DISSERTATION

MODIFIED BURG ALGORITHMS FOR MULTIVARIATE SUBSET AUTOREGRESSION

Submitted by

Adão Alexandre Trindade

Department of Statistics

In partial fulfillment of the requirements

for the degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Fall 2000

COLORADO STATE UNIVERSITY

August 2, 2000

WE HEREBY RECOMMEND THAT THE DISSERTATION MODIFIED BURG ALGORITHMS FOR MULTIVARIATE SUBSET AUTOREGRESSION PREPARED UNDER OUR SUPERVISION BY ADÃO ALEXANDRE TRINDADE BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY.

Committee on Graduate Work

Committee Member

Committee Member

Committee Member

Adviser

Department Head

ABSTRACT

MODIFIED BURG ALGORITHMS FOR MULTIVARIATE SUBSET AUTOREGRESSION

We devise an algorithm that extends Burg's original method for recursive modeling of univariate autoregressions on a *full set* of lags, to multivariate modeling on a *subset* of lags. The key step in the algorithm involves minimizing the sum of the norm of the forward and backward *prediction error residual vectors*, as a function of the *reflection coefficient* matrices. We show that this sum has a global minimum, and give an explicit expression for the minimizer. By modifying the manner in which the reflection coefficients are calculated, this algorithm will also give the well-known Yule-Walker estimates. Based on recently proposed subset extensions to existing full set counterparts, two other algorithms that involve modifying the reflection coefficient calculation are also presented. Using simulated data, all four algorithms are compared with respect to the size of the Gaussian likelihood produced by each respective model. We find that the Burg and *Vieira-Morf* algorithms tend to perform better than the others for all configurations of roots of the autoregressive polynomial, averaging higher likelihoods with smaller variability across a large number of realizations.

We extend existing asymptotic central limit type results for three common vector autoregressive process estimators, to the subset case. First, consistency and asymptotic normality are established for the *least squares* estimator. This is extended to Yule-Walker, by virtue of the similarity in the closed forms for the two estimators.

Taking advantage of the fact that the Yule-Walker and Burg estimates can be calculated recursively via nearly identical algorithms, we then show these two differ by terms of order at most $O_p(1/n)$. In this way the Burg estimator inherits the same asymptotics as both Yule-Walker and least squares.

Saddlepoint approximations to the distributions of the Yule-Walker and Burg autoregressive coefficient estimators, when sampling from a subset Gaussian AR(p) with only one non-zero lag, are given. In this context, each estimator can be written as a ratio of quadratic forms in normal random variables. The larger bias and variance in the distribution of the Yule-Walker estimator, particularly evident at low sample sizes and when the AR coefficient is close to ± 1, agrees with its tendency to give lower likelihoods, as noted earlier. Empirical probability density histograms of the sampling distributions, for these two as well as the *maximum likelihood* estimator, provide further confirmation of the superiority of Burg over Yule-Walker in the vicinity of ± 1. Relative error comparisons between the saddlepoint approximations and the empirical cumulative distribution functions, show close agreement.

In conclusion, we elaborate on the logic employed in writing the computer programs that implement the Burg algorithm in the univariate and bivariate modeling settings. The central idea involves the formation of a tree of *nodes* connected by pointers. Due to the recursive nature of the algorithm, where modeling on larger subset sizes relies on that of smaller ones, each node harbors information on the modeling problem as applied to the particular subset of lags it represents. Starting at nodes of just one lag, the program follows pointers to those of larger numbers of lags, pausing at each to build the necessary modeling information. Termination occurs at the top of the

tree, when applying the algorithm to the unique node that represents the subset of lags upon which modeling was originally desired.

Adão Alexandre Trindade
Department of Statistics
Colorado State University
Fort Collins, Colorado 80523
Fall 2000

ACKNOWLEDGEMENTS

This work would not have been possible without the patient guidance of my advisors, Professors Peter J. Brockwell and Richard A. Davis, to whom I am deeply grateful. I'm especially thankful to Dr. Brockwell for the generous supply of problems to work on, thus ensuring I was never stuck on any particular one for very long. To Dr. Davis for his help with the asymptotic theory, and the opportunity to work on a research assistantship from which I learned so much about how a master probabilist works.

I would also like to acknowledge the financial support of my advisors through their NSF grant DMS-9972015, under which a substantial portion of this work was conducted.

I would like to thank Professor Ronald W. Butler for the many helpful comments and suggestions on the saddlepoint approximations.

I would like to thank my remaining committee members, Professors Azimi-Sadjadi, Mielke, and Givens, some of whom agreed to act as substitutes at very short notice.

Želeo bih da zahvalim Tomi i Zori, koji su mi pružili utočište za dušu i telo, dom daleko od doma.

A finalizar, gostaria de agradecer o meu irmão Cláudio pelas várias conversas filosóficas que tanto contribuem para que me mantenha "atento".

DEDICATION

Najdražoj Milici, čija su nesebična podrška i nesalomljiva vera bile nepresušan izvor inspiracije.

Aos meus pais, Vital e Guida, que embora o Atlântico nos separe, voçês estão sempre comigo.

CONTENTS

LIST OF FIGURES

Chapter 1

MULTIVARIATE SUBSET AUTOREGRESSIVE BURG ALGORITHMS

1.1 Introduction

A fundamental problem in time series analysis is the forecasting of future observations, based on some subset of past observations. If the covariance function of the process is known, one can obtain *best linear predictors* (having smallest mean squared error among the class of all linear predictors) by solving the well-known Yule-Walker equations. In the ensuing work, this will be termed the *prediction problem*. Typically however, one simply has a set of data and no knowledge of the covariance function of the underlying stochastic process that generated the observed realization. Here, modeling the observed data will be a necessary first step before implementing any type of model-based forecasting technique. We will call this the *modeling problem*.

By its very nature, the forecasting problem has a clear and concise solution. The modeling problem on the other hand, is plagued with deep theoretical and philosophical issues. These can range from how the covariance function should be estimated, to model selection and how best to estimate the parameters thereof. Further complications may arise when this agreed best estimation method is not amenable to practical implementation, such as when maximum likelihood estimation is used to model a multivariate process of high dimensionality.

Our goal in this chapter is to introduce and compare some parameter estimation methods/algorithms for a particular class of multivariate time series models, called *vector autoregressive (VAR)*. The d-dimensional time series $\{\mathbf{X}_t\}$ is said to follow the VAR process of order p, if it satisfies the following relation:

$$\mathbf{X}_t = \Phi(1)\mathbf{X}_{t-1} + \cdots + \Phi(p)\mathbf{X}_{t-p} + \mathbf{Z}_t,$$

where $\{\mathbf{Z}_t\}$ is a sequence of zero-mean uncorrelated random vectors, each with covariance matrix Σ. We call the process $\{\mathbf{Z}_t\}$ *white noise*, and write $\mathbf{Z}_t \sim \mathrm{WN}(\mathbf{0}, \Sigma)$.

In a VAR model of order p therefore, the current value of the series is a function of the previous p values, perturbed by a random amount. We say that we are modeling the series on the lagged set $\{1, \ldots, p\}$. One can generalize this concept to modeling on a lagged *subset* $K = \{k_1, \ldots, k_m\} \subseteq \{1, \ldots, p\}$, where the coefficient matrices pertaining to the lags not present in the set K, are constrained to be zero. Such models are called *subset vector autoregressive (SVAR)*.

Similarly, we can think of predicting the current value of a process $\{\mathbf{X}_t\}$ based on the previous p values, by the relation:

$$\hat{\mathbf{X}}_t = \Phi(1)\mathbf{X}_{t-1} + \cdots + \Phi(p)\mathbf{X}_{t-p}.$$

Again, one can generalize this idea to prediction based on a *subset* $K = \{k_1, \ldots, k_m\}$ of the past lags $\{1, \ldots, p\}$.

The algorithms we propose in this work, will be applicable to SVAR modeling and subset prediction, and will be recursive in nature. A major advantage of recursion, is that it involves inversion of matrices whose dimension does not exceed d^2.

1.1.1 Some early work

One of the most common SVAR modeling procedures, is the Yule-Walker or Durbin-Levinson-Whittle algorithm (introduced in section 1.3). A derivation of this algo-

rithm for univariate series, was first given by Penm and Terrell (1982). A variety of methods with superior properties have been proposed for non-subset (full set) modeling. Burg's algorithm for AR modeling (Burg (1978)), which minimizes the sums of squares of the forward and backward prediction errors, has been found to generally give models with better spectral resolution than Yule-Walker. Morf *et al.* (1978), Strand (1977), and others (see Jones (1978)), generalize Burg's algorithm to full set multivariate modeling. More recently, Brockwell and Dahlhaus (1998) proposed multivariate subset versions of the Yule-Walker algorithm, Burg's algorithm, as well as those of Morf *et al.*, and Strand. Our objective in this chapter will be to extend their work on some of these algorithms, and investigate the relative performances of these various SVAR modeling methods.

1.1.2 Applications of subset prediction/modeling

Forecasting with missing observations: This is a very natural situation in which this subset methodology is appropriate if the covariance function of the process is known.

Modeling of causal seasonal models of the form:

$$(1 - \psi B^s)\,(1 - \phi_1 B - \cdots - \phi_p B^p)\,X_t = Z_t.$$

These models will, for $s > p + 1$, be subset AR models of the type

$$1 - \phi_1 B - \cdots - \phi_p B^p - \psi B^s + \psi\phi_1 B^{s+1} + \cdots + \psi\phi_1 B^{s+p},$$

and thus $K = \{1, \ldots, p, s, s + 1, \ldots, s + p\}$.

Obtaining initial estimates for constrained MLE of subset VAR models: Obtaining good initial estimates for maximum likelihood estimation where some of the autoregressive coefficients are constrained to be zero, is highly

desirable due to the nonlinear nature of the likelihood equations. The subset Durbin-Levinson-Whittle algorithm, is very fast compared with constrained maximum likelihood estimation, and can therefore be used to provide such estimates. As an alternative to maximum likelihood estimation, the primary motivation of this work is then in finding fast, recursive VAR modeling algorithms that produce models with high likelihoods.

h-step prediction: If the covariance function of the process is known, and prediction h steps ahead based on the previous m observations is required, we can take $K = \{h, h+1, \ldots, h+m-1\}$. This generalizes immediately to h-step prediction on any subset of past observations.

1.1.3 Selection of "best" subset model

There is a growing body of literature on the subject of selection of the "best" subset model from all possible subset AR models. Sarkar and Sharma (1997), and Sarkar and Kanjilal (1995) propose a method for selecting a reduced subset from the full set, using singular value decomposition and QR orthonormal decomposition with column pivoting factorization of a matrix. Terrell and Zhang (1997) introduce so called *projection modulus* statistics which respond to the exclusion of important lags by producing high residual variances in an appropriate Hilbert space. Yu and Lin (1991) improve upon a method of Hokstad by employing the inverse autocorrelation function to select the first tentative model. Some early methods were proposed in Duong (1984) and McClave (1978).

A large part of these research efforts have concentrated on reducing the size of the candidate model set, in order to implement a feasible search in real time. From this reduced pool of a few "good" candidate models, one could then use the subset

modeling algorithms we will present to obtain good initial estimates for constrained maximum likelihood estimation.

Although not usually done, the recursive nature of these algorithms would also permit a direct search on all 2^{k_m} candidate model sets to be carried out. One could start by searching all k_m subsets of size 1, and selecting the best based on some criterion such as AICC. Then all $\begin{pmatrix} k_m \\ 2 \end{pmatrix}$ subsets of size 2 could be searched, and the best model thus far obtained updated; etc.

1.2 The prediction problem

Suppose $\{\mathbf{X}_t,\ t = 0, \pm 1, \pm 2, \cdots\}$ is a zero-mean weakly stationary d-variate time series with $\mathbf{X}_t = (X_{t,1}, \cdots, X_{t,d})'$, and covariance function

$$< \mathbf{X}_{t+h}, \mathbf{X}_t >:= E[\mathbf{X}_{t+h}\mathbf{X}_t'] = \Gamma(h), \quad h = 0, \pm 1, \pm 2, \ldots . \tag{1.1}$$

The best (forward in time) linear predictor of $\mathbf{X}_t$ on the subset of lags $K = \{k_1, \cdots, k_m\}$, is defined as

$$\hat{\mathbf{X}}_t(K) = \sum_{j \in K} \Phi_K(j)\mathbf{X}_{t-j} \tag{1.2}$$

Best here means that $\mathbf{E}||\mathbf{X}_t - \hat{\mathbf{X}}_t(K)||^2$ is minimized, where $||\cdot||$ is the Euclidean norm. Using simple orthogonality relations, we see that the coefficient matrices $\Phi_K(j)$ are found from the Yule-Walker equations

$$\sum_{j \in K} \Phi_K(j)\Gamma(k-j) = \Gamma(k),\ k \in K \tag{1.3}$$

(see for example Brockwell and Davis (1991), p. 421-422, with $P(\mathbf{X}_{n+1}|\mathbf{X}_1, \cdots, \mathbf{X}_n)$ replaced by $P(\mathbf{X}_t|\mathbf{X}_{t-k},\ k \in K)$), with mean squared error covariance matrix

$$\begin{aligned} U_K &= E[(\mathbf{X}_t - \hat{\mathbf{X}}_t(K))(\mathbf{X}_t - \hat{\mathbf{X}}_t(K))'] \\ &= \Gamma(0) - \sum_{j \in K} \Phi_K(j)\Gamma(j)'. \end{aligned} \tag{1.4}$$

Analogously, we define the best backward in time linear predictor of $\mathbf{X}_t$ on the subset of lags K as

$$\hat{\mathbf{X}}_t^{(b)}(K) = \sum_{j \in K} \Psi_K(j) \mathbf{X}_{t+j}$$

with resulting Yule-Walker equations and mean squared error covariance matrix (obtained by replacing $P(\mathbf{X}_0|\mathbf{X}_1, \cdots, \mathbf{X}_n)$ with $P(\mathbf{X}_t|\mathbf{X}_{t+k},\ k \in K)$ in the above reference) given by

$$\sum_{j \in K} \Psi_K(j) \Gamma(k-j)' = \Gamma(k)',\ k \in K, \tag{1.5}$$

and

$$\begin{aligned} V_K &= E[(\mathbf{X}_t - \hat{\mathbf{X}}_t^{(b)}(K))(\mathbf{X}_t - \hat{\mathbf{X}}_t^{(b)}(K))'] \\ &= \Gamma(0) - \sum_{j \in K} \Psi_K(j) \Gamma(j). \end{aligned} \tag{1.6}$$

Remark 1.2.1 *In the univariate case there is no distinction between the forward and backward prediction problem (based on the same subset K). This is because when $d = 1$, the equations (1.3) for $\{\Phi_K(j), j \in K\}$ are the same as (1.5) for $\{\Psi_K(j), j \in K\}$, and (1.4) for U_K the same as (1.6) for V_K.*

Definition 1.2.1 *A multivariate stationary process $\{\mathbf{X}_t\}$ is said to be of* full rank *if the covariance matrix of any finite collection of random vectors is non-singular.*

Remark 1.2.2 *As will be shown in Chapter 2, the coefficient matrices $\Phi_K(j)$ and $\Psi_K(j),\ j \in K$, of the forward and backward prediction problems, will be unique if $\{\mathbf{X}_t\}$ is of full rank.*

1.3 The modeling problem

Given n observations $\mathbf{x}_1, \cdots, \mathbf{x}_n$ of the zero-mean random vectors $\mathbf{X}_1, \cdots, \mathbf{X}_n$, we define the empirical analogue of the covariance matrix at lag h to be

$$\hat{\Gamma}(h) \;=\; \begin{cases} \frac{1}{n}\sum_{t=1}^{n-h} \mathbf{x}_{t+h}\mathbf{x}_t' & \text{, if } h \geq 0 \\ \Gamma(-h)' & \text{, if } h < 0 \end{cases} \tag{1.7}$$

In order to define an inner-product in the space of empirical observations, Brockwell and Dahlhaus (1998) proceed by defining $\mathbf{x}_t = 0$ for $t \leq 0$ and $t > n$. With this, they define the $(d \times \infty)$ array $\mathbf{y} = \{\mathbf{x}_j,\ j = 0, \pm 1, \cdots\}$, and define $\mathbf{y}_t$ to be the array obtained by shifting the columns of $\mathbf{y}$ t places to the left, i.e. $\mathbf{y}_t = \{\mathbf{x}_{t+j},\ j = 0, \pm 1, \cdots\}$. We view $\mathbf{y}_t$ as a d-dimensional column vector, whose elements are infinite-dimensional row vectors, with finitely many non-zero entries. The set of all such row vectors constitutes an inner-product space, if we define the inner-product of any two row vectors $\mathbf{u} = \{\mathbf{u}_j\}$ and $\mathbf{v} = \{\mathbf{v}_j\}$ as

$$< \mathbf{u}, \mathbf{v} >= \frac{1}{n}\sum_{j=-\infty}^{\infty} \mathbf{u}_j\mathbf{v}_j. \tag{1.8}$$

With this set-up, we have for example

$$< \mathbf{y}_{t+h}, \mathbf{y}_t >= \hat{\Gamma}(h),$$

since $< \mathbf{y}_{t+h}, \mathbf{y}_t >$ is the matrix of inner-products whose (i, j)-element is the inner product of the i^{th} row of $\mathbf{y}_{t+h}$, with the j^{th} row of $\mathbf{y}_t$.

Solving the Yule-Walker equations (1.3) and (1.4), with $\Gamma(\cdot)$ replaced by $\hat{\Gamma}(\cdot)$, we obtain the subset (based on the subset of lags $K = \{k_1, \ldots, k_m\}$) vector autoregressive, SVAR(K), model for the data:

$$\mathbf{X}_t = \sum_{i \in K} \hat{\Phi}_K(i)\mathbf{X}_{t-i} + \mathbf{Z}_t, \tag{1.9}$$

where $\{\mathbf{Z}_t\}$ is a sequence of zero-mean uncorrelated random vectors, each with covariance matrix $\hat{U}_K$, i.e. $\mathbf{Z}_t \sim \mathrm{WN}(\mathbf{0}, \hat{U}_K)$, given by

$$\hat{U}_K = \hat{\Gamma}(0) - \sum_{i \in K} \hat{\Phi}_K(i)\hat{\Gamma}(i)'. \tag{1.10}$$

The corresponding *backward* subset vector autoregressive model for the data, based on the subset $K^* = \{k_m - k_{m-1}, \ldots, k_m - k_1, k_m\}$ of lags is

$$\mathbf{X}_t = \sum_{j \in K^*} \hat{\Psi}_{K^*}(j)\mathbf{X}_{t+j} + \mathbf{Z}_t, \tag{1.11}$$

where $\mathbf{Z}_t \sim \mathrm{WN}(\mathbf{0}, \hat{V}_{K^*})$, given by

$$\hat{V}_{K^*} = \hat{\Gamma}(0) - \sum_{j \in K^*} \hat{\Psi}_{K^*}(j)\hat{\Gamma}(j). \tag{1.12}$$

Remark 1.3.1 *Finding $\hat{\Phi}_K$ involves solving an equation of the form,*

$$\Phi_K \hat{G}_K = \hat{\Gamma}_K, \tag{1.13}$$

where $\hat{G}_K$ and $\hat{\Gamma}_K$ are matrices of empirical autocovariances arranged in particular ways (see section 2.2 for details). There are many solutions to this equation if $\hat{G}_K$ is singular, but every solution gives the same linear predictor when substituted into (1.2). Non-causal solutions, $\{\hat{\Phi}_K(i),\ i \in K\}$, may be obtained, suggesting when they occur, that the data is not well fitted by a subset vector autoregression with lags in K. Whether the solution is causal or not, the expression (1.2) with each $\Phi_K(i)$ replaced by $\hat{\Phi}_K(i)$, will give the best linear predictor under the assumption that the sample autocovariances are equal to the true autocovariances. Similarly for the backward modeling/prediction problem.

Definition 1.3.1 (Causality) *The SVAR(K) model,*

$$\mathbf{X}_t = \sum_{i \in K} \Phi_K(i)\mathbf{X}_{t-i} + \mathbf{Z}_t, \quad \{\mathbf{Z}_t\} \sim \mathit{WN}(\mathbf{0}, \Sigma), \tag{1.14}$$

is said to be causal *(or more specifically to be a causal function of* $\{\mathbf{Z}_t\}$*), if there exists a sequence of matrices* $\{\Upsilon_0, \Upsilon_1, \dots\}$ *such that,*

$$\mathbf{X}_t = \sum_{j=0}^{\infty} \Upsilon_j \mathbf{Z}_{t-j}.$$

Definition 1.3.2 (VAR characteristic polynomial) *The* VAR characteristic polynomial*,* $|\Phi(z)|$*, for model (1.14), is defined to be the polynomial in* z *of degree* dk_m *given by,*

$$|\Phi(z)| = |I_d - \Phi_K(k_1)z - \cdots - \Phi_K(k_m)z^{k_m}|.$$

From Brockwell and Davis (1991), theorem 11.3.1, a SVAR(K) model is causal if all roots of its VAR characteristic polynomial are greater than one in magnitude. Causality is a property often desired in a model; without it the parameters are non-identifiable for Gaussian likelihood and other inherently second order estimation methods.

The *subset Durbin-Levinson-Whittle algorithm*, as presented in Brockwell and Dahlhaus (1998), can be employed to provide a recursive solution to the empirical Yule-Walker equations:

Algorithm 1.3.1 (The subset Durbin-Levinson-Whittle algorithm)

$$\begin{aligned}
\hat{\Phi}_K(k_m) &= \left(\hat{\Gamma}(k_m) - \sum_{i\in J} \hat{\Phi}_J(i)\hat{\Gamma}(k_m - i)\right)\hat{V}_{J^*}^{-1} \\
\hat{\Phi}_K(i) &= \hat{\Phi}_J(i) - \hat{\Phi}_K(k_m)\hat{\Psi}_{J^*}(k_m - i),\ i \in J \\
\hat{\Psi}_{K^*}(k_m) &= \left(\hat{\Gamma}(k_m)' - \sum_{j\in J^*} \hat{\Psi}_{J^*}(j)\hat{\Gamma}(k_m - j)'\right)\hat{U}_J^{-1} \\
\hat{\Psi}_{K^*}(j) &= \hat{\Psi}_{J^*}(j) - \hat{\Psi}_{K^*}(k_m)\hat{\Phi}_J(k_m - j),\ j \in J^* \\
\hat{U}_K &= \hat{U}_J - \hat{\Phi}_K(k_m)\hat{V}_{J^*}\hat{\Phi}_K(k_m)' \\
\hat{V}_{K^*} &= \hat{V}_{J^*} - \hat{\Psi}_{K^*}(k_m)\hat{U}_J\hat{\Psi}_{K^*}(k_m)'
\end{aligned} \tag{1.15}$$

Where $\hat{U}_J^{-1}$ *and* $\hat{V}_{J^*}^{-1}$ *denote generalized inverses of* $\hat{U}_J$ *and* $\hat{V}_{J^*}$ *respectively. The forward subsets of lags are* $K = \{k_1, \cdots, k_m\}$, $J = \{k_1, \cdots, k_{m-1}\}$, *and the backward subsets of lags are* $K^* = \{k_m - k_{m-1}, \cdots, k_m - k_1, k_m\}$, $J^* = \{k_m - k_{m-1}, \cdots, k_m - k_1\}$. *The algorithm begins at* $m = 1$ *(any appropriate subset of size one), with* $\hat{U}_\emptyset = \hat{\Gamma}(0) = \hat{V}_\emptyset$.

1.4 Burg-type Algorithms

In Brockwell and Dahlhaus (1998), a different kind of recursion for the best predictors based on *forward* and *backward* residuals was presented. The forward prediction residual is defined as

$$\boldsymbol{\varepsilon}_K(t) = \mathbf{X}_t - \hat{\mathbf{X}}_t(K),$$

and the backward prediction residual as

$$\boldsymbol{\eta}_K(t) = \mathbf{X}_t - \hat{\mathbf{X}}_t^{(b)}(K).$$

The covariance matrices of these residuals are respectively U_K and V_K defined in 1.4 and 1.6. The empirical analogues of these prediction errors are

$$\hat{\boldsymbol{\varepsilon}}_K(t) = \mathbf{y}_t - \hat{\mathbf{y}}_t(K),$$

and

$$\hat{\boldsymbol{\eta}}_K(t) = \mathbf{y}_t - \hat{\mathbf{y}}_t^{(b)}(K),$$

where

$$\hat{\mathbf{y}}_t(K) = \sum_{j \in K} \Phi_K(j)\mathbf{y}_{t-j},$$

and

$$\hat{\mathbf{y}}_t^{(b)}(K) = \sum_{j \in K} \Psi_K(j) \mathbf{y}_{t+j}.$$

With these definitions and inner-products defined as in (1.8), Brockwell and Dahlhaus (1998) present the following prediction error solution to the empirical Yule-Walker (YW) equations (1.9) - (1.12):

Algorithm 1.4.1 (Prediction error solution of the YW equations)

$$\begin{aligned}
\hat{\Phi}_K(k_m) &= \langle \hat{\boldsymbol{\varepsilon}}_J(t), \hat{\boldsymbol{\eta}}_{J^*}(t-k_m) \rangle \hat{V}_{J^*}^{-1} && (1.16)\\
\hat{\Phi}_K(i) &= \hat{\Phi}_J(i) - \hat{\Phi}_K(k_m)\hat{\Psi}_{J^*}(k_m - i),\ i \in J \\
\hat{\Psi}_{K^*}(k_m) &= \hat{V}_{J^*}\hat{\Phi}_K(k_m)'\hat{U}_J^{-1} && (1.17)\\
\hat{\Psi}_{K^*}(j) &= \hat{\Psi}_{J^*}(j) - \hat{\Psi}_{K^*}(k_m)\hat{\Phi}_J(k_m - j),\ j \in J^* \\
\hat{U}_K &= \hat{U}_J - \hat{\Phi}_K(k_m)\hat{V}_{J^*}\hat{\Phi}_K(k_m)' \\
\hat{V}_{K^*} &= \hat{V}_{J^*} - \hat{\Psi}_{K^*}(k_m)\hat{U}_J\hat{\Psi}_{K^*}(k_m)' \\
\hat{\boldsymbol{\varepsilon}}_K(t) &= \hat{\boldsymbol{\varepsilon}}_J(t) - \hat{\Phi}_K(k_m)\hat{\boldsymbol{\eta}}_{J^*}(t-k_m) && (1.18)\\
\hat{\boldsymbol{\eta}}_{K^*}(t) &= \hat{\boldsymbol{\eta}}_{J^*}(t) - \hat{\Psi}_{K^*}(k_m)\hat{\boldsymbol{\varepsilon}}_J(t+k_m) && (1.19)
\end{aligned}$$

with initial conditions,

$$\begin{aligned}
\hat{\boldsymbol{\varepsilon}}_\emptyset(t) &= \hat{\boldsymbol{\eta}}_\emptyset(t) = \mathbf{y}_t,\ t = 0, \pm 1, \pm 2, \cdots \\
\hat{U}_\emptyset &= \hat{\Gamma}(0) = \hat{V}_\emptyset.
\end{aligned}$$

Remark 1.4.1 *These recursions are equivalent to the Durbin-Levinson-Whittle algorithm. Thus each line of algorithm 1.3.1 is interchangeable with the corresponding line of algorithm 1.4.1. In particular, equations (1.15) and (1.17) are interchangeable, providing a simple relationship between* $\hat{\Phi}_K(k_m)$ *and* $\hat{\Psi}_{K^*}(k_m)$ *which we will use frequently.*

This type of algorithm is patterned after that introduced by Burg (1978) for univariate modeling on a *full set* of lags rather than a *subset*. Burg's contribution was to show that the maximum entropy spectral density estimator of a univariate stationary process over all densities g satisfying the constraints

$$\int_{-\pi}^{\pi} e^{ih\lambda} g(\lambda) d\lambda = \hat{\gamma}(h), \quad h = 0, \pm 1, \dots, \pm p,$$

is exactly that of an AR(p) process. However, replacing $\gamma(\cdot)$ by $\hat{\gamma}(\cdot)$, where

$$\hat{\gamma}(h) = \frac{1}{n} \sum_{t=1}^{n-h} \mathbf{x}_t \mathbf{x}_{t+h}, \quad \text{and noting that} \quad \gamma(h) = \lim_{n\to\infty} \frac{1}{2n+1} \sum_{t=-n}^{n} \mathbf{x}_t \mathbf{x}_{t+h},$$

is roughly equivalent to assuming that the unobserved data $\{\dots, \mathbf{x}_{-1}, \mathbf{x}_0, \mathbf{x}_{n+1}, \dots\}$ is zero since, under ergodicity conditions, $\gamma(h)$ is the mean square limit as $n \to \infty$ of $\sum_{t=-n}^{n} \mathbf{x}_t \mathbf{x}_{t+h}/(2n+1)$. The sudden transition from (usually) non-zero value to zero at the edges of the observation domain, typically results in poor estimates of the coefficients $\Phi_K(k_m)$ and $\Psi_{K^*}(k_m)$. To alleviate this problem in the context of univariate AR(p) models, Burg suggested making no assumptions about unobserved data; $\hat{\phi}_p$ should minimize the sum of squares of the forward and backward residuals over the range of data values where these are defined. That is, choose $\hat{\phi}_p$ to minimize

$$\frac{1}{2(n-p)} \sum_{p+1}^{n} \left(\hat{\boldsymbol{\varepsilon}}_K(t)^2 + \hat{\boldsymbol{\eta}}_K(t-p)^2 \right).$$

This is commonly referred to as Burg's algorithm, and it was generalized to multivariate models by Morf *et al.* (1978), Strand (1977), and others (see Jones (1978)).

In line with these ideas, Brockwell and Dahlhaus (1998) propose the following multivariate subset versions of these algorithms:

Algorithm 1.4.2 (A subset Vieira-Morf algorithm)

Use algorithm 1.4.1 with (1.16) replaced by

$$\hat{\Phi}_K(k_m) = \hat{U}_J^{1/2} R \hat{V}_{J^*}^{-1/2},$$

where

$$R = \left(\sum_{t=k_m+1}^{n} \hat{\boldsymbol{\varepsilon}}_J(t) \hat{\boldsymbol{\varepsilon}}_J(t)' \right)^{-1/2} \left(\sum_{t=k_m+1}^{n} \hat{\boldsymbol{\varepsilon}}_J(t) \hat{\boldsymbol{\eta}}_{J^*}(t-k_m)' \right) \left(\sum_{t=k_m+1}^{n} \hat{\boldsymbol{\eta}}_{J^*}(t-k_m) \hat{\boldsymbol{\eta}}_{J^*}(t-k_m)' \right)^{-1/2}$$

Algorithm 1.4.3 (A subset Nuttall-Strand algorithm)

Use algorithm 1.4.1 with (1.16) replaced by

$$\hat{\Phi}_K(k_m) = \hat{U}_J^{1/2} R \, \hat{V}_J^{-1/2},$$

where

$$\begin{aligned} & vec(R) \\ = \quad & \left[I_d \otimes \left(\sum_{t=k_m+1}^{n} \hat{\boldsymbol{\varepsilon}}_J(t) \hat{\boldsymbol{\varepsilon}}_J(t)' \right) + \left(\sum_{t=k_m+1}^{n} \hat{\boldsymbol{\eta}}_{J^*}(t-k_m) \hat{\boldsymbol{\eta}}_{J^*}(t-k_m)' \right) \otimes I_d \right]^{-1} \\ & vec \left(2 \sum_{t=k_m+1}^{n} \hat{\boldsymbol{\varepsilon}}_J(t) \hat{\boldsymbol{\eta}}_{J^*}(t-k_m)' \right) \\ \Longrightarrow \quad & vec(\hat{\Phi}_K(k_m)) = \left(\hat{V}_J^{-1/2} \otimes \hat{U}_J^{1/2} \right) vec(R). \end{aligned}$$

Obtain $\hat{\Psi}_{K^*}(k_m)$ not according to (1.17), but according to

$$\hat{\Psi}_{K^*}(k_m) = \hat{V}_J^{1/2} R' \, \hat{U}_J^{-1/2}.$$

Algorithm 1.4.4 (A subset analogue of Burg's algorithm)

Use algorithm 1.4.1 with (1.16) replaced by the condition that $\hat{\Phi}_K(k_m)$ and $\hat{\Psi}_{K^*}(k_m)$ minimize the scalar quantity

$$S_K = \sum_{t=k_m+1}^{n} \left[\hat{\boldsymbol{\varepsilon}}_K(t)' \hat{\boldsymbol{\varepsilon}}_K(t) + \hat{\boldsymbol{\eta}}_{K^*}(t-k_m)' \hat{\boldsymbol{\eta}}_{K^*}(t-k_m) \right], \tag{1.20}$$

with $\hat{\boldsymbol{\varepsilon}}_K(t)$ and $\hat{\boldsymbol{\eta}}_{K^*}(t-k_m)$ as in (1.18) and (1.19), and $\hat{\Psi}_{K^*}(k_m)$ constrained to satisfy (1.17).

