AMSI VACATION RESEARCH SCHOLARSHIPS 2019–20

EXPLORE THE MATHEMATICAL SCIENCES THIS SUMMER

A New ARMA Model Fitting Algorithm for Big Time Series Data

Luke Yerbury

Supervised by Ali Eshragh and Glen Livingston

University of Newcastle

Vacation Research Scholarships are funded jointly by the Department of Education and Training and the Australian Mathematical Sciences Institute.

Abstract

Big data matrix compression via non-uniform sampling schemes such as leverage score sampling, provide excellent alternatives to naïve computation that result in high-quality numerical implementations and strong theoretical guarantees. In the context of autoregressive (`AR`) time series models, a highly efficient algorithm to approximate leverage scores of the underlying regressors has recently been established. For more general `ARMA` models, unlike the `AR` model, the likelihood function is a non-convex nonlinear function which makes the problem more complicated. One approach considered here is utilising the Hannan-Rissanen (`HR`) algorithm, which allows unobserved white noise to be approximated using high order `AR` models. An investigation into the optimal order of this intermediary `AR` model and the effectiveness of the trimming step in the `HR` algorithm has been conducted.

Contents

1 Introduction

A *time series* is a sequence of random variables indexed according to the order they are observed in time. Considered here are regularly sampled time series, some examples for which include daily closing stock prices, daily sunspot counts, monthly average temperature in Newcastle, etc. The primary objective of time series analysis is to forecast the future behaviour of a system via the construction of suitable models. Such modelling can be approached from one of the *frequency domain* or *time domain* perspectives. Frequency domain exploits methods from Fourier Analysis to build models that focus on the periodic variations in the data. The time domain approach observes the correlation between the time series and lagged versions of itself to build parametric functions of past and present values. The latter is more prevalent for its simplicity and typically superior performance, and the models featured in this report belong to that framework.

A time series is said to be *stationary* when the mean function and variance are independent of time, and the autocovariance depends only on the time difference. Time series with this property can have an *autoregressive moving average* (`ARMA`) model fitted to them. The autoregressive (`AR`) component refers to linear regression of current values of the time series against particular lagged values where significant correlation was identified. The moving average (`MA`) component involves linear regression of current values against previous *white noise* - which are uncorrelated, zero mean, equal variance random variables representing variation not explained by the series itself. Introduced by Box and Jenkins (1976), these simple models are still widely used to great effect.

An ever increasing capacity to collect large masses of data has required reconsideration of typical approaches to data analysis. The fitting of `ARMA` models involves solving many ordinary least squares problems, and in the context of big time series data, this can create a significant computational bottleneck. Randomised Numerical Linear Algebra (RandNLA) employs random sub-sampling routines to develop improved algorithms for large-scale linear algebra problems such as matrix multiplcation, regression and low rank matrix approximations (Drineas and Mahoney, 2017). By intelligent sampling, the matrices involved in these problems can be compressed in such a way that they still retain important properties of the original matrix. Calculations subsequently performed using the compressed matrices are then not only more efficient, but also theoretically accurate with high probability (Mahoney, 2011, Woodruff, 2014). Sampling based on leverage scores has been shown to aptly identify non-uniformity in data, ultimately providing strong theoretical guarantees and high quality numerical implementations (Drineas et al. (2012)). The `LSAR` algorithm, introduced by Eshragh et al. (2019), uses leverage score based sampling to estimate the order and parameters of an `AR` model, a special

case of an ARMA model without the MA component. Naïvely, computation of the leverage scores in this ordinary least squares (OLS) regression context is as costly as solving the original OLS problem. By exploiting the Toeplitz structure of the design matrix, Eshragh et al. developed a method to recursively calculate the leverage scores during the model fitting process. They also developed theoretical relative error bounds for these recursively calculated leverage scores with high probability, and conducted empirical testing to demonstrate the effectiveness of the final algorithm compared to state-of-the-art alternatives.

A natural progression upon establishing LSAR is to extend the algorithm to the LSARMA, allowing for the inclusion of MA terms. This is much less straightforward due to the non-convex, non-linear nature of the likelihood function for ARMA parameters. One solution to this problem is to use the Hannan-Rissanen (HR) algorithm. This algorithm exploits the equivalence between invertible MA(q) and AR(∞) models to fit a large order AR model to the data, the residuals from which can then be used in place of the unobserved white noise in OLS parameter estimation. An optional extra *trimming* step improves the original estimates. The aim of this report was to investigate the characteristics of the HR algorithm within the context of big time series data to inform algorithmic decisions in the formulation of LSARMA. Of primary concern was understanding how large the order of the intermediate AR model should be for various models, and understanding the effectiveness of the trimming step. Simulated data from known ARMA processes were used for this investigation.

Statement of Authorship

Eshragh and Livingston devised and supervised the project. They provided initial code frameworks which were altered and extended by Yerbury. Yerbury performed the simulations and interpreted results with support from Livingston.

2 Background Theory

This section provides an overview of the relevant theory behind the simulations performed for this project.

2.1 Autoregressive Moving Average Models

A time series $\{X_t;\, t = 0, \pm1, \pm2, \ldots\}$ is a sequence of random variables indexed according to the order they are observed in time. A realisation of the random variable X_t is denoted x_t. The time series X_t is called (weakly) stationary if:

(i) the mean function $\mathbb{E}[X_t]$ is constant, and;

(ii) the autocovariance function $Cov(X_t, X_{t+h})$ only depends on the lag h (independent of time t).

A time series process is `ARMA`(p, q) if it is stationary and

$$X_t \;= \phi_1 X_{t-1} + \cdots + \phi_p X_{t-p} + \theta_1 W_{t-1} + \cdots + \theta_q W_{t-q} + W_t$$

where $\phi_p \neq 0$, $\theta_q \neq 0$ and the time series $\{W_t;\, t = 0, \pm1, \pm2, \ldots\}$ is a Gaussian white noise process, meaning $\mathbb{E}[W_t] = 0$ and $Cov(W_t, W_s) = \delta_{ts}\sigma_W^2$ where δ_{ts} is the Kronecker delta.
Another way of expressing the above `ARMA` model is through the use of the *backshift operator* B. We can define $BY_t = Y_{t-1}$, which naturally extends to powers with $B^k Y_t = Y_{t-k}$. Hence we can define the *autoregressive operator* $\Phi_p(B) = 1 - \phi_1 B - \ldots - \phi_p B^p$ and the *moving average operator* $\Theta_p(B) = 1 + \theta_1 B + \ldots + \theta_q B^q$ to express the model as

$$\Phi_p(B) X_t = \Theta_q(B) W_t\,.$$

Additionally, an `ARMA`(p, q) process is said to be *invertible* if the time series can be written as:

$$W_t = \sum_{j=0}^{\infty} \pi_j X_{t-j}, \qquad \text{where } \pi_0 = 1 \text{ and } \sum_{j=0}^{\infty} |\pi_j| < \infty$$

Equivalently, the process is invertible if and only if the roots of the `MA` polynomial $\Theta_q(z)$, for $z \in \mathbb{C}$, lie outside the unit circle. Analagous to this definition, the process is said to be *causal* if it can be written as:

$$X_t = \sum_{j=0}^{\infty} \psi_j W_{t-j}, \qquad \text{where } \psi_0 = 1 \text{ and } \sum_{j=0}^{\infty} |\psi_j| < \infty$$

or again, if the roots of the `AR` polynomial $\Phi_p(z)$ lie outside the unit circle.
[Shumway and Stoffer, 2017]

