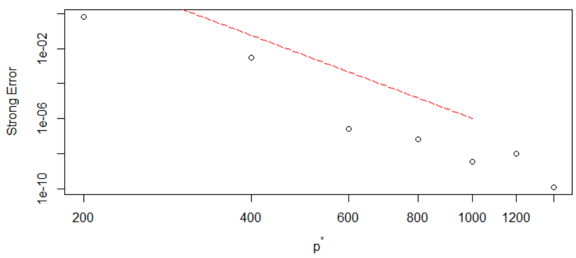
Figure 5.3: Strong error of ARMA(2,1) autoregressive parameter estimates with respect to T ($p* = 600$)

p^*	ϕ_1	ϕ_2
200	3.220160e-04	3.021746e-04
400	7.593333e-09	8.193333e-09
600	6.666667e-11	7.333333e-11
800	6.666667e-12	6.666667e-12
1000	0.000000e+00	0.000000e+00
1200	0.000000e+00	0.000000e+00
1400	0.000000e+00	0.000000e+00

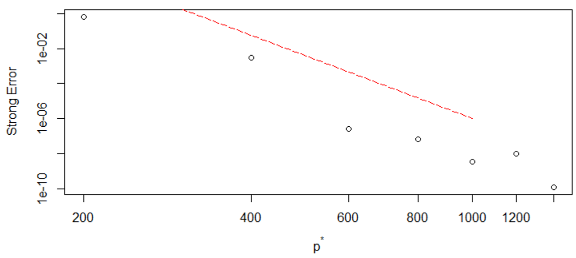
Table 5.1: Strong error of ARMA(2,1) auto-regressive parameter estimates with respect to $p*$ ($T = 6000 + p^*$)

5.1.2 Extension to ARMA(1,1) and Mixed root ARMA(2,1)

We have also carried out similar simulations for the ARMA(2,1) case with mixed root AR component and and ARMA(1,1) non-stationary case. The results of these tests support the hypotheses 2, 3. These results suggest that the hypothesis formed may hold in the general ARMA(p,q) and mixed root cases. Note that the reference lines in figure 5.4 is, as previously a function of ${p^*}^{-12}$. The reference line in figure 5.5 however is a function of $\exp(-18T)$. This may indicate that, for ARMA processes with unstable AR component, there exists some relationship between the strong order of convergence and the order p of the AR component.



(a) $\hat{\phi}_1$



(b) $\hat{\phi}_2$

Figure 5.4: Strong error of mixed root ARMA(2,1) auto-regressive parameter estimates with respect to $p*$ ($T = 6000 + p^*$)

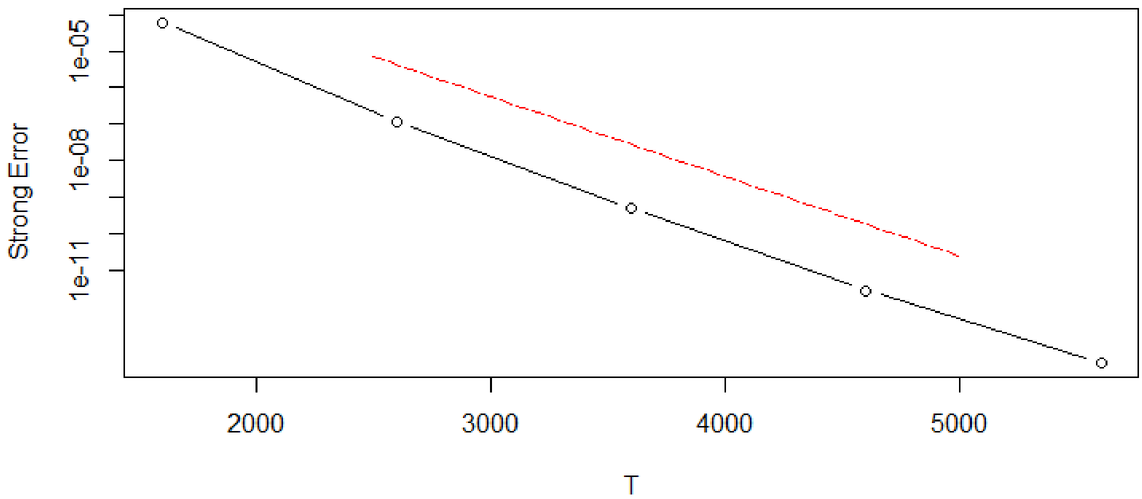


Figure 5.5: Strong error of ARMA(1,1) auto-regressive parameter estimates with respect to T ($p^* = 600$)

5.2 Convergence of moving average parameter estimates

5.2.1 Convergence with respect to p*

In the common stationary ARMA case, the moving average coefficient estimates are asymptotically normal such that in the MA(1) case.

$$\sqrt{T}(\theta - \hat{\theta}) \sim \mathcal{N}\left(0, 1 - \theta^2\right) \tag{5.1}$$

Derivation of this distribution is included in section 4.3.

Recall the ARMA(2,1) case as in the previous section. The corresponding $\hat{\theta}$ are displayed in figure 5.6. In this figure we can see that the θ estimates approach a symmetrical distribution about θ^s for $p* \geq 600$. The set of sample moments for each distribution are shown in figure 5.7 and give a more detailed insight into the convergence of parameter estimates. The sample moments appear to converge to those of the MA(1) process described in Equation 5.1. The difference being in this case that we have $T - p^*$ substituted for T. This leads us to the following hypothesis.

Hypothesis 4. The moving average coefficient estimates $\hat{\theta}$ of an non-stationary ARMA(p, 1) process obtained under the AR(p^*) method converge in distribution as $p^* \to \infty$ such that

$$\lim_{p* \to \infty} \hat{\theta} \xrightarrow{d} \mathcal{N}\left(\theta^s, \frac{1 - \theta^{s2}}{T - p^*}\right)$$

Remark. $T - p^*$ in this context is the number of fitting points available after conditioning the estimation on the initial p^* observations. ie. residual sum of squares is calculated for the final $T - p^*$ observations of the time series.