Notation

In order to differentiate between the Yule-Walker and this Burg solution of the modeling problem, denote the former estimates by topping them with *hats* (ˆ), and the latter with *tildes* (˜).

Remark 1.4.2 *By definition,*

$$\hat{\boldsymbol{\varepsilon}}_J = \mathbf{x}_t - \hat{\Phi}_J(k_1)\mathbf{x}_{t-k_1} - \cdots - \hat{\Phi}_J(k_{m-1})\mathbf{x}_{t-k_{m-1}},$$

and therefore $\hat{\boldsymbol{\varepsilon}}_J \neq 0$ *for* $t \in \mathcal{A}_1$*, where* $\mathcal{A}_1 = \{1, \ldots, n + k_{m-1}\}$*. In addition, all components in* $\hat{\boldsymbol{\varepsilon}}_J$ *are non-zero for* $t \in \mathcal{A}_2$*, where* $\mathcal{A}_2 = \{1 + k_{m-1}, \ldots, n\}$*.*

Also by definition,

$$\hat{\boldsymbol{\eta}}_{J^*} = \mathbf{x}_{t-k_m} - \hat{\Psi}_{J^*}(k_m - k_{m-1})\mathbf{x}_{t-k_{m-1}} - \cdots - \hat{\Psi}_{J^*}(k_m - k_1)\mathbf{x}_{t-k_1},$$

and therefore $\hat{\boldsymbol{\eta}}_{J^*} \neq 0$ *for* $t \in \mathcal{B}_1$*, where* $\mathcal{B}_1 = \{k_1 + 1, \ldots, n + k_m\}$*. In addition, all components in* $\hat{\boldsymbol{\eta}}_{J^*}$ *are non-zero for* $t \in \mathcal{B}_2$*, where* $\mathcal{B}_2 = \{1 + k_m, \ldots, n + k_1\}$*.*

In view of this, (1.16) becomes:

$$\hat{\Phi}_K(k_m) = \left(\frac{1}{n}\sum_{t\in\mathcal{A}_1\cup\mathcal{B}_1} \hat{\boldsymbol{\varepsilon}}_J(t)\hat{\boldsymbol{\eta}}_{J^*}(t-k_m)'\right)\hat{V}_{J^*}^{-1} = \left(\frac{1}{n}\sum_{t=1}^{n+k_m} \hat{\boldsymbol{\varepsilon}}_J(t)\hat{\boldsymbol{\eta}}_{J^*}(t-k_m)'\right)\hat{V}_{J^*}^{-1}$$

For the Burg ajustment to this Yule-Walker solution, we must prevent any of the $\hat{\boldsymbol{\varepsilon}}_J$ *and* $\hat{\boldsymbol{\eta}}_{J^*}$ *components in the above summation from becoming zero, so we need:*

$$t \in \mathcal{A}_2 \cap \mathcal{B}_2, \implies t = \{k_m + 1, \ldots, n\}.$$

1.5 The minimization problem

Using calculus to minimize (1.20) with respect to $\hat{\Phi}_K(k_m)$, Brockwell and Dahlhaus (1998) find that in the univariate case

$$\tilde{\Phi}_K(k_m) = \frac{\tilde{U}_J\left(\tilde{U}_J + \tilde{V}_{J^*}\right)\sum_{t=1+k_m}^{n}\tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)}{\sum_{t=k_m+1}^{n}\left(\tilde{U}_J^2\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)^2 + \tilde{V}_{J^*}^2\tilde{\boldsymbol{\varepsilon}}_J(t)^2\right)}. \tag{1.21}$$

As already stated, our primary objective in this chapter is to find the general multivariate solution to (1.20) and investigate its performance relative to the remaining three VAR process estimation methods. Substituting for $\tilde{\boldsymbol{\varepsilon}}_K(t)$ and $\tilde{\boldsymbol{\eta}}_{K^*}(t-k_m)$ from (1.18) and (1.19) respectively, into the expression for S_K, we obtain

$$\begin{aligned} S_K(\Phi_K(k_m)) &= \sum_{t=k_m+1}^{n} \big[(\tilde{\boldsymbol{\varepsilon}}_J(t) - \Phi_K(k_m)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m))' ((\tilde{\boldsymbol{\varepsilon}}_J - \Phi_K(k_m)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)) \\ &\quad + (\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m) - \Psi_{K^*}(k_m)\tilde{\boldsymbol{\varepsilon}}_J(t))' (\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m) - \Psi_{K^*}(k_m)\tilde{\boldsymbol{\varepsilon}}_J(t))\big] . \end{aligned}$$

Using relation (1.17) to substitute for $\Psi_{K^*}(k_m)$ in terms of $\Phi_K(k_m)$, expanding the resulting expressions, and noting the symmetry of the covariance matrices, leads to

$$\begin{aligned} S_K(\Phi_K(k_m)) &= \sum_{t=k_m+1}^{n} [\tilde{\boldsymbol{\varepsilon}}_J(t)'\tilde{\boldsymbol{\varepsilon}}_J(t) + \tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)'\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m) \\ &\quad - 2\tilde{\boldsymbol{\varepsilon}}_J(t)'\Phi_K(k_m)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m) \\ &\quad + \tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)'\Phi_K(k_m)'\Phi_K(k_m)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m) \\ &\quad - 2\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)'\tilde{V}_{J^*}\tilde{\Phi}_K(k_m)'\tilde{U}_J^{-1}\tilde{\boldsymbol{\varepsilon}}_J(t) \\ &\quad + \tilde{\boldsymbol{\varepsilon}}_J(t)'\tilde{U}_J^{-1}\Phi_K(k_m)\tilde{V}_{J^*}^2\Phi_K(k_m)'\tilde{U}_J^{-1}\tilde{\boldsymbol{\varepsilon}}_J(t)\Big] . \end{aligned}$$

This expression can written in the more manageable form

$$S_K(X) = \sum_{t=k_m+1}^{n} [\mathbf{a}'\mathbf{a} + \mathbf{b}'\mathbf{b} - 2\mathbf{a}'X\mathbf{b} + \mathbf{b}'X'X\mathbf{b} - 2\mathbf{d}'X'\mathbf{c} + \mathbf{c}'XEX'\mathbf{c}] , \qquad (1.22)$$

where, using lowercase for vectors and uppercase for matrices,

$$\begin{aligned} \mathbf{a} &= \tilde{\boldsymbol{\varepsilon}}_J(t), \\ \mathbf{b} &= \tilde{\boldsymbol{\eta}}_{J^*}(t-k_m), \\ \mathbf{c} &= \tilde{U}_J^{-1}\tilde{\boldsymbol{\varepsilon}}_J(t), \\ \mathbf{d} &= \tilde{V}_{J^*}\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m), \\ E &= \tilde{V}_{J^*}^2, \\ X &= \Phi_K(k_m). \end{aligned} \qquad (1.23)$$

1.6 Finding the minimum

Using the results and identities in appendix A, and noting that S_K is a matrix scalar function, we can now obtain the differentials of each component of (1.22):

- $$\begin{aligned} \mathbf{a}'X\mathbf{b} &= tr(\mathbf{a}'X\mathbf{b}), \quad \text{since this expression is a scalar} \\ &= tr((\mathbf{a}'X)\mathbf{b}) \\ &= tr(\mathbf{b}\mathbf{a}'X), \quad \text{by K-2} \end{aligned}$$

$$\Rightarrow \partial(\mathbf{a}'X\mathbf{b}) = (vec(\mathbf{a}\mathbf{b}'))'vec(\partial X), \quad \text{by D-7} \tag{1.24}$$

- $$\begin{aligned} \partial(\mathbf{a}'X'X\mathbf{b}) &= \partial(\mathbf{a}'X')X\mathbf{b} + \mathbf{a}'X'\partial(X\mathbf{b}), \quad \text{by D-7} \\ &= \mathbf{a}'(\partial X)'X\mathbf{b} + \mathbf{a}'X'(\partial X)\mathbf{b}, \quad \text{by D-2, D-3, and D-4} \\ &= tr(\mathbf{a}'(\partial X)'X\mathbf{b}) + tr(\mathbf{a}'X'(\partial X)\mathbf{b}), \quad \text{each summand a scalar} \\ &= tr((\partial X)'X\mathbf{b}\mathbf{a}') + tr(\mathbf{b}\mathbf{a}'X'(\partial X)), \quad \text{by K-2} \\ &= tr(\mathbf{a}\mathbf{b}'X'\partial X) + tr(\mathbf{b}\mathbf{a}'X'\partial X), \quad \text{by K-3} \\ &= tr[(\mathbf{a}\mathbf{b}'X' + \mathbf{b}\mathbf{a}'X')\partial X] \\ &= [vec(X\mathbf{b}\mathbf{a}' + X\mathbf{a}\mathbf{b}')]'vec\,\partial X, \quad \text{by K-4} \end{aligned}$$

$$\Rightarrow \partial(\mathbf{a}'X'X\mathbf{b}) = [vec(X(\mathbf{b}\mathbf{a}' + \mathbf{a}\mathbf{b}'))]'vec\,\partial X \tag{1.25}$$

•

$$\begin{aligned}
\partial(\mathbf{a}'XEX'\mathbf{b}) &= \mathbf{a}'(\partial X)EX'\mathbf{b} + \mathbf{a}'XE(\partial X)'\mathbf{b}, \quad \text{by D-2, D-3, D-4} \\
&= tr[\mathbf{a}'(\partial X)EX'\mathbf{b} + \mathbf{a}'XE(\partial X)'\mathbf{b}] \\
&= tr[EX'\mathbf{b}\mathbf{a}'(\partial X)] + tr[(\partial X)'\mathbf{b}\mathbf{a}'XE], \quad \text{by K-2, K-3, K-4} \\
&= tr[EX'\mathbf{b}\mathbf{a}'(\partial X)] + tr[E'X'\mathbf{a}\mathbf{b}'(\partial X)], \quad \text{by K-3} \\
&= tr[(EX'\mathbf{b}\mathbf{a}' + E'X'\mathbf{a}\mathbf{b}')\partial X]
\end{aligned}$$

$$\Rightarrow \partial(\mathbf{a}'XEX'\mathbf{b}) = [vec(\mathbf{a}\mathbf{b}'XE + \mathbf{b}\mathbf{a}'XE)]'vec\,\partial X \quad \text{, by K-4.} \tag{1.26}$$

By D-8, the differential of a sum is the sum of the differentials, so that

$$\partial(S_K) = \sum_{t=k_m+1}^{n} \partial\left(\mathbf{a}'\mathbf{a} + \mathbf{b}'\mathbf{b} - 2\mathbf{a}'X\mathbf{b} + \mathbf{b}'X'X\mathbf{b} - 2\mathbf{d}'X'\mathbf{c} + \mathbf{c}'XEX'\mathbf{c}\right).$$

By applying (1.24)-(1.26), we now obtain the differential of each term:

$$\begin{aligned}
\partial S_K &= \sum_{t=k_m+1}^{n} [0 + 0 - 2(vec(\mathbf{a}\mathbf{b}'))'vec(\partial X) + 2(vec(X\mathbf{b}\mathbf{b}'))'vec(\partial X) \\
&\quad - 2(vec(\mathbf{c}\mathbf{d}'))'vec(\partial X) + 2(vec(\mathbf{c}\mathbf{c}'XE))'vec(\partial X)],
\end{aligned}$$

which by K-1 and D-8 becomes

$$\partial S_K = \left[vec\left(-2(\sum_t \mathbf{a}\mathbf{b}') + 2(\sum_t X\mathbf{b}\mathbf{b}') - 2(\sum_t \mathbf{c}\mathbf{d}') + 2(\sum_t \mathbf{c}\mathbf{c}'XE)\right)\right]' vec(\partial X),$$

where for ease of notation, $\sum_t$ will denote $\sum_{t=k_m+1}^{n}$. Noting that X and E are independent of t, we have

$$\partial S_K = \left[vec\left(-2(\sum_t \mathbf{a}\mathbf{b}') + 2X(\sum_t \mathbf{b}\mathbf{b}') - 2(\sum_t \mathbf{c}\mathbf{d}') + 2(\sum_t \mathbf{c}\mathbf{c}')XE\right)\right]' vec(\partial X). \tag{1.27}$$

Comparing with (A.1), we immediately see that the Jacobian matrix of S_K at X is

$$\left[vec\left(-2(\sum_t \mathbf{ab}') + 2(\sum_t X\mathbf{bb}') - 2(\sum_t \mathbf{cd}') + 2(\sum_t \mathbf{cc}'XE) \right) \right]'$$

Premultiplying the second summand of the Jacobian by the $(d \times d)$ identity matrix I_d, equating to zero, and solving for X, leads to

$$vec\left(I_d X(\sum_t \mathbf{bb}') \right) + vec\left((\sum_t \mathbf{cc}')XE \right) = vec\left((\sum_t \mathbf{ab}') + (\sum_t \mathbf{cd}') \right).$$

Now apply K-6 to each summand on the LHS to obtain

$$\left[(\sum_t \mathbf{bb}')' \otimes I_d \right] vec\, X + \left[E' \otimes (\sum_t \mathbf{cc}') \right] vec\, X = vec\left((\sum_t \mathbf{ab}') + (\sum_t \mathbf{cd}') \right)$$

$$\Rightarrow vec\, X = \left[(\sum_t \mathbf{bb}')' \otimes I_d + E' \otimes (\sum_t \mathbf{cc}') \right]^{-1} vec\left((\sum_t \mathbf{ab}') + (\sum_t \mathbf{cd}') \right), \tag{1.28}$$

for any generalized inverse

$$\left[(\sum_t \mathbf{bb}')' \otimes I_d + E' \otimes (\sum_t \mathbf{cc}') \right]^{-1} \text{ of } \left[(\sum_t \mathbf{bb}')' \otimes I_d + E' \otimes (\sum_t \mathbf{cc}') \right].$$

Finally, substituting back to our variables of interest from (1.23), and noting symmetries and independence of t in some of the terms, we obtain (in vec form) the value of the matrix $\Phi_K(k_m)$ that minimizes S_K (note that this matrix is $\tilde{\Phi}_K(k_m)$ by definition):

$$\begin{aligned} &vec(\tilde{\Phi}_K(k_m)) = \\ &\quad \left[\left(\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)' \right) \otimes I_d \right. \\ &\quad \left. + \tilde{V}_{J^*}^2 \otimes \tilde{U}_J^{-1} \left(\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\varepsilon}}_J(t)' \right) \tilde{U}_J^{-1} \right]^{-1} \\ &\quad vec\left[\left(\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)' \right) + \tilde{U}_J^{-1} \left(\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)' \right) \tilde{V}_{J^*} \right]. \end{aligned} \tag{1.29}$$

Setting $d = 1$ for the unidimensional case, we see immediately that (1.29) becomes

$$\begin{aligned}\tilde{\Phi}_K(k_m) &= \frac{\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m) + \frac{\tilde{V}_{J^*}}{\tilde{U}_J}\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)}{\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)^2 + \frac{\tilde{V}_{J^*}^2}{\tilde{U}_J^2}\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\varepsilon}}_J(t)^2} \\ &= \frac{\tilde{U}_J\left(\tilde{U}_J + \tilde{V}_{J^*}\right)\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)}{\tilde{U}_J^2\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)^2 + \tilde{V}_{J^*}^2\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\varepsilon}}_J(t)^2},\end{aligned}$$

which is identical to (1.21).

Remark 1.6.1 *An alternative way to obtain a "good" estimate of* $\Phi_K(k_m)$*, might be to consider the matrix quantity*

$$T_K = \sum_{t=k_m+1}^{n} \left[\hat{\boldsymbol{\varepsilon}}_K(t)\hat{\boldsymbol{\varepsilon}}_K(t)' + \hat{\boldsymbol{\eta}}_{K^*}(t-k_m)\hat{\boldsymbol{\eta}}_{K^*}(t-k_m)'\right],$$

viewing $\hat{\boldsymbol{\varepsilon}}_K(t)\hat{\boldsymbol{\varepsilon}}_K(t)'$ *and* $\hat{\boldsymbol{\eta}}_{K^*}(t-k_m)\hat{\boldsymbol{\eta}}_{K^*}(t-k_m)'$ *as error covariance matrices of some sort. The criterion of* A-Optimality *in linear models would then minimize the trace of* T_K *in order to find an optimal estimate for* $\tilde{\Phi}_K(k_m)$*. By identity K-7, the trace of the sum is the sum of the traces, so that*

$$\begin{aligned}tr(T_K) &= \sum_{t=k_m+1}^{n} \left[tr\left(\hat{\boldsymbol{\varepsilon}}_K(t)\hat{\boldsymbol{\varepsilon}}_K(t)'\right) + tr\left(\hat{\boldsymbol{\eta}}_{K^*}(t-k_m)\hat{\boldsymbol{\eta}}_{K^*}(t-k_m)'\right)\right] \\ &= S_K,\end{aligned}$$

so that the solution (1.29) is also A-Optimal *in the sense just described.*

1.7 Global optimality of the solution

From (1.27) we have

$$\begin{aligned}\partial S_K &= vec\left(-2(\sum_t \mathbf{ab}')\right)' vec(\partial X) + vec\left(2I_d X(\sum_t \mathbf{bb}')\right)' vec(\partial X) \\ &\quad - vec\left(2(\sum_t \mathbf{cd}')\right)' vec(\partial X) + vec\left(2(\sum_t \mathbf{cc}')XE\right)' vec(\partial X).\end{aligned}$$

Using identity K-6 on the terms in the summand that involve X,

$$\begin{aligned}\partial S_K &= \left[vec\left(-2(\sum_t \mathbf{ab}') - 2(\sum_t \mathbf{cd}')\right)\right]' vec(\partial X) \\ &+ \left[\left(2(\sum_t \mathbf{bb}') \otimes I_d\right) vec\, X\right]' vec(\partial X) \\ &+ \left[\left(2E \otimes (\sum_t \mathbf{cc}')\right) vec\, X\right]' vec(\partial X).\end{aligned}$$

Taking transposes and noting that most of the terms are symmetric yields

$$\begin{aligned}\partial S_K &= \left[vec\left(-2(\sum_t \mathbf{ab}') - 2(\sum_t \mathbf{cd}')\right)\right]' vec(\partial X) \\ &+ (vec\, X)' \left(2(\sum_t \mathbf{bb}') \otimes I_d\right) vec(\partial X) \\ &+ (vec\, X)' \left(2E \otimes (\sum_t \mathbf{cc}')\right) vec(\partial X).\end{aligned}$$

Taking differentials again, and using D-10 gives

$$\begin{aligned}\partial^2 S_K &= (vec\, \partial X)' \left(2(\sum_t \mathbf{bb}') \otimes I_d\right) vec(\partial X) \\ &\quad + (vec\, \partial X)' \left(2E \otimes (\sum_t \mathbf{cc}')\right) vec(\partial X) \\ &= (vec\, \partial X)' \left(2(\sum_t \mathbf{bb}') \otimes I_d + 2E \otimes (\sum_t \mathbf{cc}')\right) vec(\partial X).\end{aligned}$$

Rewriting in terms of our variables by substituting back from (1.23), comparing with (A.2), and noting that the resulting matrix is already symmetric, gives the Hessian

$$\begin{aligned}H(\Phi_K(k_m)) &= 2\left(\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)'\right) \otimes I_d \\ &\quad + 2\tilde{V}_{J^*}^2 \otimes \tilde{U}_J^{-1} \left(\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\varepsilon}}_J(t)'\right) \tilde{U}_J^{-1} \qquad (1.30)\end{aligned}$$

Setting $d = 1$ for the unidimensional case, we see immediately that

$$H(\Phi_K(k_m)) = \frac{\partial^2 S_K}{\partial \tilde{\Phi}_K(k_m)^2} = 2 \sum_{t=k_m+1}^{n} \left(\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)^2 + \frac{\tilde{V}_{J^*}^2}{\tilde{U}_J^2} \tilde{\boldsymbol{\varepsilon}}_J(t)^2 \right);$$

and this is clearly positive, thus showing that (1.29) does indeed minimize S_K.

In the general case, theorems A.1.3 and A.1.4 essentially tell us that if we can show the Hessian to be positive semi-definite (psd) for all $\Phi_K(k_m)$, then S_K will have a global minimum at $\tilde{\Phi}_K(k_m)$, as given in (1.29). This minimum will be unique if the Hessian is positive definite (pd) for all $\Phi_K(k_m)$. Since (1.30) is independent of $\Phi_K(k_m)$, we need only show that it is pd or psd as it stands. To this end, we use the results on the definiteness of symmetric matrices in appendix A.3:

The matrices $\tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\varepsilon}}_J(t)'$ and $\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)'$ are, by their very nature, psd. To see this, note that for any $\mathbf{z} \in \mathbb{R}^d$,

$$\mathbf{z}'\tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\varepsilon}}_J(t)'\mathbf{z} = \left(\mathbf{z}'\tilde{\boldsymbol{\varepsilon}}_J(t)\right)\left(\mathbf{z}'\tilde{\boldsymbol{\varepsilon}}_J(t)\right)' = \left(\mathbf{z}'\tilde{\boldsymbol{\varepsilon}}_J(t)\right)^2 \geq 0.$$

Similarly for $\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)'$. Since I_d is pd,

$$\left(\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)' \right) \otimes I_d$$

is psd (M-4). By M-5, both $\tilde{V}_{J^*}^2$ and $\tilde{U}_J^{-1}$ are psd. By M-6 and M-4, we then have

$$\tilde{V}_{J^*}^2 \otimes \tilde{U}_J^{-1} \left(\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\varepsilon}}_J(t)' \right) \tilde{U}_J^{-1}$$

psd. Thus far we have shown that H is the sum of two psd matrices, and by M-3 this is again psd. If either of the two terms comprised of sums of matrices of prediction error residual vectors is pd, then H will be also, and theorems A.1.3 and A.1.4 will then guarantee that S_K has a *unique global minimum* at $\tilde{\Phi}_K(k_m)$.

Alternatively, we can argue as follows: Since $S_K = S_K(\mathbf{u})$ is a non-negative quadratic form in the components of $\mathbf{u} \equiv vec\,\Phi_K(k_m)$, it is expressible as

$$(\mathbf{u} - \boldsymbol{\beta})'\Omega(\mathbf{u} - \boldsymbol{\beta}) + \delta,$$

where the vector $\boldsymbol{\beta}$ is independent of $\mathbf{u}$, Ω is a non-negative definite matrix, and the scalar $\delta \geq 0$. Thus $S_K(\mathbf{u})$ has a global minimum value, namely δ, attained when $\mathbf{u} = \boldsymbol{\beta} + \mathbf{v}$, where $\mathbf{v}$ is any vector in the null space, $\mathcal{N}(\Omega)$, of Ω. If Ω is non-singular, there is therefore a unique minimizing value of $\Phi_K(k_m)$. In any case, if $\mathbf{u}_1$ and $\mathbf{u}_2$ are two minimizing values of $S_K(\mathbf{u})$, then

$$\mathbf{u}_1 - \mathbf{u}_2 \in \mathcal{N}(\Omega), \quad \text{and} \quad S_K(\mathbf{u}_1) = S_K(\mathbf{u}_2).$$

1.8 Some Monte Carlo comparisons of the Yule-Walker and Burg algorithms

We first present three examples to compare the performance of the *Yule-Walker* (algorithm 1.3.1), *Burg* (algorithm 1.4.4), and, in the univariate case, *Maximum Likelihood* subset AR and VAR modeling, applied to simulated data sets with Gaussian noise. Since one of the aims of the Yule-Walker and Burg algorithms is to provide fast and simple algorithms for obtaining models with high Gaussian likelihoods, it is of considerable interest to compare the likelihoods achieved by each.

If $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is a realization from the zero-mean $d-$variate full rank Gaussian process $\{\mathbf{X}_t\}$, with

$$\Gamma_n = \mathbb{E}\left([\mathbf{X}_1', \ldots, \mathbf{X}_n']'[\mathbf{X}_1', \ldots, \mathbf{X}_n']\right),$$

we obtain the likelihood,

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_n) = (2\pi)^{-dn/2} \left|\Gamma_n\right|^{-1/2} \exp\left\{-\frac{1}{2}[\mathbf{x}_1', \ldots, \mathbf{x}_n']\Gamma_n^{-1}[\mathbf{x}_1', \ldots, \mathbf{x}_n']'\right\}.$$

If $\{\mathbf{X}_t\}$ is the causal subset VAR process

$$\mathbf{X}_t = \sum_{i \in K} \Phi_K(i)\mathbf{X}_{t-i} + \mathbf{Z}_t, \quad \{\mathbf{Z}_t\} \sim \text{IID N}(\mathbf{0}, \Sigma),$$

we can write the likelihood for $n > k_m$ as,

$$\begin{aligned} f(\mathbf{x}_1, \ldots, \mathbf{x}_n) &= f(\mathbf{x}_1, \ldots, \mathbf{x}_{k_m}) f_{\mathbf{X}_{k_m+1}|\mathbf{X}_t, t\le k_m}(\mathbf{x}_{k_m+1}|\mathbf{x}_t, t \le k_m) \\ &\quad \cdots f_{\mathbf{X}_n|\mathbf{X}_t, t\le n-1}(\mathbf{x}_n|\mathbf{x}_t, t \le n-1). \end{aligned}$$

The first factor is

$$f(\mathbf{x}_1, \ldots, \mathbf{x}_{k_m}) = (2\pi)^{-dk_m/2} \left|\Gamma_{k_m}\right|^{-1/2} \exp\left\{-\frac{1}{2}[\mathbf{x}_1', \ldots, \mathbf{x}_{k_m}']\Gamma_{k_m}^{-1}[\mathbf{x}_1', \ldots, \mathbf{x}_{k_m}']'\right\},$$

where $\Gamma_{k_m} = \mathbb{E}\left([\mathbf{X}_1', \ldots, \mathbf{X}_{k_m}']'[\mathbf{X}_1', \ldots, \mathbf{X}_{k_m}']\right)$.

The remaining $n - k_m$ factors are

$$\begin{aligned} &f_{\mathbf{X}_t|\mathbf{X}_s, s<t}(\mathbf{x}_t|\mathbf{x}_s, s < t) = \\ &(2\pi)^{-d/2} \left|\Sigma\right|^{-1/2} \exp\left\{-\frac{1}{2}\left[\mathbf{x}_t - \sum_{j\in K} \Phi_K(j)\mathbf{x}_{t-j}\right]' \Sigma^{-1} \left[\mathbf{x}_t - \sum_{j\in K} \Phi_K(j)\mathbf{x}_{t-j}\right]\right\}, \end{aligned}$$

for $t = k_m + 1, \ldots, n$, since conditional on $\mathbf{X}_s,\ s < t$,

$$\mathbf{X}_t \sim \mathrm{N}_d\left(\sum_{j\in K} \Phi_K(j)\mathbf{X}_{t-j},\ \Sigma\right).$$

Putting all this together, and viewing the likelihood as a function of Σ, for fixed autoregressive coefficients, we obtain the -2 log likelihood:

$$\begin{aligned} &\mathcal{L}(\Sigma) = \\ &\quad nd\log(2\pi) + \log\left|\Gamma_{k_m}\right| + (n - k_m)\log\left|\Sigma\right| + \left[\mathbf{x}_1', \ldots, \mathbf{x}_{k_m}'\right]\Gamma_{k_m}^{-1}\left[\mathbf{x}_1', \ldots, \mathbf{x}_{k_m}'\right]' \\ &\quad + \sum_{t=k_m+1}^{n} \left[\mathbf{x}_t - \sum_{j\in K} \Phi_K(j)\mathbf{x}_{t-j}\right]' \Sigma^{-1} \left[\mathbf{x}_t - \sum_{j\in K} \Phi_K(j)\mathbf{x}_{t-j}\right]. \end{aligned} \tag{1.31}$$

In the univariate case (i.e. if $d = 1$), where

$$\{\Phi_K(k_1), \ldots, \Phi_K(k_m)\} \equiv \{\phi_K(k_1), \ldots, \phi_K(k_m)\}, \quad \text{and} \quad \Sigma \equiv \sigma^2,$$

we can set $\sigma^{-2}\Gamma_{k_m} \equiv G_{k_m}$, which is free of σ^2. This gives the -2 log likelihood for the data:

$$\mathcal{L}(\sigma^2) = n\log(2\pi\sigma^2) + \log|G_{k_m}| + \frac{1}{\sigma^2}\left\{ [\mathbf{x}_1, \ldots, \mathbf{x}_{k_m}] G_{k_m}^{-1} [\mathbf{x}_1, \ldots, \mathbf{x}_{k_m}]' + \sum_{t=k_m+1}^{n} \left(\mathbf{x}_t - \sum_{j\in K} \Phi_K(j)\mathbf{x}_{t-j} \right)^2 \right\}.$$

We can now obtain the maximum likelihood estimate of σ^2, by differentiating the above expression to give

$$\hat{\sigma}^2_{\text{MLE}} = \frac{1}{n}\left\{ [\mathbf{x}_1, \ldots, \mathbf{x}_{k_m}] G_{k_m}^{-1} [\mathbf{x}_1, \ldots, \mathbf{x}_{k_m}]' + \sum_{t=k_m+1}^{n} \left(\mathbf{x}_t - \sum_{j\in K} \Phi_K(j)\mathbf{x}_{t-j} \right)^2 \right\}. \tag{1.32}$$

The expression in braces is usually called the *residual sum of squares (RSS)*, and therefore $\hat{\sigma}^2_{\text{MLE}} = RSS/n$.

Therefore, for a given set of estimated autoregressive coefficients and white noise variance, one can always improve on (or do no worse than) the Gaussian likelihood for the observed data, by substituting RSS/n for the white noise variance estimate. In the ensuing examples, we will compare likelihoods obtained via the two algorithms, and proceed to improve upon each by doing just this. We will denote the variance estimate obtained in the usual way from the algorithm, by $\hat{\sigma^2}_{AL}$ or $\hat{\Sigma}_{AL}$. The variance estimate obtained via maximization of the likelihood function (or equivalently, minimization of $\mathcal{L}(\Sigma)$) with respect to Σ for the fixed set of estimated AR/VAR coefficients, will de denoted by $\hat{\sigma^2}_{ML}$ or $\hat{\Sigma}_{ML}$.

In the multivariate setting, it is difficult (perhaps impossible) to obtain an explicit expression for the maximum likelihood estimator of the white noise covariance matrix Σ given the $\Phi_K(\cdot)$'s, as was achieved in (1.32) for the univariate case. We may begin understanding the difficulties involved if we recall that Σ is connected to the

process autocovariance function $\Gamma(\cdot)$ via the causal representation,

$$\Gamma(h) = \sum_{j=0}^{\infty} \Psi_{h+j} \Sigma \Psi_j'. \tag{1.33}$$

In the multivariate examples of dimension 2 that follow, we use a direct search for the minimizing white noise matrix, expressing the objective function $\mathcal{L}(\Sigma)$ as a function of the three components of Σ. The Hooke and Jeeves algorithm, Hooke and Jeeves (1961), constrained to yield a positive definite solution, is employed to perform the search. Further details are given in appendix C.