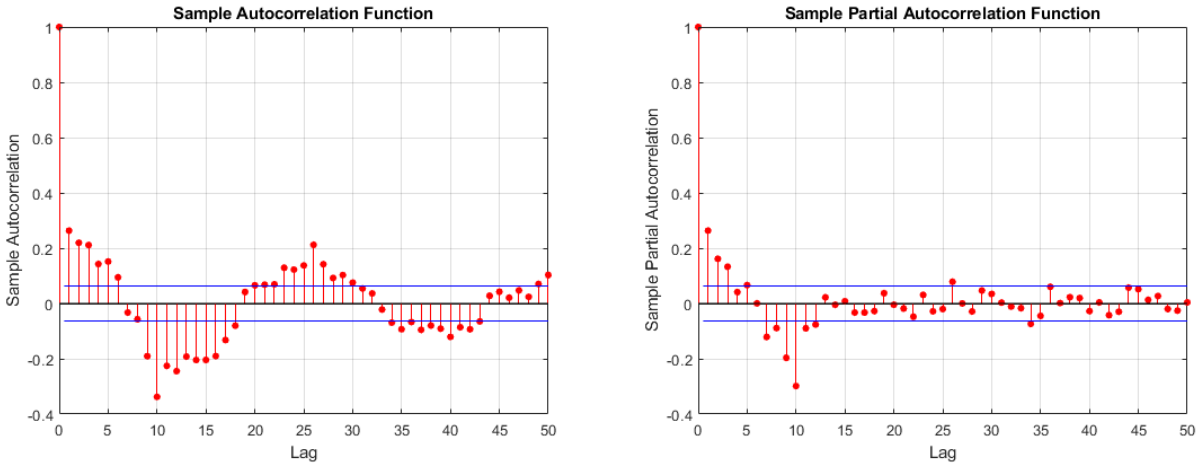

Figure 1: The sample ACF and PACF for an $\texttt{ARMA}(10, 3)$ process.

	$\texttt{AR}(p)$	$\texttt{MA}(q)$	$\texttt{ARMA}(p, q)$
ACF	Tails off	Cuts off after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off

Table 1: Behaviour of ACF and PACF for various `ARMA` Models (Shumway & Stoffer, 2017)

2.1.1 Estimating Model Orders

The *orders* p and q are commonly estimated for unknown processes using the *autocorrelation function* (ACF) and *partial autocorrelation function* (PACF) in conjunction. See Figure 1 for example ACF and PACF plots where the blue lines are 95% zero-confidence bounds. Table 1 indicates the behaviour expected in these plots for various `ARMA` models. For pure `AR` or `MA` models, estimates for p and q are readily obtained as the greatest non-zero (partial) autocorrelation, which acts as a cut-off, with all (partial) autocorrelations thereafter being zero. For mixed `ARMA` models, it is not always clear which one is tailing off or cutting off. In such cases, these plots become tools that provide approximate starting points for p and q. Subsequently, models with values of p and q within a reasonable neighbourhood should be constructed and compared using a suitable criterion, such as AIC or prediction error.

2.1.2 Estimating the Model Parameters

Subsequent to estimating the orders of a time series process, there are $p+q+1$ parameters to estimate including the coefficients ϕ_i, θ_i and the white noise variance σ_W^2. Let $x_1, \ldots, x_n$ be a sample from an $\texttt{ARMA}(p, q)$ process. The maximum likelihood estimate (MLE) of $(\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q, \sigma_W^2)^\intercal$ can be

obtained by supposing the x_t are from a given distribution (Gaussian for instance). Unfortunately it is not possible to find an analytical solution for the MLE due to the non-linear, non-convex nature of the likelihood surface. Numerical procedures must be employed instead and these methods require good initial values to converge.

An alternative is to use the typical regression *least squares estimate* given by $\hat{\boldsymbol{\psi}} = (\boldsymbol{Z}^\intercal \boldsymbol{Z})^{-1} \boldsymbol{Z}^\intercal \boldsymbol{x}$, where we suppose that $\boldsymbol{\psi} = (\phi_1, \dots \phi_p, \theta_1, \dots \theta_q)^\intercal$, $\boldsymbol{x} = (x_{p+1}, x_{p+2}, \dots x_n)^\intercal$ and for $p > q$

$$\boldsymbol{Z} = \begin{pmatrix} x_p & x_{p-1} & \cdots & x_1 & w_p & \cdots & w_{p-q+2} & w_{p-q+1} \\ x_{p+1} & x_p & \cdots & x_2 & w_{p+1} & \cdots & w_{p-q+3} & w_{p-q+2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n-1} & x_{n-2} & \cdots & x_{n-p} & w_{n-1} & \cdots & w_{n-q+1} & w_{n-q} \end{pmatrix}.$$

If $q > p$ then,

$$\boldsymbol{Z} = \begin{pmatrix} x_q & x_{q-1} & \cdots & x_{q-p+1} & w_q & \cdots & w_2 & w_1 \\ x_{q+1} & x_q & \cdots & x_{q-p+2} & w_{q+1} & \cdots & w_3 & w_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n-1} & x_{n-2} & \cdots & x_{n-p} & w_{n-1} & \cdots & w_{n-q+1} & w_{n-q} \end{pmatrix}.$$

However, the white noise series w_t is unobserved, so must be replaced with approximations $\hat{w}_t$. The Hannan-Rissanen Algorithm can be applied to produce those estimates. This algorithm exploits the representation of any invertible $\texttt{MA}(q)$ model as an $\texttt{AR}(\infty)$ (cf. Section 2.1). The algorithm will now be described in detail (Hannan and Rissanen, 1982 and Brockwell and Davis, 2006).

Step 1: Fit a high order $\texttt{AR}(\tilde{p})$ model to the data

An `AR` model with an appropriately high order, $\tilde{p}$ - usually taken to be greater than the $\max\{p, q\}$ - is fitted to the data by minimising the conditional log-likelihood function which is equivalent to computing the least squares estimate $\hat{\boldsymbol{\beta}} = (\boldsymbol{Y}^\intercal \boldsymbol{Y})^{-1} \boldsymbol{Y}^\intercal \boldsymbol{y}$, where $\boldsymbol{\beta} = (\phi_1, \dots, \phi_{\tilde{p}})^\intercal$, $\boldsymbol{y} = (x_{\tilde{p}+1}, x_{\tilde{p}+2}, \dots, x_n)^\intercal$ and

$$\boldsymbol{Y} = \begin{pmatrix} x_{\tilde{p}} & x_{\tilde{p}-1} & \cdots & x_1 \\ x_{\tilde{p}+1} & x_{\tilde{p}} & \cdots & x_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{n-1} & x_{n-2} & \cdots & x_{n-\tilde{p}} \end{pmatrix}.$$

Step 2: Use the residuals from the long AR model as estimates for the w_t

The residuals can then be used as estimates for the unobserved white noise in the design matrix $\boldsymbol{Z}$, i.e. $\hat{w}_t = \boldsymbol{y}_{t-\tilde{p}} - \boldsymbol{Y}_{(t-\tilde{p},*)}\hat{\boldsymbol{\beta}}$ for $t \in (\tilde{p}+1, \ldots, n)$, where $\boldsymbol{Y}_{(m,*)}$ denotes the m^{th} row of matrix $\boldsymbol{Y}$. Necessarily the first $\tilde{p}$ white noise estimates remain unknown, and can be made zero for subsequent use in the $\boldsymbol{Z}$ matrices mentioned earlier, or excluded from the OLS calculations. Estimates obtained from either of these methods are asymptotically equivalent. If they are to be excluded then $\hat{\boldsymbol{\psi}} = (\boldsymbol{Z}^\intercal \boldsymbol{Z})^{-1}\boldsymbol{Z}^\intercal \boldsymbol{x}$ with $\boldsymbol{x} = (x_{\tilde{p}+q+1}, x_{\tilde{p}+q+2}, \ldots x_n)^\intercal$ and

$$\boldsymbol{Z} = \begin{pmatrix} x_{\tilde{p}+q} & x_{\tilde{p}+q-1} & \cdots & x_{\tilde{p}+q+1-p} & \hat{w}_{\tilde{p}+q} & \cdots & \hat{w}_{\tilde{p}+2} & \hat{w}_{\tilde{p}+1} \\ x_{\tilde{p}+q+1} & x_{\tilde{p}+q} & \cdots & x_{\tilde{p}+q+2-p} & \hat{w}_{\tilde{p}+q+1} & \cdots & \hat{w}_{\tilde{p}+3} & \hat{w}_{\tilde{p}+2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n-1} & x_{n-2} & \cdots & x_{n-p} & \hat{w}_{n-1} & \cdots & \hat{w}_{n-q+1} & \hat{w}_{n-q} \end{pmatrix}.$$

Note: If either $p = 0$ or $q = 0$, $\boldsymbol{Z}$ contains only the last q columns or first p columns respectively. An estimate for the variance of the white noise can also be obtained as $\hat{\sigma}^2_W = (\boldsymbol{x} - \boldsymbol{Z}\hat{\boldsymbol{\psi}})/(n - \tilde{p} - q)$.