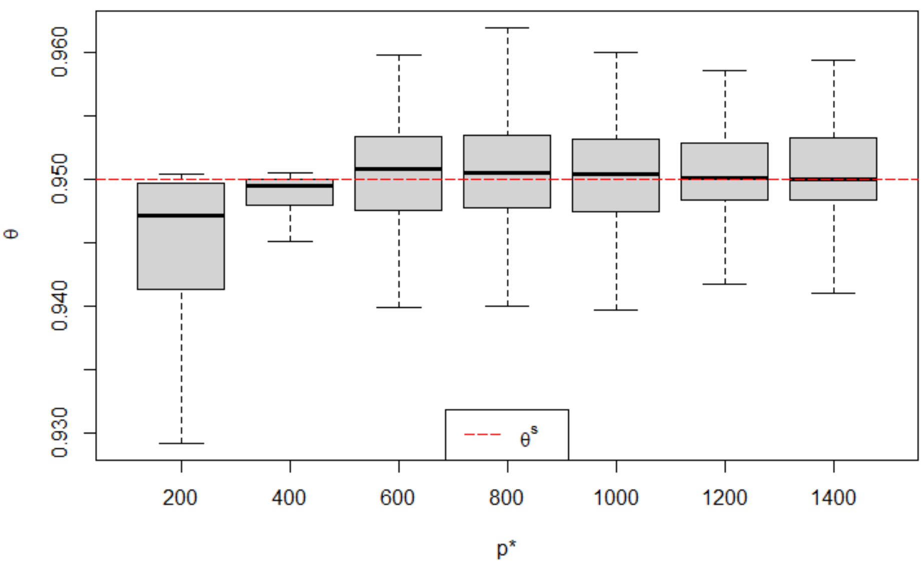
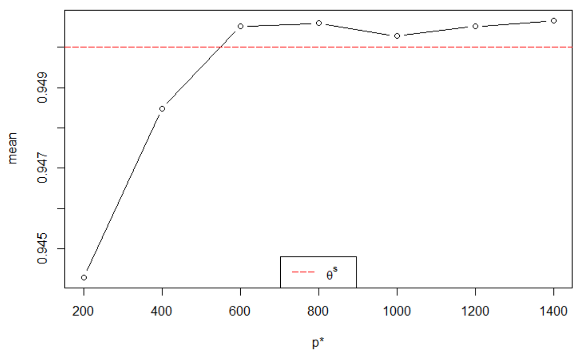
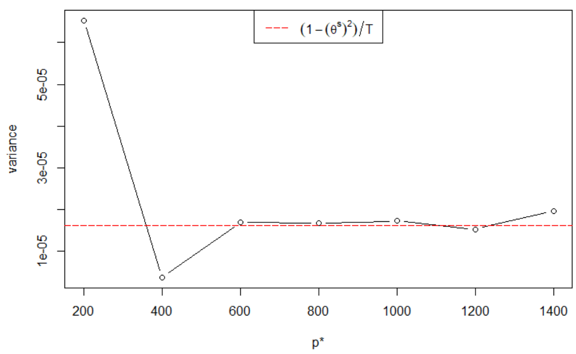


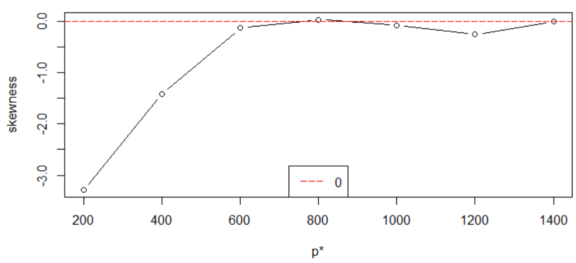
Figure 5.6: ARMA(2,1) moving average coefficient estimates with respect to p^*



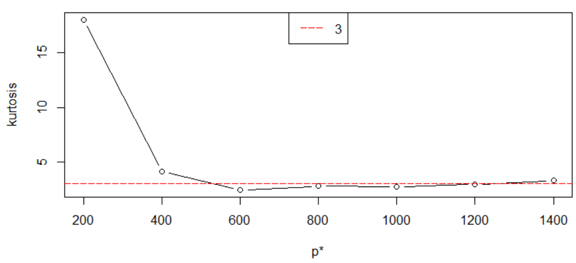
(a) Mean



(b) Variance



(c) Skewness



(d) Kurtosis

Figure 5.7: Sample Statistics of ARMA(2,1) moving average coefficient estimates with respect to p^*

5.2.2 Convergence with respect to T

By hypothesis 4, we would expect that in taking sample estimates for a range of T that the corresponding sample variances would follow approximately $\frac{1-\theta^{s2}}{T-p^*}$. Figure 5.8 verifies this and leads us to the hypothesis.

Hypothesis 5. The moving average coefficient estimates $\hat{\theta}$ of an non-stationary ARMA$(p,1)$ process obtained under the AR(p^*) method converge strongly as $p^*, T \to \infty$ such that

$$\lim_{T,p^* \to \infty} \hat{\theta} \xrightarrow{p} \theta^s$$

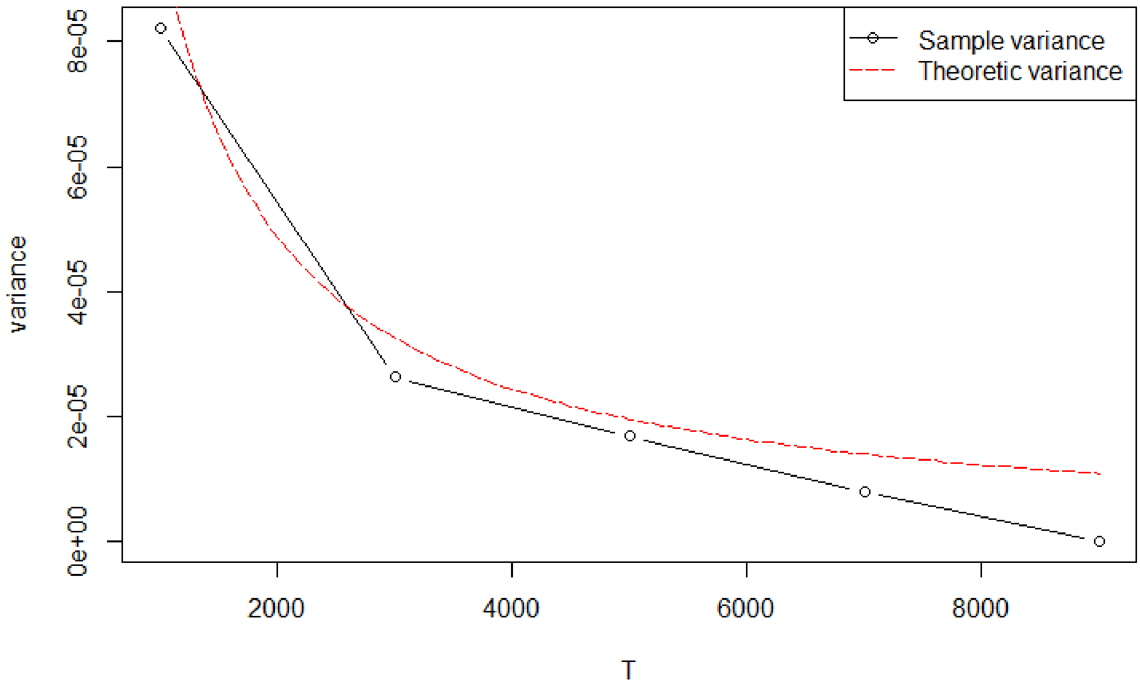


Figure 5.8: Sample variance of θ coefficient estimates

5.2.3 Variance of coefficient estimates with respect to θ^s

Based on the expression for the variance of coefficient estimates in hypothesis 4. Repeating the Monte Carlo simulations as above and varying the values of θ in the DGP, we have generated the sample statistics in table 5.2. Note that in this case that in generating the data we take θ to be the non-invertible, equivalent MA process of θ^s, and where the coefficients are distributed with a mean approximately equal θ^s. The sample variances resemble well the theoretical variances calculated from the term specified in hypothesis 4.