Example 1.8.1 *200 observations were simulated from the causal subset AR(11) model*

$$X_t - 0.98X_{t-1} + 0.924X_{t-2} - 0.138X_{t-4} + 0.0033X_{t-7} - 0.5X_{t-8} - 0.12X_{t-11} = Z_t \tag{1.34}$$

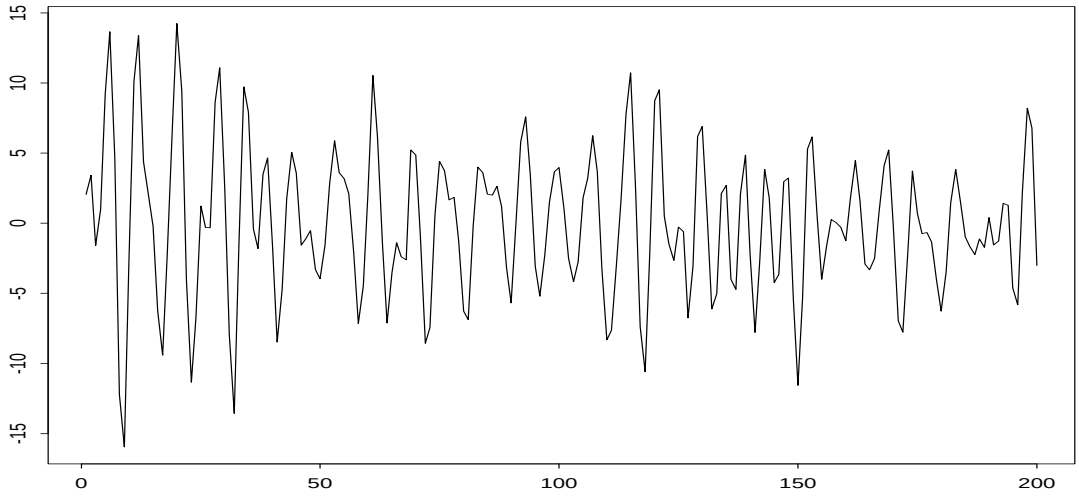
where $\{Z_t\} \sim$ *WN(0,4). The data is displayed in figure 1.1.*

Table 1.1: Parameters and estimates for the data of example 1.8.1.

Parameter	Parameter Estimates by Method			
	Truth	Yule-Walker	Burg	Maximum Likelihood
$\Phi_K(1)$	0.9800	0.8958	0.9066	0.9048
$\Phi_K(2)$	-0.9240	-0.8525	-0.8860	-0.8890
$\Phi_K(4)$	0.1380	0.0419	0.0319	0.0376
$\Phi_K(7)$	-0.0033	-0.0549	-0.0941	-0.0862
$\Phi_K(8)$	0.5000	0.4137	0.4675	0.4759
$\Phi_K(11)$	0.1200	0.0927	0.1160	0.1193
σ^2_{AL}	4	4.6813	3.8253	
σ^2_{ML}	4	3.9002	3.8330	3.8242
$\mathcal{L}(\sigma^2_{AL})$		847.804	842.193	
$\mathcal{L}(\sigma^2_{ML})$		844.666	842.192	842.006

Here we have $m = 6$ *and* $K = \{1, 2, 4, 7, 8, 11\}$. *Table 1.1 shows the true parameter values, contrasted against the estimates obtained via Yule-Walker, Burg, and maximum likelihood modeling. We note that the likelihood of the Burg estimates based on*

Figure 1.1: Plot of the simulated subset AR(11) data set of example 1.8.1



both σ^2_{AL} *and* σ^2_{ML}*, are higher than is the case for Yule-Walker. A possible reason for the higher likelihoods based on* $\tilde{\sigma}^2_{AL}$ *compared to* $\hat{\sigma}^2_{AL}$*, is the observed tendency of* $\tilde{\sigma}^2_{AL}$ *to be closer to* $\tilde{\sigma}^2_{ML}$ *than is the case for Yule-Walker.*

Example 1.8.2 *500 observations were simulated from the causal subset VAR(3) model of dimension 2*

$$\mathbf{X}_t - \begin{bmatrix} -0.4 & 1.3 \\ -0.3 & 1.2 \end{bmatrix} \mathbf{X}_{t-1} - \begin{bmatrix} 0.4 & -0.4 \\ -0.06 & 0.05 \end{bmatrix} \mathbf{X}_{t-3} = \mathbf{Z}_t \tag{1.35}$$

where $\{\mathbf{Z}_t\} \sim WN\left(\mathbf{0}, \Sigma = \begin{bmatrix} 1633 & 2043 \\ 2043 & 3024 \end{bmatrix}\right)$.

Here we have $m = 2$ *and* $K = \{1, 3\}$*. Table 1.2 shows the true parameter values, contrasted against the estimates obtained via Yule-Walker and Burg modeling. The results are similar to the univariate example, with the likelihoods based on the respective algorithm-obtained white noise covariance matrix estimates being superior*

Table 1.2: Parameters and estimates by method for the data of example 1.8.2.

Parameter	Parameter Estimates by Method		
	Truth	Yule-Walker	Burg
$\Phi_K(1)$	$\begin{bmatrix} -0.4 & 1.3 \\ -0.3 & 1.2 \end{bmatrix}$	$\begin{bmatrix} -0.471 & 1.348 \\ -0.303 & 1.202 \end{bmatrix}$	$\begin{bmatrix} -0.476 & 1.355 \\ -0.309 & 1.209 \end{bmatrix}$
$\Phi_K(3)$	$\begin{bmatrix} 0.4 & -0.4 \\ -0.06 & 0.05 \end{bmatrix}$	$\begin{bmatrix} 0.455 & -0.421 \\ 0.049 & -0.026 \end{bmatrix}$	$\begin{bmatrix} 0.464 & -0.430 \\ 0.054 & -0.031 \end{bmatrix}$
Σ_{AL}	$\begin{bmatrix} 1633 & 2043 \\ 2043 & 3024 \end{bmatrix}$	$\begin{bmatrix} 1560.3 & 1885.4 \\ 1885.4 & 2793.2 \end{bmatrix}$	$\begin{bmatrix} 1448.6 & 1790.0 \\ 1790.0 & 2713.0 \end{bmatrix}$
Σ_{ML}	$\begin{bmatrix} 1633 & 2043 \\ 2043 & 3024 \end{bmatrix}$	$\begin{bmatrix} 1456.4 & 1800.6 \\ 1800.6 & 2727.4 \end{bmatrix}$	$\begin{bmatrix} 1456.3 & 1800.4 \\ 1800.4 & 2727.0 \end{bmatrix}$
$\mathcal{L}(\Sigma_{AL})$		9594.35	9591.85
$\mathcal{L}(\Sigma_{ML})$		9591.91	9591.84

for Burg. Again we note the likelihood for the Burg estimates based on the estimated Σ_{AL} being substantially closer to those based on Σ_{ML} than is the case for Yule-Walker.

Example 1.8.3 *500 observations were simulated from the causal subset VAR(7) model of dimension 2*

$$\mathbf{X}_t - \begin{bmatrix} -0.28 & 1.29 \\ -0.62 & 1.63 \end{bmatrix} \mathbf{X}_{t-1} - \begin{bmatrix} 0.42 & -0.45 \\ 0.54 & -0.58 \end{bmatrix} \mathbf{X}_{t-4} - \begin{bmatrix} -0.19 & 0.20 \\ -0.30 & 0.31 \end{bmatrix} \mathbf{X}_{t-5} - \begin{bmatrix} -0.11 & 0.08 \\ -0.22 & 0.20 \end{bmatrix} \mathbf{X}_{t-7} = \mathbf{Z}_t \tag{1.36}$$

where $\{\mathbf{Z}_t\} \sim WN\left(\mathbf{0}, \Sigma = \begin{bmatrix} 27.5 & 28.2 \\ 28.2 & 30.6 \end{bmatrix}\right)$.

Here we have $m = 4$ *and* $K = \{1, 4, 5, 7\}$ *Table 1.3 shows the true parameter values, contrasted against the estimates obtained via Yule-Walker and Burg modeling. Once*

Table 1.3: Parameters and estimates by method for the data of example 1.8.3.

Parameter	Parameter Estimates by Method		
	Truth	Yule-Walker	Burg
$\Phi_K(1)$	$\begin{bmatrix} -0.28 & 1.29 \\ -0.62 & 1.63 \end{bmatrix}$	$\begin{bmatrix} -0.260 & 1.287 \\ -0.612 & 1.634 \end{bmatrix}$	$\begin{bmatrix} -0.223 & 1.256 \\ -0.584 & 1.618 \end{bmatrix}$
$\Phi_K(4)$	$\begin{bmatrix} 0.42 & -0.45 \\ 0.54 & -0.58 \end{bmatrix}$	$\begin{bmatrix} 0.754 & -0.802 \\ 0.916 & -0.961 \end{bmatrix}$	$\begin{bmatrix} 0.708 & -0.725 \\ 0.876 & -0.889 \end{bmatrix}$
$\Phi_K(5)$	$\begin{bmatrix} 0.19 & 0.20 \\ -0.30 & 0.31 \end{bmatrix}$	$\begin{bmatrix} -0.075 & 0.097 \\ -0.188 & 0.189 \end{bmatrix}$	$\begin{bmatrix} 0.068 & -0.073 \\ -0.053 & 0.021 \end{bmatrix}$
$\Phi_K(7)$	$\begin{bmatrix} -0.11 & 0.08 \\ -0.22 & 0.20 \end{bmatrix}$	$\begin{bmatrix} -0.004 & -0.020 \\ -0.104 & 0.091 \end{bmatrix}$	$\begin{bmatrix} 0.090 & -0.123 \\ -0.018 & -0.000 \end{bmatrix}$
Σ_{AL}	$\begin{bmatrix} 27.5 & 28.2 \\ 28.2 & 30.6 \end{bmatrix}$	$\begin{bmatrix} 33.4 & 33.9 \\ 33.9 & 36.1 \end{bmatrix}$	$\begin{bmatrix} 28.3 & 28.9 \\ 28.9 & 31.1 \end{bmatrix}$
Σ_{ML}	$\begin{bmatrix} 27.5 & 28.2 \\ 28.2 & 30.6 \end{bmatrix}$	$\begin{bmatrix} 28.4 & 29.0 \\ 29.0 & 31.2 \end{bmatrix}$	$\begin{bmatrix} 28.3 & 28.8 \\ 28.8 & 31.1 \end{bmatrix}$
$\mathcal{L}(\Sigma_{AL})$		4763.65	4754.51
$\mathcal{L}(\Sigma_{ML})$		4757.04	4754.51

again we note that the likelihood for the Burg estimates is higher than that for Yule-Walker. Note also the dramatic improvement in estimated white noise covariance matrix and likelihood for Yule-Walker, when we use $\hat{\Sigma}_{ML}$.

1.9 Monte Carlo comparisons of Yule-Walker, Burg, Vieira-Morf, and Nuttall-Strand algorithms

Since these four algorithms are frequently used as quick and easy VAR estimation methods, we present a simulation study of the relative performance of each in terms of the size of its Gaussian likelihood. We examine first the univariate setting.

1.9.1 Univariate case

We simulate 1,000 realizations from univariate models, with $\{Z_t\} \sim$ IID N$(0, 1)$, and various configurations of roots of the autoregressive polynomial. For each realization, the Yule-Walker, Burg, Vieira-Morf, and Nuttall-Strand solutions are obtained, and the respective Gaussian -2 log likelihoods computed based on the RSS/n white noise variance estimate. The maximum likelihood solution is also obtained, and its -2 log likelihood subtracted from that of each of the four algorithms, to give what we will call the *net -2 log likelihood ($\mathcal{NL}$).* This maximum likelihood solution is computed using the true parameter values as initial guesses to the Hooke and Jeeves minimization routine. If for a particular realization the likelihood of a model arrived at via one of the algorithms was higher than that obtained for the model with the true parameter values, those parameter estimates were used as initial guesses instead. See appendix C for more details.

Example 1.9.1 *100 observations were simulated from the causal subset AR(3) model*

$$(1 + 0.5B)(1 - (0.1 - 0.3i)B)(1 - (0.1 + 0.3i)B)X_t = Z_t. \tag{1.37}$$

The roots of the AR polynomyal are -2 and $1 \pm 3i$, with moduli 2 and $\sqrt{10}$, respectively. The summaries and plots of table 1.4 and left side of figure 1.2, shows Yule-Walker giving lower $\mathcal{NL}$'s about $1/3$ of the time, but with a somewhat higher mean and variance than the remaining 3 methods.

Example 1.9.2 *100 observations were simulated from the causal subset AR(4) model*

$$(1 + 0.98B)(1 - 0.98B)(1 + 0.98iB)(1 - 0.98iB)X_t = Z_t \tag{1.38}$$

Table 1.4: Summary statistics by method for the data of example 1.9.1

Method	Mean of $\mathcal{NL}$	Median of $\mathcal{NL}$	Std. Dev. of $\mathcal{NL}$	% of realizations with lowest $\mathcal{NL}$
Yule-Walker	0.011	0.002	0.027	33.3
Burg	0.003	0.001	0.007	17.7
Vieira-Morf	0.003	0.001	0.007	26.5
Nuttall-Strand	0.006	0.002	0.010	22.5

The roots of the AR polynomial are ± 1.0204 *and* $\pm 1.0204i$*. The summaries and plots of table 1.5 and right side of figure 1.2, show that Yule-Walker now performs poorly, with substantially higher mean and variance than the remaining 3 methods. Since the model contains only one non-zero autoregressive coefficient, the Burg and Nuttall-Strand estimators are algebraically identical in this case.*

Table 1.5: Summary statistics by method for the data of example 1.9.2

Method	Mean of $\mathcal{NL}$	Median of $\mathcal{NL}$	Std. Dev. of $\mathcal{NL}$	% of realizations with lowest $\mathcal{NL}$
Yule-Walker	1.629	0.994	1.84	14.3
Burg and Nuttall-Strand	0.112	0.053	0.17	38.5
Vieira-Morf	0.108	0.052	0.16	47.2

Example 1.9.3 *100 observations were simulated from the causal subset AR(4) model*

$$(1 + 0.98B)(1 - 0.95B^3)X_t = Z_t \tag{1.39}$$

The roots of the AR polynomyal are $-0.5086 \pm 0.8809i$ *(with modulus* 1.0172*),* 1.0172*, and* -1.0204*. The summaries and plots of table 1.6 and the left side of figure 1.3, show that now Morf and Burg have similar performance, which is substantially better than Yule-Walker and Nuttall-Strand.*

Figure 1.2: Boxplots and barplots for the data of example 1.9.1 (left), and example 1.9.2 (right)

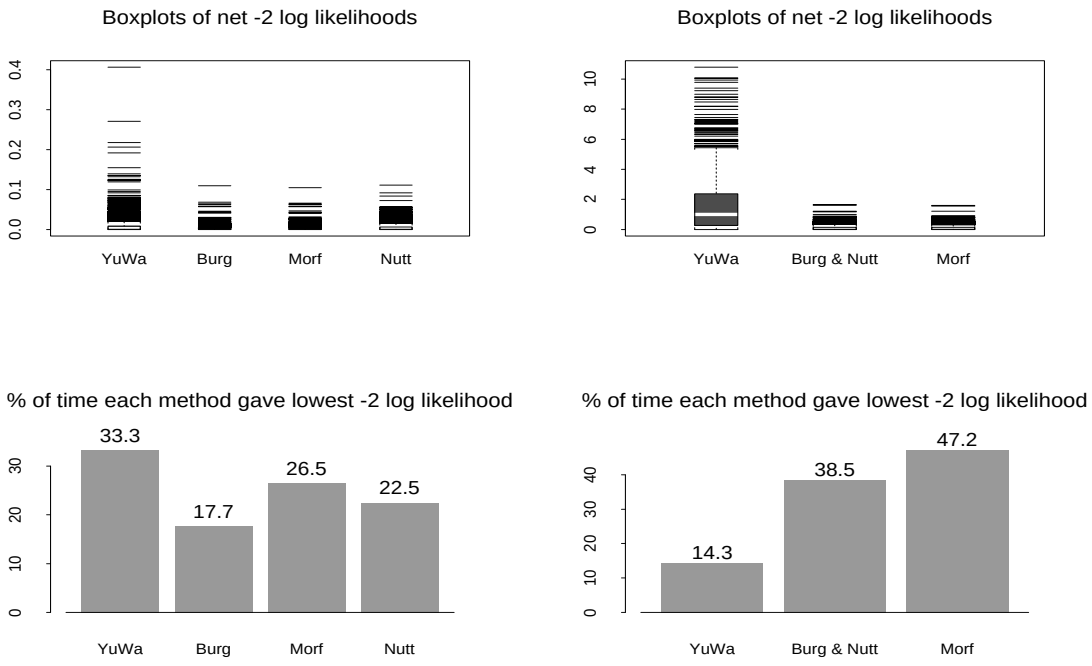


Table 1.6: Summary statistics by method for the data of example 1.9.3

Method	Mean of $\mathcal{NL}$	Median of $\mathcal{NL}$	Std. Dev. of $\mathcal{NL}$	% of realizations with lowest $\mathcal{NL}$
Yule-Walker	5.941	3.710	6.508	6.5
Burg	0.513	0.284	0.761	42.8
Vieira-Morf	0.513	0.285	0.766	45.3
Nuttall-Strand	9.747	6.982	9.336	5.4

Example 1.9.4 *100 observations were simulated from the causal subset AR(4) model*

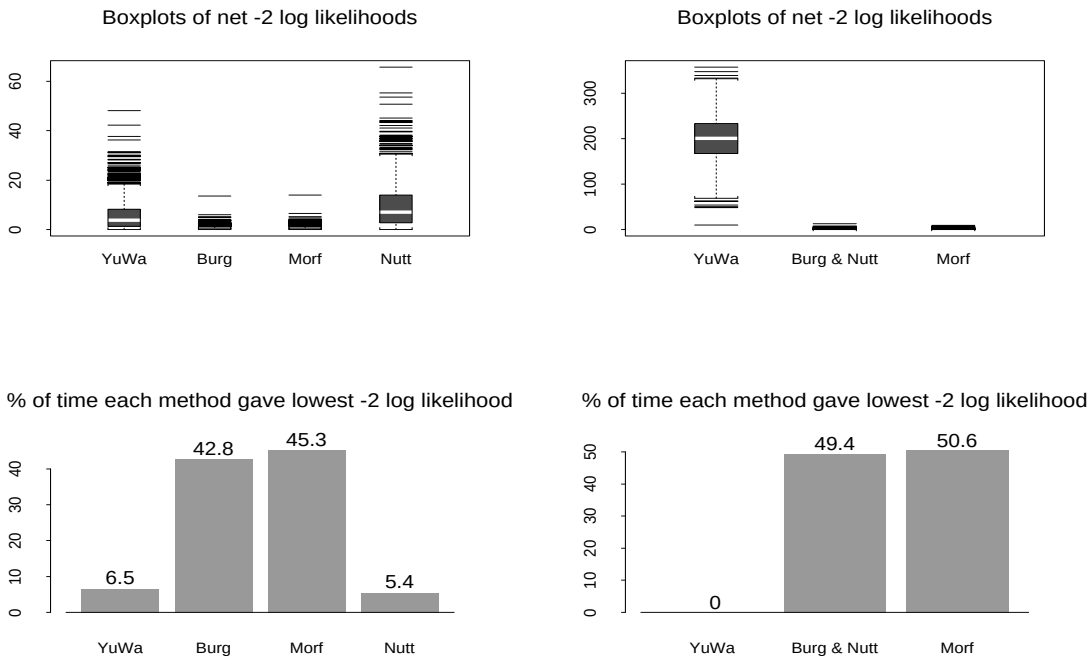
$$(1 - 0.95B^2)(1 + 0.98B)(1 - 0.98B)X_t = Z_t \tag{1.40}$$

The roots of the AR polynomyal are ± 1.0204 *and* ± 1.0260. *From the summaries and plots of table 1.7 and right side of figure 1.3, it is evident that Yule-Walker's performance is far inferior to the remaining methods, particularly Vieira-Morf. Once again, Burg and Nuttall-Strand yield identical solutions.*

Table 1.7: Summary statistics by method for the data of example 1.9.4

Method	Mean of $\mathcal{NL}$	Median of $\mathcal{NL}$	Std. Dev. of $\mathcal{NL}$	% of realizations with lowest $\mathcal{NL}$
Yule-Walker	200.18	200.802	48.83	0.0
Burg and Nuttall-Strand	0.38	0.139	0.80	49.4
Vieira-Morf	0.32	0.133	0.64	50.6

Figure 1.3: Boxplots and barplots for the data of example 1.9.3 (left), and example 1.9.4 (right)



Restricting to the univariate case, we compared the performance with various configurations of roots of the AR polynomial. With roots far from the unit circle in the complex plane, Yule-Walker's performance is comparable with the remaining methods. As the roots approach the unit circle and the real axis, we see the Burg and Vieira-Morf solutions giving consistently higher likelihoods. Apart from the special scenarios where it coincides with Burg, the Nuttall-Strand method performs similarly to Yule-Walker. On the whole, the Burg and Vieira-Morf methods perform better than the rest, tending to give higher likelihoods with smaller variability across a large number of realizations.

1.9.2 Multivariate case

Motivated by the changing results of the modeling algorithms in the face of different configurations of roots of the autoregressive polynomial, we seek to investigate this behavior for analogous scenarios in the bivariate case. Appendix B details the methods used to find the VAR coefficients that correspond to models with specified characteristic polynomials.

Due to the difficulties involved in finding maximum likelihood solutions in the multivariate setting, we concentrate on bivariate models with subset size one. 200 realizations are then simulated from each, with noise $\mathbf{Z}_t \sim \mathrm{N}_2(\mathbf{0}, I_2)$, and configurations of roots of the VAR characteristic polynomial that mimic those of the univariate examples. For each realization, the $\mathcal{NL}$ for each of the four algorithms is obtained in the same manner as in the univariate examples. Unlike the univariate case though, the search for the maximum likelihood estimates is carried out simultaneously for the coefficients and the white noise covariance matrix.

Example 1.9.5 *100 observations were simulated from the causal bivariate subset VAR(2) model*

$$\mathbf{X}_t - \begin{bmatrix} 0.547 & -0.300 \\ 0.700 & -0.457 \end{bmatrix} \mathbf{X}_{t-2} = \mathbf{Z}_t,$$

with characteristic polynomial

$$|\Phi(z)| = (1 - 0.25z^2)(1 + 0.16z^2).$$

having roots ± 2 *and* $\pm 2.5i$*. The summaries and plots of table 1.8 and left side of figure 1.4, show the Burg, Vieira-Morf, and Nuttall-Strand methods giving similar means and variances for the* $\mathcal{NL}$*'s. Yule-walker has a somewhat larger mean and variance.*

Table 1.8: Summary statistics by method for the data of example 1.9.5

Method	Mean of $\mathcal{NL}$	Median of $\mathcal{NL}$	Std. Dev. of $\mathcal{NL}$	% of realizations with lowest $\mathcal{NL}$
Yule-Walker	0.137	0.076	0.168	12.5
Burg	0.030	0.021	0.027	25.5
Vieira-Morf	0.028	0.018	0.029	32.0
Nuttall-Strand	0.028	0.020	0.029	30.0

Example 1.9.6 *100 observations were simulated from the bivariate causal subset VAR(2) model*

$$\mathbf{X}_t - \begin{bmatrix} 1.0091 & -0.3000 \\ 0.7000 & -1.0670 \end{bmatrix} \mathbf{X}_{t-2} = \mathbf{Z}_t,$$

with characteristic polynomial

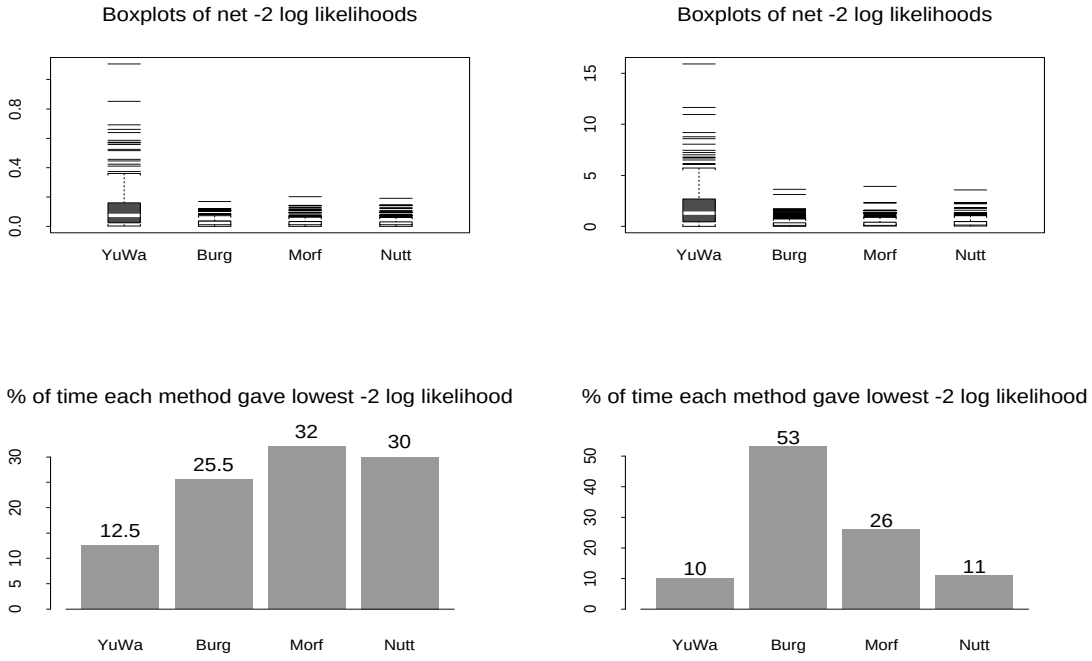
$$|\Phi(z)| = (1 + 0.98^2 z^2)(1 - 0.95^2 z^2)$$

having roots ± 1.0526 *and* $\pm 1.0204i$*. The summaries and plots of table 1.9 and right side of figure 1.4, show that Yule-Walker now gives lower* $\mathcal{NL}$*'s about 10% of the time, but with substantially higher mean and dispersion than the remaining 3 methods, which continue to perform comparably.*

Table 1.9: Summary statistics by method for the data of example 1.9.6

Method	Mean of $\mathcal{NL}$	Median of $\mathcal{NL}$	Std. Dev. of $\mathcal{NL}$	% of realizations with lowest $\mathcal{NL}$
Yule-Walker	2.07	1.29	2.39	10.0
Burg	0.33	0.20	0.45	53.0
Vieira-Morf	0.37	0.22	0.45	26.0
Nuttall-Strand	0.40	0.26	0.46	11.0

Figure 1.4: Boxplots and barplots for the data of example 1.9.5 (left), and example 1.9.6 (right)



Example 1.9.7 *100 observations were simulated from the bivariate causal subset VAR(2) model*

$$\mathbf{X}_t - \begin{bmatrix} 0.4 & -1.2 \\ 0.9 & -0.4 \end{bmatrix} \mathbf{X}_{t-2} = \mathbf{Z}_t,$$

with characteristic polynomial

$$|\Phi(z)| = (1 + 0.92z^4)$$

having roots $\pm 0.722 \pm 0.722i$ *(with modulus* 1.0211*). The summaries and plots of table 1.10 and the left side of figure 1.5, show that now the mean and variance for the* $\mathcal{NL}$*'s of the Burg, Nuttall-Strand, and Vieira-Morf methods, is substantially lower than Yule-Walker.*

Table 1.10: Summary statistics by method for the data of example 1.9.7

Method	Mean of $\mathcal{NL}$	Median of $\mathcal{NL}$	Std. Dev. of $\mathcal{NL}$	% of realizations with lowest $\mathcal{NL}$
Yule-Walker	2.551	1.744	2.527	10.0
Burg	0.538	0.339	0.617	56.0
Vieira-Morf	0.610	0.393	0.630	20.0
Nuttall-Strand	0.608	0.387	0.635	14.0

Example 1.9.8 *100 observations were simulated from the bivariate causal subset VAR(2) model*

$$\mathbf{X}_t - \begin{bmatrix} 1.4135 & -0.3000 \\ 0.7000 & 0.4969 \end{bmatrix} \mathbf{X}_{t-2} = \mathbf{Z}_t,$$

with characteristic polynomial

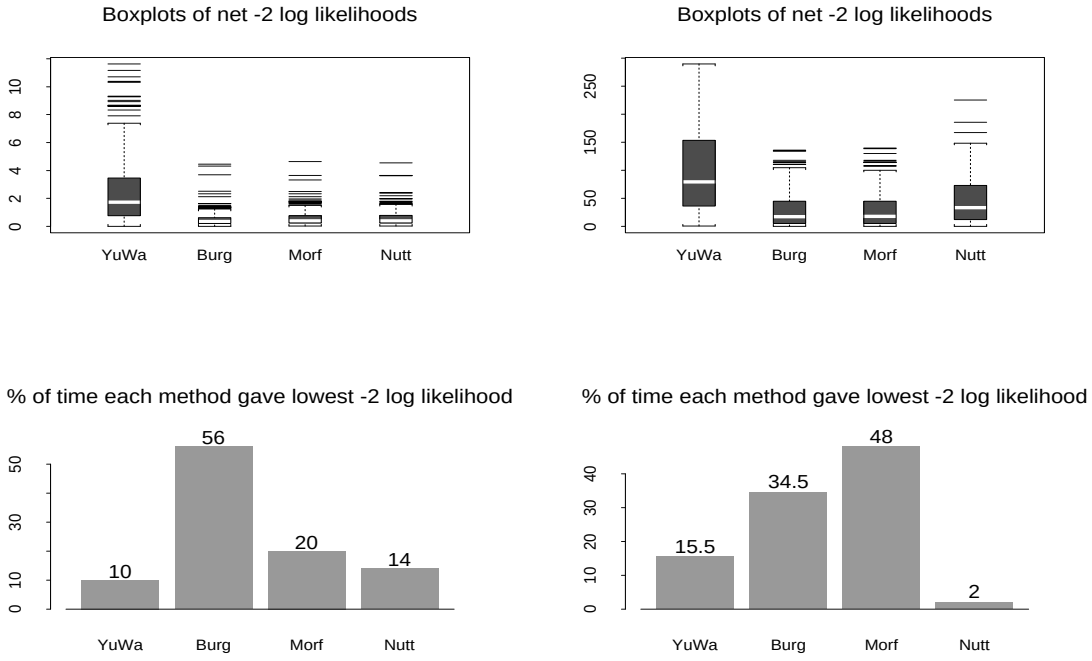
$$|\Phi(z)| = (1 - 0.98^2 z^2)(1 - 0.95^2 z^2)$$

having roots ± 1.0204 *and* ± 1.0260*. From the summaries and plots of table 1.11 and right side of figure 1.5, we see that the performance of Burg and Vieira-Morf is much better than that of the remaining two methods. Unlike the remaining examples though, a curious event occurred here in that 239 realizations were simulated instead of 200. In 39 of those realizations, the Burg solution, although causal, produced a negative definite white noise covariance matrix. These 39 realizations were omitted from the simulation results.*

Table 1.11: Summary statistics by method for the data of example 1.9.4

Method	Mean of $\mathcal{NL}$	Median of $\mathcal{NL}$	Std. Dev. of $\mathcal{NL}$	% of realizations with lowest $\mathcal{NL}$
Yule-Walker	97.7	79.5	72.7	15.5
Burg	29.9	17.1	32.5	34.5
Vieira-Morf	29.8	18.1	32.2	48.0
Nuttall-Strand	46.9	33.1	42.3	2.0

Figure 1.5: Boxplots and barplots for the data of example 1.9.7 (left), and example 1.9.8 (right)



These multivariate examples display similar behavior to that seen in the univariate case. The main theme continues to be that Yule-Walker's performance is inferior to the remaining three methods. Burg and Vieira-Morf emerge as clear winners here too, albeit closely followed by Nuttall-Strand.

Chapter 2

ASYMPTOTIC NORMALITY OF SOME SUBSET VECTOR AUTOREGRESSIVE PROCESS ESTIMATORS

2.1 Introduction

In this chapter we establish asymptotic normality for the distribution of some of the more common *subset vector autoregressive (SVAR)* process estimators, namely: Least Squares (LS), Yule-Walker (YW, algorithm 1.3.1), and the Burg estimator introduced in chapter 1 (algorithm 1.4.4). This is achieved by first finding the asymptotic distribution of the subset Least Squares estimator, and showing its asymptotic equivalence with the subset Yule-Walker. This equivalence is then extended to the subset Burg estimator via the subset Yule-Walker, which thus inherits all central limit theorems applicable to the other two.