Step 3: Trim the parameter estimates

Step 3 would not be completed if the `HR` algorithm is only used to obtain initial estimates for numerical MLE. This step seeks to improve the parameter estimates, and produces estimates with the same asymptotic distribution as the maximum likelihood estimators.

Now let $g = \max\{p, q\}$ and note that $\hat{\boldsymbol{\psi}} = (\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\theta}})$. New estimates for the white noise ω_t are then obtained recursively as

$$\hat{\omega}_t = \begin{cases} 0, & 1 \le t \le g \\ x_t - \sum_{j=1}^{p} \hat{\phi}_j x_{t-j} - \sum_{k=1}^{q} \hat{\theta}_k \hat{\omega}_{t-k}, & g < t \le n \end{cases}$$

These are then used to form

$$\zeta_t = \begin{cases} 0, & 1 \le t \le g \\ \sum_{j=1}^{p} \hat{\phi}_j \zeta_{t-j} + \hat{\omega}_t, & g < t \le n \end{cases} \qquad \eta_t = \begin{cases} 0, & 1 \le t \le g \\ -\sum_{k=1}^{q} \hat{\theta}_k \eta_{t-k} + \hat{\omega}_t, & g < t \le n \end{cases}$$

where $\hat{\Phi}(B)\zeta_t = \hat{\omega}_t$, $\hat{\Theta}(B)\eta_t = \hat{\omega}_t$ and hence $\hat{\Phi}(B)\zeta_t = \hat{\Theta}(B)\eta_t$.

Let $\hat{\boldsymbol{\psi}}^\dagger$ be the regression estimate for $\boldsymbol{\psi}^\dagger$ found when regressing $\hat{\omega}_t$ on $\{\zeta_{t-1}, \ldots, \zeta_{t-p}, \eta_{t-1}, \ldots, \eta_{t-q}\}$ for $t \in (g+1, \ldots, n)$. Then the trimmed parameter estimate is $\hat{\boldsymbol{\psi}} + \hat{\boldsymbol{\psi}}^\dagger$. This step can be repeated using the residuals derived from the trimmed parameter estimates until the estimates, or $\hat{\sigma}^2_W$ converge.

3 Results and Discussion

As already indicated, this project aimed to investigate two major characteristics of the HR algorithm and the interplay between them. The first is regarding the optimal order of the intermediate AR(p) model, and the second concerns the trimming step and its effectiveness. This section explores the results and insights from the various simulations that were conducted using MATLAB on The University of Newcastle's Research Compute Grid (RCG).

3.1 The Optimal Value of $\tilde{p}$

The equivalence between invertible MA(q) and AR(∞) models may lead one to expect that larger values of $\tilde{p}$ would perform strictly better than smaller values. Of course this equivalence does not consider finite samples, as Broersen (2000) noted, 'Practice and simulations, however, have shown that the best AR order in estimation is finite and depends on the true process parameters and on the number of observations.' As seen in Figure 2, there is clearly an optimal value achieved for some $\tilde{p}$ that is much smaller than the available sample size. Broersen focused on this question of order in the context of Durbin's method (essentially the first two steps of the HR algorithm) and even derived some theoretical approaches to find the optimal order. The sliding window technique he described is effective in the small simulations he performs to investigate the finite sample behaviour, however it is not practical in the context of big data due to the computational requirements. In Hannan and Rissanen (1982), 10 000 data points is considered 'very large' and Broersen never ran simulations on even that scale. This might well be due to the assumption that finite sample behaviour is no longer worth investigating for larger sample sizes where asymptotic behaviour is expected to dominate, or is due to the associated computational demands.

The HR algorithm was used (without trimming) to obtain parameter estimates for simulated data produced from a collection of 21 ARMA models with known orders and parameters, for all values of $\tilde{p}$ ranging between $p+q+1$ and 500. This collection of models was comprised of 7 sets of orders, (p, q), with 3 different sets of parameters for each. The parameters were generated randomly in MATLAB using a try/catch loop to ensure the models were both invertible and causal. Three sample sizes were tested for each of the 21 models, namely 250 000, 1 000 000 and 5 000 000. The first two sample sizes were chosen for computational convenience, with the last sample size being more representative of the order of modern big data regimes. Table 2 shows that only those models with $p \geq q$ were considered in this project. This is due to a difficulty encountered in extending the LSAR algorithm to include an MA component. The recursive calculation of leverage scores is not yet understood for the cases where

$p < q$, and this will constitute future work.

In order to determine which value of $\tilde{p}$ resulted in the *best* parameter estimates, the *relative error of the parameter estimates* was recorded for each value, that is,

$$\mathrm{RE}_{\hat{\psi}} = \frac{\|\boldsymbol{\psi} - \hat{\boldsymbol{\psi}}\|}{\|\boldsymbol{\psi}\|}\,.$$

For future instances where models are to be fitted to data from unknown processes, an alternative measure of *best* also under consideration was the *relative error of the white noise estimates*, that is,

$$\mathrm{RE}_{\hat{\omega}} = \frac{\|\hat{\boldsymbol{\omega}} - \hat{\boldsymbol{w}}\|}{\|\hat{\boldsymbol{w}}\|}\,.$$

This measure is something that could be conditioned on within the `LSARMA` algorithm to choose an optimal $\tilde{p}$. The intuition here is that after obtaining white noise estimates from the `AR` model, estimates subsequently obtained by fitting the full `ARMA` model should have changed minimally if the optimal $\tilde{p}$ value was chosen. Suboptimal values of $\tilde{p}$ would be expected to produce white noise estimates that differ more significantly from the subsequent, full `ARMA`, white noise estimates. The efficacy of this measure was confirmed during this project and is explored in more detail later.

Due to the random nature of the simulated data, computations were repeated 400 times so averaging would produce a smoothing effect on both of the relative error estimates. This value of 400 would ideally have been larger, but was chosen due to the individual-user submission restrictions on the RCG and the time constraints of the project. For many models, 400 repetitions was sufficient to produce relatively smooth plots with clear systemic minima (see Figure 7), rather than the erratic minima observed for individual cases - an outcome of simulation. Many still required smoothing for the identification of clear minima, and so a smoothing process was applied to the averaged data for each model and sample size. Of those models requiring smoothing, particular models displayed significant and/or periodic variations (see Figure 8 and 9) and it would be interesting to see if these remain even after conducting many more repetitions. An out-of-the-box smoothing function (`smooth`) was used in MATLAB which employed 'local regression using weighted linear least squares and a 2nd degree polynomial model'. This was chosen after experimenting and comparing plots with other alternatives, where it was deemed to provide the best results across the board.