θ^s	Theoretical Variance	Sample Variance
.95	1.625e-05	1.373991e-05
.9	3.166667e-05	3.499874e-05
.8	6e-05	4.844922e-05
.5	1.25e-04	1.083156e-04

Table 5.2: Variance of non-stationary ARMA(2,1) moving average coefficient estimates; $T = 6000$, $p^* = 600$

Interestingly, hypotheses 4 , 5 imply Independence of AR and MA coefficient estimates of a non-stationary ARMA process under AR(p^*) method. We can investigate this to some extent by taking the sample pearson correlation coefficients of the AR and MA coefficient estimates.Table 5.3 presents such sample correlations. This data corresponds to the Monte Carlo simulation as in figure 5.8.

The resulting correlations show no particular pattern or trend with respect to T. We have that the they are, in most cases, approximately equal .05 in absolute value. This may indicate that MA coefficient estimates are weakly correlated with the AR coefficient estimates, however further evidence would be required to develop our hypothesis further.

T	$\rho_{\phi_1,\theta}$	$\rho_{\phi_2,\theta}$
1600	-0.0478	0.0486
3600	-0.0543	0.0572
5600	0.0579	-0.064
7600	0.113	-0.133
9600	0.0255	-0.054

Table 5.3: Sample correlations of coefficient estimates; $p* = 600$

Using parameter estimates to create stationary time series

Given the Claims we have made thus far, the estimation of an ARMA(p,q) process under the AR(p*) method provide a good approximation for sufficiently large p*,T.

The strong consistency of auto-regressive parameters in particular allows us to extend the method in a meaningful way. We can use these coefficient estimates to filter the unstable roots in the auto-regressive characteristic polynomial, creating a stationary series. We can then employ well known ML methods to the transformed data to complete our estimation.

This is analogous to the ideas of Integrated or ARIMA models.

The AR(p^*) method involves conditioning the likelihood on the inital p^* values of the time series. The following filtering procedures employ this method and hence, require conditioning in the same manner in order to estimate the AR(p^*) coefficients accurately. However, the filtered data is assumed stationary and so we can apply the initial conditions of the stationary case as discussed in section **??**. Theoretically speaking, this should provide a increased precision in the coefficient estimates.

6.1 Purely Explosive auto-regressive component

Consider the ARMA(p,q) model expressed in terms of the characteristic polynomial.

$$\phi(L)y_t = \theta(L)u_t \tag{6.1}$$

Define z_t such that,

$$z_t = \phi(L)y_t = y_t - \phi_1 y_{t-1} - \phi_2 y_{t-2} \cdots - \phi_p y_{t-p} \tag{6.2}$$

Substituting 6.2 into equation 6.1 we have have

$$z_t = \theta(L)u_t \tag{6.3}$$

From 6.3 we can conclude that z_t is a stationary moving-average process described exactly by $\theta(L)u_t$ and can be estimated as a MA(q) by maximum likelihood methods. On this basis we can define our filtering procedure for ARMA processes with purely explosive AR components.

Definition 7 (Filtering estimation method for a purely explosive ARMA process)**.** Given a finite, non-stationary series $\{y_t\}$ generated from an ARMA(p,q) process which has a purely explosive AR component, the filtering estimation method is defined by the following steps.

Step 1.
Apply the AR($p*$) method to the series $\{y_t\}$ and store estimates $\hat{\phi}_1, \hat{\phi}_2, \ldots, \hat{\phi}_p$

Step 2.
Generate series $\{\hat{z}_t\}$ such that

$$\hat{z}_t = y_t - \hat{\phi}_1 y_{t-1} - \hat{\phi}_2 y_{t-2} \cdots - \hat{\phi}_p y_{t-p} \quad \forall\ t = p+1, p+2, \ldots T$$

Step 3.
Fit an MA(q) model to the series $\{z_t\}$ by maximum likelihood under the assumptions and initial conditions of a stationary process. Store $\hat{\theta}_1^z, \hat{\theta}_2^z, \ldots, \hat{\theta}_q^z$ coefficient estimates.

The resulting estimation of the ARMA(p,q) model is

$$y_t = \hat{\phi}_1 y_{t-1} + \hat{\phi}_2 y_{t-2} \cdots + \hat{\phi}_p y_{t-p} + u_t + \hat{\theta}_1^z u_{t-1} + \hat{\theta}_2^z u_{t-1} + \cdots + \hat{\theta}_q^z u_{t-1}$$

6.1.1 Example: ARMA(2,1)

Consider the ARMA(2,1) process:

$$y_t = 1.990950 y_{t-1} - 1.00553 y_{t-2} + u_t + \frac{1}{.95} u_{t-1}$$

where $r_{1,2} = 0.9900003 \pm 0.1199993i$. A series $\{y_t\}$ generated from this process is shown in figure 6.1a.

Step 1.
Applying the AR(p*) method with p* = 600 we have the coefficient estimates

Step 2.
$\{z_t\}$ can be estimated as

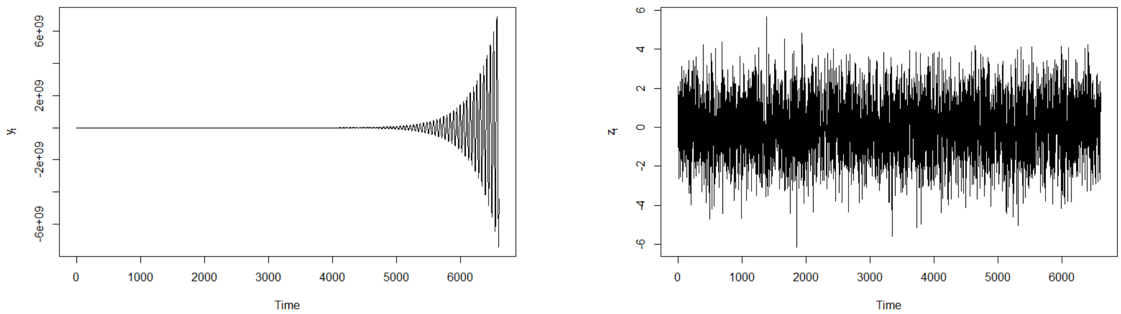
$$\hat{z}_t = y_t - \hat{\phi}_1 y_{t-1} - \hat{\phi}_2 y_{t-2} \quad \forall\ t = 3, 4, \ldots T$$

we have $\hat{\phi}_1, \hat{\phi}_2$ from step 1. Hence we can generate the series $\{\hat{z}_t\}$. The series is plotted in figure 6.1b.