A partial derivation of the asymptotics for the multivariate full set LS estimator can be found in Lutkepohl (1993), section 3.2. In this chapter, we generalize this to the subset case, using the theory of martingales to complete the derivation. In Hannan (1970), section 6.2, we can find a development of the multivariate YW asymptotics. However, we choose instead to generalize the univariate full set arguments of Brockwell and Davis (1991) to the multivariate subset case. The asymptotics for the multivariate full set Burg estimator, was recently presented by Hainz (1994). We extend this derivation to the subset case.

Condition 1

We assume that $\{\mathbf{X}_t = (X_{t,1}, \ldots, X_{t,d})',\ t = 0, \pm 1, \pm 2, \ldots\}$ is a zero-mean d-variate stationary ergodic stochastic process of full rank, with finite variance, and $(d \times d)$ covariance matrix at lag h

$$\mathbf{E}\left[\mathbf{X}_{t+h}\mathbf{X}_t'\right] \equiv \Gamma(h).$$

Condition 2

In some cases, we shall assume that $\{\mathbf{X}_t\}$ follows the causal SVAR(K) model,

$$\mathbf{X}_t = \sum_{i \in K} \Phi_K(i)\mathbf{X}_{t-i} + \mathbf{Z}_t, \quad \{\mathbf{Z}_t\} \sim \text{IID}(\mathbf{0}, \Sigma), \tag{2.1}$$

where $K = \{k_1, \ldots, k_m\}$, and Σ is non-singular.

Remark 2.1.1 *Any process satisfying Condition 2 also satisfies Condition 1. To see this, we note first that the IID sequence* $\{\mathbf{Z}_t\}$ *is stationary and ergodic. Since* $\{\mathbf{X}_t\}$ *is a function of this sequence through the causality property, it is also stationary and ergodic. To show model (2.1) is of full rank, suppose the linear combination* $\boldsymbol{\alpha}_0'\mathbf{X}_t + \boldsymbol{\alpha}_1'\mathbf{X}_{t-1} + \ldots + \boldsymbol{\alpha}_l'\mathbf{X}_{t-l} = 0$ *a.s., with* $\boldsymbol{\alpha}_0 \neq \mathbf{0}$. *Taking the variance of both sides, and by the assumed causality, we have*

$$\begin{aligned} 0 &= Var(\boldsymbol{\alpha}_0'\mathbf{Z}_t + \boldsymbol{\alpha}_0'\Phi_K(k_1)\mathbf{X}_{t-k_1} + \ldots + \boldsymbol{\alpha}_1'\mathbf{X}_{t-1} + \ldots + \boldsymbol{\alpha}_l'\mathbf{X}_{t-l}) \\ &= Var(\boldsymbol{\alpha}_0'\mathbf{Z}_t) + Var(\boldsymbol{\alpha}_1'\mathbf{X}_{t-1} + \ldots + \boldsymbol{\alpha}_0'\Phi_K(k_1)\mathbf{X}_{t-k_1} + \ldots + \boldsymbol{\alpha}_l'\mathbf{X}_{t-l}) \\ &\geq Var(\boldsymbol{\alpha}_0'\mathbf{Z}_t) = \boldsymbol{\alpha}_0'\Sigma\boldsymbol{\alpha}_0. \end{aligned}$$

Therefore, we must have $\boldsymbol{\alpha}_0'\Sigma\boldsymbol{\alpha}_0 = 0$, *which by the positive definiteness of* Σ *implies* $\boldsymbol{\alpha}_0 = \mathbf{0}$. *This contradicts the initial hypothesis, and hence* $\{\mathbf{X}_t\}$ *must be of full rank.*

From Chapter 1, section 1.2, we know the best linear forecast of $\mathbf{X}_t$ on the lagged subset K is

$$\hat{\mathbf{X}}_t(K) = \sum_{i \in K} \Phi_K(i)\mathbf{X}_{t-i},$$

with mean squared error $U_K(\equiv \Sigma)$; and the best backward linear forecast on the lagged subset K^*

$$\hat{\mathbf{X}}_t^{(b)}(K^*) = \sum_{j \in K^*} \Psi_{K^*}(j)\mathbf{X}_{t+j},$$

with mean squared error V_{K^*}.

Before proceeding, let us introduce the following notation for this chapter:

- As in the previous chapter, we will need to distinguish between estimators of coefficients and MSE's obtained via different methods. Thus we will continue to top those obtained via YW with *hats* ($\hat{}$), and Burg with *tildes* ($\tilde{}$) (there should be no confusion with the usual estimators of covariances, $\hat{\Gamma}(h)$, since these are not algorithm specific). In addition, denote the LS estimators by topping them with *breves* ($\breve{}$).

- Define in block form the $(d \times dm)$ matrix of coefficient matrices,

$$\Phi_K := [\Phi_K(k_1), \Phi_K(k_2), \ldots, \Phi_K(k_{m-1}), \Phi_K(k_m)].$$

- $\boldsymbol{\alpha}_K$:=$vec(\Phi_K)$.

- Define in block form the $(d \times dm)$ matrix of process autocovariances,

$$\Gamma_K := [\Gamma(k_1), \Gamma(k_2), \ldots, \Gamma(k_{m-1}), \Gamma(k_m)].$$

2.2 The subset Yule-Walker estimator

From chapter 1, the Yule-Walker equations for the forward prediction problem are

$$\sum_{i \in K} \Phi_K(i)\Gamma(k-i) - \Gamma(k) = 0, \quad k \in K \tag{2.2}$$

$$\Gamma(0) - \sum_{i \in K} \Phi_K(i)\Gamma(i)' = U_K, \tag{2.3}$$

which can be written in compact block matrix form as

$$[I_d, -\Phi_K(k_1), -\Phi_K(k_2), \ldots, -\Phi_K(k_{m-1}), -\Phi_K(k_m)]\, R_K = [U_K, 0, \ldots, 0]\,, \qquad (2.4)$$

where

$$R_K = \begin{bmatrix} \Gamma(0) & \Gamma(k_1) & \cdots & \Gamma(k_{m-1}) & \Gamma(k_m) \\ \Gamma(k_1)' & \Gamma(0) & \cdots & \Gamma(k_{m-1}-k_1) & \Gamma(k_m-k_1) \\ \Gamma(k_2)' & \Gamma(k_2-k_1)' & \cdots & \Gamma(k_{m-1}-k_2) & \Gamma(k_m-k_2) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \Gamma(k_{m-1})' & \Gamma(k_{m-1}-k_1)' & \cdots & \Gamma(0) & \Gamma(k_m-k_{m-1}) \\ \Gamma(k_m)' & \Gamma(k_m-k_1)' & \cdots & \Gamma(k_m-k_{m-1})' & \Gamma(0) \end{bmatrix}$$

$$\stackrel{\text{def}}{=} \left[\begin{array}{c|cccc} \Gamma(0) & \Gamma(k_1) & \Gamma(k_2) & \cdots & \Gamma(k_{m-1}) \quad \Gamma(k_m) \\ \hline \Gamma(k_1)' & & & & \\ \Gamma(k_2)' & & & & \\ \vdots & & G_K & & \\ \Gamma(k_{m-1})' & & & & \\ \Gamma(k_m)' & & & & \end{array}\right], \quad \text{say.}$$

For the backward prediction problem, the Yule-Walker equations are

$$\sum_{j\in K^*} \Psi_{K^*}(j)\Gamma(j-k) - \Gamma(k)' = 0, \quad k \in K^* \qquad (2.5)$$

$$\Gamma(0) - \sum_{j\in K^*} \Psi_{K^*}(j)\Gamma(j) = V_{K^*}, \qquad (2.6)$$

which can also be written in block matrix form as

$$[-\Psi_{K^*}(k_m), -\Psi_{K^*}(k_m-k_1), \ldots, -\Psi_{K^*}(k_m-k_{m-1}), I_d]\, R_K = [0, \ldots, 0, V_{K^*}]\,.$$

We can express (2.2)-(2.3) in the reduced block matrix format of (2.4) as

$$\Phi_K G_K = \Gamma_K \quad \text{and} \quad U_K = \Gamma(0) - \Phi_K \Gamma_K'$$

Taking vecs of both sides of the first expression leads to,

$$\begin{aligned} vec(\Phi_K G_K) &= vec(\Gamma_K) \\ \Rightarrow (G_K' \otimes I_d) vec(\Phi_K) &= vec(\Gamma_K), \quad \text{by identity K-6 of chapter 1} \\ \Rightarrow vec(\Phi_K) &= (G_K \otimes I_d)^{-1} vec(\Gamma_K), \quad \text{noting the symmetry of } G_K \\ \Rightarrow \boldsymbol{\alpha}_K &= (G_K^{-1} \otimes I_d)\ vec(\Gamma_K), \quad \text{by K-8} \\ \Rightarrow \Phi_K &= \Gamma_K G_K^{-1}, \end{aligned}$$

Remark 2.2.1 *Since* $\mathbf{X}_t$ *is of full rank,* G_K *will be non-singular and the solution* Φ_K *unique.*

This leads to the YW estimates of Φ_K (in vec and unvec form) and U_K, respectively,

$$\hat{\boldsymbol{\alpha}}_K = (\hat{G}_K^{-1} \otimes I_d)\ vec(\hat{\Gamma}_K) \tag{2.7}$$

$$\hat{\Phi}_K = \hat{\Gamma}_K \hat{G}_K^{-1} \tag{2.8}$$

$$\hat{U}_K = \hat{\Gamma}(0) - \hat{\Phi}_K \hat{\Gamma}_K'. \tag{2.9}$$

The estimated YW model for (2.1) is therefore

$$\mathbf{X}_t = \sum_{i \in K} \hat{\Phi}_K(i) \mathbf{X}_{t-i} + \mathbf{Z}_t, \quad \{\mathbf{Z}_t\} \sim \text{IID}(\mathbf{0}, \hat{U}_K). \tag{2.10}$$

2.3 The subset Least Squares estimator

The ensuing derivation of the asymptotics of the LS estimator, closely parallels the argument presented in Lutkepohl (1993), section 3.2.1. Throughout this section, we will assume *Condition 2* holds.

We begin by noting that model (2.1) can be written as,

$$\mathbf{X}_t = [\Phi_K(k_1), \ldots, \Phi_K(k_m)] \begin{bmatrix} \mathbf{X}_{t-k_1} \\ \vdots \\ \mathbf{X}_{t-k_m} \end{bmatrix} + \mathbf{Z}_t, \quad \{\mathbf{Z}_t\} \sim \text{IID}(\mathbf{0}, \Sigma).$$

For the set of random vectors $\{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$ from this model, we can write the above concisely in block matrix form,

$$\underbrace{[\mathbf{X}_{k_m+1}, \ldots, \mathbf{X}_n]}_{Y(d\times(n-k_m))} = \underbrace{[\Phi_K(k_1), \ldots, \Phi_K(k_m)]}_{\Phi_K(d\times dm)} \underbrace{\left(\begin{bmatrix} \mathbf{X}_{k_m+1-k_1} \\ \vdots \\ \mathbf{X}_1 \end{bmatrix}, \ldots, \begin{bmatrix} \mathbf{X}_{n-k_1} \\ \vdots \\ \mathbf{X}_{n-k_m} \end{bmatrix} \right)}_{M_K(dm\times(n-k_m))} + \underbrace{[\mathbf{Z}_{k_m+1}, \ldots, \mathbf{Z}_n]}_{Z(d\times(n-k_m))},$$

which in the compact notation of the under-braces becomes,

$$Y = \Phi_K M_K + Z.$$

Defining $\mathbf{y} \equiv vec(Y)$ and $\mathbf{z} \equiv vec(Z)$, take vecs of both sides of the above equation to obtain

$$\begin{aligned} vec(Y) &= vec(\Phi_K M_K) + vec(Z) \\ \Rightarrow \mathbf{y} &= (M_K' \otimes I_d)\, \boldsymbol{\alpha}_K + \mathbf{z}. \end{aligned}$$

Letting Σ_Z denote the covariance matrix of $\mathbf{z}$, we see that

$$\begin{aligned} \Sigma_Z &= \mathbb{E}\left(\begin{bmatrix} \mathbf{Z}_{k_m+1} \\ \vdots \\ \mathbf{Z}_n \end{bmatrix} \left[\mathbf{Z}_{k_m+1}', \ldots, \mathbf{Z}_n'\right] \right) = \mathbb{E} \begin{bmatrix} \mathbf{Z}_{k_m+1}\mathbf{Z}_{k_m+1}' & & 0 \\ & \ddots & \\ 0 & & \mathbf{Z}_n\mathbf{Z}_n' \end{bmatrix} \\ &= \begin{bmatrix} \Sigma & & 0 \\ & \ddots & \\ 0 & & \Sigma \end{bmatrix} = I_{n-k_m} \otimes \Sigma. \end{aligned}$$

The LS estimator seeks to find the $\boldsymbol{\alpha}_K$ which minimizes the scalar expression

$$S(\boldsymbol{\alpha}_K) = \mathbf{z}'\Sigma_Z^{-1}\mathbf{z} = \mathbf{z}'\left(I_{n-k_m} \otimes \Sigma^{-1}\right)\mathbf{z} = \mathrm{Tr}\left[(Y - \Phi_K M_K)'\Sigma^{-1}(Y - \Phi_K M_K)\right].$$

Following the argument in Lutkepohl (1993), section 3.2.1, we are led to the normal equations

$$(M_K M_K' \otimes \Sigma^{-1})\breve{\boldsymbol{\alpha}}_K = (M_K \otimes \Sigma^{-1})\mathbf{y},$$

with solution

$$\breve{\boldsymbol{\alpha}}_K = \left((M_K M_K')^{-1} M_K \otimes I_d\right) \mathbf{y} \tag{2.11}$$

$$= \boldsymbol{\alpha}_K + \left((M_K M_K')^{-1} M_K \otimes I_d\right) \mathbf{z} \tag{2.12}$$

$$= vec\left(Y M_K'(M_K M_K')^{-1}\right). \tag{2.13}$$

Equation (2.13) implies that

$$\breve{\Phi}_K = Y M_K'(M_K M_K')^{-1} \tag{2.14}$$

$$= (\Phi_K M_K + Z) M_K'(M_K M_K')^{-1}$$

$$= \Phi_K + Z M_K'(M_K M_K')^{-1}. \tag{2.15}$$

2.4 The asymptotic distribution of the subset LS estimator

In this section we establish asymptotic normality for the distribution of the subset LS estimator, by extending the arguments given in Lutkepohl (1993), section 3.2.2, to the subset case. This result will later be extended to the YW and Burg estimators. We begin with the following lemma:

Lemma 2.4.1 *For the process* $\{\mathbf{X}_t\}$ *satisfying* Condition 2*,*

(a) $\frac{M_K M_K'}{n} \xrightarrow{p} G_K$.

(b) $\frac{1}{\sqrt{n}} vec\,(Z M_K') \xrightarrow{d} \mathrm{N}(\mathbf{0}, G_K \otimes \Sigma)$.

Proof

(a) By definition, the $(i,j)^{th}$, $j \geq i$, block entry of the symmetric matrix $\frac{M_K M_K'}{n}$, is

$$\frac{1}{n} \sum_{t=k_m+1}^{n} \mathbf{X}_{t-k_i} \mathbf{X}_{t-k_j}',$$

which can be written as,

$$\frac{1}{n}\sum_{t=1}^{n-(k_j-k_i)} \mathbf{X}_{t+k_j-k_i}\mathbf{X}_t' + o_p(1) = \hat{\Gamma}(k_j-k_i) + o_p(1).$$

From Brockwell and Davis (1991), theorem 11.2.1, and for any integer h, $\hat{\Gamma}(h) \xrightarrow{p} \Gamma(h)$, where convergence in probability of random matrices means convergence in probability of all components of the matrix, and therefore,

$$\frac{1}{n}\sum_{t=k_m+1}^{n} \mathbf{X}_{t-k_i}\mathbf{X}_{t-k_j}' \xrightarrow{p} \Gamma(k_j-k_i),$$

which is precisely the $(i,j)^{th},\ j \geq i$, block entry of the matrix G_K.

(b) We will use a martingale central limit theorem in conjunction with the Cramer-Wold device to establish this result. We begin by noting that the $(d \times dm)$ matrix ZM_K' is given by,

$$\begin{aligned} ZM_K' &= \left[\sum_{t=k_m+1}^{n} \mathbf{Z}_t\mathbf{X}_{t-k_1}', \ldots, \sum_{t=k_m+1}^{n} \mathbf{Z}_t\mathbf{X}_{t-k_m}'\right] \\ &= \mathbf{Z}_{k_m+1}\left[\mathbf{X}_{k_m+1-k_1}', \ldots, \mathbf{X}_1'\right] + \mathbf{Z}_{k_m+2}\left[\mathbf{X}_{k_m+2-k_1}', \ldots, \mathbf{X}_2'\right] \\ &\quad + \cdots + \underbrace{\mathbf{Z}_n}_{(d\times 1)}\underbrace{\left[\mathbf{X}_{n-k_1}', \ldots, \mathbf{X}_{n-k_m}'\right]}_{(1\times dm)}. \end{aligned}$$

Defining the vector of length d^2m,

$$\begin{aligned} \mathbf{U}_t &\equiv vec\left(\mathbf{Z}_t\left[\mathbf{X}_{t-k_1}', \ldots, \mathbf{X}_{t-k_m}'\right]\right) \\ &= \begin{bmatrix} \mathbf{X}_{t-k_1} \\ \vdots \\ \mathbf{X}_{t-k_m} \end{bmatrix} \otimes \mathbf{Z}_t, \quad \text{by K-10}, \\ \Longrightarrow vec\left(ZM_K'\right) &= \sum_{t=k_m+1}^{n} \mathbf{U}_t = \sum_{t=1}^{n} \mathbf{U}_t + O_p(1). \end{aligned}$$

Then for any $\boldsymbol{\lambda} \in \mathbb{R}^{d^2m}$, we have, defining the scalar $W_{n,t} \equiv \frac{1}{\sqrt{n}}\,\boldsymbol{\lambda}'\,\mathbf{U}_t$,

$$\begin{aligned} \frac{1}{\sqrt{n}}\,\boldsymbol{\lambda}'\,vec\left(ZM_K'\right) &= \sum_{t=1}^{n} \frac{1}{\sqrt{n}}\,\boldsymbol{\lambda}'\,\mathbf{U}_t + O_p(1/\sqrt{n}) \\ &= \sum_{t=1}^{n} W_{n,t} + O_p(1/\sqrt{n}). \end{aligned}$$

Letting $\mathbf{X}_t = \sum_{j=0}^{\infty} \Upsilon_j \mathbf{Z}_{t-j}$ be the causal representation of $\mathbf{X}_t$, we see that $\mathbf{Z}_t$ is independent of $\{\mathbf{X}_{t-k_1}, \ldots, \mathbf{X}_{t-k_m}\}$, and therefore $\mathbb{E}(\mathbf{U}_t) = \mathbf{0}$. Defining $\mathcal{F}_t$ to be the sigma-field generated by $\{\mathbf{Z}_k : k \leq t\}$, i.e.

$$\mathcal{F}_t = \sigma\left(\mathbf{Z}_k : k \leq t\right),$$

it follows immediately that $\{W_{n,t}\},\ t = 1, \ldots, n$, is a martingale difference sequence, and hence uncorrelated. That is,

$$\begin{aligned}
\mathbb{E}\left(W_{n,t} \mid \mathcal{F}_{t-1}\right) &= \mathbb{E}\left(\frac{1}{\sqrt{n}}\, \boldsymbol{\lambda}' \begin{bmatrix} \mathbf{X}_{t-k_1} \\ \vdots \\ \mathbf{X}_{t-k_m} \end{bmatrix} \otimes \mathbf{Z}_t \mid \mathcal{F}_{t-1}\right) \\
&= \frac{1}{\sqrt{n}}\, \boldsymbol{\lambda}' \left(\mathbb{E} \begin{bmatrix} \mathbf{X}_{t-k_1} \\ \vdots \\ \mathbf{X}_{t-k_m} \end{bmatrix} \otimes \underbrace{\mathbb{E}\, \mathbf{Z}_t}_{\mathbf{0}}\right) \\
&= 0.
\end{aligned}$$

The sequence $\{\mathbf{U}_t\}$ is therefore also uncorrelated.

In the above calculation, we used the fact that if X and Y are independent random matrices,

$$\mathbb{E}(X \otimes Y) = \mathbb{E}(X) \otimes \mathbb{E}(Y).$$

This is a simple property of the Kronecker Product operation, since, by independence,

$$\begin{aligned}
X \otimes Y &= \left[X_{ij} Y\right]_{(i,j)=1}^{d} \\
\Longrightarrow \mathbb{E}\left(X \otimes Y\right) &= \left[\mathbb{E}\left(X_{ij} Y\right)\right]_{(i,j)=1}^{d} = \left[\mathbb{E}(X_{ij})\mathbb{E}(Y)\right]_{(i,j)=1}^{d} \\
&= \mathbb{E}(X) \otimes \mathbb{E}(Y).
\end{aligned}$$

Noting that the covariance matrix of $\mathbf{U}_t$ is,

$$\begin{aligned}
& \mathbb{E}\left(\mathbf{U}_t\mathbf{U}_t'\right) \\
= \; & \mathbb{E}\left(\begin{bmatrix} \mathbf{X}_{t-k_1} \\ \vdots \\ \mathbf{X}_{t-k_m} \end{bmatrix} \otimes \mathbf{Z}_t\right)\left(\left[\mathbf{X}_{t-k_1}', \ldots, \mathbf{X}_{t-k_m}'\right] \otimes \mathbf{Z}_t'\right), \quad \text{by K-5} \\
= \; & \mathbb{E}\left(\begin{bmatrix} \mathbf{X}_{t-k_1} \\ \vdots \\ \mathbf{X}_{t-k_m} \end{bmatrix}\left[\mathbf{X}_{t-k_1}', \ldots, \mathbf{X}_{t-k_m}'\right] \otimes \mathbf{Z}_t\mathbf{Z}_t'\right), \quad \text{by K-9} \\
= \; & \mathbb{E}\left(\begin{bmatrix} \mathbf{X}_{t-k_1}\mathbf{X}_{t-k_1}' & \cdots & \mathbf{X}_{t-k_1}\mathbf{X}_{t-k_m}' \\ \vdots & \ddots & \vdots \\ \mathbf{X}_{t-k_m}\mathbf{X}_{t-k_1}' & \cdots & \mathbf{X}_{t-k_m}\mathbf{X}_{t-k_m}' \end{bmatrix} \otimes \mathbf{Z}_t\mathbf{Z}_t'\right) \\
= \; & \mathbb{E}\left(\begin{bmatrix} \mathbf{X}_{t-k_1}\mathbf{X}_{t-k_1}' & \cdots & \mathbf{X}_{t-k_1}\mathbf{X}_{t-k_m}' \\ \vdots & \ddots & \vdots \\ \mathbf{X}_{t-k_m}\mathbf{X}_{t-k_1}' & \cdots & \mathbf{X}_{t-k_m}\mathbf{X}_{t-k_m}' \end{bmatrix}\right) \otimes \mathbb{E}\left(\mathbf{Z}_t\mathbf{Z}_t'\right) \\
= \; & G_K \otimes \Sigma,
\end{aligned}$$

we have,

$$\mathrm{Var}(W_{n,t}) = \frac{1}{n}\boldsymbol{\lambda}'\mathbb{E}\left(\mathbf{U}_t\mathbf{U}_t'\right)\boldsymbol{\lambda} = \frac{1}{n}\boldsymbol{\lambda}'\left(G_K \otimes \Sigma\right)\boldsymbol{\lambda}. \tag{2.16}$$

The sequence $\{W_{n,t}\}$ is in the prerequisite form for application of a martingale central limit theorem. In this context, we use theorem 3.2 and corollary 3.1 of Hall and Heyde (1980). Accordingly, we need only check the following three conditions:

(i) $\sum_{t=1}^{n}\mathbb{E}\left[W_{n,t}^2 \mid \mathcal{F}_{t-1}\right] \xrightarrow{p} \boldsymbol{\lambda}'\left(G_K \otimes \Sigma\right)\boldsymbol{\lambda}$.

(ii) $\max_{1\le t\le n}|W_{n,t}| \xrightarrow{p} 0$.

(iii) $\mathbb{E}\left(\max_{1\le t\le n} W_{n,t}^2\right)$ is bounded in n.

Proof of (i):

Since $\{\mathbb{E}\left(\boldsymbol{\lambda}'\mathbf{U}_t\mathbf{U}_t'\boldsymbol{\lambda} \mid \mathcal{F}_{t-1}\right)\}$ is a stationary ergodic sequence, the ergodic theorem implies,

$$\begin{aligned}
\sum_{t=1}^{n}\mathbb{E}\left[W_{n,t}^2 \mid \mathcal{F}_{t-1}\right] &= \frac{1}{n}\sum_{t=1}^{n}\mathbb{E}\left(\boldsymbol{\lambda}'\mathbf{U}_t\mathbf{U}_t'\boldsymbol{\lambda} \mid \mathcal{F}_{t-1}\right) \\
&\xrightarrow{p} \mathbb{E}\left[\mathbb{E}\left(\boldsymbol{\lambda}'\mathbf{U}_t\mathbf{U}_t'\boldsymbol{\lambda} \mid \mathcal{F}_{t-1}\right)\right] \\
&= \boldsymbol{\lambda}'\mathbb{E}\left(\mathbf{U}_t\mathbf{U}_t'\right)\boldsymbol{\lambda} \\
&= \boldsymbol{\lambda}'\left(G_K \otimes \Sigma\right)\boldsymbol{\lambda}.
\end{aligned}$$

Proof of (ii):

We have,

$$\begin{aligned}
\Pr\left(\max_{1\le t\le n}\left|\boldsymbol{\lambda}'\mathbf{U}_t\right|/\sqrt{n} > \epsilon\right) &= \Pr\left(\max_{1\le t\le n}\left|\boldsymbol{\lambda}'\mathbf{U}_t\right| > \epsilon\sqrt{n}\right) \\
&= \Pr\left(\bigcup_{t=1}^{n}\left\{\left|\boldsymbol{\lambda}'\mathbf{U}_t\right| > \epsilon\sqrt{n}\right\}\right) \\
&\le \sum_{t=1}^{n}\Pr\left(\left|\boldsymbol{\lambda}'\mathbf{U}_t\right| > \epsilon\sqrt{n}\right) \\
&= n\ \Pr\left(\left|\boldsymbol{\lambda}'\mathbf{U}_1\right| > \epsilon\sqrt{n}\right) \\
&= n\,\mathbb{E}\left[\mathrm{I}_{\left\{\left|\boldsymbol{\lambda}'\mathbf{U}_1\right| > \epsilon\sqrt{n}\right\}}\right] \\
&\le \frac{n}{n\epsilon^2}\mathbb{E}\left[\left|\boldsymbol{\lambda}'\mathbf{U}_1\right|^2\mathrm{I}_{\left\{\boldsymbol{\lambda}'\mathbf{U}_1\right| > \epsilon\sqrt{n}\}}\right] \\
&\xrightarrow{p} 0,
\end{aligned}$$

by the finite variance of $\{\mathbf{X}_t\}$.

Proof of (iii):

Since $\{W_{n,t}\}$ is identically distributed,

$$\begin{aligned}
\mathbb{E}\left(\max_{1\le t\le n}\left|W_{n,t}\right|^2\right) &= \frac{1}{n}\mathbb{E}\left(\max_{1\le t\le n}\left|\boldsymbol{\lambda}'\mathbf{U}_t\right|^2\right) \\
&\le \frac{1}{n}\sum_{t=1}^{n}\mathbb{E}\left|\boldsymbol{\lambda}'\mathbf{U}_t\right|^2 \\
&= \mathbb{E}\left|\boldsymbol{\lambda}'\mathbf{U}_1\right|^2.
\end{aligned}$$

Therefore, invoking theorem 3.2 and corollary 3.1 of Hall and Heyde (1980), we have

$$\frac{1}{\sqrt{n}}\boldsymbol{\lambda}' vec\left(ZM_K'\right) \xrightarrow{d} \mathrm{N}\left(0, \boldsymbol{\lambda}'\left(G_K \otimes \Sigma\right)\boldsymbol{\lambda}\right).$$

Finally, since $\boldsymbol{\lambda}$ was arbitrarily chosen from $\mathbb{R}^{d^2m}$, application of the Cramer-Wold device (Brockwell and Davis (1991), proposition 6.3.1), gives

$$\frac{1}{\sqrt{n}} vec\left(ZM_K'\right) \xrightarrow{d} \mathrm{N}(\mathbf{0}, G_K \otimes \Sigma).$$

□

The following theorem establishes the weak consistency and asymptotic normality of the subset LS estimator.

Theorem 2.4.1

(Consistency and Central Limit Theorem for the subset LS estimator)

The LS estimators of the coefficients in the SVAR model (2.1), satisfy

(a)

$$\breve{\Phi}_K \xrightarrow{p} \Phi_K.$$

(b)

$$\sqrt{n}\left(\breve{\boldsymbol{\alpha}}_K - \boldsymbol{\alpha}_K\right) \xrightarrow{d} \mathrm{N}(\mathbf{0}, G_K^{-1} \otimes \Sigma). \tag{2.17}$$

Proof

(a) From (2.15) we have,

$$\begin{aligned} \breve{\Phi}_K - \Phi_K &= ZM_K'(M_K M_K')^{-1} \\ &= \frac{ZM_K'}{n}\left(\frac{M_K M_K'}{n}\right)^{-1} \end{aligned}$$

By lemma 2.4.1, part (a), the term in brackets converges in probability to a nonsingular quantity; while part (b) $\Rightarrow \frac{ZM_K'}{n} \xrightarrow{p} 0$. Therefore, $\breve{\Phi}_K - \Phi_K \xrightarrow{p} 0$.

(b) From (2.12), lemma 2.4.1[part (a)], and the continuous mapping theorem,

$$\begin{aligned}\sqrt{n}\left(\breve{\boldsymbol{\alpha}}_K - \boldsymbol{\alpha}_K\right) &= \sqrt{n}\left((M_K M_K')^{-1} M_K \otimes I_d\right)\mathbf{z} \\ &= \left(\left(\frac{M_K M_K'}{n}\right)^{-1} \otimes I_d\right)\frac{1}{\sqrt{n}}\left(M_K \otimes I_d\right)\mathbf{z} \\ &\xrightarrow{d} \left(G_K^{-1} \otimes I_d\right)\,\mathcal{N},\end{aligned}$$

where $\mathcal{N} \sim \mathrm{N}(\mathbf{0}, G_K \otimes \Sigma)$, since lemma 2.4.1[part (b)] implies that

$$\frac{1}{\sqrt{n}}\left(M_K \otimes I_d\right)\mathbf{z} = \frac{1}{\sqrt{n}} vec\left(ZM_K'\right) \xrightarrow{d} \mathrm{N}(\mathbf{0}, G_K \otimes \Sigma).$$

Therefore, and by successive applications of identity K-9 in appendix A.2,

$$\begin{aligned}\sqrt{n}\left(\breve{\boldsymbol{\alpha}}_K - \boldsymbol{\alpha}_K\right) &\xrightarrow{d} \mathrm{N}\left(\mathbf{0}, \left(G_K^{-1} \otimes I_d\right)\left(G_K \otimes \Sigma\right)\left(G_K^{-1} \otimes I_d\right)'\right) \\ &= \mathrm{N}\left(\mathbf{0}, \left(G_K^{-1} \otimes \Sigma\right)\right).\end{aligned}$$

□

2.5 The asymptotic distribution of the subset YW estimator

In this section we establish analogous results of asymptotic normality for the subset YW estimators. The results and respective proofs in this section, are an extension of Brockwell and Davis (1991), theorem 8.1.1, to the multivariate subset case. We begin with the following lemma:

Lemma 2.5.1 *For the process* $\{\mathbf{X}_t\}$ *satisfying* Condition 2*,*

(a) $\sqrt{n}\left[\hat{\Gamma}_K - \frac{YM_K'}{n}\right] \xrightarrow{p} 0$*, and,* $\frac{YM_K'}{n} \xrightarrow{p} \Gamma_K$.