$n = 250\,000$	$n = 1\,000\,000$	$n = 5\,000\,000$

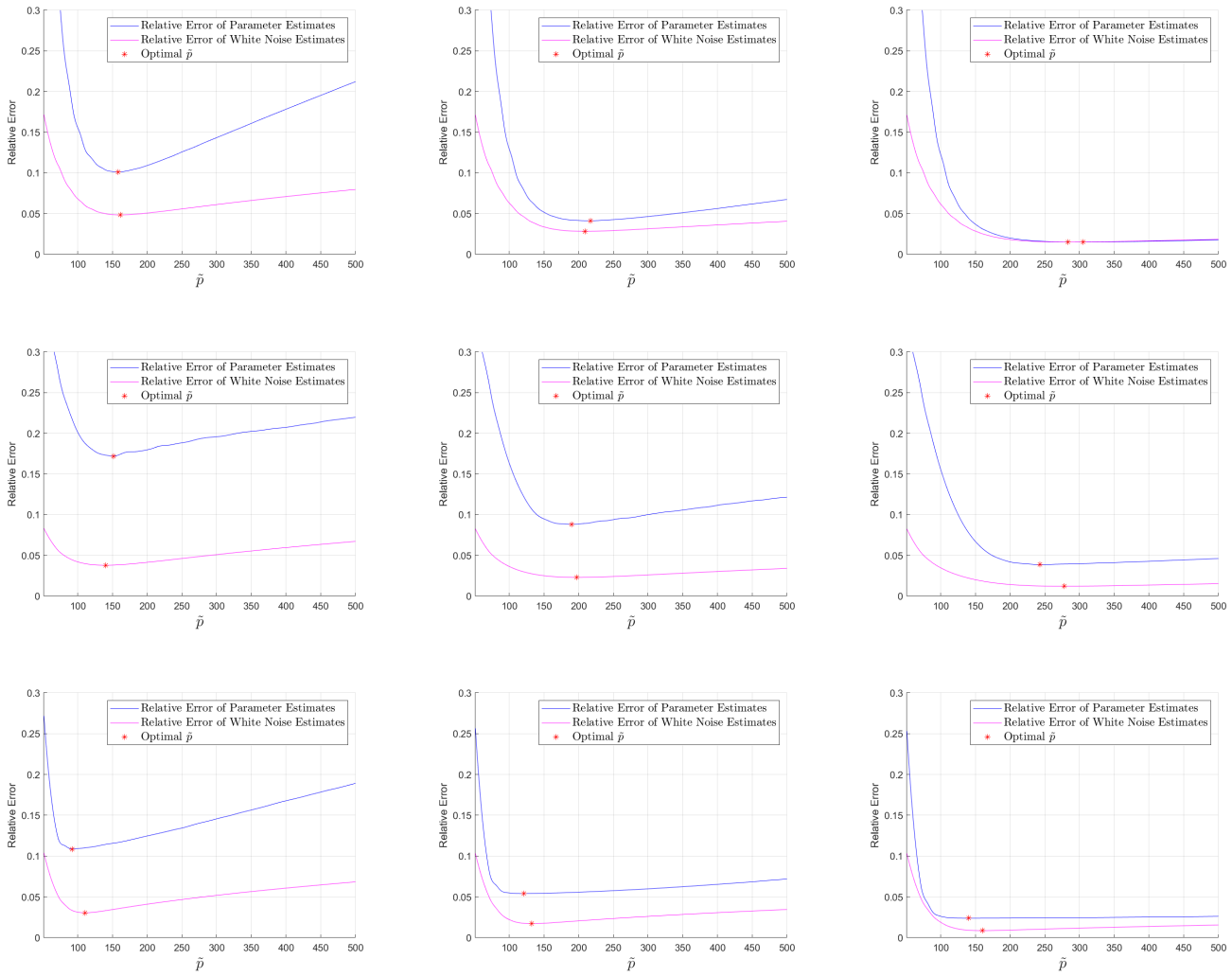

Figure 2: All figures show both the relative error of the parameter and white noise estimates from the `HR` algorithm without trimming, for three different `ARMA`$(30, 20)$ models (Models 1, 2 and 3 in Table 2). Each row represents a different random set of parameters and each column represents a different sample size: 250 000, 1 000 000 and 5 000 000 from left to right. The optimal value of $\tilde{p}$ according to each of the measures is also presented. See Figures 5 and 6 for similar plots with different models.

Observations

Every combination of model and sample size produced essentially convex $\text{RE}_{\hat{\psi}}$ curves with a clear minimum after averaging and smoothing. This suggests that for any given model and sample size, there exists an optimal value of $\tilde{p}$. The relative error associated with underestimating the optimal value appears worse than that associated with overestimation, something that becomes more pronounced as n increases and the minimum "flattens out." It is also worth noting that in general, the variance of $\text{RE}_{\hat{\psi}}$ was greater when the optimal order was underestimated than overestimated (see Figure 7). As expected, the relative error associated with the optimal order tends to decreases as n (the available information) increases. These behaviours can be seen in Figure 2 and 7.

The $\text{RE}_{\hat{\omega}}$ curves emulated the behaviour of the $\text{RE}_{\hat{\psi}}$ curves quite well. They show similar convexity, decreasing relative error with increasing n and flattening out of the minimum with n, to that of $\text{RE}_{\hat{\psi}}$. However, the optimal orders from $\text{RE}_{\hat{\omega}}$ underestimated the value given by $\text{RE}_{\hat{\psi}}$ 67% of the time (Figure 10) suggesting the measure may have some bias.

Figure 3 shows the optimal $\tilde{p}$ values for each model and sample size[1]. The optimal value of $\tilde{p}$ was known to depend on the sample size, and this is quite clear from Figures 3 and 12 and Table 2 which show almost unanimously that it increases with sample size. Models 10 and 19 show some peculiar defiance of this trend, which may simply be due to the application of smoothing. As mentioned, more repetitions would enhance these estimates. It is also quite clear that as p and/or q increases, so too does the optimal order. The similarity between the $\texttt{ARMA}(60, 40)$ and $\texttt{ARMA}(100, 30)$ optimal orders may indicate that this effect is less pronounced for values of p and q of that scale, or it is a side-effect of the random parameters and small number of parameter sets.

The optimal value was also known to depend on the model parameters, but the extent to which this is the case is quite striking. Consider $\texttt{ARMA}(20, 15)$ in Figure 3 when $n = 5\,000\,000$ for instance, where the distance between optimal orders is over 300. This suggests that approximating the value of $\tilde{p}$ purely from the sample size and estimated orders p and q would not be viable, despite a linear model fitting the data reasonably well (Figure 11). In the future it would be worth investigating many more than 3 sets of parameters for each model to see how the optimal $\tilde{p}$ values distribute, and what property of the parameter set controls this. This could also address the suggestion from this initial plot that the variance of the optimal order also increases as p and q increase.

[1]For the sake of clarity, Figure 12 shows the range for each sample size separately.