Step 3.
Estimating the stationary series as z_t as an MA(1) model, applying Maximum likelihood. We find $\hat{\theta} = 0.953252$ and $\hat{\sigma^2} = 1.1276$

The final estimation of the ARMA(2,1) model is therefore

$$\hat{z}_t = \hat{\theta} u_t$$
$$y_t - \hat{\phi}_1 y_{t-1} - \hat{\phi}_2 y_{t-2} = u_t + \hat{\theta} u_{t-1}$$
$$y_t - 1.99095 y_{t-1} + 1.00553 y_{t-2} = u_t + 0.953252 u_{t-1}$$



(a) Observed Series y_t

(b) Filtered Series z_t

Figure 6.1

6.2 Mixed root autoregressive component

In the case of mixed roots, it is possible to use the coefficient estimates to further estimate the roots of the auto-regressive process and remove the unstable root. This creates a stationary time series which can be estimated as a stable and invertible ARMA process. This requires a slightly different transformation than in the purely explosive case.

Consider an ARMA(p,q) process such that

$$\phi(L)y_t = \theta(L)u_t$$

If we define r_i as the roots of the AR characteristic polynomial and r_j^u to be the roots r_i such that $|r_i| \leq 1$ and similarly r_k^s to be the roots r_i such that $|r_i| > 1$, we have that, assuming the AR component has mixed roots,

$$\begin{aligned} \prod_{i=1}^{p}(\frac{1}{r_i}L - 1)y_t &= \theta(L)u_t \\ \prod_{j=1}^{p_u}(\frac{1}{r_j^u}L - 1)\prod_{k=1}^{p_s}(\frac{1}{r_k^s}L - 1)y_t &= \theta(L)u_t \\ \phi^u(L)\phi^s(L)y_t &= \theta(L)u_t \end{aligned}$$

where $p_s, p_u \neq 0$, $p_s + p_u = p$ and

$$\begin{aligned} \phi^u(L) &= \prod_{i=1}^{p_u}(\frac{1}{r_j^u}L - 1) \\ |r_j^u| &\leq 1 \;\; \forall \; j = 1,2\ldots,p_u \\ \phi^s(L) &= \prod_{k=1}^{p_s}(\frac{1}{r_k^s}L - 1) \\ |r_k^s| &> 1 \;\; \forall \; k = 1,\ldots,p_s \end{aligned}$$

Define z_t such that

$$z_t = \phi^u(L)y_t \tag{6.4}$$

Substituting zt into equation 6.5 we have that

$$\phi^s(L)z_t = \theta(L)u_t \tag{6.5}$$

which is a stable and invertible ARMA(p_s , q) process which we can estimate by applying usual maximum likelihood to the series $\{z_t\}$. From here we can define our filtering procedure for the mixed root AR case.

Definition 8 (Filtering estimation method for a purely explosive ARMA process)**.** Given a finite, non-stationary series $\{y_t\}$ generated from an ARMA(p,q) process which has a mixed root AR component, the filtering estimation method is defined by the following steps.

Step 1.
Apply the AR($p*$) method to the series $\{y_t\}$ and store estimates $\hat{\phi}_1, \hat{\phi}_2, \ldots, \hat{\phi}_p$

Step 2.
Find the roots of the associated characteristic equation $\phi(\hat{L})$ and identify the non stationary roots $\hat{r_i^s}$.

Step 3.
Generate series $\{\hat{z_t}\}$ where

$$\hat{z_t} = \phi^u(L)y_t \;\; \forall \; t = p_u + 1, p_u + 2, \ldots T$$

Step 4.
Fit an ARMA(p_s,q) model to the series $\{z_t\}$ by maximum likelihood. This will return the model

$$\phi^s(\hat{L})\hat{z_t} = \theta(\hat{L})u_t$$

Step 5.
Substitute 8 into the equation to get our final model for y_t

$$\begin{aligned}\phi^s(\hat{L})(\phi^u(\hat{L})y_t) &= \theta(\hat{L})u_t \\ \phi(\hat{L})y_t &= \theta(\hat{L})u_t\end{aligned}$$

6.2.1 Example: ARMA(2,1)

Take as an example the ARMA(2,1) process with $\sigma^2 = 1$

$$\begin{aligned}y_t &= 1.9121y_{t-1} - 0.9118y_{t-2} + u_t + \frac{1}{.9}u_{t-1} \\ (\frac{1}{1.1}L - 1)(\frac{1}{.997}L - 1)y_t &= (\frac{1}{.9}L + 1)u_t\end{aligned}$$

we have that

$$\begin{aligned}p_u &= 1 \\ r_u &= .997 \\ \phi^u(L) &= (\frac{L}{.997} - 1)\end{aligned}$$

$$\begin{aligned}p_s &= 1 \\ r_s &= 1.1 \\ \phi^u(L) &= (\frac{L}{1.1} - 1)\end{aligned}$$

A series y_t generated by the ARMA process is shown in figure 6.2a.
Step 1.
Applying the AR(p*) procedure to y_t returns

$$\hat{\phi}_1 = 1.9120999$$
$$\hat{\phi}_2 = -0.9118264$$

Step 2.
using the polyroot function in R we can find and store $\hat{r_u}$. In this example that the bias of the coefficient estimates in table **Step 3.**

	Bias
ϕ_1	-2.249636e-10
ϕ_2	2.214696e-10
r_u	4.414447e-11

$$\hat{z_t} = \frac{1}{\hat{r_u}} y_{t-1} - y_t$$

Generating $\{\hat{z_t}\}$ from $\hat{r_u}$ we get the series in figure 6.2b.

Step 4.
Using R arima function we fit an ARMA(1,1) $\{z_t\}$ using maximum likelihood estimation. This returns estimates:

$$\hat{r_s} = \frac{1}{phi^z} = 1.108936$$
$$\hat{\theta} = 0.8999501$$
$$\hat{\sigma^2} = 1.201201$$

Step 5.