(b) $\sqrt{n}\left[\hat{G}_K^{-1} - \left(\frac{M_K M_K'}{n}\right)^{-1}\right] \xrightarrow{p} 0.$

Proof

(a) Now,

$$\begin{aligned} YM_K' &= [\mathbf{X}_{k_m+1}, \ldots, \mathbf{X}_n] \begin{bmatrix} \mathbf{X}'_{k_m+1-k_1} & \cdots & \mathbf{X}'_1 \\ \vdots & \ddots & \vdots \\ \mathbf{X}'_{n-k_1} & \cdots & \mathbf{X}'_{n-k_m} \end{bmatrix} \\ &= \left[\sum_{t=k_m+1}^{n} \mathbf{X}_t \mathbf{X}'_{t-k_1}, \ldots, \sum_{t=k_m+1}^{n} \mathbf{X}_t \mathbf{X}'_{t-k_m}\right], \end{aligned}$$

and,

$$\hat{\Gamma}_K = [\hat{\Gamma}(k_1), \ldots, \hat{\Gamma}(k_m)] = \left[\frac{1}{n}\sum_{t=1}^{n-k_1} \mathbf{X}_{t+k_1}\mathbf{X}'_t, \ldots, \frac{1}{n}\sum_{t=1}^{n-k_m} \mathbf{X}_{t+k_m}\mathbf{X}'_t\right],$$

so that the j^{th}, $1 \le j \le m$, block matrix entry of $\sqrt{n}\left[\hat{\Gamma}_K - \frac{YM_K'}{n}\right]$ has the form

$$\begin{aligned} &\frac{1}{\sqrt{n}}\left[\sum_{t=1}^{n-k_j} \mathbf{X}_{t+k_j}\mathbf{X}'_t - \sum_{t=k_m+1}^{n} \mathbf{X}_t \mathbf{X}'_{t-k_j}\right] \\ = &\frac{1}{\sqrt{n}}\left[\sum_{t=1}^{n-k_j} \mathbf{X}_{t+k_j}\mathbf{X}'_t - \sum_{t=k_m+1-k_j}^{n-k_j} \mathbf{X}_{t+k_j}\mathbf{X}'_t\right] \\ = &\frac{1}{\sqrt{n}}\sum_{t=1}^{k_m-k_j} \mathbf{X}_{t+k_j}\mathbf{X}'_t \xrightarrow{p} 0, \end{aligned}$$

by re-indexing the right summand.

Similarly, the j^{th}, block matrix entry of $\left[\hat{\Gamma}_K - \frac{YM_K'}{n}\right]$ is

$$\frac{1}{n}\sum_{t=1}^{k_m-k_j} \mathbf{X}_{t+k_j}\mathbf{X}_t \xrightarrow{p} 0, \quad \text{as } n \to \infty.$$

(b) From Brockwell and Davis (1991), proposition 6.1.2, it suffices to show that

$$\left\| \sqrt{n}\ \left[\hat{G}_K^{-1} - \left(\frac{M_K M_K'}{n}\right)^{-1}\right] \right\|_2 \xrightarrow{p} 0,$$

where for matrix A, $\|A\|_2$ denotes the Euclidean norm of $vec(A)$:

$$\begin{aligned}
&\sqrt{n}\ \left\|\hat{G}_K^{-1} - \left(\frac{M_K M_K'}{n}\right)^{-1}\right\|_2 \\
&= \left\|\hat{G}_K^{-1}\sqrt{n}\ \left(\frac{M_K M_K'}{n} - \hat{G}_K\right)\left(\frac{M_K M_K'}{n}\right)^{-1}\right\|_2 \\
&\leq \left\|\hat{G}_K^{-1}\right\|_2 . \left\|\sqrt{n}\ \left(\frac{M_K M_K'}{n} - \hat{G}_K\right)\right\|_2 . \left\|\left(\frac{M_K M_K'}{n}\right)^{-1}\right\|_2,
\end{aligned}$$

the inequality following from Cauchy-Schwarz for matrix norms (see for example Lutkepohl (1996)[p. 111]). Now, since $\hat{\Gamma}(h) \xrightarrow{p} \Gamma(h)$ for any integer h, we have by the continuous mapping theorem that $\hat{G}_K^{-1} \xrightarrow{p} G_K^{-1}$. Also, by lemma 2.4.1, part (a), and again using the continuous mapping theorem, $\left(\frac{M_K M_K'}{n}\right)^{-1} \xrightarrow{p} G_K^{-1}$. Finally, employing a similar argument to the proof of part (a), the $(i,j)^{\text{th}},\ 1 \leq i \leq j \leq m$, block entry of $\sqrt{n}\ \left(\frac{M_K M_K'}{n} - \hat{G}_K\right)$ can be written as,

$$\frac{1}{\sqrt{n}}\left[\sum_{t=1}^{n-(k_j-k_i)} \mathbf{X}_{t+k_j-k_i}\mathbf{X}_t' - \sum_{t=1}^{n-(k_j-k_i)} \mathbf{X}_{t+k_j-k_i}\mathbf{X}_t'\right] + o_p(1) \xrightarrow{p} 0, \quad \text{as } n \to \infty.$$

Thus, the right hand side of the above norm inequality

$$\xrightarrow{p} \|G_K^{-1}\|_2 . \|0\|_2 . \|G_K^{-1}\|_2 = 0,$$

$$\implies \sqrt{n}\ \left\|\hat{G}_K^{-1} - \left(\frac{M_K M_K'}{n}\right)^{-1}\right\|_2 \xrightarrow{p} 0.$$

□

We are now ready for the main result in this section:

Theorem 2.5.1

(Consistency and Central Limit Theorem for the subset YW estimator)

The YW estimators of the coefficients of SVAR model (2.1), satisfy

(a) $\hat{\Phi}_K \xrightarrow{p} \Phi_K$.

(b) $\sqrt{n}\,(\hat{\boldsymbol{\alpha}}_K - \boldsymbol{\alpha}_K) \xrightarrow{d} \mathrm{N}(\mathbf{0}, G_K^{-1} \otimes \Sigma)$.

(c) $\hat{U}_K \xrightarrow{p} \Sigma$.

Proof

(a) From (2.8), $\hat{\Phi}_K = \hat{\Gamma}_K \hat{G}_K^{-1}$. Since $\hat{\Gamma}(h) \xrightarrow{p} \Gamma(h)$, for every integer h, $\hat{\Gamma}_K \xrightarrow{p} \Gamma_K$, and $\hat{G}_K \xrightarrow{p} G_K$. Therefore, by the continuous mapping theorem, $\hat{\Phi}_K \xrightarrow{p} \Gamma_K G_K^{-1} \equiv \Phi_K$.

(b) The subset LS and YW estimators are respectively:

$$\breve{\boldsymbol{\alpha}}_K = \left((M_K M_K')^{-1} M_K \otimes I_d\right) \mathbf{y}, \quad \text{and} \quad \hat{\boldsymbol{\alpha}}_K = (\hat{G}_K^{-1} \otimes I_d)\, vec(\hat{\Gamma}_K).$$

In light of theorem 2.4.1 part (b), and the fact that

$$\sqrt{n}(\hat{\boldsymbol{\alpha}}_K - \breve{\boldsymbol{\alpha}}_K) = \left[\sqrt{n}(\hat{\boldsymbol{\alpha}}_K - \boldsymbol{\alpha}_K) - \sqrt{n}(\breve{\boldsymbol{\alpha}}_K - \boldsymbol{\alpha}_K)\right],$$

it suffices to show, by Brockwell and Davis (1991) proposition 6.3.3 for example, that $\sqrt{n}(\hat{\boldsymbol{\alpha}}_K - \breve{\boldsymbol{\alpha}}_K) \xrightarrow{p} 0$. Thus, from (2.8) and (2.13),

$$\begin{aligned}
\sqrt{n}(\hat{\Phi}_K - \breve{\Phi}_K) &= \sqrt{n}\left[\hat{\Gamma}_K \hat{G}_K^{-1} - Y M_K'(M_K M_K')^{-1}\right] \\
&= \sqrt{n}\left[\hat{\Gamma}_K - \frac{Y M_K'}{n}\right] \hat{G}_K^{-1} + \left(\frac{Y M_K'}{n}\right) \sqrt{n}\left[\hat{G}_K^{-1} - \left(\frac{M_K M_K'}{n}\right)^{-1}\right] \\
&\xrightarrow{p} 0, \quad \text{by lemma 2.5.1.}
\end{aligned}$$

(c) From (2.9), $\hat{U}_K = \hat{\Gamma}(0) - \hat{\Phi}_K \hat{\Gamma}_K'$. From part (a), $\hat{\Gamma}(0) \xrightarrow{p} \Gamma(0)$, $\hat{\Phi}_K \xrightarrow{p} \Phi_K$, and $\hat{\Gamma}_K \xrightarrow{p} \Gamma_K$; so that by the continuous mapping theorem,

$$\hat{U}_K \xrightarrow{p} \Gamma(0) - \Phi_K \Gamma_K' \equiv \Sigma.$$

□

2.6 The asymptotic distribution of the subset Burg estimator

In this section, we prove that the multivariate subset Burg estimator has the same asymptotic distribution as the YW estimator. Our strategy is to show that the two estimators differ by terms of order at most $O_p(1/n)$ (which in particular implies a difference of order $o_p(1/\sqrt{n})$). Applying Brockwell and Davis (1991) proposition 6.3.3, then gives convergence in distribution to the same limiting random vector. The arguments that follow are a generalization to the subset case of the results presented in Hainz (1994). We begin with a lemma:

Lemma 2.6.1 *Let $\{X_n\}$ be a tight sequence of invertible $(d \times d)$ random matrices, and A a constant invertible $(d \times d)$ matrix. Then*

$$X_n = A + O_p(1/n) \implies X_n^{-1} = A^{-1} + O_p(1/n).$$

Proof

For any invertible matrix B, let $g_k(.)$ be the continuous differentiable mapping from $\mathbb{R}^{d^2} \longrightarrow \mathbb{R}$, that takes the k^{th} element of $vec(B)$ to the k^{th} element of $vec(B^{-1})$, i.e.

$$g_k(vec(B)) = \left[vec(B^{-1})\right]_k .$$

Then, applying the random vector version of Fuller (1996) corollary 5.1.5, with $s = 1$ and $r_n = \frac{1}{n}$ to $vec(X_n)$, gives

$$\left[vec(X_n^{-1})\right]_k \equiv g_k(vec(X_n)) = g_k(vec(A)) + O_p(1/n) \equiv \left[vec(A^{-1})\right]_k + O_p(1/n).$$

Applying this component-wise with $k = \{1, \ldots, d^2\}$ in turn, gives the required result.

□

Theorem 2.6.1 *If* $\{\mathbf{X}_t\}$ *satisfies* Condition 1*, then the Burg and YW estimators of the coefficients and MSE's of the forward and backward subset prediction problems, satisfy:*

(a) $\tilde{\hat{\Phi}}_K = \hat{\Phi}_K + O_p(1/n)$.

(b) $\tilde{\hat{\Psi}}_{K^*} = \hat{\Psi}_{K^*} + O_p(1/n)$.

(c) $\tilde{U}_K = \hat{U}_K + O_p(1/n)$.

(d) $\tilde{V}_{K^*} = \hat{V}_{K^*} + O_p(1/n)$.

In addition, we have the following auxiliary relationships:

(e)

$$\frac{1}{n}\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\varepsilon}}_J(t)' = \tilde{U}_J + O_p(1/n).$$

(f)

$$\frac{1}{n}\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)' = \tilde{V}_{J^*} + O_p(1/n).$$

(g)

$$\frac{1}{n}\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)' = \hat{\Phi}_K(k_m)\hat{V}_{J^*} + O_p(1/n).$$

Proof

We will use induction on the size of the set K of the subset prediction problem. We begin at level 1 ($m = 1$) with K consisting of a single positive integer, k_m. Note that $J = \emptyset = J^*$. Now from the YW algorithm,

$$\begin{aligned} \hat{\Phi}_K(k_m) &= \hat{\Gamma}(k_m)\hat{\Gamma}(0)^{-1} \\ \Longrightarrow vec\left(\hat{\Phi}_K(k_m)\right) &= \left[\hat{\Gamma}(0)^{-1} \otimes I_d\right] vec\left(\hat{\Gamma}(k_m)\right). \end{aligned} \tag{2.18}$$

From the Burg algorithm and (1.29), we have

$$
\begin{aligned}
& vec\left(\tilde{\Phi}_K(k_m)\right) \\
&= \left[\left(\sum_{t=1+k_m}^{n} \mathbf{X}_{t-k_m}\mathbf{X}'_{t-k_m}\right) \otimes I_d + \hat{\Gamma}(0)^2 \otimes \hat{\Gamma}(0)^{-1} \left(\sum_{t=1+k_m}^{n} \mathbf{X}_t\mathbf{X}'_t\right) \hat{\Gamma}(0)^{-1}\right]^{-1} \\
& \quad vec\left[\left(\sum_{t=1+k_m}^{n} \mathbf{X}_t\mathbf{X}'_{t-k_m}\right) + \hat{\Gamma}(0)^{-1} \left(\sum_{t=1+k_m}^{n} \mathbf{X}_t\mathbf{X}'_{t-k_m}\right) \hat{\Gamma}(0)\right].
\end{aligned}
$$

Multiplying and dividing by $\frac{1}{n}$ gives

$$
\begin{aligned}
& vec\left(\tilde{\Phi}_K(k_m)\right) = \\
& \left[\left(\frac{1}{n}\sum_{t=1+k_m}^{n} \mathbf{X}_{t-k_m}\mathbf{X}'_{t-k_m}\right) \otimes I_d + \hat{\Gamma}(0)^2 \otimes \hat{\Gamma}(0)^{-1} \left(\frac{1}{n}\sum_{t=1+k_m}^{n} \mathbf{X}_t\mathbf{X}'_t\right) \hat{\Gamma}(0)^{-1}\right]^{-1} \\
& vec\left[\left(\frac{1}{n}\sum_{t=1+k_m}^{n} \mathbf{X}_t\mathbf{X}'_{t-k_m}\right) + \hat{\Gamma}(0)^{-1} \left(\frac{1}{n}\sum_{t=1+k_m}^{n} \mathbf{X}_t\mathbf{X}'_{t-k_m}\right) \hat{\Gamma}(0)\right],
\end{aligned}
$$

and upon recalling that for $h \geq 0$, $\hat{\Gamma}(h) = \frac{1}{n}\sum_{t=1}^{n-h} \mathbf{X}_{t+h}\mathbf{X}'_t$, we see that

$$
\begin{aligned}
& vec\left(\tilde{\Phi}_K(k_m)\right) \\
&= \left[\left(\hat{\Gamma}(0) + O_p(1/n)\right) \otimes I_d + \hat{\Gamma}(0)^2 \otimes \hat{\Gamma}(0)^{-1} \left(\hat{\Gamma}(0) + O_p(1/n)\right) \hat{\Gamma}(0)^{-1}\right]^{-1} \\
& \quad vec\left[\hat{\Gamma}(k_m) + \hat{\Gamma}(0)^{-1}\hat{\Gamma}(k_m)\hat{\Gamma}(0)\right] \\
&= \left[\hat{\Gamma}(0) \otimes I_d + \hat{\Gamma}(0)^2 \otimes \hat{\Gamma}(0)^{-1} + O_p(1/n)\right]^{-1} \left[I_{d^2} + \hat{\Gamma}(0) \otimes \hat{\Gamma}(0)^{-1}\right] \\
& \quad vec\left(\hat{\Gamma}(k_m)\right), \qquad (2.19)
\end{aligned}
$$

using K-1 and K-6 of appendix A.2 on the *vec* term. Consider now the inverse term in the above equation. From lemma 2.6.1,

$$
\begin{aligned}
& \left[\hat{\Gamma}(0) \otimes I_d + \hat{\Gamma}(0)^2 \otimes \hat{\Gamma}(0)^{-1} + O_p(1/n)\right]^{-1} \\
&= \left[\hat{\Gamma}(0) \otimes I_d + \hat{\Gamma}(0)^2 \otimes \hat{\Gamma}(0)^{-1}\right]^{-1} + O_p(1/n).
\end{aligned}
$$

Applying identities K-9 and K-11 leads to the factorization

$$\begin{aligned}
&\left[\hat{\Gamma}(0)\otimes I_d + \hat{\Gamma}(0)^2\otimes\hat{\Gamma}(0)^{-1}\right]^{-1} + O_p(1/n)\\
&= \left[\left(I_d\otimes I_d + \hat{\Gamma}(0)\otimes\hat{\Gamma}(0)^{-1}\right)\left(\hat{\Gamma}(0)\otimes I_d\right)\right]^{-1} + O_p(1/n)\\
&= \left[\hat{\Gamma}(0)\otimes I_d\right]^{-1}\left[I_{d^2} + \hat{\Gamma}(0)\otimes\hat{\Gamma}(0)^{-1}\right]^{-1} + O_p(1/n),
\end{aligned}$$

where we can easily see that $I_d \otimes I_d$ coalesces into I_{d^2}. Finally, applying identity K-8 to the first inverse gives

$$\begin{aligned}
&\left[\hat{\Gamma}(0)\otimes I_d + \hat{\Gamma}(0)^2\otimes\hat{\Gamma}(0)^{-1}\right]^{-1} + O_p(1/n)\\
&= \left[\hat{\Gamma}(0)^{-1}\otimes I_d\right]\left[I_{d^2} + \hat{\Gamma}(0)\otimes\hat{\Gamma}(0)^{-1}\right]^{-1} + O_p(1/n).
\end{aligned}$$

We can now substitute the above into (2.19) to give,

$$\begin{aligned}
vec\left(\tilde{\Phi}_K(k_m)\right) &= \left[\hat{\Gamma}(0)\otimes I_d\right]^{-1}\left[I_{d^2} + \hat{\Gamma}(0)\otimes\hat{\Gamma}(0)^{-1}\right]^{-1}\left[I_{d^2} + \hat{\Gamma}(0)\otimes\hat{\Gamma}(0)^{-1}\right]\\
&\quad vec\left(\hat{\Gamma}(k_m)\right) + O_p(1/n)\\
&= \left[\hat{\Gamma}(0)\otimes I_d\right]^{-1} vec\left(\hat{\Gamma}(k_m)\right) + O_p(1/n)\\
&= vec\left(\hat{\Phi}_K(k_m)\right) + O_p(1/n), \quad \text{from (2.18)}.
\end{aligned}$$

We therefore have that $\tilde{\Phi}_K(k_m) = \hat{\Phi}_K(k_m) + O_p(1/n)$, which implies $\tilde{\Phi}_K = \hat{\Phi}_K + O_p(1/n)$.

From the prediction error solution of the Yule-Walker equations (algorithm 1.4.1), $\tilde{\Phi}_K(k_m)$ and $\tilde{\Psi}_{K^*}(k_m)$ are linked via:

$$\tilde{\Psi}_{K^*}(k_m) = \tilde{V}_{J^*}\tilde{\Phi}_K(k_m)'\tilde{U}_J^{-1} \text{ which implies } \tilde{\Psi}_{K^*}(k_m)\tilde{U}_J = \tilde{V}_{J^*}\tilde{\Phi}_K(k_m)', \qquad (2.20)$$

(and similarly for the YW estimators), so that with $\tilde{V}_{J^*} = \Gamma(0) = \tilde{U}_J$,

$$\begin{aligned}
\tilde{\Psi}_{K^*}(k_m) &= \Gamma(0)\left[\hat{\Phi}_K(k_m) + O_p(1/n)\right]'\Gamma(0)^{-1}\\
&= \Gamma(0)\hat{\Phi}_K(k_m)'\Gamma(0)^{-1} + O_p(1/n)\\
&= \hat{\Psi}_{K^*}(k_m) + O_p(1/n).
\end{aligned}$$

Now for the MSE's, we have from algorithm 1.4.1 that

$$\begin{aligned}
\tilde{U}_K &= \tilde{U}_J - \tilde{\Phi}_K(k_m)\tilde{V}_{J^*}\tilde{\Phi}_K(k_m)' \\
&= \left[I_d - \tilde{\Phi}_K(k_m)\tilde{\Psi}_{K^*}(k_m)\right]\tilde{U}_J, \quad \text{from (2.20)}, \\
&= \left[I_d - \left(\hat{\Phi}_K(k_m)\hat{\Psi}_{K^*}(k_m) + O_p(1/n)\right)\right]\hat{\Gamma}(0) \\
&= \left[I_d - \hat{\Phi}_K(k_m)\hat{\Psi}_{K^*}(k_m)\right]\hat{\Gamma}(0) + O_p(1/n) \\
&= \hat{U}_K + O_p(1/n),
\end{aligned}$$

and similarly,

$$\begin{aligned}
\tilde{V}_{K^*} &= \tilde{V}_{J^*} - \tilde{\Psi}_{K^*}(k_m)\tilde{U}_J\tilde{\Psi}_{K^*}(k_m)' \\
&= \left[I_d - \tilde{\Psi}_{K^*}(k_m)\tilde{\Phi}_K(k_m)\right]\tilde{V}_{J^*}, \quad \text{from the transpose of (2.20)}, \\
&= \left[I_d - \left(\hat{\Psi}_{K^*}(k_m)\hat{\Phi}_K(k_m) + O_p(1/n)\right)\right]\hat{\Gamma}(0) \\
&= \left[I_d - \hat{\Psi}_{K^*}(k_m)\hat{\Phi}_K(k_m)\right]\hat{\Gamma}(0) + O_p(1/n) \\
&= \hat{V}_{K^*} + O_p(1/n).
\end{aligned}$$

For the auxiliary relationships, we have from algorithm 1.4.1

$$\begin{aligned}
\frac{1}{n}\sum_{t=k_m+1}^{n}\tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\varepsilon}}_J(t)' &= \frac{1}{n}\sum_{t=k_m+1}^{n}\mathbf{X}_t\mathbf{X}_t' \\
&= \hat{\Gamma}(0) + O_p(1/n) \\
&= \tilde{U}_J + O_p(1/n).
\end{aligned}$$

$$\begin{aligned}
\frac{1}{n}\sum_{t=k_m+1}^{n}\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)' &= \frac{1}{n}\sum_{t=k_m+1}^{n}\mathbf{X}_{t-k_m}\mathbf{X}_{t-k_m}' \\
&= \hat{\Gamma}(0) + O_p(1/n) \\
&= \tilde{V}_{J^*} + O_p(1/n).
\end{aligned}$$

$$\begin{aligned}
\frac{1}{n}\sum_{t=k_m+1}^{n}\tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)' &= \frac{1}{n}\sum_{t=k_m+1}^{n}\mathbf{X}_t\mathbf{X}_{t-k_m}' \\
&= \hat{\Gamma}(k_m) \\
&= \hat{\Phi}_K(k_m)\hat{V}_{J^*},
\end{aligned}$$

the last equality following from the first line of algorithm 1.3.1. Finally,

$$\begin{aligned}\frac{1}{n}\sum_{t=k_m+1}^{n}\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)\tilde{\boldsymbol{\varepsilon}}_J(t)' &= \frac{1}{n}\sum_{t=k_m+1}^{n}\mathbf{X}_{t-k_m}\mathbf{X}_t' \\ &= \hat{\Gamma}(-k_m) = \hat{\Gamma}(k_m)' \\ &= \hat{\Psi}_{K^*}(k_m)\hat{U}_J,\end{aligned}$$

where the last equality follows similarly from the third line of algorithm 1.3.1.

This ends the inductive step when K is any subset of size one. Now suppose the theorem holds for all subsets K of size $\leq m-1$, and consider $K = \{k_1, \ldots, k_m\}$. Recalling that $J = \{k_1, \ldots, k_{m-1}\}$ and $J^* = \{k_m - k_{m-1}, \ldots, k_m - k_1\}$, introduce the following additional notation for sets of lags G and H, where $H \in \{J, J^*, K^*\}$: $G(H)$ is obtained from G in the same manner that H is obtained from K. For example:

- $J(J)$ is to J what J is to K, i.e. $J(J) = \{k_1, \ldots, k_{m-2}\}$.
- $J(J^*)$ is to J what J^* is to K, i.e. $J(J^*) = \{k_{m-1} - k_{m-2}, \ldots, k_{m-1} - k_1\}$.
- $J^*(J)$ is to J^* what J is to K, i.e. $J^*(J) = \{k_m - k_{m-1}, \ldots, k_m - k_2\}$.
- $J^*(J^*)$ is to J^* what J^* is to K, i.e. $J^*(J^*) = \{k_2 - k_1, \ldots, k_{m-1} - k_1\}$.
- $J^*(K^*)$ is to J^* what K^* is to K, i.e. $J^*(K^*) = \{k_2 - k_1, \ldots, k_m - k_1\}$.

By the inductive hypothesis, we then have:

[H-1] $\tilde{\Phi}_J = \hat{\Phi}_J + O_p(1/n)$.

[H-2] $\tilde{\Psi}_{J^*} = \hat{\Psi}_{J^*} + O_p(1/n)$.

[H-3] $\tilde{U}_J = \hat{U}_J + O_p(1/n)$.

[H-4] $\tilde{V}_{J^*} = \hat{V}_{J^*} + O_p(1/n)$.

[H-5] $\frac{1}{n}\sum_{t=k_{m-1}+1}^{n} \tilde{\boldsymbol{\varepsilon}}_{J(J)}(t)\tilde{\boldsymbol{\varepsilon}}_{J(J)}(t)' = \tilde{U}_{J(J)} + O_p(1/n)$.

[H-6] $\frac{1}{n}\sum_{t=k_{m-1}+1}^{n} \tilde{\boldsymbol{\eta}}_{J(J^*)}(t-k_{m-1})\tilde{\boldsymbol{\eta}}_{J(J^*)}(t-k_{m-1})' = \tilde{V}_{J(J^*)} + O_p(1/n)$.

[H-7] $\frac{1}{n}\sum_{t=k_{m-1}+1}^{n} \tilde{\boldsymbol{\varepsilon}}_{J(J)}(t)\tilde{\boldsymbol{\eta}}_{J(J^*)}(t-k_{m-1})' = \hat{\Phi}_J(k_{m-1})\hat{V}_{J(J^*)} + O_p(1/n)$.

[H-8]

$$\frac{1}{n}\sum_{t=k_m-k_1+1}^{n} \tilde{\boldsymbol{\varepsilon}}_{J^*(J^*)}(t)\tilde{\boldsymbol{\varepsilon}}_{J^*(J^*)}(t)' = \tilde{U}_{J^*(J^*)} + O_p(1/n),$$

which upon re-indexing

$$\Longrightarrow \quad \frac{1}{n}\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\varepsilon}}_{J^*(J^*)}(t-k_1)\tilde{\boldsymbol{\varepsilon}}_{J^*(J^*)}(t-k_1)' = \tilde{U}_{J^*(J^*)} + O_p(1/n).$$

[H-9]

$$\frac{1}{n}\sum_{t=k_m-k_1+1}^{n} \tilde{\boldsymbol{\eta}}_{J^*(J)}(t-k_m+k_1)\tilde{\boldsymbol{\eta}}_{J^*(J)}(t-k_m+k_1)' = \tilde{V}_{J^*(J)} + O_p(1/n),$$

which upon re-indexing

$$\Longrightarrow \frac{1}{n}\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\eta}}_{J^*(J)}(t-k_m)\tilde{\boldsymbol{\eta}}_{J^*(J)}(t-k_m)' = \tilde{V}_{J^*(J)} + O_p(1/n).$$

[H-10]

$$\begin{aligned} \frac{1}{n}\sum_{t=k_m-k_1+1}^{n} \quad & \tilde{\boldsymbol{\varepsilon}}_{J^*(J^*)}(t) \quad \tilde{\boldsymbol{\eta}}_{J^*(J)}(t-k_m+k_1)' \\ = \quad & \hat{\Phi}_{J^*(K^*)}(k_m-k_1)\hat{V}_{J^*(J)} + O_p(1/n), \end{aligned}$$

which upon re-indexing

$$\begin{aligned} \Longrightarrow \frac{1}{n}\sum_{t=k_m+1}^{n} \quad & \tilde{\boldsymbol{\varepsilon}}_{J^*(J^*)}(t-k_1) \quad \tilde{\boldsymbol{\eta}}_{J^*(J)}(t-k_m)' \\ = \quad & \hat{\Phi}_{J^*(K^*)}(k_m-k_1)\hat{V}_{J^*(J)} + O_p(1/n). \end{aligned}$$

[H-11] From (2.20), $\tilde{V}_{J(J^*)}\tilde{\Phi}_J(k_{m-1})' = \tilde{\Psi}_{J(K^*)}(k_{m-1})\tilde{U}_{J(J)}$. Also holds with for YW estimators i.e. can replace tildes with hats throughout.

[H-12] From (2.20), $\tilde{U}_J = \left[I_d - \tilde{\Phi}_J(k_{m-1})\tilde{\Psi}_{J(K^*)}(k_{m-1})\right]\tilde{U}_{J(J)}$.

[H-13] $\tilde{\boldsymbol{\varepsilon}}_J(t) = \tilde{\boldsymbol{\varepsilon}}_{J(J)}(t) - \tilde{\Phi}_J(k_{m-1})\tilde{\boldsymbol{\eta}}_{J(J^*)}(t - k_{m-1})$.

[H-14] $\tilde{\boldsymbol{\eta}}_{J^*}(t - k_m) = \tilde{\boldsymbol{\eta}}_{J^*(J)}(t - k_m) - \tilde{\Psi}_{J^*}(k_m - k_1)\tilde{\boldsymbol{\varepsilon}}_{J^*(J^*)}(t - k_1)$.

[H-15] From (2.20), $\tilde{\Psi}_{J^*}(k_m - k_1)\tilde{U}_{J^*(J^*)} = \tilde{V}_{J^*(J)}\tilde{\Phi}_{J^*(K^*)}(k_m - k_1)'$. Also holds with for YW estimators i.e. can replace tildes with hats throughout.

[H-16] Set $K = J$ and $k = k_m - j$, $j \in J^*$ ($\Rightarrow k \in J$), in the YW equation (2.2) to obtain:

$$\sum_{i \in J} \hat{\Phi}_J(i)\hat{\Gamma}(k_m - j - i) = \hat{\Gamma}(k_m - j), \quad \text{for every } j \in J^*.$$

[H-17] From (2.20), $\tilde{V}_{J^*} = \left[I_d - \tilde{\Psi}_{J^*}(k_m - k_1)\tilde{\Psi}_{J^*(K^*)}(k_m - k_1)\right]\tilde{V}_{J^*(J)}$.

It is easy to show that H-5 - H-10 hold when $m = 2$, i.e. $K = \{k_1, k_2\}$. We note that in this case, $J = \{k_1\}$, $J^* = \{k_2 - k_1\} = J^*(K^*)$, and $J(J) = J(J^*) = J^*(J) = J^*(J^*) = \emptyset$. Thus:

- For H-5,

$$\begin{aligned} \frac{1}{n}\sum_{t=k_{m-1}+1}^{n} \tilde{\boldsymbol{\varepsilon}}_{J(J)}(t)\tilde{\boldsymbol{\varepsilon}}_{J(J)}(t)' &= \frac{1}{n}\sum_{t=k_1+1}^{n} \mathbf{X}_t\mathbf{X}_t' = \hat{\Gamma}(0) + O_p(1/n) \\ &= \tilde{U}_{J(J)} + O_p(1/n), \end{aligned}$$

  and similarly for H-8.

- For H-6,

$$\begin{aligned}\frac{1}{n}\sum_{t=k_{m-1}+1}^{n}\tilde{\boldsymbol{\eta}}_{J(J^*)}(t-k_{m-1})\tilde{\boldsymbol{\eta}}_{J(J^*)}(t-k_{m-1})' &= \frac{1}{n}\sum_{t=k_1+1}^{n}\mathbf{X}_{t-k_1}\mathbf{X}'_{t-k_1}\\ &= \hat{\Gamma}(0)+O_p(1/n)\\ &= \tilde{V}_{J(J^*)}+O_p(1/n),\end{aligned}$$

  and similarly for H-9.