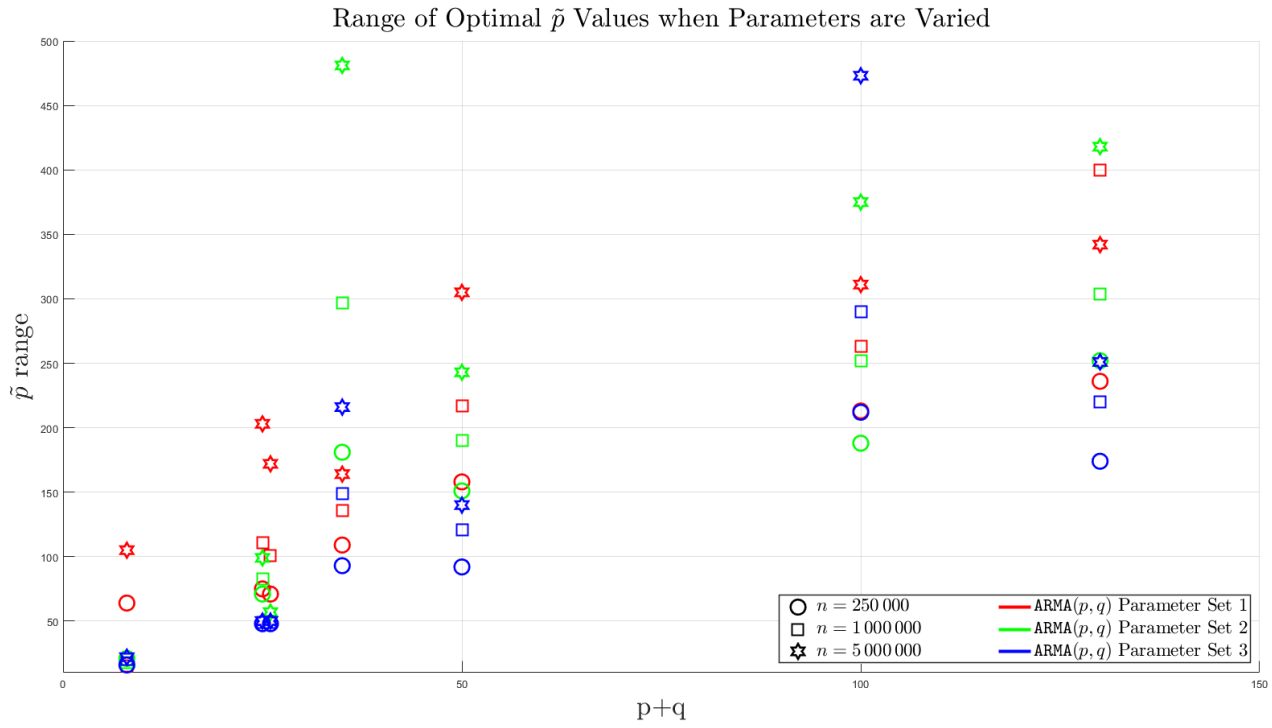

Figure 3: Each column of points represents a set of orders (p, q). Within a given set of orders, the three colours represent three different sets of parameters. There are also three point types for each sample size.

3.2 The Effectiveness of Trimming

It was hypothesised that there would be a trade-off between the choice of $\tilde{p}$ and the effectiveness of the trimming step. Under time constraints, this was not explored fully and will be considered in future work. Shown in Figure 4 is the $\text{RE}_{\hat{\psi}}$ from one single run of the HR algorithm over the given range of $\tilde{p}$ values, with multiple trimming steps. The initial estimate involved no trimming, so is similar to what has been seen throughout the previous section. In this scenario, trimming clearly had the effect of improving the relative error of the parameter estimates, however the magnitude of that effect depended on the value of $\tilde{p}$. Here the optimal order was 181, and for values of $\tilde{p}$ close to this order, the first trimming step improved the estimates by ~ 0.005. Subsequent trimming steps saw the relative error converge, as Hannan and Rissanen suggested it would. More interestingly, for small values of $\tilde{p}$ that produce poorer initial estimates, the first trimming step halved the relative error (reduction of ~ 0.15) and about three trimming steps were sufficient for the relative error to converge to the same point. This suggests that the choice of $\tilde{p}$ might not be so important if trimming behaves this way in

general.

In the context of the proposed `LSARMA` algorithm, it is not immediately apparent whether the estimates should be trimmed until convergence after choosing some small $\tilde{p}$, or if the optimal $\tilde{p}$ should be roughly approximated with subsequently little or no trimming. The computational demand for each approach will need to be considered.

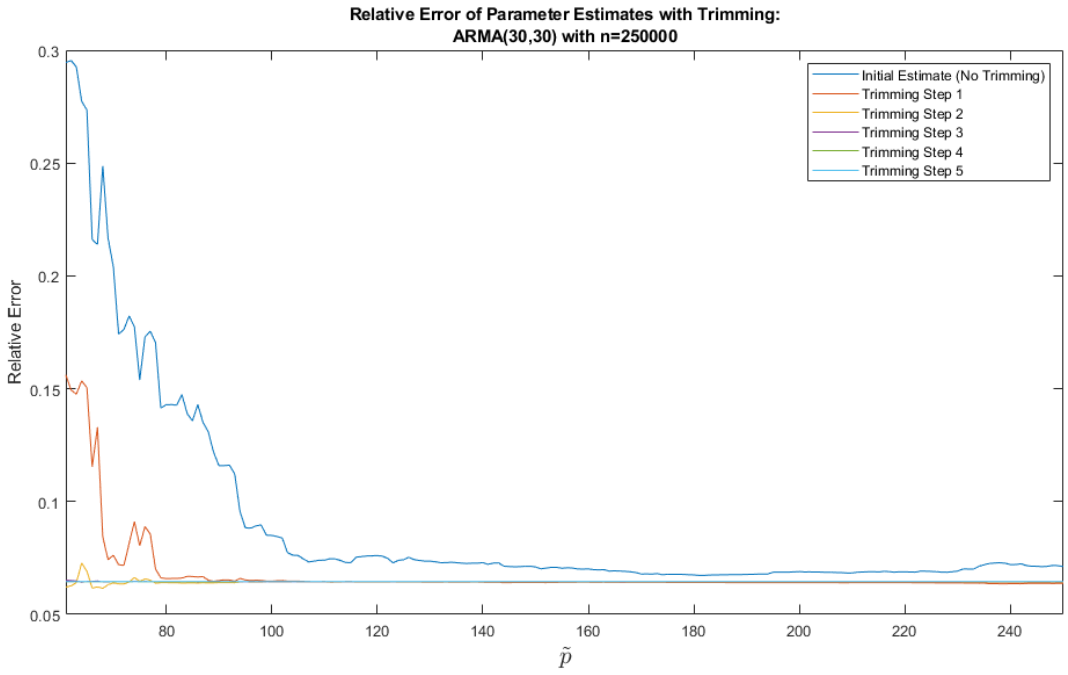

Figure 4: This plot shows $\mathrm{RE}_{\hat{\psi}}$ for each step of trimming over a range of $\tilde{p}$ values.

4 Conclusion

The aim of this project was to gain a deeper understanding of some properties of the `HR` algorithm in order to effectively employ it within the proposed `LSARMA` algorithm. This was accomplished by running the algorithm for various data sets simulated from known processes and analysing the output. The two features under consideration were the trimming step and the choice of order for the intermediate `AR` model. Although the optimal order can not be readily estimated if p, q and n are known, the relative error of the white noise estimates ($\text{RE}_{\hat{\omega}}$) can be used to provide a reasonable approximation. Subject to further investigation into the trimming step, an accurate value of $\tilde{p}$ may not be required.

For future work, applying the `HR` algorithm to models where $q > p$ will need to be considered for completeness and `LSARMA` needs to be extended to include such models. Trimming needs to be investigated further for various models and the computational demands compared to approximating an optimal $\tilde{p}$ with no or reduced trimming. Also of interest is the distribution of optimal $\tilde{p}$ values for given pairs of orders (p, q) where parameters are allowed to vary, and exploring what property of the parameters is responsible for that. Theoretical error bounds and rates of convergence have been derived for the `LSAR` algorithm, and the goal is to obtain similar for the `LSARMA`.

Acknowledgements

Special thanks to Glen Livingston for his enthusiastic support throughout this project, and AMSI for funding the research.

5 References

Box, G. E. P., & Jenkins, G. M. (1976). *Time series analysis: Forecasting and control*. Holden-Day.

Drineas, P., & Mahoney, M. W. (2017, December 24). Lectures on Randomized Numerical Linear Algebra.

Mahoney, M. W. (2011, November 15). Randomized algorithms for matrices and data.

Woodruff, D. P. (2014). Sketching as a Tool for Numerical Linear Algebra. *Foundations and Trends® in Theoretical Computer Science*, *10*(1-2), 1–157.

Drineas, P., Magdon-Ismail, M., Mahoney, M. W., & Woodruff, D. P. (2012, December 4). Fast approximation of matrix coherence and statistical leverage.

Eshragh, A., Roosta, F., Nazari, A., & Mahoney, M. W. (2019, November 27). LSAR: Efficient Leverage Score Sampling Algorithm for the Analysis of Big Time Series Data.