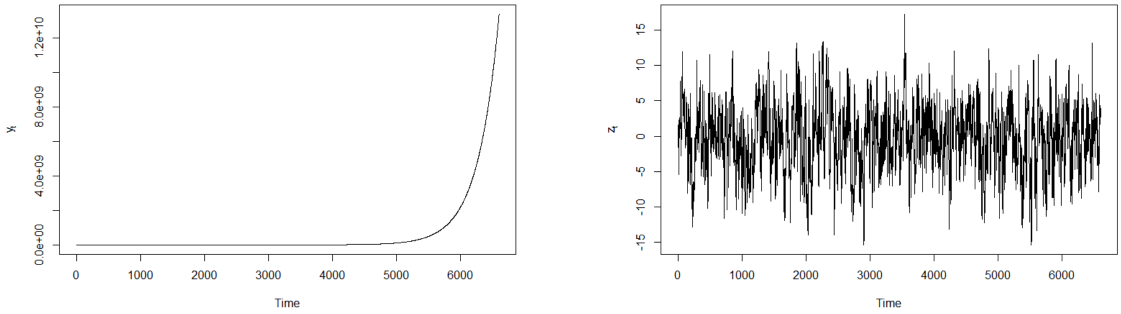
$$\phi^s(\hat{L})\hat{z_t} = \theta(\hat{L})\hat{u_t}$$
$$(\frac{1}{\hat{r_u}}L - 1)(\frac{1}{\hat{r_s}}L - 1) = (1 + \hat{\theta}L)\hat{u_t}$$

Expanding this equation will give us our final model

$$y_t = 1.907488 y_{t-1} - 0.9072006 y_{t-2} + u_t + .89995 u_{t-1}$$

6.3 On the AR(p*) and filtering methods

Naturally, having established the doability of the AR(p^*) and filtering methods, the need to compare the two methods arises. The first thing worth noting about the methods

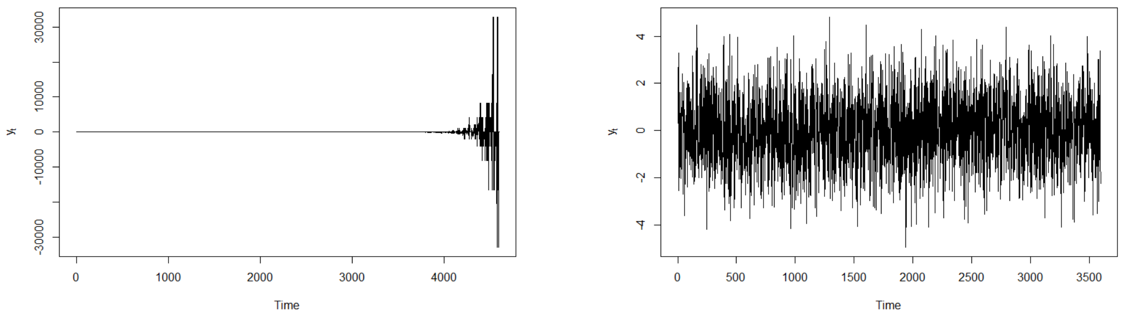


(a) Observed Series y_t

(b) Filtered Series z_t

Figure 6.2

is that the feasibility of the filtering method depends on the accuracy of the absolute bias of the AR(p^*) auto-regressive coefficient estimates. For any ARMA process of autoregressive order p, denote the error of each coefficient estimates as $\epsilon_i = \hat{\phi}_i - \phi_i, i = 1,2,\ldots p$. The resulting filtered series $\hat{z}_t = \phi(\hat{L})y_t = \phi(L)y_t + \epsilon_1 y_{t-1} + \epsilon_2 y_{t-2} + \ldots \epsilon_p y_{t-p}$. The estimate $\hat{z}_t$ will be stationary assuming $\epsilon_i y_{t-i}$ is sufficiently small **for all i,t**. Given the series is explosive, the absolute value of the observations y_t will increase over time and so the latter portion of the resulting filtered series $\hat{z}_t$ is liable to violate the stationarity condition 2.4 if the $\epsilon_i y_{t-i}$ become large relative to the true z_t.



(a) filtering in presence of significant error

(b) filtering in absence of error

Figure 6.3

Clearly, 6.3a is no longer representative of the true, stationary MA(1) process z_t and so applying maximum likelihood techniques which assume stationarity are ineffective. The true MA(1) process is such that $\theta^s = .95, \sigma^2 = 1.0526$. If we ignore the stationarity assumption and apply maximum likelihood, we find $\hat{\theta} = -.0524$. The AR(p^*) estimate on the other hand is exact, equaling 0.95

For given $T, p*$ it is straight forward to generate a series of coefficient estimates for both methods. In doing so, we find that the variance of coefficient estimates is lowest when filtering for all AR coefficient estimates and estimating the residual $\theta(L)$ as a stationary series is applied, as we have done in case 1.

In the purely explosive case, we have that the MA coefficient estimates are comparable in bias, but that the variance of coefficient estimates reduces by 10.5%.

	AR(p*)	Filtered
Sample Mean	0.9507	0.9508
Sample Variance	1.557567e-5	1.393848e-5

Table 6.1: Sample statistics of AR(p*) and filtered moving average coefficient estimates of an ARMA process with purely explosive AR component ($T = 6600, p^* = 600$)

As established in section 5.2.1, for p^* sufficiently large, the moving average coefficient estimates of an ARMA(p,1) behave comparably to that of the stationary MA(1). Given the filtered series z_t is stationary, we know the distribution of coefficient estimates of an MA(1) has variance $\frac{(1-\theta^{s2})}{T}$. The corresponding distribution of AR(p^*) estimates is approximate well by $\frac{(1-\theta^{s2})}{T-p^*}$. From here we can approximate the reduction in variance is as the absolute difference between the two variance terms. In this case we have $p^* = 6000, T = 6600$ and the approximate difference is $9.\dot{0}9\%$.

The same procedure for multiple values of T are displayed in figure 6.4. Assuming hypothesis 4 holds, we would expect that the variance of the moving average coefficient estimates for an ARMA(p,1) process would be greater under the AR(p^*) method than under the Filtering method for all T. The difference in variance would be much more significant for small sample sizes, with the difference between the two expressions strictly decreasing with respect to T.

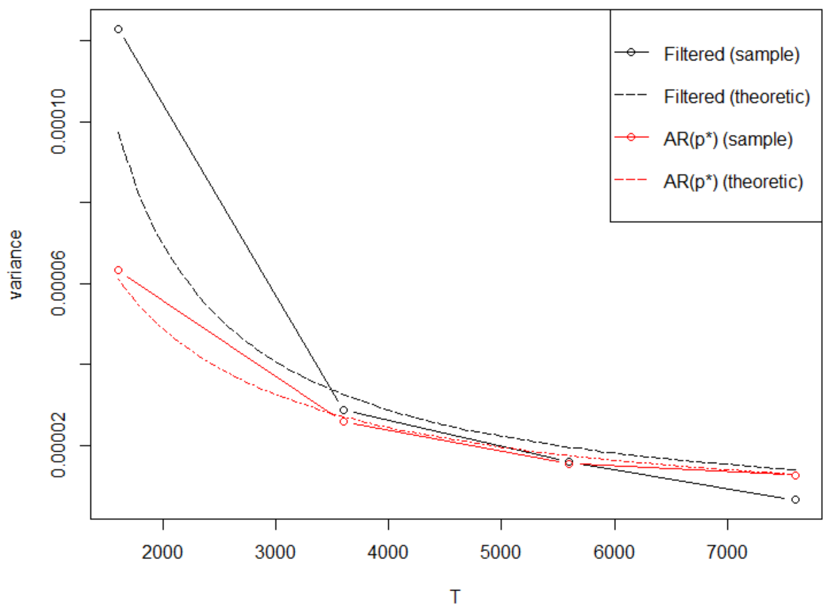


Figure 6.4: Comparison of variance of moving average coefficient estimates

In the mixed root case, we have two possible methods filtering methods. The first is to filter all AR coefficients and estimate the residual MA process using stationary MA estimation procedure, as in the purely explosive case. The other is to filter the unstable roots and estimate the residual ARMA process using stationary ARMA estimation procedure, as in section 6.1.