- For H-7,

$$\begin{aligned}\frac{1}{n}\sum_{t=k_{m-1}+1}^{n}\tilde{\boldsymbol{\varepsilon}}_{J(J)}(t)\;\;\tilde{\boldsymbol{\eta}}_{J(J^*)}(t-k_{m-1})' &= \frac{1}{n}\sum_{t=k_1+1}^{n}\mathbf{X}_t\mathbf{X}'_{t-k_1}=\hat{\Gamma}(k_1)\\ &= \hat{\Phi}_{k_1}(k_1)\hat{\Gamma}(0),\quad \text{from first line of algorithm 1.3.1}\\ &= \hat{\Phi}_J(k_{m-1})\hat{V}_{J(J^*)},\end{aligned}$$

  and similarly for H-10.

We will complete the inductive argument by showing in order:

(i)
$$\frac{1}{n}\sum_{t=k_m+1}^{n}\tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\varepsilon}}_J(t)'=\tilde{U}_J+O_p(1/n).$$

(ii)
$$\frac{1}{n}\sum_{t=k_m+1}^{n}\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)'=\tilde{V}_{J^*}+O_p(1/n).$$

(iii)
$$\frac{1}{n}\sum_{t=k_m+1}^{n}\tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)'=\hat{\Phi}_K(k_m)\hat{V}_{J^*}+O_p(1/n).$$

(iv) $\tilde{\Phi}_K(k_m)=\hat{\Phi}_K(k_m)+O_p(1/n).$

(v) $\tilde{\Psi}_{K^*}(k_m) = \hat{\Psi}_{K^*}(k_m) + O_p(1/n)$.

(vi) $\tilde{\Phi}_K(i) = \hat{\Phi}_K(i) + O_p(1/n)$, for every $i \in J$.

(vii) $\tilde{\Psi}_{K^*}(j) = \hat{\Psi}_{K^*}(j) + O_p(1/n)$, for every $j \in J^*$.

(viii) $\tilde{U}_K = \hat{U}_K + O_p(1/n)$.

(ix) $\tilde{V}_{K^*} = \hat{V}_{K^*} + O_p(1/n)$.

We now begin these demonstrations.

(i) From H-13,

$$
\begin{aligned}
&\tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\varepsilon}}_J(t)' \\
&\quad = \left(\tilde{\boldsymbol{\varepsilon}}_{J(J)}(t) - \tilde{\Phi}_J(k_{m-1})\tilde{\boldsymbol{\eta}}_{J(J^*)}(t-k_{m-1})\right) \\
&\qquad \left(\tilde{\boldsymbol{\varepsilon}}_{J(J)}(t)' - \tilde{\boldsymbol{\eta}}_{J(J^*)}(t-k_{m-1})'\tilde{\Phi}_J(k_{m-1})'\right) \\
&\quad = \tilde{\boldsymbol{\varepsilon}}_{J(J)}(t)\tilde{\boldsymbol{\varepsilon}}_{J(J)}(t)' - \tilde{\boldsymbol{\varepsilon}}_{J(J)}(t)\tilde{\boldsymbol{\eta}}_{J(J^*)}(t-k_{m-1})'\tilde{\Phi}_J(k_{m-1})' \\
&\qquad - \tilde{\Phi}_J(k_{m-1})\tilde{\boldsymbol{\eta}}_{J(J^*)}(t-k_{m-1})\tilde{\boldsymbol{\varepsilon}}_{J(J)}(t)' \\
&\qquad + \tilde{\Phi}_J(k_{m-1})\tilde{\boldsymbol{\eta}}_{J(J^*)}(t-k_{m-1})\tilde{\boldsymbol{\eta}}_{J(J^*)}(t-k_{m-1})'\tilde{\Phi}_J(k_{m-1})',
\end{aligned}
$$

and thus

$$
\begin{aligned}
&\frac{1}{n}\sum_{t=k_m+1}^{n}\tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\varepsilon}}_J(t)' \\
&= \left(\frac{1}{n}\sum_{t=k_{m-1}+1}^{n}\tilde{\boldsymbol{\varepsilon}}_{J(J)}(t)\tilde{\boldsymbol{\varepsilon}}_{J(J)}(t)'\right) \\
&\quad - \left(\frac{1}{n}\sum_{t=k_{m-1}+1}^{n}\tilde{\boldsymbol{\varepsilon}}_{J(J)}(t)\tilde{\boldsymbol{\eta}}_{J(J^*)}(t-k_{m-1})'\right)\tilde{\Phi}_J(k_{m-1})' \\
&\quad - \tilde{\Phi}_J(k_{m-1})\left(\frac{1}{n}\sum_{t=k_{m-1}+1}^{n}\tilde{\boldsymbol{\varepsilon}}_{J(J)}(t)\tilde{\boldsymbol{\eta}}_{J(J^*)}(t-k_{m-1})'\right)' \\
&\quad + \tilde{\Phi}_J(k_{m-1})\left(\frac{1}{n}\sum_{t=k_{m-1}+1}^{n}\tilde{\boldsymbol{\eta}}_{J(J^*)}(t-k_{m-1})\tilde{\boldsymbol{\eta}}_{J(J^*)}(t-k_{m-1})'\right)\tilde{\Phi}_J(k_{m-1})' \\
&\quad + O_p(1/n),
\end{aligned}
$$

and using H-5, H-7, H-7, H-6, respectively in each of the bracketed summands above, gives

$$
\begin{aligned}
&\frac{1}{n}\sum_{t=k_m+1}^{n}\tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\varepsilon}}_J(t)' \\
&= \tilde{U}_{J(J)} - \hat{\Phi}_J(k_{m-1})\hat{V}_{J(J^*)}\tilde{\Phi}_J(k_{m-1})' - \tilde{\Phi}_J(k_{m-1})\hat{V}_{J(J^*)}\hat{\Phi}_J(k_{m-1})' \\
&\quad + \tilde{\Phi}_J(k_{m-1})\hat{V}_{J(J^*)}\tilde{\Phi}_J(k_{m-1})' + O_p(1/n).
\end{aligned}
$$

By H-1 and H-4, we can interchange Burg and YW estimators to within $O_p(1/n)$, so that

$$
\begin{aligned}
&\frac{1}{n}\sum_{t=k_m+1}^{n}\tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\varepsilon}}_J(t)' \\
&= \tilde{U}_{J(J)} - \tilde{\Phi}_J(k_{m-1})\tilde{V}_{J(J^*)}\tilde{\Phi}_J(k_{m-1})' - \tilde{\Phi}_J(k_{m-1})\tilde{V}_{J(J^*)}\tilde{\Phi}_J(k_{m-1})' \\
&\quad + \tilde{\Phi}_J(k_{m-1})\tilde{V}_{J(J^*)}\tilde{\Phi}_J(k_{m-1})' + O_p(1/n) \\
&= \tilde{U}_{J(J)} - \tilde{\Phi}_J(k_{m-1})\tilde{V}_{J(J^*)}\tilde{\Phi}_J(k_{m-1})' + O_p(1/n) \\
&= \tilde{U}_{J(J)} - \tilde{\Phi}_J(k_{m-1})\tilde{\Psi}_{J(K^*)}(k_{m-1})\tilde{U}_{J(J)} + O_p(1/n), \quad \text{by H-11} \\
&= \tilde{U}_J + O_p(1/n), \quad \text{by H-12.}
\end{aligned}
$$

(ii) From H-14,

$$
\begin{aligned}
&\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)' \\
&= \left(\tilde{\boldsymbol{\eta}}_{J^*(J)}(t-k_m) - \tilde{\Psi}_{J^*}(k_m-k_1)\tilde{\boldsymbol{\varepsilon}}_{J^*(J^*)}(t-k_1)\right) \\
&\quad \left(\tilde{\boldsymbol{\eta}}_{J^*(J)}(t-k_m)' - \tilde{\boldsymbol{\varepsilon}}_{J^*(J^*)}(t-k_1)'\tilde{\Psi}_{J^*}(k_m-k_1)'\right) \\
&= \tilde{\boldsymbol{\eta}}_{J^*(J)}(t-k_m)\tilde{\boldsymbol{\eta}}_{J^*(J)}(t-k_m)' \\
&\quad - \left(\tilde{\boldsymbol{\varepsilon}}_{J^*(J^*)}(t-k_1)\tilde{\boldsymbol{\eta}}_{J^*(J)}(t-k_m)'\right)'\tilde{\Psi}_{J^*}(k_m-k_1)' \\
&\quad - \tilde{\Psi}_{J^*}(k_m-k_1)\left(\tilde{\boldsymbol{\varepsilon}}_{J^*(J^*)}(t-k_1)\tilde{\boldsymbol{\eta}}_{J^*(J)}(t-k_m)'\right) \\
&\quad + \tilde{\Psi}_{J^*}(k_m-k_1)\left(\tilde{\boldsymbol{\varepsilon}}_{J^*(J^*)}(t-k_1)\tilde{\boldsymbol{\varepsilon}}_{J^*(J^*)}(t-k_1)'\right)\tilde{\Psi}_{J^*}(k_m-k_1)',
\end{aligned}
$$

and thus

$$
\begin{aligned}
&\frac{1}{n}\sum_{t=k_m+1}^{n}\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)' = \\
&\quad \left(\frac{1}{n}\sum_{t=k_m+1}^{n}\tilde{\boldsymbol{\eta}}_{J^*(J)}(t-k_m)\tilde{\boldsymbol{\eta}}_{J^*(J)}(t-k_m)'\right) \\
&\quad - \left(\frac{1}{n}\sum_{t=k_m+1}^{n}\tilde{\boldsymbol{\varepsilon}}_{J^*(J^*)}(t-k_1)\tilde{\boldsymbol{\eta}}_{J^*(J)}(t-k_m)'\right)'\tilde{\Psi}_{J^*}(k_m-k_1)' \\
&\quad - \tilde{\Psi}_{J^*}(k_m-k_1)\left(\frac{1}{n}\sum_{t=k_m+1}^{n}\tilde{\boldsymbol{\varepsilon}}_{J^*(J^*)}(t-k_1)\tilde{\boldsymbol{\eta}}_{J^*(J)}(t-k_m)'\right) \\
&\quad + \tilde{\Psi}_{J^*}(k_m-k_1)\left(\frac{1}{n}\sum_{t=k_m+1}^{n}\left(\tilde{\boldsymbol{\varepsilon}}_{J^*(J^*)}(t-k_1)\tilde{\boldsymbol{\varepsilon}}_{J^*(J^*)}(t-k_1)'\right)\right)\tilde{\Psi}_{J^*}(k_m-k_1)'.
\end{aligned}
$$

Using H-9, H-10, H-10, H-8, respectively in each of the bracketed summands above, gives

$$
\begin{aligned}
&\frac{1}{n}\sum_{t=k_m+1}^{n}\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)' \\
&= \tilde{V}_{J^*(J)} - \overbrace{\hat{V}_{J^*(J)}\hat{\Phi}_{J^*(K^*)}(k_m-k_1)'}^{=\hat{\Psi}_{J^*}(k_m-k_1)\hat{U}_{J^*(J^*)}, \text{ from H-15}}\tilde{\Psi}_{J^*}(k_m-k_1)' \\
&\quad - \tilde{\Psi}_{J^*}(k_m-k_1)\hat{\Phi}_{J^*(K^*)}(k_m-k_1)\hat{V}_{J^*(J)} \\
&\quad + \tilde{\Psi}_{J^*}(k_m-k_1)\tilde{U}_{J^*(J^*)}\tilde{\Psi}_{J^*}(k_m-k_1)' + O_p(1/n).
\end{aligned}
$$

By the inductive hypothesis (H-1 - H-4), we can interchange YW and Burg estimators to within $O_p(1/n)$, so that

$$\begin{aligned}
&\frac{1}{n}\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)' \\
&= \tilde{V}_{J^*(J)} - \tilde{\Psi}_{J^*}(k_m-k_1)\tilde{U}_{J^*(J^*)}\tilde{\Psi}_{J^*}(k_m-k_1)' \\
&\quad - \tilde{\Psi}_{J^*}(k_m-k_1)\tilde{\Phi}_{J^*(K^*)}(k_m-k_1)\tilde{V}_{J^*(J)} \\
&\quad + \tilde{\Psi}_{J^*}(k_m-k_1)\tilde{U}_{J^*(J^*)}\tilde{\Psi}_{J^*}(k_m-k_1)' + O_p(1/n) \\
&= \left[I_d - \tilde{\Psi}_{J^*}(k_m-k_1)\tilde{\Phi}_{J^*(K^*)}(k_m-k_1)\right]\tilde{V}_{J^*(J)} + O_p(1/n) \\
&= \tilde{V}_{J^*} + O_p(1/n), \quad \text{by H-17.}
\end{aligned}$$

(iii) By definition,

$$\tilde{\boldsymbol{\varepsilon}}_J(t) = \mathbf{X}_t - \sum_{i\in J}\tilde{\Phi}_J(i)\mathbf{X}_{t-i},$$

and

$$\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m) = \mathbf{X}_{t-k_m} - \sum_{j\in J^*}\tilde{\Psi}_{J^*}(j)\mathbf{X}_{t-k_m+j},$$

and this implies

$$
\begin{aligned}
&\frac{1}{n}\sum_{t=k_m+1}^{n}\tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)' \\
&= \overbrace{\left(\frac{1}{n}\sum_{t=k_m+1}^{n}\mathbf{X}_t\mathbf{X}'_{t-k_m}\right)}^{\hat{\Gamma}(k_m)} - \sum_{j\in J^*}\overbrace{\left(\frac{1}{n}\sum_{t=k_m+1}^{n}\mathbf{X}_t\mathbf{X}'_{t-k_m+j}\right)}^{\hat{\Gamma}(k_m-j)+O_p(1/n)}\tilde{\Psi}_{J^*}(j)' \\
&\quad - \sum_{i\in J}\tilde{\Phi}_J(i)\underbrace{\left(\frac{1}{n}\sum_{t=k_m+1}^{n}\mathbf{X}_{t-i}\mathbf{X}'_{t-k_m}\right)}_{\hat{\Gamma}(k_m-i)+O_p(1/n)} \\
&\quad + \sum_{j\in J^*}\sum_{i\in J}\tilde{\Phi}_J(i)\underbrace{\left(\frac{1}{n}\sum_{t=k_m+1}^{n}\mathbf{X}_{t-i}\mathbf{X}'_{t-k_m+j}\right)}_{\hat{\Gamma}(k_m-j-i)+O_p(1/n)}\tilde{\Psi}_{J^*}(j)' \\
&= \hat{\Gamma}(k_m) - \sum_{j\in J^*}\hat{\Gamma}(k_m-j)\tilde{\Psi}_{J^*}(j)' - \sum_{i\in J}\tilde{\Phi}_J(i)\hat{\Gamma}(k_m-i) \\
&\quad + \sum_{j\in J^*}\underbrace{\left(\sum_{i\in J}\hat{\Gamma}(k_m-j-i)\right)}_{=\hat{\Gamma}(k_m-j),\text{ by H-16}}\tilde{\Psi}_{J^*}(j)' + O_p(1/n) \\
&= \hat{\Gamma}(k_m) - \sum_{i\in J}\hat{\Phi}_J(i)\hat{\Gamma}(k_m-i) + O_p(1/n),
\end{aligned}
$$

since $\tilde{\Phi}_J(i) = \hat{\Phi}_J(i) + O_p(1/n)$. Thus by the first line of algorithm 1.3.1

$$
\frac{1}{n}\sum_{t=k_m+1}^{n}\tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)' = \hat{\Phi}_K(k_m)\hat{V}_{J^*} + O_p(1/n).
$$

(iv) Taking (1.29), applying identities K-1 and K-6 to the *vec* term, and multiplying and dividing by $\frac{1}{n}$, gives

$$
\begin{aligned}
& vec\left(\tilde{\Phi}_K(k_m)\right) \\
&= \Bigg[\overbrace{\left(\frac{1}{n}\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)'\right)}^{=\tilde{V}_{J^*}+O_p(1/n),\text{ by (ii)}} \otimes I_d \\
&\quad + \tilde{V}_{J^*}^2 \otimes \tilde{U}_J^{-1} \overbrace{\left(\frac{1}{n}\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\varepsilon}}_J(t)'\right)}^{=\tilde{U}_J+O_p(1/n),\text{ by (i)}} \tilde{U}_J^{-1} \Bigg]^{-1} \\
&\quad \left[I_{d^2} + \tilde{V}_{J^*} \otimes \tilde{U}_J^{-1}\right] vec \underbrace{\left(\frac{1}{n}\sum_{t=k_m+1}^{n} \tilde{\boldsymbol{\varepsilon}}_J(t)\tilde{\boldsymbol{\eta}}_{J^*}(t-k_m)'\right)}_{=\hat{\Phi}_K(k_m)\hat{V}_{J^*}+O_p(1/n),\text{ by (iii)}} \\
&= \left[\tilde{V}_{J^*} \otimes I_d + \tilde{V}_{J^*}^2 \otimes \tilde{U}_J^{-1} + O_p(1/n)\right]^{-1} \left[I_{d^2} + \tilde{V}_{J^*} \otimes \tilde{U}_J^{-1}\right] \\
&\quad vec\left(\hat{\Phi}_K(k_m)\hat{V}_{J^*}\right) + O_p(1/n) \\
&= \left[\tilde{V}_{J^*} \otimes I_d + \tilde{V}_{J^*}^2 \otimes \tilde{U}_J^{-1}\right]^{-1} \left[I_{d^2} + \tilde{V}_{J^*} \otimes \tilde{U}_J^{-1}\right] \left(\hat{V}_{J^*} \otimes I_d\right) \\
&\quad vec\left(\hat{\Phi}_K(k_m)\right) + O_p(1/n),
\end{aligned}
$$

where the last equality follows by applying lemma 2.6.1 to the bracketed inverse term, and identity K-6 to the *vec* operator. By inductive hypothesis H-4, we can replace $\left(\hat{V}_{J^*} \otimes I_d\right)$ with $\left(\hat{V}_{J^*} \otimes I_d\right) + O_p(1/n)$, to give

$$
\begin{aligned}
& vec\left(\tilde{\Phi}_K(k_m)\right) \\
&= \left[\tilde{V}_{J^*} \otimes I_d + \tilde{V}_{J^*}^2 \otimes \tilde{U}_J^{-1}\right]^{-1} \underbrace{\left[I_{d^2} + \tilde{V}_{J^*} \otimes \tilde{U}_J^{-1}\right] \left(\tilde{V}_{J^*} \otimes I_d\right)}_{=\tilde{V}_{J^*}\otimes I_d + \tilde{V}_{J^*}^2 \otimes \tilde{U}_J^{-1},\text{ by K-11 and K-9}} \\
&\quad vec\left(\hat{\Phi}_K(k_m)\right) + O_p(1/n) \\
&= vec\left(\hat{\Phi}_K(k_m)\right) + O_p(1/n).
\end{aligned}
$$

(v) From (2.20),

$$\begin{aligned}
\tilde{\Psi}_{K^*}(k_m) &= \tilde{V}_{J^*}\tilde{\Phi}_K(k_m)'\tilde{U}_J^{-1} \\
&= \hat{V}_{J^*}\hat{\Phi}_K(k_m)'\hat{U}_J^{-1} + O_p(1/n), \quad \text{by H-3, H-4, and (iv)}, \\
&= \hat{\Psi}_{K^*}(k_m) + O_p(1/n), \quad \text{again from (2.20)}.
\end{aligned}$$

(vi) From the algorithm,

$$\begin{aligned}
\tilde{\Phi}_K(i) &= \tilde{\Phi}_J(i) - \tilde{\Phi}_K(k_m)\tilde{\Psi}_{J^*}(k_m - i), \quad \text{for every } i \in J, \\
&= \hat{\Phi}_J(i) - \hat{\Phi}_K(k_m)\hat{\Psi}_{J^*}(k_m - i) + O_p(1/n), \quad \text{by H-1, (iv), and H-2}, \\
&= \hat{\Phi}_K(i) + O_p(1/n),
\end{aligned}$$

where the last line follows again from the algorithm, but now applied to the YW estimators.

(vii) Similarly, from the algorithm,

$$\begin{aligned}
\tilde{\Psi}_{K^*}(j) &= \tilde{\Psi}_{J^*}(j) - \tilde{\Psi}_{K^*}(k_m)\tilde{\Phi}_J(k_m - j), \quad \text{for every } j \in J^*, \\
&= \hat{\Psi}_{J^*}(j) - \hat{\Psi}_{K^*}(k_m)\hat{\Phi}_J(k_m - j) + O_p(1/n), \quad \text{by H-2, (v), and H-1}, \\
&= \hat{\Psi}_{K^*}(j) + O_p(1/n),
\end{aligned}$$

again from the algorithm applied to the YW estimators.

(viii) From the algorithm,

$$\begin{aligned}
\tilde{U}_K &= \tilde{U}_J - \tilde{\Phi}_K(k_m)\tilde{V}_{J^*}\tilde{\Phi}_K(k_m)' \\
&= \left[I_d - \tilde{\Phi}_K(k_m)\tilde{\Psi}_{K^*}(k_m)\right]\tilde{U}_J, \quad \text{from (2.20)}, \\
&= \left[I_d - \hat{\Phi}_K(k_m)\hat{\Psi}_{K^*}(k_m)\right]\hat{U}_J + O_p(1/n), \quad \text{from (iv), (v), and H-3}, \\
&= \hat{U}_K + O_p(1/n),
\end{aligned}$$

from the algorithm applied to the YW estimators.

(ix) Using almost identical arguments to the above, we have from the algorithm

$$\begin{aligned}
\tilde{V}_{K^*} &= \tilde{V}_{J^*} - \tilde{\Psi}_{K^*}(k_m)\tilde{U}_J\tilde{\Psi}_{K^*}(k_m)' \\
&= \left[I_d - \tilde{\Psi}_{K^*}(k_m)\tilde{\Phi}_K(k_m)\right]\tilde{V}_{J^*}, \quad \text{from (2.20)}, \\
&= \left[I_d - \hat{\Psi}_{K^*}(k_m)\hat{\Phi}_K(k_m)\right]\hat{V}_{J^*} + O_p(1/n), \quad \text{from (iv), (v), and H-4}, \\
&= \hat{V}_{K^*} + O_p(1/n).
\end{aligned}$$

This completes the induction argument, and therefore the statement of the theorem holds for an arbitrary set of lags K.

□

Theorem 2.6.2 (Asymptotic distribution of the subset Burg estimator) *The Burg estimators of the coefficients and white noise variance of SVAR model (2.1), satisfy*

(a) $\sqrt{n}\,(\tilde{\boldsymbol{\alpha}}_K - \boldsymbol{\alpha}_K) \xrightarrow{d} \mathrm{N}(\mathbf{0}, G_K^{-1} \otimes \Sigma)$.

(b) $\tilde{U}_K \xrightarrow{p} \Sigma$.

Proof

Theorem 2.6.1 states that the Burg and YW estimators for the forward and backward prediction problems, differ by terms of order $O_p(1/n)$ when $\{\mathbf{X}_t\}$ satisfies *Condition 1*. As mentioned at the beginning of this section, application of Brockwell and Davis (1991) proposition 6.3.3, then gives convergence in distribution/probability to the same limiting random vectors. These limiting distributions were presented in theorem 2.5.1. Since $\{\mathbf{X}_t\}$ satisfying *Condition 2* also satisfies *Condition 1*, these limiting distributions extend to the Burg estimators of the coefficients and white noise variance of SVAR model (2.1).

□

Chapter 3

SADDLEPOINT APPROXIMATIONS TO THE DISTRIBUTIONS OF THE YULE-WALKER AND BURG COEFFICIENT ESTIMATORS OF SUBSET AR MODELS WITH SUBSET SIZE ONE

3.1 Introduction

A notable feature of the simulation results of Chapter 1 is that the Gaussian likelihoods for models fitted via the Burg method tend to be consistently larger than those fitted via Yule-Walker, particularly as the roots of the AR polynomial approach the unit circle. Comparing the distributions of Yule-Walker, Burg, and maximum (Gaussian) likelihood estimators in some special cases, should provide further insight into their different finite-sample performances, and the question of whether or not the densities of the Burg and maximum likelihood estimators are "closer" in some sense than those of Yule-Walker and maximum likelihood.

In this chapter we compute saddlepoint approximations to the probability distribution and density functions of the Yule-Walker and Burg estimators of the autoregressive coefficient in a Gaussian AR(p) model, where the coefficients of the first $p-1$ lags are zero (henceforth abbreviated as a SAR(p) model). We obtain simulation-based estimates of the probability density function for these two as well as the maximum likelihood estimator, and proceed to compare all three.

The saddlepoint approximation in this context was originally discussed by Daniels (1956), in which he derived the density of the Burg estimator for an AR(1). Phillips (1978), obtained the Edgeworth and saddlepoint approximations to the density of the least squares estimator. Durbin (1980), explored the approximate distribution of partial serial correlation coefficients, which included the Yule-Walker estimator. Using Edgeworth approximations, Ochi (1983) obtained asymptotic expansions to terms of order n^{-1} for the distribution of the generalized AR(1) coefficient estimator $\hat{\phi}(c_1, c_2)$ presented in the next section. More recently, Butler and Paolella (1998) have obtained saddlepoint approximations to ratios of quadratic forms in normal random variables. The development in this chapter parallels their technique.

3.2 SAR(p) Model Parameter Estimation

Consider estimating the parameters in the zero-mean causal univariate Gaussian subset AR(p) model:

$$X_t = \phi X_{t-p} + Z_t, \quad \{Z_t\} \sim \text{ IID N}(0, \sigma^2). \tag{3.1}$$

Given observations $x_1, \cdots, x_n$ from a time series, and defining

$$\sigma^2_{AL}(\phi) \equiv (1 - \phi^2)\hat{\gamma}_0,$$

the least squares estimator of ϕ is from (2.14),

$$\begin{aligned} \hat{\phi}_{LS} &= \frac{\sum_{t=p+1}^{n} x_t x_{t-p}}{\sum_{t=1}^{n-p} x_t^2} \\ &= \frac{\sum_{t=1+p}^{n} x_t x_{t-p}}{\sum_{t=p+1}^{n-p} x_t^2 + \sum_{t=1}^{p} x_t^2}. \end{aligned} \tag{3.2}$$

The Yule-Walker algorithm (1.3.1) gives the estimates

$$\begin{aligned}
\hat{\phi}_{YW} &= \frac{\sum_{t=1+p}^{n} x_t x_{t-p}}{\sum_{t=p+1}^{n-p} x_t^2 + \sum_{t=1}^{p} x_t^2 + \sum_{t=n-p+1}^{n} x_t^2} \qquad (3.3)\\
&= \frac{\frac{1}{n}\sum_{t=1+p}^{n} x_t x_{t-p}}{\frac{1}{n}\sum_{t=1}^{n} x_t^2}\\
&= \frac{\hat{\gamma}_p}{\hat{\gamma}_0} \qquad (3.4)\\
\hat{\sigma}^2_{YW} &= (1-\hat{\phi}^2_{YW})\hat{\gamma}_0 = \sigma^2_{AL}(\hat{\phi}_{YW}),
\end{aligned}$$

while from the Burg algorithm (1.4.4) we obtain

$$\begin{aligned}
\hat{\phi}_{BG} &= \frac{\sum_{t=1+p}^{n} x_t x_{t-p}}{\sum_{t=p+1}^{n-p} x_t^2 + \frac{1}{2}\sum_{t=1}^{p} x_t^2 + \frac{1}{2}\sum_{t=n-p+1}^{n} x_t^2} \qquad (3.5)\\
&= \frac{2\hat{\gamma}_p}{\hat{\gamma}_0 + \frac{1}{n}\sum_{t=p+1}^{n-p} x_t^2}\\
&= \frac{2\hat{\gamma}_p}{\hat{\gamma}_0 + a_p}, \text{ where } a_p = \frac{1}{n}\sum_{t=p+1}^{n-p} x_t^2\\
\hat{\sigma}^2_{BG} &= (1-\hat{\phi}^2_{BG})\hat{\gamma}_0 = \sigma^2_{AL}(\hat{\phi}_{BG}).
\end{aligned}$$

Remark 3.2.1 *Since $a_p \leq \hat{\gamma}_0$, we easily see that $|\hat{\phi}_{BG}| \geq |\hat{\phi}_{YW}|$, and thus $\sigma^2_{AL}(\hat{\phi}_{BG}) \leq \sigma^2_{AL}(\hat{\phi}_{YW})$. Also note that the Burg estimator of a SAR(p) coincides with the Nuttall-Strand estimator.*

Remark 3.2.2 *Since we are assuming a zero-mean process, we have opted not to mean-correct the data prior to parameter estimation. This is an unrealistic assumption in practice, but will help us fix the comparisons between the various estimators, as well as examine the relative performance of the saddlepoint approximations to the sampling distributions.*

From (3.2), (3.3), and (3.5), and defining the generalized estimator

$$\hat{\phi}(c_1, c_2) = \frac{\sum_{t=1+p}^{n} x_t x_{t-p}}{\sum_{t=p+1}^{n-p} x_t^2 + c_1\sum_{t=1}^{p} x_t^2 + c_2\sum_{t=n-p+1}^{n} x_t^2},$$

we see that $\hat{\phi}(1,0)$, $\hat{\phi}(1,1)$, and $\hat{\phi}(\frac{1}{2},\frac{1}{2})$, are the least squares, Yule-Walker, and Burg estimators respectively.

From Brockwell and Davis (1991) problem 8.7, the *-2 Log Likelihood* for observations $X_1,\ldots,X_n$ from model (3.1), is given by

$$\mathcal{L}(\phi,\sigma^2) = n\log(2\pi\sigma^2) + \log|G_p| + \frac{1}{\sigma^2}\left[\mathbf{X}_p' G_p^{-1}\mathbf{X}_p + \sum_{t=p+1}^{n}(X_t - \phi X_{t-p})^2\right], \quad (3.6)$$

where

$$\begin{aligned}
\mathbf{X}_p' &= [X_1,\ldots,X_p]' \\
G_p &= \sigma^{-2}\Gamma_p = \frac{\gamma_0}{\sigma^2}I_p = (1-\phi^2)^{-1}I_p \\
&\Rightarrow \quad |G_p| = (1-\phi^2)^{-p} \ \text{ and } \ G_p^{-1} = (1-\phi^2)I_p,
\end{aligned}$$

and thus

$$\mathcal{L}(\phi,\sigma^2) = n\log(2\pi\sigma^2) - p\log(1-\phi^2) + \frac{1}{\sigma^2}\left[(1-\phi^2)\sum_{t=1}^{p}X_t^2 + \sum_{t=p+1}^{n}(X_t-\phi X_{t-p})^2\right].$$

Comparing this with equation (8.7.4) in Brockwell and Davis (1991), we see immediately that the expression in square brackets must be the residual sum of squares (RSS), ie.

$$RSS = \sum_{t=1}^{n}(X_t - \hat{X}_t)^2/r_{t-1} = (1-\phi^2)\sum_{t=1}^{p}X_t^2 + \sum_{t=p+1}^{n}(X_t-\phi X_{t-p})^2. \quad (3.7)$$

Defining $\sigma^2_{ML}(\phi) \equiv \frac{RSS(\phi)}{n}$, and expanding the above, we see that $\sigma^2_{ML}(\phi) = \hat{\gamma}_0 - 2\hat{\gamma}_p\phi + a_p\phi^2$. Since the maximum likelihood estimator (MLE) of σ^2 for fixed ϕ is RSS/n, ignoring constants we obtain the *reduced -2 log likelihood*:

$$\begin{aligned}
\mathcal{RL}(\phi) &= n\log(\hat{\gamma}_0 - 2\hat{\gamma}_p\phi + a_p\phi^2) - p\log(1-\phi^2) \quad (3.8)\\
&\propto \log\left(\frac{\sigma^2_{ML}(\phi)^n}{\sigma^2_{AL}(\phi)^p}\right). \quad (3.9)
\end{aligned}$$

Differentiating (3.8), we find that the MLE of ϕ ($\hat{\phi}_{ML}$) is a root of the cubic

$$\phi^3 - \frac{(n-2p)\hat{\gamma}_p}{(n-p)a_p}\phi^2 - \frac{na_p + p\hat{\gamma}_0}{(n-p)a_p}\phi + \frac{n\hat{\gamma}_p}{(n-p)a_p}. \quad (3.10)$$

Example 3.2.1 *500 observations were simulated from model (3.1) with* $p = 3$, $\phi = 0.74$, *and* $\{Z_t\} \sim$ IID $N(0, 1)$. *To 4 decimal places, we obtained the following estimates:* $\hat{\phi}_{YW} = 0.7055$ $(\mathcal{RL} = -42.2659)$, $\hat{\phi}_{BG} = 0.7125$ $(\mathcal{RL} = -42.3555)$, $\hat{\phi}_{ML} = 0.7153$ $(\mathcal{RL} = -42.3635)$.