Shumway, R. H., & Stoffer, D. S. (2017). *Time Series Analysis and Its Applications: With R Examples* (4th ed.). Springer Texts in Statistics. Springer International Publishing.

Hannan, E. J., & Rissanen, J. (1982). Recursive Estimation of Mixed Autoregressive-Moving Average Order. *Biometrika*, *69*(1), 81–94. doi:10.2307/2335856

Brockwell, P. J., & Davis, R. A. (2006, April 10). *Introduction to Time Series and Forecasting*. Springer Science & Business Media.

Broersen, P.M.T. (2000, August). Autoregressive model orders for Durbin's MA and ARMA estimators. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, *48*(8). doi:http://dx.doi.org/10.1109/78.852025

6 Appendix

			Optimal $\tilde{p}$					
			250 000		1 000 000		5 000 000	
Model	p	q	$\text{RE}_{\hat{\psi}}$	$\text{RE}_{\hat{\omega}}$	$\text{RE}_{\hat{\psi}}$	$\text{RE}_{\hat{\omega}}$	$\text{RE}_{\hat{\psi}}$	$\text{RE}_{\hat{\omega}}$
1	30	20	158	161	217	209	305	283
2	30	20	151	140	190	197	243	278
3	30	20	92	110	121	132	140	160
4	14	12	71	107	101	162	172	276
5	14	12	48	60	49	70	57	83
6	14	12	48	49	49	56	50	66
7	13	12	75	102	111	159	203	268
8	13	12	71	73	83	87	99	105
9	13	12	48	51	49	59	50	68
10	4	4	66	27	28	29	29	30
11	4	4	27	25	28	28	28	29
12	4	4	25	24	29	27	29	29
13	60	40	213	212	263	264	311	329
14	60	40	188	193	252	246	375	381
15	60	40	212	276	290	402	473	500
16	20	15	109	115	136	143	164	175
17	20	15	181	122	297	192	481	320
18	20	15	93	132	149	191	216	275
19	100	30	236	215	400	265	342	319
20	100	30	252	231	304	274	418	354
21	100	30	174	187	220	217	251	271

Table 2: This reference table summarises the optimal values of $\tilde{p}$ found for the various combinations of parameters, orders and sample sizes according to both the relative error of the parameter estimates (shown in pink) and the relative error of the white noise estimates (shown in blue).

$n = 250\,000$	$n = 1\,000\,000$	$n = 5\,000\,000$

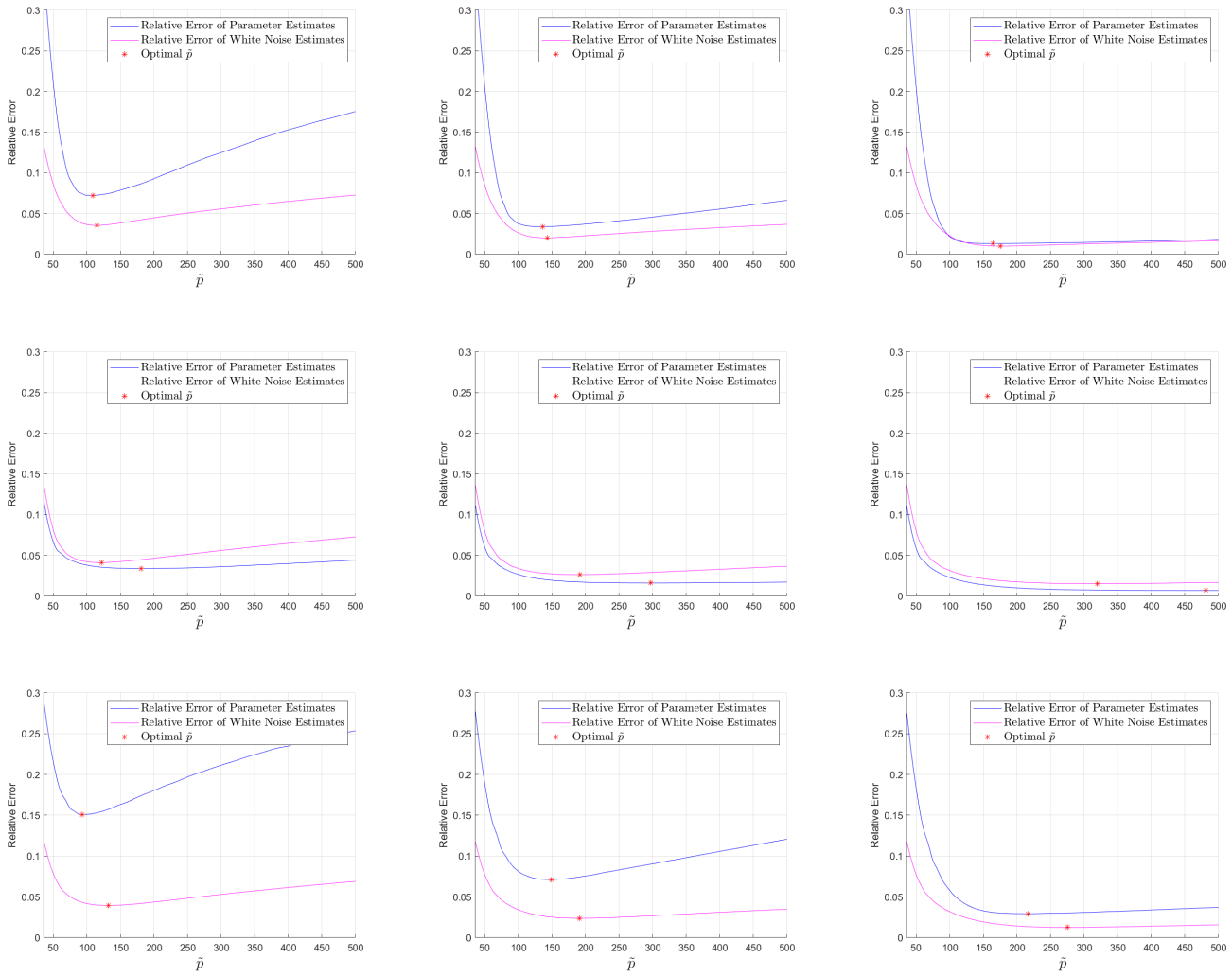

Figure 5: As for Figure 2 with `ARMA`$(20, 15)$ models instead (Models 16, 17 and 18 in Table 2).