In filtering all AR coefficients, we find again comparable bias of MA coefficient estimates but a 20% reduction in variance when applying the filtered estimation in comparison to the AR(p^*) coefficient estimates.

	AR(p*)	filter $\phi(L)$
Sample Mean	0.9504628	0.9502876
Sample Variance	1.8745e-5	1.4995e-5

Table 6.2: Sample statistics of AR(p*) and filtered moving average coefficient estimates of an ARMA process with mixed root AR component ($T = 6600, p^* = 600$)

In applying the filtering of solely unstable AR roots, we find that the AR(p^*) method performs better in estimating AR coefficients and MA coefficients. We have that the strong error is much lower but we do have slightly more bias in the moving average coefficient estimates. This issue of bias is avoided in applying the filtering of all AR roots. ie. in applying the filtering method for purely explosive models.

These preliminary tests would indicate that, of the three estimation procedures discussed, filtering the entire AR component, irrespective of whether the AR component is purely explosive or has mixed roots, will minimise the bias and variance of coefficient estimates

	AR(p^*)	Filter $\phi^u(L)$
ϕ_1	3.85×10^{-7}	3.89×10^{-3}
ϕ_2	3.86×10^{-7}	3.9×10^{-3}

Table 6.3: Strong error of AR coefficient estimates

	AR(p^*)	Filter $\phi^u(L)$
Sample mean	0.949662	0.950014
Sample variance	1.2118e-05	1.8393e-05

Table 6.4: Sample statistics of MA coefficient estimates

Computational errors and inaccuracies

Throughout this project and in various simulations, we have encountered a number of computational errors which are very difficult to avoid when working with non-stationary process. These problems have limited us in the extent to which we have been able to investigate the behaviour of non-stationary time series and I feel its important to discuss these issues.

7.1 Non-Stationary Data and Numerical accuracies

Recall 2.3, 2.4 as our definition of Stationarity. Non-Stationary time series violates one, or both of these conditions.

Implicit in this definition is the notion that the absolute value of the observations from a non-stationary series tend to infinity as $t \to \infty$. The implication of this for model fitting is that, for time series with large number of observations, the absolute value of the observations can be so large and differ so greatly from the initial values that we can encounter numerical errors if we are not careful.

Even a process with AR coefficients which are only slightly stable and with standard normal error process can return observations greater than 10^{18} if the process runs for over 6,000 time steps, as we have done for much of this project. Observations like this in particular when they are summed (for the purposed of sum of squares) can cause numerical overflow or round off error. This issue is, as such, implicit in the estimation of non-stationary ARMA processes.

Figure 6.3 and surrounding discussion denotes how small biases in estimation can result in significant errors in a filtered series. We have observed that, for an AR process with unstable root with lower absolute value, this error can occur when the series is filtered using exact inputs. The result implying that the accuracy of finite precision numerical representation is not always enough to guarantee inaccuracies are avoided.

Figures 7.1 and 7.2 demonstrate another example of how large observations can impact estimation. In this example, the surface of the optimisation criterion (sum of squared errors) deteriorates as T reaches large values. This renders the estimation unworkable. Note that, in this example, the AR parameters are fixed and exact.

7.2 Optimisation errors

The conditional likelihood function of a non-stationary ARMA process is known to have local minima apart from the global minim where the optimal estimator is observed. This remains the case with the AR(p^*) approximation of the process.

The optimisation method utilised throughout this project has been the quasi-Newton "BFGS" (Broyden, Fletcher, Goldfarb and Shanno) method. This method performs adequately. However, the method is susceptible to returning coefficient estimates which correspond to a local minimum in the conditional likelihood. Where unusual estimates have been returned, a multidimensional grid search has been employed to correct the errors. In practical use, and particularly if higher order ARMA processes are to estimated, more sophisticated global search algorithms may be required.

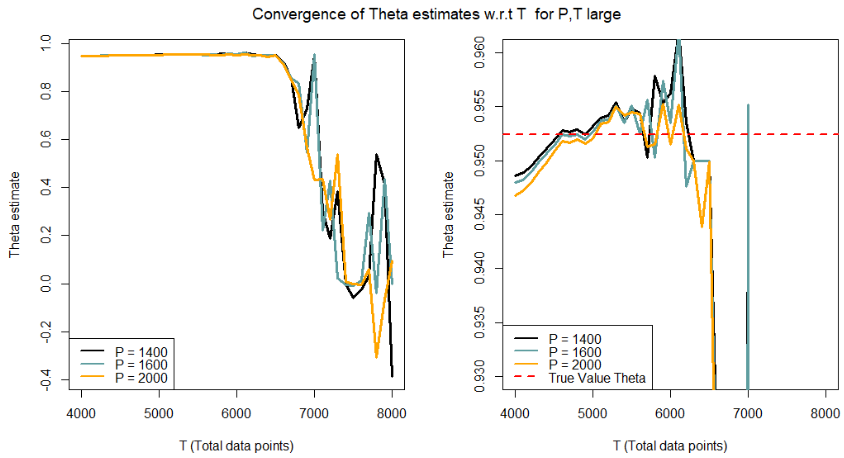


Figure 7.1

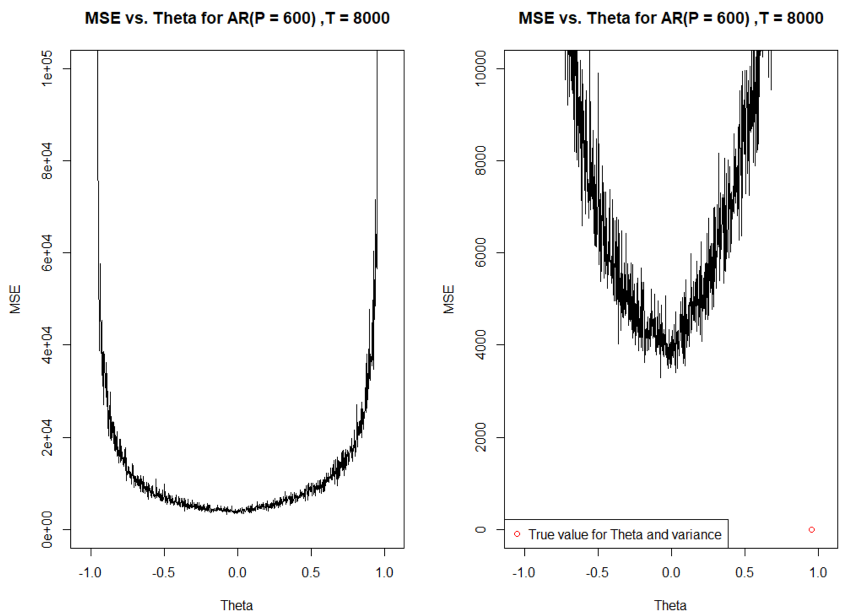


Figure 7.2

Conclusions and Further Research

The original motivation for this paper was to investigate whether the AR(p^*) method would provide an adequate approximation to a non-stationary ARMA(p,q) process and to, based on the behaviour of the approximation, form hypotheses which may be extended to the general non-stationary ARMA case. I feel the hypotheses we have formed have been considered and insightful. I also believe that the methods we have defined and examined in this paper provide a basis for an exciting tool for forecasting problems.