Figure 3.1: Plot of $\sigma^2_{ML}(\phi)$ (short dashes and bounded below), $\sigma^2_{AL}(\phi)$ (long dashes and bounded above), and a scaled and re-centered $\mathcal{RL}(\phi)$ (solid line), for the simulated data of example 3.2.1.

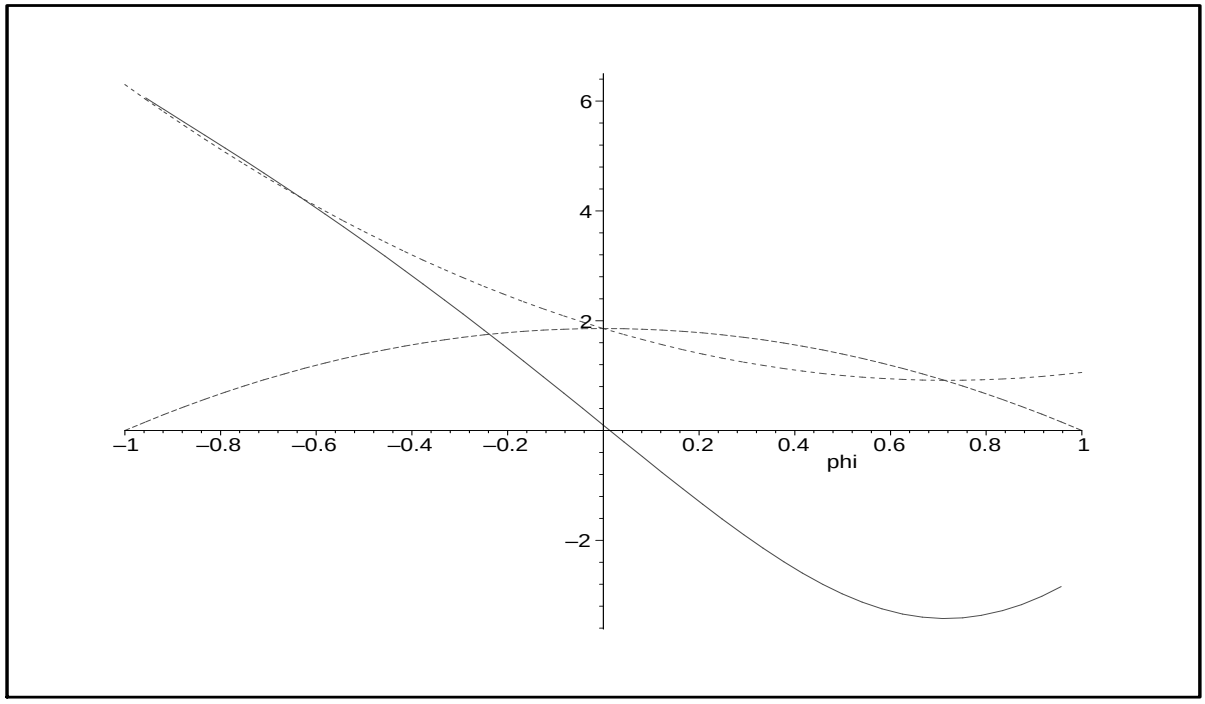


Figure 3.1 shows the two variance curves $\sigma^2_{ML}(\phi)$ and $\sigma^2_{AL}(\phi)$ for the data of example 3.2.1, overlaid by the $\mathcal{RL}(\phi)$ curve (suitably scaled and centered to fit on the

same figure). The plot is typical of the shapes of the variance curves. Both are quadratics in ϕ; $\sigma^2_{AL}(\phi)$ bounded above with roots at ± 1, $\sigma^2_{ML}(\phi)$ bounded below with $\sigma^2_{ML}(\phi) \geq 0$ for all $|\phi| \leq 1$ (since by (3.7) it is a sum of squares). The curves intersect where

$$\begin{aligned} \sigma^2_{ML}(\phi) - \sigma^2_{AL}(\phi) &= 0 \\ \Rightarrow \phi\left((a_p + \hat{\gamma}_0)\phi - 2\hat{\gamma}_p\right) &= 0 \\ \Rightarrow \phi &= 0 \quad \text{or} \quad \phi = \frac{2\hat{\gamma}_p}{\hat{\gamma}_0 + a_p} \equiv \tilde{\phi}. \end{aligned}$$

Remark 3.2.3 *Thus the Burg white noise variance estimate coincides with the RSS/n variance estimate, ie.* $\sigma^2_{ML}(\hat{\phi}_{BG}) = \sigma^2_{AL}(\hat{\phi}_{BG})$.

It is clear from (3.4) that the Yule-Walker algorithm always gives a causal solution for model (3.1). An easy geometric argument enables us to conclude likewise for the Burg estimate $\hat{\phi}_{BG}$: Since $\sigma^2_{ML}(\phi) \geq 0$ for $|\phi| \leq 1$, it always intersects the curve $\sigma^2_{AL}(\phi)$ at $\phi = 0$ and $\phi = \hat{\phi}_{BG}$ in this causal region, so that we must have $|\hat{\phi}_{BG}| \leq 1$. An immediate consequence is that $2|\hat{\gamma}_p| \leq \hat{\gamma}_0 + a_p$.

3.3 Saddlepoint Approximating the Distribution of $\hat{\phi}(\mathbf{c_1}, \mathbf{c_2})$

A realization $\mathbf{X} = [X_1, \ldots, X_n]'$ from model (3.1), has the multivariate normal distribution

$$\mathbf{X} \sim \mathrm{N}_n(\mathbf{0}, \Gamma_n),$$

with probability density function

$$f_{\mathbf{X}}(\mathbf{x}) = (2\pi)^{n/2}|\Gamma_n|^{-1/2} \exp\left\{-\frac{1}{2}\mathbf{x}'\Gamma_n^{-1}\mathbf{x}\right\},$$

where

$$\Gamma_n = \begin{bmatrix} \gamma(0) & \cdots & \gamma(n-1) \\ \vdots & \ddots & \vdots \\ \gamma(n-1) & \cdots & \gamma(0) \end{bmatrix} = \frac{\sigma^2}{1-\phi^2} J_n,$$

and J_n is the $(n \times n)$ matrix whose $(i,j)^{th}$ entry is

$$J_n(i,j) = \begin{cases} \phi^k, & \text{if } |i-j| = kp,\ k = 0, 1, \ldots, [n/p], \\ 0, & \text{otherwise,} \end{cases}$$

and $[z]$ denotes the greatest integer less than or equal to z.

Defining the $(i,j)^{th}$ entry of the $(n \times n)$ matrix A to be

$$A(i,j) = \begin{cases} \frac{1}{2}, & \text{if } |i-j| = p, \\ 0, & \text{otherwise,} \end{cases}$$

and that of $(n \times n)$ matrix B to be

$$B(i,j) = \begin{cases} c_1, & \text{if } i = j \text{ and } 1 \le i \le p, \\ 1, & \text{if } i = j \text{ and } p+1 \le i \le n-p, \\ c_2, & \text{if } i = j \text{ and } n-p+1 \le i \le n, \\ 0, & \text{otherwise,} \end{cases}$$

we can express the generic estimator $\hat{\phi}(c_1, c_2)$ as a ratio of quadratic forms in normal random variables

$$\hat{\phi}(c_1, c_2) = \frac{\mathbf{X}'A\mathbf{X}}{\mathbf{X}'B\mathbf{X}} = \frac{Q_1}{Q_2}.$$

The joint moment generating function (mgf) of Q_1 and Q_2 is given by

$$M(s,t) = \mathbb{E} \exp\{sQ_1 + tQ_2\} = \mathbb{E} \exp\{\mathbf{X}' (sA + tB) \mathbf{X}\} = \mathbb{E} \exp\{\mathbf{X}'C\mathbf{X}\}, \quad (3.11)$$

with $C = sA + tB$. Therefore, we have

$$\begin{aligned} M(s,t) &= \int_{\mathbb{R}^n} (2\pi)^{n/2} |\Gamma_n|^{-1/2} \exp\left\{ -\frac{1}{2} \mathbf{x}' \left(\Gamma_n^{-1} - 2C \right) \mathbf{x} \right\} d\mathbf{x} \\ &= \frac{|\Gamma_n|^{-1/2}}{\left| (\Gamma_n^{-1} - 2C)^{-1} \right|^{-1/2}} \\ &\quad \underbrace{\int_{\mathbb{R}^n} (2\pi)^{n/2} \left| \left(\Gamma_n^{-1} - 2C \right)^{-1} \right|^{-1/2} \exp\left\{ -\frac{1}{2} \mathbf{x}' \left(\Gamma_n^{-1} - 2C \right) \mathbf{x} \right\} d\mathbf{x}}_{1} \\ &= |I_n - 2\Gamma_n C|^{-1/2} \\ &= |I_n - 2\Gamma_n (sA + tB)|^{-1/2}, \end{aligned}$$

defined for all s and t such that $|I_n - 2\Gamma_n (sA + tB)| > 0$.

3.3.1 Some preliminary results

Suppose $g(\cdot)$ is a real-valued function, and $X(s)$ a square matrix viewed as a function of the scalar variable s. The chain rule for obtaining the derivative of $g(X(s))$ with respect to s is, using the notation of Lutkepohl (1996)

$$\begin{aligned}
\frac{\partial g(X(s))}{\partial s} &= \frac{\partial g(X)}{\partial vec(X)'} \frac{\partial vec(X(s))}{\partial s} \\
&= vec\left(\frac{\partial g(X)}{\partial X}\right)' vec\left(\frac{\partial X(s)}{\partial s}\right) \\
&= vec\left(\left(\left(\frac{\partial g(X)}{\partial X}\right)'\right)'\right)' vec\left(\frac{\partial X(s)}{\partial s}\right) \\
&= \mathrm{Tr}\left[\left(\frac{\partial g(X)}{\partial X}\right)' \frac{\partial X(s)}{\partial s}\right], \qquad (3.12)
\end{aligned}$$

where we have used the fact that for square matrices A and B,

$$vec(A')'vec(B) = \mathrm{Tr}(AB) = \mathrm{Tr}(BA),$$

and the shorthand, $vec(X)' \equiv (vec(X))'$.

Applying this to $M(s,t)$, we obtain

$$\begin{aligned}
\frac{\partial M(s,t)}{\partial t} &= \mathrm{Tr}\left[\left(\frac{\partial \left|I_n - 2\Gamma_n(sA+tB)\right|^{-1/2}}{\partial \left(I_n - 2\Gamma_n(sA+tB)\right)}\right)' \frac{\partial (I_n - 2\Gamma_n(sA+tB))}{\partial t}\right] \\
&= -\frac{1}{2}\left|I_n - 2\Gamma_n(sA+tB)\right|^{-3/2} \left|I_n - 2\Gamma_n(sA+tB)\right| \\
&\quad \mathrm{Tr}\left[\left(I_n - 2\Gamma_n(sA+tB)\right)^{-1}(-2\Gamma_n B)\right] \\
&= \left|I_n - 2\Gamma_n(sA+tB)\right|^{-1/2} \mathrm{Tr}\left[\left(I_n - 2\Gamma_n(sA+tB)\right)^{-1}\Gamma_n B\right]. \qquad (3.13)
\end{aligned}$$

Likewise,

$$\frac{\partial M(s,t)}{\partial s} = \left|I_n - 2\Gamma_n(sA+tB)\right|^{-1/2} \mathrm{Tr}\left[\left(I_n - 2\Gamma_n(sA+tB)\right)^{-1}\Gamma_n A\right].$$

Thus,

$$\mathbf{E}Q_1 = \left.\frac{\partial M(s,t)}{\partial s}\right|_{s=0=t} = \mathrm{Tr}(\Gamma_n A),$$

and

$$\mathbf{E}Q_2 = \left.\frac{\partial M(s,t)}{\partial t}\right|_{s=0=t} = \mathrm{Tr}(\Gamma_n B). \tag{3.14}$$

Defining $Y_r \equiv Q_1 - rQ_2 = \mathbf{X}'(A - rB)\mathbf{X}$, we obtain by linearity of the trace,

$$\mathbf{E}Y_r = \mathrm{Tr}\left[\Gamma_n(A - rB)\right].$$

3.3.2 The Cumulative Distribution Function (cdf)

In deriving the cdf of $\hat{\phi}(c_1, c_2)$, we will use the notion of *the constructed random variable at zero*, as in Butler and Paolella (1998):

$$F(r) = P\left(\frac{Q_1}{Q_2} \leq r\right) = P(Q_1 - rQ_2 \leq 0) = P(Y_r \leq 0),$$

where $r \in (r_L, r_U) \subseteq (-1, 1)$ lies in the interior of the support of $\hat{\phi}(c_1, c_2)$. These lower and upper bounds of the support satisfy

$$r_L = \min\left\{\frac{\mathbf{x}'A\mathbf{x}}{\mathbf{x}'B\mathbf{x}} : \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq \mathbf{0}\right\},$$

and

$$r_U = \max\left\{\frac{\mathbf{x}'A\mathbf{x}}{\mathbf{x}'B\mathbf{x}} : \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \neq \mathbf{0}\right\}.$$

Defining $\mathbf{z} = B^{1/2}\mathbf{x}$, and noting that both A and B are symmetric, we can rewrite these optimization expressions for ratios of quadratic forms as

$$r_L = \min\left\{\frac{\mathbf{z}'B^{-1/2}AB^{-1/2}\mathbf{z}}{\mathbf{z}'\mathbf{z}} : \mathbf{z} \in \mathbb{R}^n, \mathbf{z} \neq \mathbf{0}\right\},$$

and

$$r_L = \min\left\{\frac{\mathbf{z}'B^{-1/2}AB^{-1/2}\mathbf{z}}{\mathbf{z}'\mathbf{z}} : \mathbf{z} \in \mathbb{R}^n, \mathbf{z} \neq \mathbf{0}\right\},$$

whence an application of the *Raleigh-Ritz Theorem* (see for example Lutkepohl (1996), section 5.2.2) gives

$$r_L = \lambda_{min}(B^{-1/2}AB^{-1/2}), \qquad r_U = \lambda_{max}(B^{-1/2}AB^{-1/2}), \tag{3.15}$$

where $\lambda_{min}(B^{-1/2}AB^{-1/2})$ and $\lambda_{max}(B^{-1/2}AB^{-1/2})$ denote respectively the smallest and largest eigenvalues of the real symmetric matrix $B^{-1/2}AB^{-1/2}$.

The mgf of Y_r is then

$$\begin{aligned} M_{Y_r}(s) &= \mathbf{E}\exp\{sY_r\} = \mathbf{E}\exp\{\mathbf{X}'(sA - srB)\mathbf{X}\} && (3.16)\\ &= |I_n - 2s\Gamma_n(A - rB)|^{-1/2} \\ &\equiv |\Omega(r,s)|^{-1/2}\,, && (3.17) \end{aligned}$$

since (3.16) is of the same format as (3.11). Note that for fixed r, $M_{Y_r}(s)$ is defined for all s such that $|\Omega(r,s)| > 0$. The cumulant generating function (cgf) of Y_r is then

$$K_{Y_r}(s) = -\frac{1}{2}\log|\Omega(r,s)|\,, \tag{3.18}$$

whence, using (3.12) and Lutkepohl (1996) equation (10) of section 10.3.3,

$$\begin{aligned} K'_{Y_r}(s) &= -\frac{1}{2}\mathrm{Tr}\left[\left(\frac{\partial \log|\Omega(r,s)|}{\partial\Omega(r,s)}\right)'\frac{\partial\Omega(r,s)}{\partial s}\right] \\ &= -\frac{1}{2}\mathrm{Tr}\left[\Omega^{-1}(r,s)(-2\Gamma_n(A - rB))\right] \\ &= \mathrm{Tr}\left[\Omega^{-1}(r,s)\Gamma_n(A - rB)\right]. \end{aligned}$$

Again from (3.12) and Lutkepohl (1996) equation (23) of section 10.3.2,

$$\begin{aligned} K''_{Y_r}(s) &= \mathrm{Tr}\left[\left(\frac{\partial \mathrm{Tr}\left[\Omega^{-1}(r,s)\Gamma_n(A - rB)\right]}{\partial\Omega(r,s)}\right)'\frac{\partial\Omega(r,s)}{\partial s}\right] \\ &= \mathrm{Tr}\left[-\Omega^{-1}(r,s)\Gamma_n(A - rB)\Omega^{-1}(r,s)(-2\Gamma_n(A - rB))\right] \\ &= 2\mathrm{Tr}\left[\left(\Omega^{-1}(r,s)\Gamma_n(A - rB)\right)^2\right]. && (3.19) \end{aligned}$$

Finally, applying (3.12) once more, and Lutkepohl (1996) equation (19) of section 10.3.2,

$$\begin{aligned}\frac{1}{2}K'''_{Y_r}(s) &= \text{Tr}\left[\left(\frac{\partial \text{Tr}\left[(\Omega^{-1}(r,s)\Gamma_n(A-rB))^2\right]}{\partial \Omega^{-1}(r,s)}\right)' \frac{\partial \Omega^{-1}(r,s)}{\partial s}\right] \\ &= \text{Tr}\left[2\Gamma_n(A-rB)\Omega^{-1}(r,s)\Gamma_n(A-rB)\frac{\partial \Omega^{-1}(r,s)}{\partial s}\right]. \qquad (3.20)\end{aligned}$$

To compute the derivative of $\Omega^{-1}(r,s)$ with respect to s, we appeal to the chain rule (2) in section 10.7, and equation (1) of section 10.6 of Lutkepohl (1996):

$$\begin{aligned}\frac{\partial vec(\Omega^{-1}(r,s))}{\partial s} &= \frac{\partial vec(\Omega^{-1}(r,s))}{\partial vec\left(\Omega(r,s)\right)'}\frac{\partial vec(\Omega(r,s))}{\partial s} \\ &= \left[-(\Omega^{-1}(r,s))' \otimes \Omega^{-1}(r,s)\right] vec\left[-2\Gamma_n(A-rB)\right],\end{aligned}$$

and therefore, from property [K-6] of section 1.6,

$$\frac{\partial \Omega^{-1}(r,s)}{\partial s} = 2\Omega^{-1}(r,s)\Gamma_n(A-rB)\Omega^{-1}(r,s). \qquad (3.21)$$

Putting this in (3.20), gives finally:

$$K'''_{Y_r}(s) = 8\text{Tr}\left[\left(\Gamma_n(A-rB)\Omega^{-1}(r,s)\right)^3\right].$$

The Lugannani and Rice approximation to the cdf of $\hat{\phi}(c_1,c_2)$ at r, can then be defined in terms of the approximation to the cdf of Y_r at 0:

$$\hat{F}(r) = \hat{F}_{Y_r}(0) = \begin{cases} \Psi(\hat{w}) + \psi(\hat{w})\left[\hat{w}^{-1} - \hat{u}^{-1}\right], & \text{if } \text{Tr}\left[\Gamma_n(A-rB)\right] \neq 0, \\ \frac{1}{2} + \frac{K'''_{Y_r}(0)}{\sqrt{72\pi}K''_{Y_r}(0)^{3/2}}, & \text{if } \text{Tr}\left[\Gamma_n(A-rB)\right] = 0, \end{cases}$$

where $\Psi(\cdot)$ and $\psi(\cdot)$ denote, respectively, the cdf and probability density function (pdf) of a standard normal random variable. Also,

$$\hat{w} = \text{sgn}(\hat{s})\sqrt{-2K_{Y_r}(\hat{s})} = \text{sgn}(\hat{s})\sqrt{\log|\Omega(r,\hat{s})|}$$
$$\hat{u} = \hat{s}\sqrt{K''_{Y_r}(\hat{s})} = \hat{s}\sqrt{2\text{Tr}\left[(\Omega^{-1}(r,\hat{s})\Gamma_n(A-rB))^2\right]},$$

and $\hat{s}$ solves the saddlepoint equation

$$K'_{Y_r}(\hat{s}) = \mathrm{Tr}\left[\Omega^{-1}(r,\hat{s})\Gamma_n(A-rB)\right] = 0, \tag{3.22}$$

in the convergence region of the cgf (a neighborhood of 0). Due to the nature of the mgf, the endpoints of this neighborhood must satisfy

$$|I_n - 2s\Gamma_n(A-rB)| = 0,$$

or, multiplying both sides by $\left(\frac{1}{2s}\right)^n$,

$$\left|\frac{1}{2s}I_n - \Gamma_n(A-rB)\right| = 0,$$

which occurs when $\frac{1}{2s}$ is any eigenvalue of $\Gamma_n(A-rB)$. Thus $\hat{s}$ is the unique solution to (3.22) in the interval

$$\frac{1}{2\lambda_{min}(\Gamma_n(A-rB))} < \hat{s} < \frac{1}{2\lambda_{max}(\Gamma_n(A-rB))}. \tag{3.23}$$

3.3.3 The Probability Density Function (pdf)

The saddlepoint approximation to the density of $\hat{\phi}(c_1,c_2)$ at r, $\hat{f}(r)$, can be expressed in terms of the saddlepoint approximation to density of random variable W_r at 0, $\hat{f}_{W_r}(0)$, where W_r is the *constructed* random variable associated with mgf

$$M_{W_r}(s) = \frac{1}{\mathbb{E}Q_2}\left.\frac{\partial M(s,t)}{\partial t}\right|_{t=-rs} = \frac{|\Omega(r,s)|^{-1/2}}{\mathrm{Tr}\left(\Gamma_n B\right)}\mathrm{Tr}\left[\Omega^{-1}(r,s)\Gamma_n B\right], \tag{3.24}$$

which follows from (3.14) and (3.13). The relationship is

$$\hat{f}(r) = \mathbb{E}(Q_2)\hat{f}_{W_r}(0) = \frac{\mathrm{Tr}\left(\Gamma_n B\right)}{\sqrt{2\pi K''_{W_r}(\hat{s})}}\exp\{K_{W_r}(\hat{s})\}, \tag{3.25}$$

and $\hat{s}$ solves the saddlepoint equation

$$K'_{W_r}(\hat{s}) = 0,$$

in the interval defined by (3.23).

The cgf is

$$\begin{aligned} K_{W_r}(s) &= \log|\Omega(r,s)|^{-1/2} - \log \mathrm{Tr}\left[\Gamma_n B\right] + \log \mathrm{Tr}\left[\Omega^{-1}(r,s)\Gamma_n B\right] \\ &= K_{Y_r}(s) - \log \mathrm{Tr}\left[\Gamma_n B\right] + \log \mathrm{Tr}\left[\Delta(r,s)\right], \end{aligned} \tag{3.26}$$

where $\Delta(r,s) \equiv \Omega^{-1}(r,s)\Gamma_n B$. Its first derivative is from (3.12), (3.21), and Lutkepohl (1996) equation (2) of section 10.3.2

$$\begin{aligned} K'_{W_r}(s) &= K'_{Y_r}(s) + \frac{\partial \log \mathrm{Tr}\left[\Delta(r,s)\right]}{\partial s} \\ &= K'_{Y_r}(s) + \frac{1}{\mathrm{Tr}\left[\Delta(r,s)\right]} \mathrm{Tr}\left[\left(\frac{\partial \mathrm{Tr}\left[\Delta(r,s)\right]}{\partial \Omega^{-1}(r,s)}\right)' \frac{\Omega^{-1}(r,s)}{\partial s}\right] \\ &= K'_{Y_r}(s) + \frac{1}{\mathrm{Tr}\left[\Delta(r,s)\right]} \mathrm{Tr}\left[(\Gamma_n B) 2\Omega^{-1}(r,s)\Gamma_n (A - rB)\Omega^{-1}(r,s)\right] \\ &= K'_{Y_r}(s) + 2\frac{\mathrm{Tr}\left[\Omega^{-1}(r,s)D(r)\Delta(r,s)\right]}{\mathrm{Tr}\left[\Delta(r,s)\right]}, \end{aligned}$$

where $D(r) \equiv \Gamma_n(A - rB)$. Note that we have established:

$$\frac{\partial \mathrm{Tr}\left[\Delta(r,s)\right]}{\partial s} = 2\mathrm{Tr}\left[\Omega^{-1}(r,s)D(r)\Delta(r,s)\right]. \tag{3.27}$$

To obtain the second derivative, we use the quotient rule on the second term of $K'_{W_r}(s)$, to give

$$\begin{aligned} K''_{W_r}(s) = K''_{Y_r}(s) &+ \frac{2}{\mathrm{Tr}\left[\Delta(r,s)\right]} \frac{\partial \mathrm{Tr}\left[\Omega^{-1}(r,s)D(r)\Delta(r,s)\right]}{\partial s} \\ &- 2\frac{\mathrm{Tr}\left[\Omega^{-1}(r,s)D(r)\Delta(r,s)\right]}{\left(\mathrm{Tr}\left[\Delta(r,s)\right]\right)^2} \frac{\partial \mathrm{Tr}\left[\Delta(r,s)\right]}{\partial s}. \end{aligned} \tag{3.28}$$

By (3.12), (3.21), and Lutkepohl (1996) equation (21) of section 10.3.2,

$$\begin{aligned} &\frac{\partial \mathrm{Tr}\left[\Omega^{-1}(r,s)D(r)\Delta(r,s)\right]}{\partial s} \\ &= \mathrm{Tr}\left[\left(\frac{\partial \mathrm{Tr}\left[\Omega^{-1}(r,s)D(r)\Delta(r,s)\right]}{\partial \Omega^{-1}(r,s)}\right)' \frac{\Omega^{-1}(r,s)}{\partial s}\right] \\ &= \mathrm{Tr}\left[\left(D(r)\Omega^{-1}(r,s)\Gamma_n B + \Gamma_n B\Omega^{-1}(r,s)D(r)\right) 2\Omega^{-1}(r,s)D(r)\Omega^{-1}(r,s)\right] \\ &= 4\mathrm{Tr}\left[\Delta(r,s)\left(\Omega^{-1}(r,s)D(r)\right)^2\right]. \end{aligned}$$

Using (3.19) and (3.27) in (3.28), gives finally

$$\begin{aligned} K''_{W_r}(s) &= 2\mathrm{Tr}\left[\left(\Omega^{-1}(r,s)D(r)\right)^2\right] + 8\frac{\mathrm{Tr}\left[\Delta(r,s)\left(\Omega^{-1}(r,s)D(r)\right)^2\right]}{\mathrm{Tr}\left[\Delta(r,s)\right]} \\ &- 4\left(\frac{\mathrm{Tr}\left[\Omega^{-1}(r,s)D(r)\Delta(r,s)\right]}{\mathrm{Tr}\left[\Delta(r,s)\right]}\right)^2. \end{aligned} \tag{3.29}$$

Substituting for $K_{W_r}(s)$ in (3.25), we obtain the saddlepoint approximation to the density of the estimator $\hat{\phi}(c_1, c_2)$:

$$\hat{f}(r) = \frac{\mathrm{Tr}\left[\Delta(r,s)\right]}{\sqrt{2\pi|\Omega(r,s)|K''_{W_r}(s)}}, \tag{3.30}$$

where, for fixed $r \in (r_L, r_U)$, $\hat{s}$ solves the saddlepoint equation

$$\mathrm{Tr}\left[\Omega^{-1}(r,\hat{s})\Gamma_n(A - rB)\right] = 0, \tag{3.31}$$

in the neighborhood of 0 defined by (3.23).

3.4 Plots of Saddlepoint Densities

In this section we compute saddlepoint approximations to the density of the estimator $\hat{\phi}(c_1, c_2)$ of ϕ in model (3.1), with $p = 2$, and sample sizes of $n = 30, 100$. We will compare the Yule-Walker ($\hat{\phi}(1,1)$) and Burg ($\hat{\phi}(\frac{1}{2}, \frac{1}{2})$) densities, as well as the asymptotic distribution, derived in Chapter 2. We will investigate the relative shapes and locations of these densities for values of ϕ ranging from 0.5 to 0.97. The range of the support for the Yule-Walker and Burg pdfs is from (3.15): ± 0.9808 and ± 1 respectively for $n = 30$; ± 0.9981 and ± 1 respectively for $n = 100$.

For saddlepoint determination in the $n = 100$ pdf plots, we will not use all of (3.24), but retain only the portion that coincides with (3.17), i.e. take

$$M_{W_r}(s) \simeq |\Omega(r,s)|^{-1/2}.$$

This greatly speeds up the computations since inverting matrices of size 100 at each iteration of the saddlepoint finding routine makes for a very slow program. Another advantage is that the saddlepoint $\hat{s}$ is the same for both cdf and pdf approximations, while the accuracy lost in the pdf plots by using this incorrect saddlepoint is essentially negligible.

Recall that both Yule-Walker and Burg have the same asymptotic distribution under model (3.1). For the univariate case being considered, this distribution is, from (2.17)

$$\sqrt{n}\left(\hat{\phi}(c_1, c_2) - \phi\right) \xrightarrow{d} \mathrm{N}\left(0, \frac{\sigma^2}{\gamma(0)}\right).$$

Since $\gamma(0) = \frac{\sigma^2}{(1-\phi^2)}$, we have

$$\begin{aligned}\sqrt{n}\left(\hat{\phi}(c_1, c_2) - \phi\right) &\xrightarrow{d} \mathrm{N}(0, 1-\phi^2), \\ \implies \hat{\phi}(c_1, c_2) &\sim \mathrm{AN}\left(\phi, \frac{1-\phi^2}{n}\right),\end{aligned}$$

with corresponding asymptotically normal density

$$f_{AN}(r) = \sqrt{\frac{n}{2\pi(1-\phi^2)}} \exp\left\{-\frac{n(r-\phi)^2}{2(1-\phi^2)}\right\}. \tag{3.32}$$

Referring to figures 3.2 and 3.3, we see that for ϕ far from 1, all three estimators have very similar densities, particularly at the larger sample size. As we gradually approach 1, we observe the mode of the Yule-Walker density occurring at smaller values of r relative to Burg and the asymptotic distribution (particularly evident at smaller sample sizes). Due to the left-skewness of Yule-Walker and Burg, this offset in the modes means that the former estimator has a larger bias than the latter. The Yule-Walker density is also substantially flatter than Burg at higher values of ϕ, indicative of a larger variability in the estimates. These findings are in agreement with what was observed in the simulations of chapter one.

Figure 3.2: Saddlepoint approximations to the densities of the estimators $\hat{\phi}(1,1)$ (Yule-Walker, dotted) and $\hat{\phi}(\frac{1}{2},\frac{1}{2})$ (Burg, dashed) of the autoregressive coefficient phi (ϕ) of model (3.1), with $p = 2$, and sample size 30. The asymptotic distribution (3.32) is shown in solid lines.