$n = 250\,000$	$n = 1\,000\,000$	$n = 5\,000\,000$

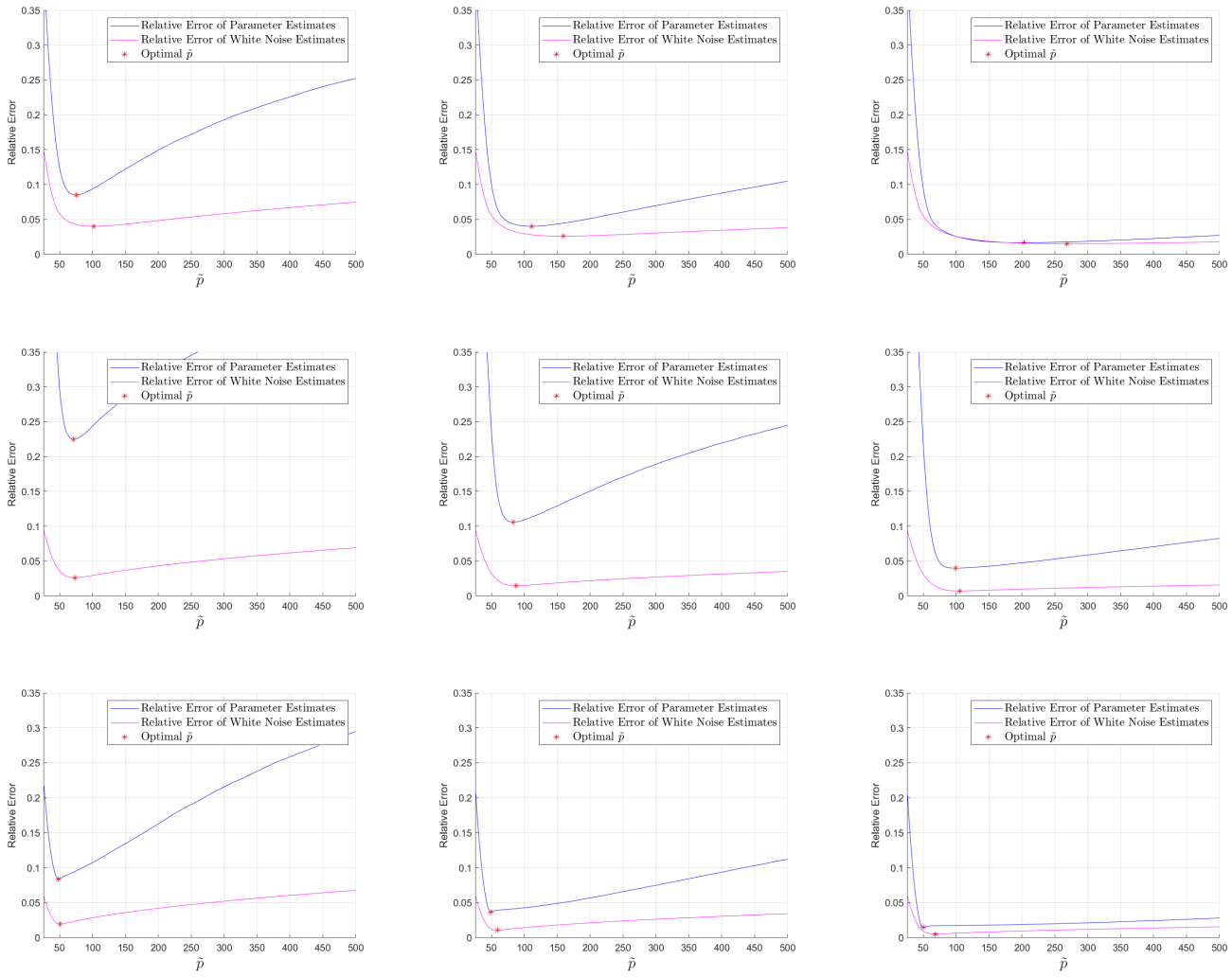

Figure 6: As for Figure 2 with `ARMA`$(13, 12)$ models instead(Models 7, 8 and 9 in Table 2).

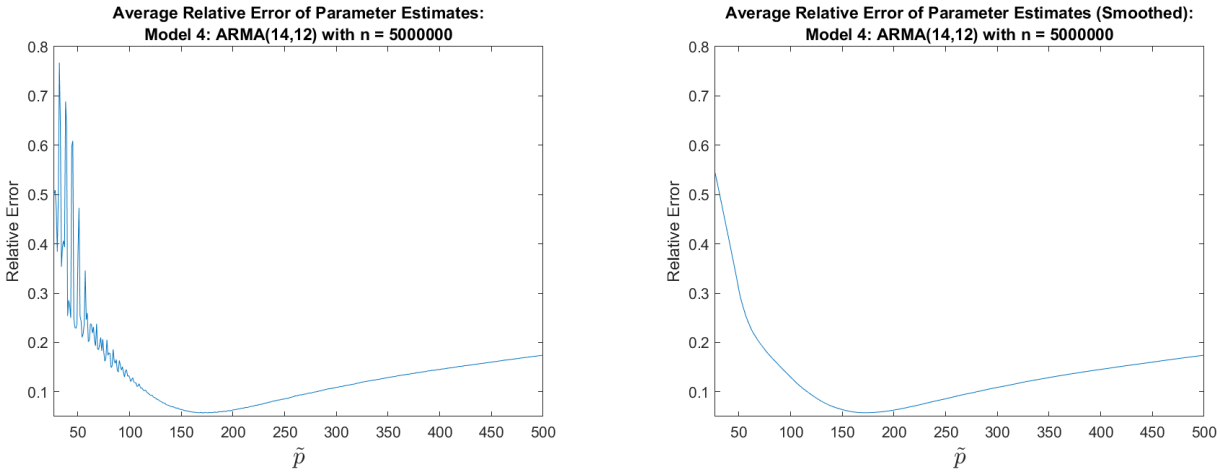

Figure 7: Example where no significant smoothing was required. Also note that even after 400 repetitions, there is significant variation in the relative error of estimates made with values of $\tilde{p}$ smaller than the optimal. That variation is not present for larger values.

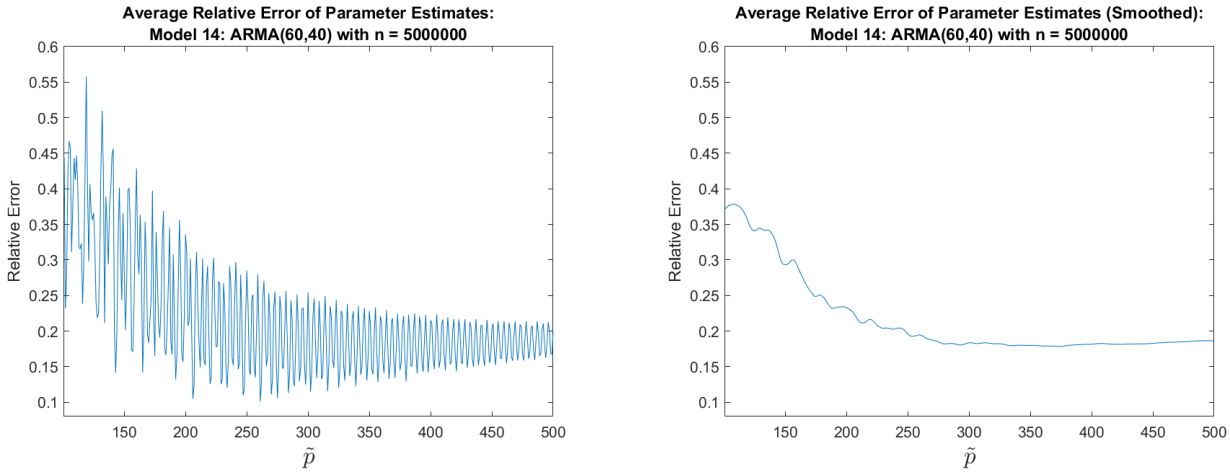

Figure 8: Substantial variation still present after averaging. In this case, smoothing is clearly required to approximate the minimum without interference from the fluctuations.

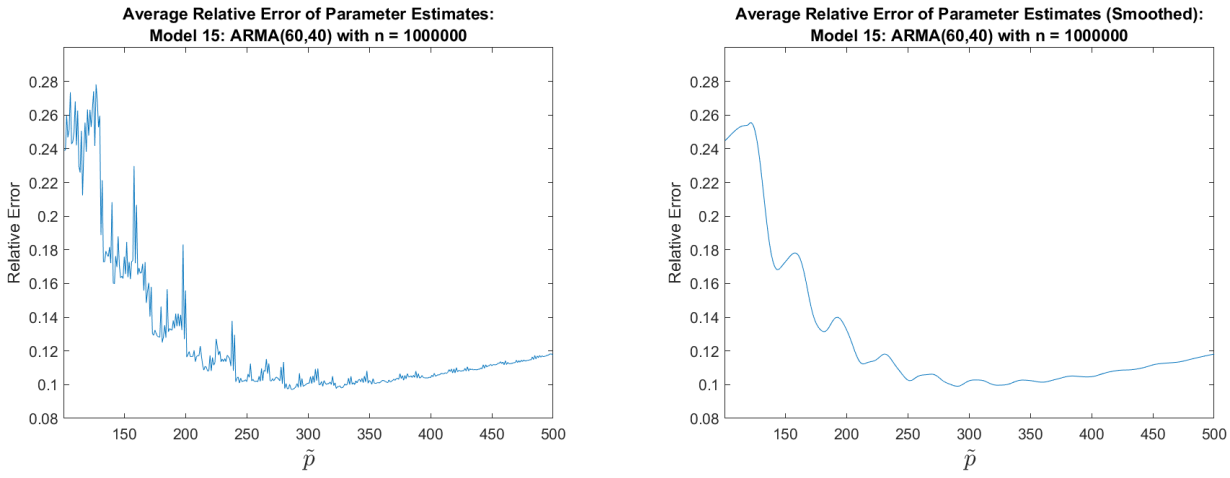

Figure 9: The variations remaining after averaging (on the left) appear somewhat periodic.