There are a number of ways in which the findings of this paper could be developed with further research and so, to conclude this paper we will discuss each of these avenues.

8.1 Theoretical Findings

This paper is motivated by the work on Hanzon & Scherrer (2019) and serves to forming conjectures that may build upon their work. The hypothesis formed through this paper are being considered and further hypotheses and conjectures are being formed. We anticipate that the results of this project would lead to useful conjectures surrounding non-stationary ARMA processes which may be proven or disproven by Professor Hanzon and Professor Scherrer. In summary, we hope that this project may be useful in developing the theory of non-stationary ARMA processes, for which there is little theory available.

8.2 Applications to Forecasting

The methods established through this paper may prove to be useful in forecasting linear time series. More examination would be required however to establish first, the viability of the methods for forecasting applications, and secondly, which method performs best and under what conditions.

The viability of the methods will likely depend on effectiveness and efficiency of the optimisation method. As discussed in section 7.2, the optimisation is susceptible to converging to local minima apart from the global minimum. Where the true coefficients are unobserved, this may pose a serious difficulty.

We would expect based upon results of 6.3 that the method of filtering $\theta(L)$ as in section 6.1 would be the preferred method for forecasting. However, an in-depth analysis would be of interest.

8.3 Generalisation to higher order ARMA processes

Through this paper, we have inspected ARMA process of order $p, q \leq 2$. A natural step forward would be to investigate whether higher order processes and SARIMA models display the same behaviour under the AR(p^*) method and if not, in what ways do they differ. Further progression of this would be to investigate if the $AR(p^*)$ method can be generalised to vector ARMA processes and linear scalar ARMA processes with non-Gaussian errors.

The equivalence of sum of squares and conditional maximum likelihood would not hold where errors are non-Gaussian case so the procedure would likely be more computationally expensive in this case.

Bibliography

[1] Lai, T. L., & Wei, C. Z. (1982). Asymptotic properties of projections with applications to stochastic regression problems. *Journal of Multivariate Analysis*, 12(3), 346-370.

[2] Lütkepohl, H. (2007). *New Introduction to Multiple Time Series Analysis.* Springer Science & Business Media.

[3] Aoki, M. (1987) *State Space Modeling of Time Series*.

[4] Hanzon, B. (2019). *A linear time series model for which the MLE does not exist-and possible Remedies*. Department of Mathematics, University College Cork.

[5] Peeters, R. L., & Hanzon, B. (1999). Symbolic computation of Fisher information matrices for parametrized state-space systems. *Automatica*, 35(6), 1059-1071.

[6] Hallin, M. (1984). Spectral factorization of nonstationary moving average processes. The Annals of Statistics, 172-192.

Examples of Code

The coding for this project has been carried out using the R coding language. Included are examples of R code used throughout this project. This is not an exhaustive catalog, of course. However, all simulations employed throughout this project can be derived as variations and combinations of the examples presented here. Versions of similar functions for various models have been included with the view that they may be useful in resulting research or in other research applications.

A.1 Data Generating Process (DGP)

The errors process of the data generating process has been simulated by using the 'rnorm()' function in R, which generates random observations from a normal distribution. Storing the errors, as well as the resulting observations of the time series, has proven to be useful throughout for various analysis, construction and reconstruction of time series.

ARMA(1,1)

```
ARMA11  = function(ar_coef , ma_coef , x , N){
  errors = c(0)
  for(t in 2:N){
    errors[t] = rnorm(1 , mean = 0 , sd = 1)
    x[t] = errors[t]  + ar_coef*x[t-1] + ma_coef*errors[t-1]
  }
  dataframes = matrix(data = c(x , errors) , ncol = 2
  , nrow = N , byrow = F)
  colnames(dataframes) = c("Yt" , "errors")
  return(dataframes)
}
```

ARMA(2,1)

```
ARMA21 = function(ar_coef , ma_coef , x , N){
  errors = c(0,0)
  for(t in 3:N){
    errors[t] = rnorm(1 , mean = 0 , sd = 1)
    x[t] = errors[t]  + ar_coef[1]*x[t-1] + ar_coef[2]*x[t-2]
```

```
    + ma_coef*errors[t-1]
  }
  dataframes = matrix(data = c(x , errors) ,
  ncol = 2 , nrow = N , byrow = F)
  colnames(dataframes) = c("Yt" , "errors")
  return(dataframes)
}
```

A.2 Estimation Criterion

As proven in section 4.1, maximising conditional likelihood is equivalent with minimising the residual sum of squares given a Gaussian error process. Sum of squares calculations are more computationally efficient and so, the estimation criterion was employed as minimum sum of squares.

ARMA(1,1)

```
crit.ARMA11 = function(v, y , p){
  phi = v[1]
  theta = v[2]

  T.final = length(y)

  y.hat = numeric((T.final- p-1))

  for (t in (p+1):T.final) {
    y.hat[t-p] = y.hat_ARMA11(phi ,theta , y , t , p)
  }
  return(sum((y[(p+1):T.final] - y.hat)^2))
}

y.hat_ARMA11 = function( phi ,theta , y , t , p){
  fit = 0
  for (i in 1:p){
    fit = fit + (phi + theta)*((-theta)^(i-1))*y[t-i]
  }
  return(fit)
}
```

ARMA(2,1)

```
 crit.ARMA21 = function(v, y , p){
  phi1 = v[1]
  phi2 = v[2]
  theta = v[3]

  T.final = length(y)
```

```
  y.hat = numeric((T.final- p-1))

  for (t in (p+1):T.final) {
    y.hat[t-p] = y.hat_ARMA21(phi1 , phi2 ,theta , y , t , p)
  }
  return(sum((y[(p+1):T.final] - y.hat)^2))
}

y.hat_ARMA21 = function( phi1 , phi2 ,theta , y , t , p){
  fit = (phi1 + theta)*y[t-1]
  for (i in 2:p){
    fit = fit + (phi2 - phi1*theta - theta*theta)*((-theta)^(i-2))*y[t-i]
  }
  return(fit)
}
```