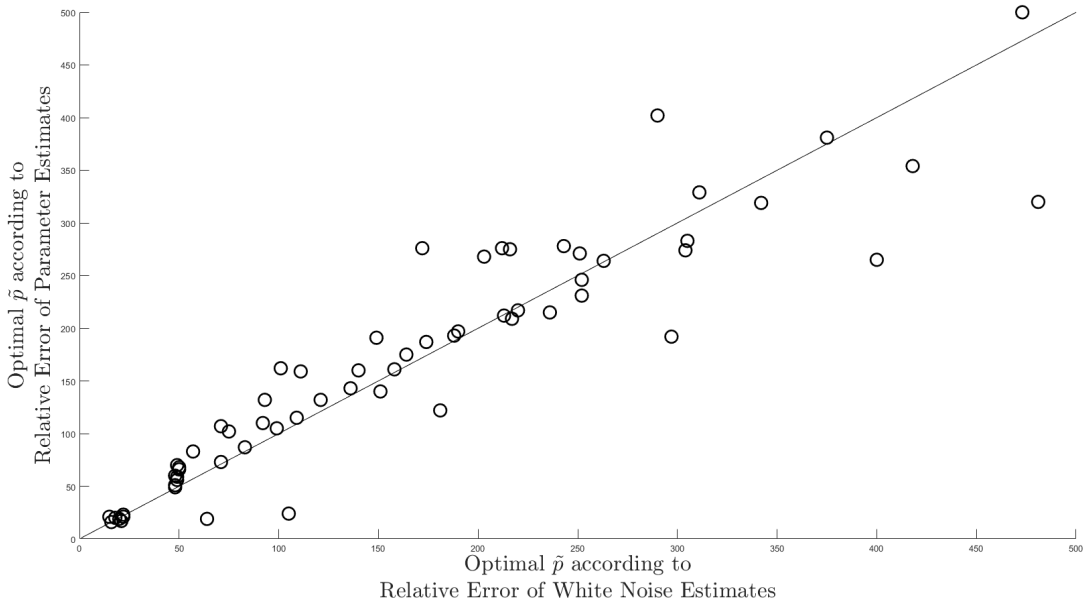

Figure 10: Comparison of measures chosen to determine the optimal $\tilde{p}$. The diagonal line represents perfect agreement between measures.

Linear Model Output

```
Call:
lm(formula = relative_par_error ~ n + p + q - 1, data = Data)

Residuals:
     Min       1Q   Median       3Q      Max
-105.92   -42.45    -8.36    26.13   303.60

Coefficients:
   Estimate Std. Error t value Pr(>|t|)
n 1.612e-05  3.835e-06   4.203 8.90e-05
p 1.059e+00  4.829e-01   2.193   0.0322
q 5.042e+00  1.095e+00   4.605 2.19e-05
---

Residual standard error: 70.86 on 60 degrees of freedom
Multiple R-squared:  0.8832,    Adjusted R-squared:  0.8774
F-statistic: 151.3 on 3 and 60 DF,  p-value: < 2.2e-16
```

Figure 11: Output from fitting a simple linear model with p, q and sample size as predictors for the optimal order according to $\text{RE}_{\hat{\psi}}$.

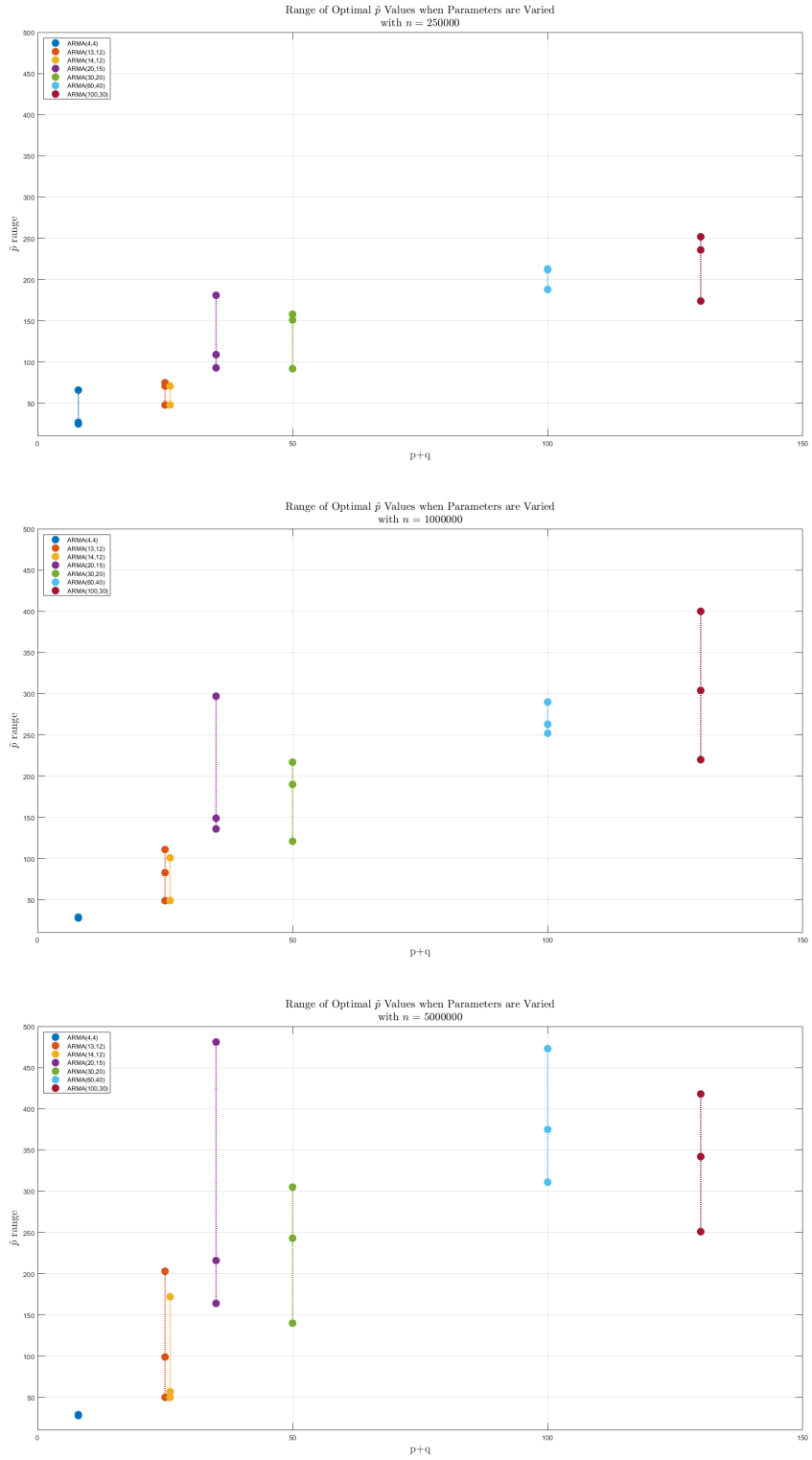

Figure 12: Optimal values of $\tilde{p}$ according to $\mathrm{RE}_{\hat{\psi}}$ for each given sample size and pair (p, q).