A.3 Monte Carlo Simulation

Much of the data presented through this paper has been generated through Monte Carlo Simulation. The following is an example of one simulation, where we are gathering coefficient estimates for a range of values for T for a given ARMA(1,1) model and p^* fixed.

```
#N = number of simulations
N = 150

#Set working directory for data
setwd("")

for (t in seq(from = 1000+P.i , by = 1000 , length.out = 5)) {
  print(t)

  optim.Est.stages.M5 = matrix(data = NA , nrow = N , ncol = 10 )
  colnames(optim.Est.stages.M5) = c("P" , "Phi" , "Theta - AR(P)" ,
  "MSE - AR(P)" , "Theta - Differencing (Phi_hat)" ,
  "MSE - Differencing (Phi_hat)"
  , "Theta - Differencing with Phi true values",
  "MSE - Differencing Phi true values"
  ,"Theta - MA1 DGP" , "MSE - MA1 DGP")

  set.seed(0404)
  for (k in 1:N){
    #generate time series Y.k and error process u.k
    dat.k = ARMA11(ar_coef = 1/.995 , ma_coef = 1/.95
    , x = c(0) , N = t)

    Y.k = dat.k[,1]
```

```
    u.k = dat.k[,2]

    #i) Estimate long AR(P) and store estimates
    m.AR = optim(par = c(1/.995,.95) , y = Y.k , p = P.i , fn = crit.ARMA11 ,
    method = "BFGS")

    optim.Est.stages.M5[k , 1 ] = P.i
    optim.Est.stages.M5[k , c(2:3)] = m.AR$par
    optim.Est.stages.M5[k , 4] = (m.AR$value)/(t-P.i)

    #ii) Difference using long AR coefficient estimates
    - estimate and store values
    u.diff.est = difference.ARMA11(ar.coef = m.AR$par[1] , y = Y.k )

    m.diff.est = arima(x = u.diff.est , order = c(0,0,1)
    , method = "ML" , include.mean = F )

    optim.Est.stages.M5[k,5] = m.diff.est$coef
    optim.Est.stages.M5[k,6] = m.diff.est$sigma2

    #iii) Difference using true AR coefficient estimates
    - estimates and store values
    u.diff.true = difference.ARMA11(ar.coef = 1/.995 , y = Y.k )

    m.diff.true = arima( x = u.diff.true , order = c(0,0,1) ,
    method = "ML",include.mean = F )

    optim.Est.stages.M5[k,7] = m.diff.true$coef
    optim.Est.stages.M5[k,8] = m.diff.true$sigma2

    #iv) Estimate MA1 process generated from error process u
    Y.MA.k = MA1.fit(u = u.k , theta = 1.05)

    m.MA = arima(x = Y.MA.k , order = c(0,0,1) ,
    method = "ML" , include.mean = F)

    optim.Est.stages.M5[k,9] = m.MA$coef
    optim.Est.stages.M5[k,10] = m.MA$sigma2

    #print iteration to track progress of code( this takes a long time to run
    - bit of insight on progress is nice to have )
    print(k)
  }

  st = paste("ARMA11 - T =", as.character(t) , ", P = ", as.character(P.i) , "
  - conv wrt T.csv" )
  write.csv(optim.Est.stages.M5 , st)
}
```

A.4 Filtering of purely explosive model

The following code corresponds to the filtering of the AR component of an ARMA process. This procedure is discussed in detail in section 6.1.

ARMA(1,1)

```
difference.ARMA11 = function( ar.coef , y){
  T.final = length(y)

  #ut values
  u = numeric(T.final-1)
  for (t in 2:T.final) {
    u[t-1] = y[t] - ar.coef*y[t-1]
  }
  return(u)
}
```

ARMA(2,1)

```
difference.ARMA21 = function( ar.coef , y){
  phi1 = ar.coef[1]
  phi2 = ar.coef[2]
  T.final = length(y)

  #ut values
  u = numeric(T.final-2)
  for (t in 3:T.final) {
    u[t-2] = y[t] - phi1*y[t-1] - phi2*y[t-2]
  }
  return(u)
}
```

A.5 Filtering unstable root from a Mixed Root AR component

The following code corresponds to the filtering of the unstable root in a mixed root AR component of an ARMA process. This procedure is discussed in detail in section 6.2.

```
#i) Estimate long AR(p) and store estimates
m.AR = optim(par = model5.coef.stable.ReAR , y = Y.k , p = P.i
, fn = crit ,  method = "BFGS")

#ii) get roots of model and store unstable root
roots = polyroot(c(1 , -m.AR$par[1:2]))

r.u = .9
```

```
# provides a control in Monte Carlo simulation to avoid roots
from previous iterations being carried forward.

r.u = Re(roots[which(abs(Re(roots)) < 1)])

optim.Est.stages.M5.ReAR[k , 6 ] = r.u

#iii) generate series z_t

z.k = c()
for(t in 2:length(Y.k)){
  z.k[t-1] =  Y.k[t] - Y.k[t-1]/r.u
}

#iv) estimate full series and store values
m.fseries.est = arima(x = z.k , order = c(1,0,1)
, method = "ML" , include.mean = F )

optim.Est.stages.M5.ReAR[k , 7:8 ] = m.fseries.est$coef
optim.Est.stages.M5.ReAR[k , 9 ] = m.fseries.est$sigma2
```

A.6 Multivariate Grid Search

As discussed in section 7.2, optimisation errors have occurred a number of time throughout this project. errors have been validated and resolved through the use of multivariate grid search.

```
#Read in data of interest and Set p*
Y.P = Y[1:9900]
P = 1900

#define search region
#AR component
delta.phi = .01
N.phi = 10

#range of search across AR coefficients
delta.phi*N.phi

#MA component
delta.theta = .01
N.theta = 10
#range of search across MA component
delta.theta*N.theta

#number of sum of squares computations
(N.phi**2)*N.theta*5
```

```
#Create sequence of parameter values
phi1.vals = seq(from = model5.coef[1] - delta.phi*(N.phi/2),
to = model5.coef[1] + delta.phi*(N.phi/2) ,
length.out = N.phi )

phi2.vals = seq(from = model5.coef[2] - delta.phi*(N.phi/2) ,
to = model5.coef[2] + delta.phi*(N.phi/2) ,
length.out = N.phi )

theta.vals = seq(from = model5.coef[3] - delta.theta*(N.theta/2) ,
to = model5.coef[3] + delta.theta*(N.theta/2),
length.out = N.theta )

# Create 3D matrix to store
crit.vals = array( unlist(NA), dim=c(N.phi, N.phi, N.theta)
,dimnames = list(phi1.vals , phi2.vals , theta.vals) )

for (i in 1:N.phi) {
  for (j in 1:N.phi) {
    for (k in 1:N.theta) {
      vi = c(phi1.i = phi1.vals[i] , phi2.j = phi2.vals[j]
      , theta.k = theta.vals[k])

      crit.vals[i,j,k] = crit(vi, y = Y.P , p = P)/(length(Y.P))
    }
  }
}
```