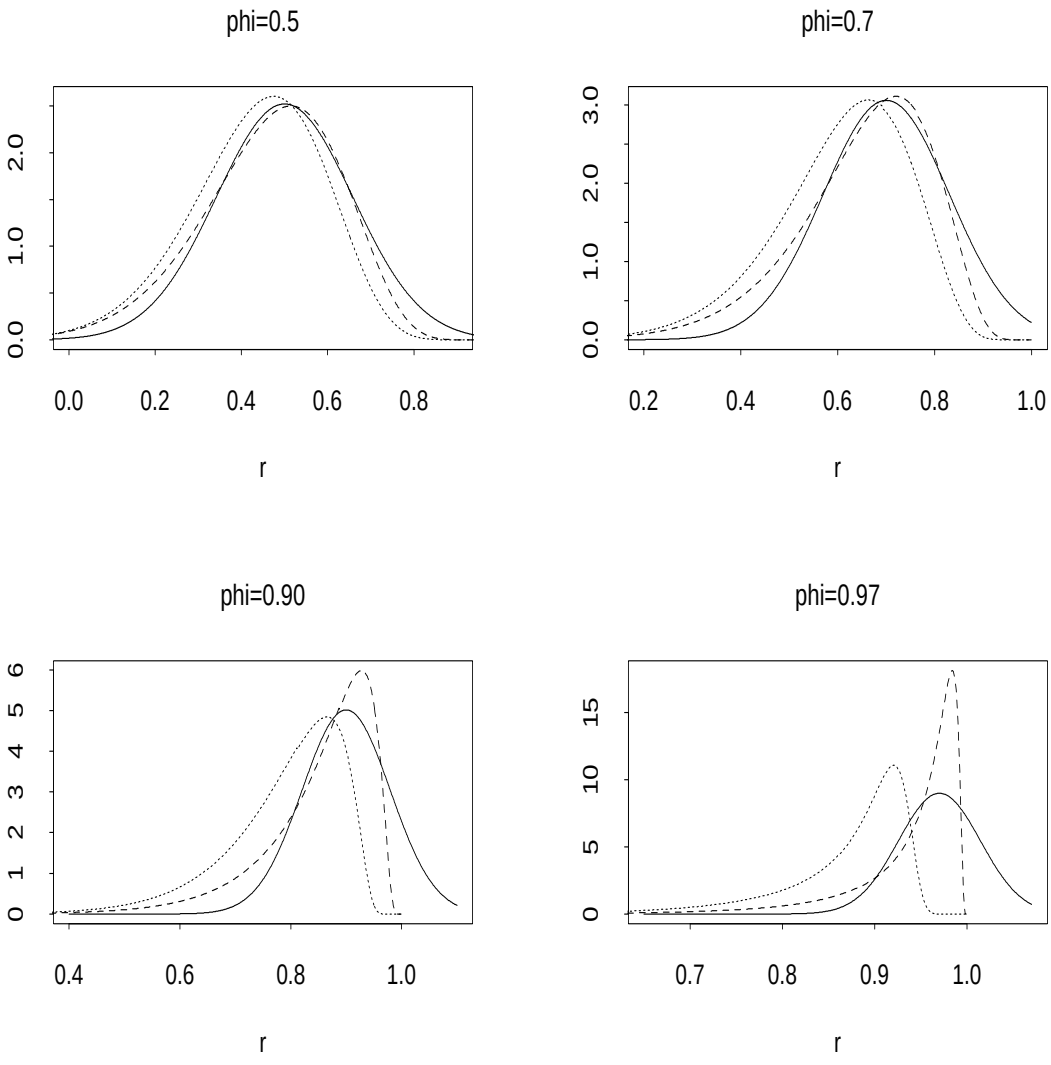
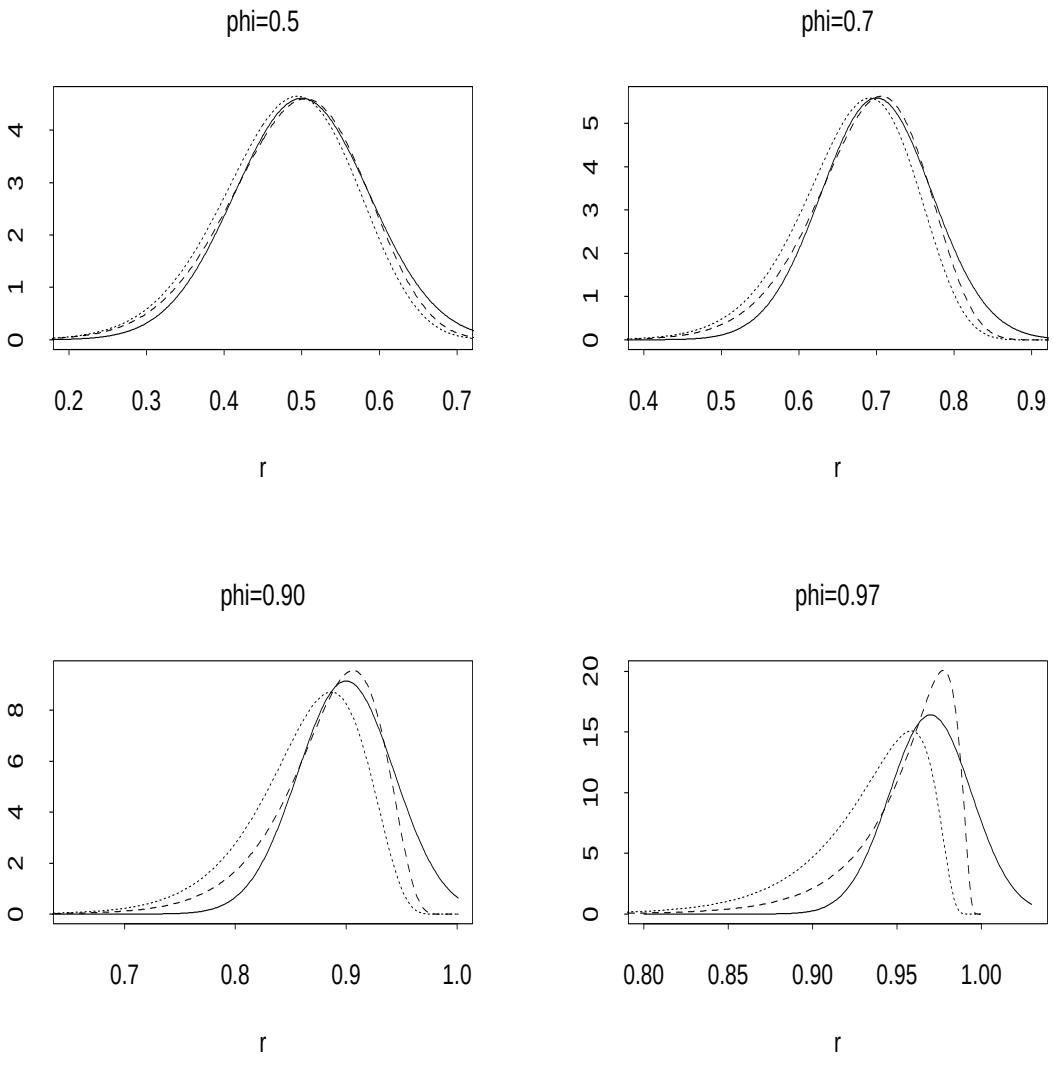


Figure 3.3: Saddlepoint approximations to the densities of the estimators $\hat{\phi}(1,1)$ (Yule-Walker, dotted) and $\hat{\phi}(\frac{1}{2},\frac{1}{2})$ (Burg, dashed) of the autoregressive coefficient phi (ϕ) of model (3.1), with $p = 2$, and sample size 100. The asymptotic distribution (3.32) is shown in solid lines.



3.5 Plots of Simulated Densities

In this section we undertake a large simulation study of the probability densities of the following estimators of ϕ in model (3.1): Yule-Walker ($\hat{\phi}(1,1)$), Burg ($\hat{\phi}(\frac{1}{2},\frac{1}{2})$), and Maximum Likelihood ($\hat{\phi}_{ML}$). As in the previous section, we will concentrate on models with $p = 2$, sample sizes of $n = 30, 100$, and values of ϕ ranging from 0.5 to 0.97. The densities are estimated by simulating $100,000$ realizations from each model, computing each respective estimator, and plotting the frequency of occurrence of each as a histogram scaled to be a probability density (the sum of the bar heights times the bar widths equal to 1). We overlay the Yule-Walker and Burg estimator histograms with the saddlepoint approximations to the pdfs of their respective distributions.

Referring to figures 3.4 - 3.7, we see that for ϕ far from 1, the densities of the three estimators are nearly coincidental. As we gradually approach 1 though, the salient feature is the way in which the Burg and Maximum Likelihood density curves remain very close together, while Yule-Walker tends to gain increasing bias and variance, particularly at lower sample sizes. This agrees with the tendency noted in chapter one of Burg to produce estimates of ϕ with a consistently higher likelihood. The saddlepoint approximation to the pdf of the Yule-Walker and Burg estimators agrees closely with the simulated pdfs.

3.6 Assessing the Accuracy of the Saddlepoint Approximations

In this section we compare the saddlepoint approximations of the cdf and pdf of the Yule-Walker and Burg estimators of ϕ in model 3.1, with simulated values. Due to the problems associated with density estimation, such a comparison is more appropriately carried out for the cdf than the pdf.

Figure 3.4: Probability density histograms of the distributions of the estimators $\hat{\phi}(1,1)$ (Yule-Walker, top), $\hat{\phi}(\frac{1}{2},\frac{1}{2})$ (Burg, middle), and $\hat{\phi}_{ML}$ (Maximum Likelihood, bottom), of the AR coefficient of model (3.1), with $p = 2$. The Yule-Walker and Burg histograms are overlaid with their respective saddlepoint approximations. Each histogram is based on $100,000$ simulated realizations, each of sample size 30. The realizations on the left side of the figure were generated from a model with $\phi = 0.5$, and on the right from one with $\phi = 0.7$.

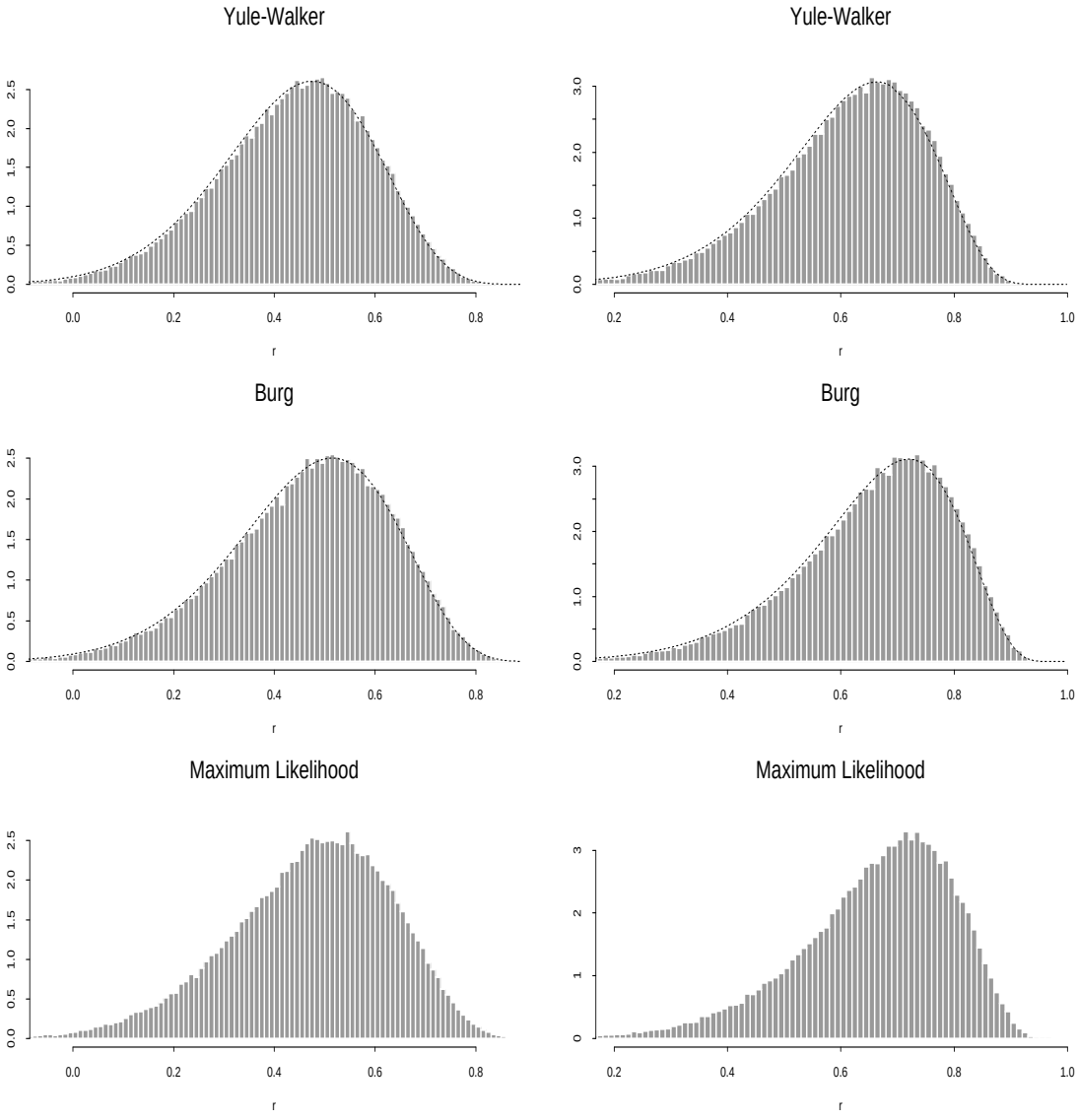


Figure 3.5: Probability density histograms of the distributions of the estimators $\hat{\phi}(1,1)$ (Yule-Walker, top), $\hat{\phi}(\frac{1}{2},\frac{1}{2})$ (Burg, middle), and $\hat{\phi}_{ML}$ (Maximum Likelihood, bottom), of the AR coefficient of model (3.1), with $p = 2$. The Yule-Walker and Burg histograms are overlaid with their respective saddlepoint approximations. Each histogram is based on $100,000$ simulated realizations, each of sample size 30. The realizations on the left side of the figure were generated from a model with $\phi = 0.9$, and on the right from one with $\phi = 0.97$.

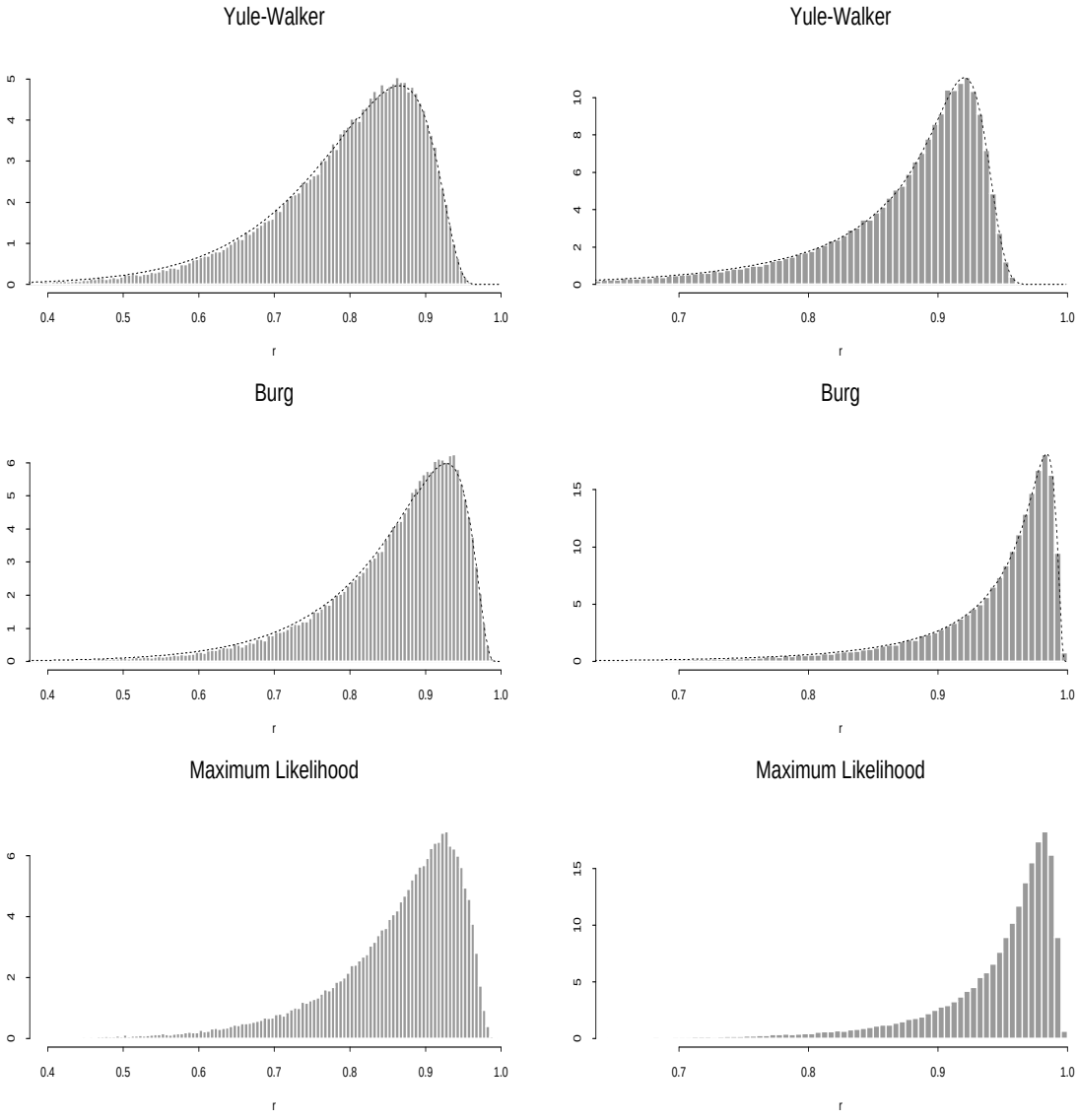


Figure 3.6: Probability density histograms of the distributions of the estimators $\hat{\phi}(1,1)$ (Yule-Walker, top), $\hat{\phi}(\frac{1}{2},\frac{1}{2})$ (Burg, middle), and $\hat{\phi}_{ML}$ (Maximum Likelihood, bottom), of the AR coefficient of model (3.1), with $p = 2$. The Yule-Walker and Burg histograms are overlaid with their respective saddlepoint approximations. Each histogram is based on $100,000$ simulated realizations, each of sample size 100. The realizations on the left side of the figure were generated from a model with $\phi = 0.5$, and on the right from one with $\phi = 0.7$.

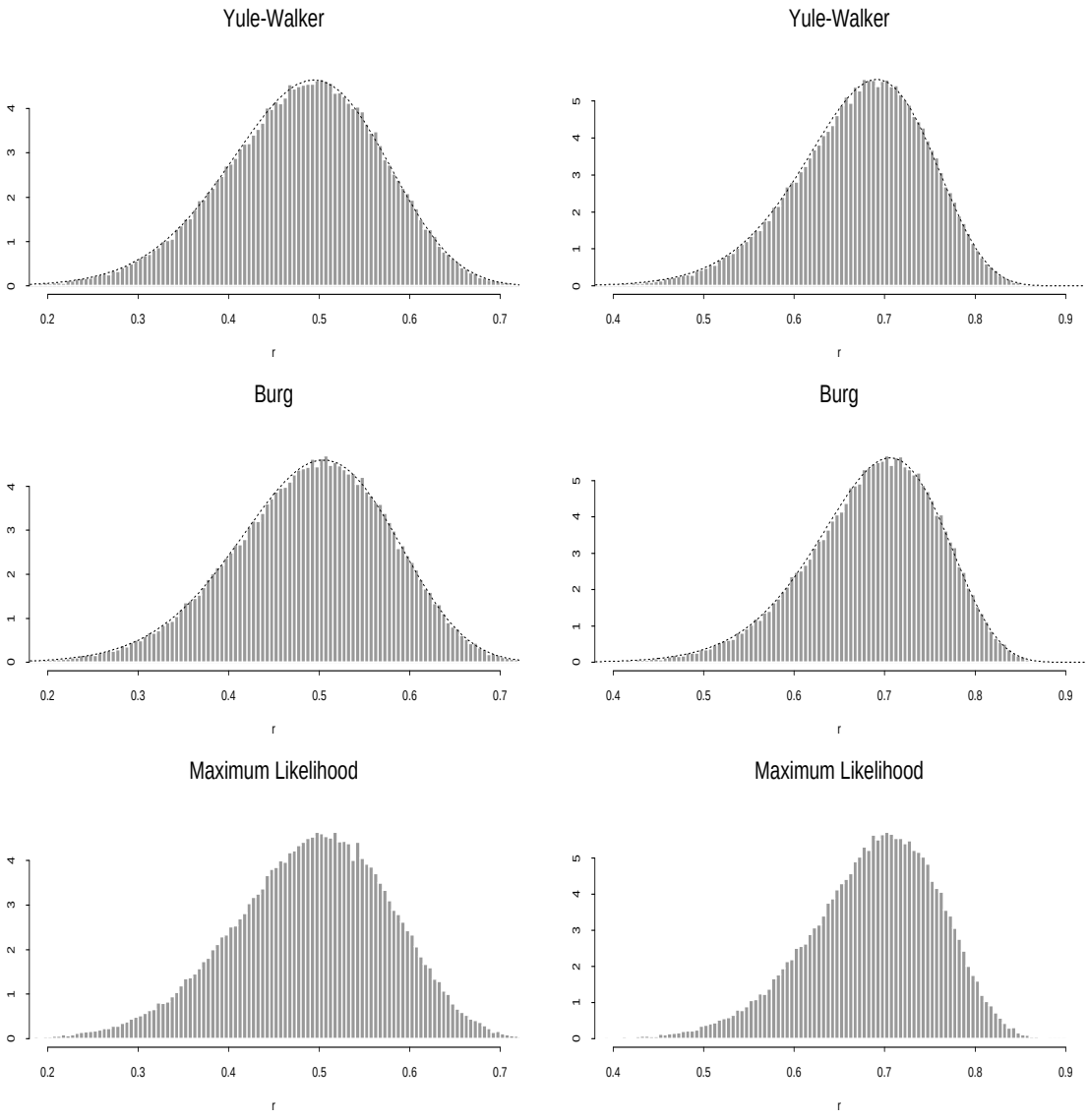
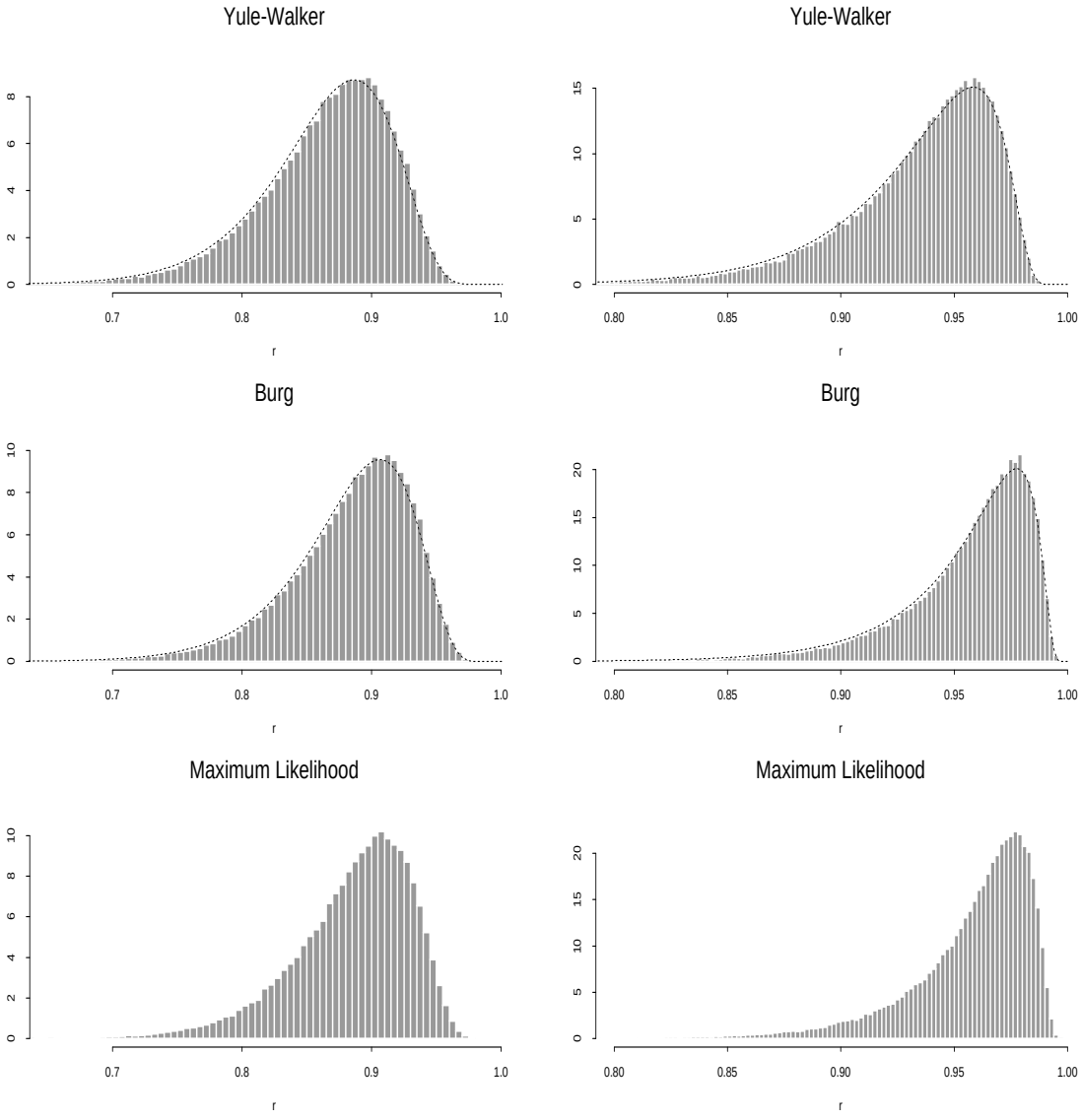


Figure 3.7: Probability density histograms of the distributions of the estimators $\hat{\phi}(1,1)$ (Yule-Walker, top), $\hat{\phi}(\frac{1}{2},\frac{1}{2})$ (Burg, middle), and $\hat{\phi}_{ML}$ (Maximum Likelihood, bottom), of the AR coefficient of model (3.1), with $p = 2$. The Yule-Walker and Burg histograms are overlaid with their respective saddlepoint approximations. Each histogram is based on $100,000$ simulated realizations, each of sample size 100. The realizations on the left side of the figure were generated from a model with $\phi = 0.9$, and on the right from one with $\phi = 0.97$.



A technique common in the saddlepoint literature, is the comparison of the *percent relative error* (PRE) in the cdfs. Denote by $\hat{F}_{sim}(r)$ and $\hat{F}_{sad}(r)$, the estimates of the true cdf $F(r)$ of the AR coefficient estimator $\hat{\phi}(c_1, c_2)$ under model 3.1, obtained via simulations and saddlepoint approximations, respectively. For $\hat{F}_{sim}(r)$ we will take the traditional empirical cdf estimator, ie. the proportion of realizations whose value is less than or equal to r. With this notation, we define the PRE at the quantile r as:

$$\text{PRE} = \begin{cases} \frac{\hat{F}_{sad}(r)-\hat{F}_{sim}(r)}{\hat{F}_{sim}(r)}100, & \hat{F}_{sim}(r) \leq 0.5, \\ \\ \frac{(1-\hat{F}_{sad}(r))-(1-\hat{F}_{sim}(r))}{1-\hat{F}_{sim}(r)}100, & \hat{F}_{sim}(r) > 0.5. \end{cases}$$

Thus, larger absolute values of PRE denote larger discrepancies between the saddlepoint approximation and simulations, while a PRE value of 0 indicates perfect agreement.

The results, presented in figures 3.8 - 3.13, show PREs generally falling in the range of $\pm 5\%$ for sample size 30, and $\pm 2\%$ for sample size 100, with somewhat higher values in the tails of the distributions. On the whole, the saddlepoint approximation for this estimator expressible as a ratio of quadratic forms in normal random variables is fairly accurate.

It is interesting to assess the robustness of the accuracy of the saddlepoint approximations under certain types of model misspecification, such as might occur when the data follows a process driven by heavy-tailed noise. In figures 3.14 - 3.19, we see what happens when the saddlepoint approximations to the sampling distributions of the Yule-Walker and Burg estimators of ϕ under Gaussian model (3.1), are used to approximate the same when the data is simulated from a process whose driving white noise follows a double exponential (Laplace) distribution. The results are similar to those of the previous examples near the center of the distributions,

Figure 3.8: Comparisons of the saddlepoint approximations with simulations for the sampling distributions of the Yule-Walker ($\hat{\phi}(1,1)$, left) and Burg ($\hat{\phi}(\frac{1}{2},\frac{1}{2})$, right) estimators of the AR coefficient $\phi = 0.5$ of model (3.1), with $p = 2$, and sample size 30. The top figures show the saddlepoint pdfs (solid) and kernel density estimates of the simulated pdfs (dashed). The bottom figures show the corresponding PREs in comparing the saddlepoint to the simulated cdfs. The estimation is based on $100,000$ simulated realizations.

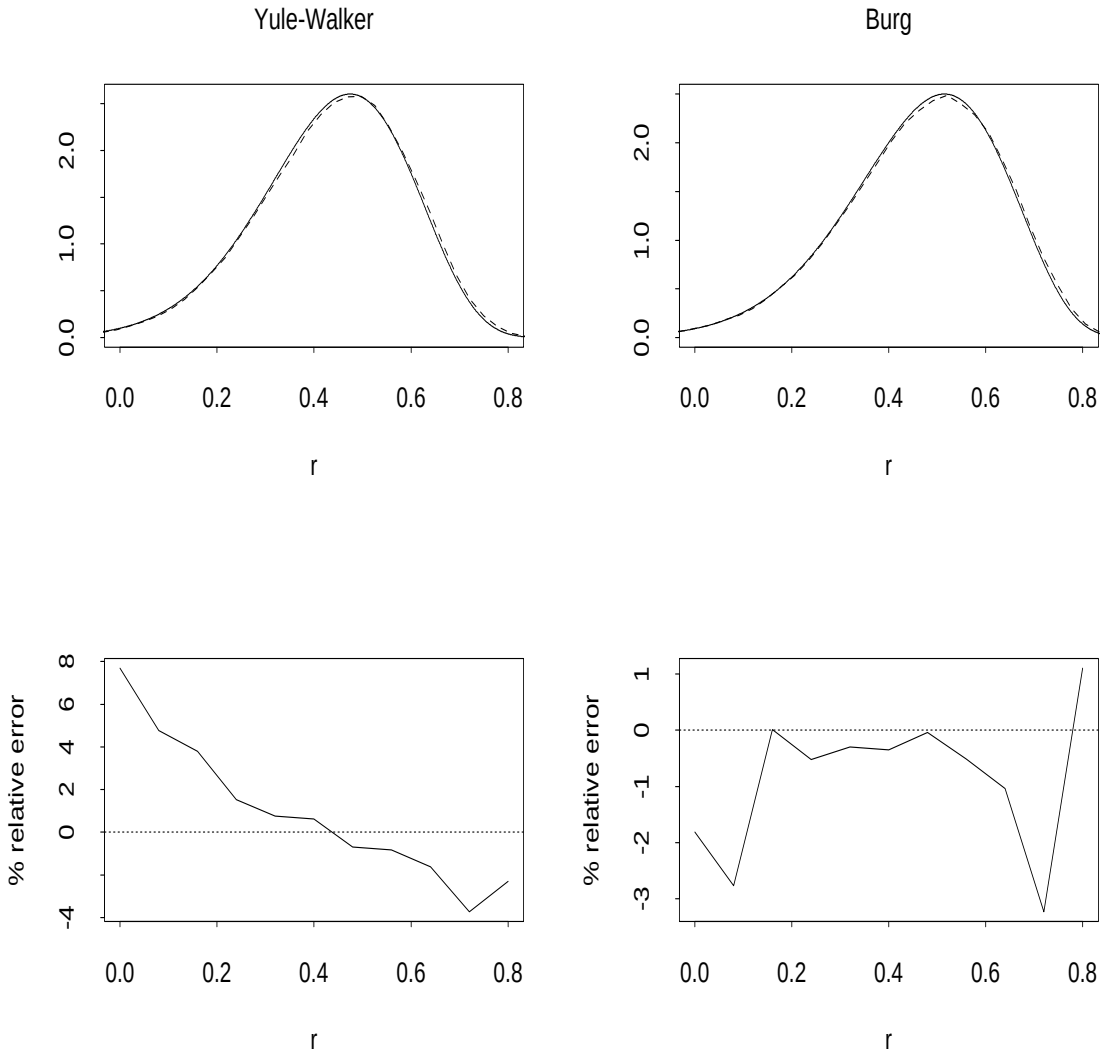


Figure 3.9: Comparisons of the saddlepoint approximations with simulations for the sampling distributions of the Yule-Walker ($\hat{\phi}(1,1)$, left) and Burg ($\hat{\phi}(\frac{1}{2},\frac{1}{2})$, right) estimators of the AR coefficient $\phi = 0.7$ of model (3.1), with $p = 2$, and sample size 30. The top figures show the saddlepoint pdfs (solid) and kernel density estimates of the simulated pdfs (dashed). The bottom figures show the corresponding PREs in comparing the saddlepoint to the simulated cdfs. The estimation is based on $100,000$ simulated realizations.

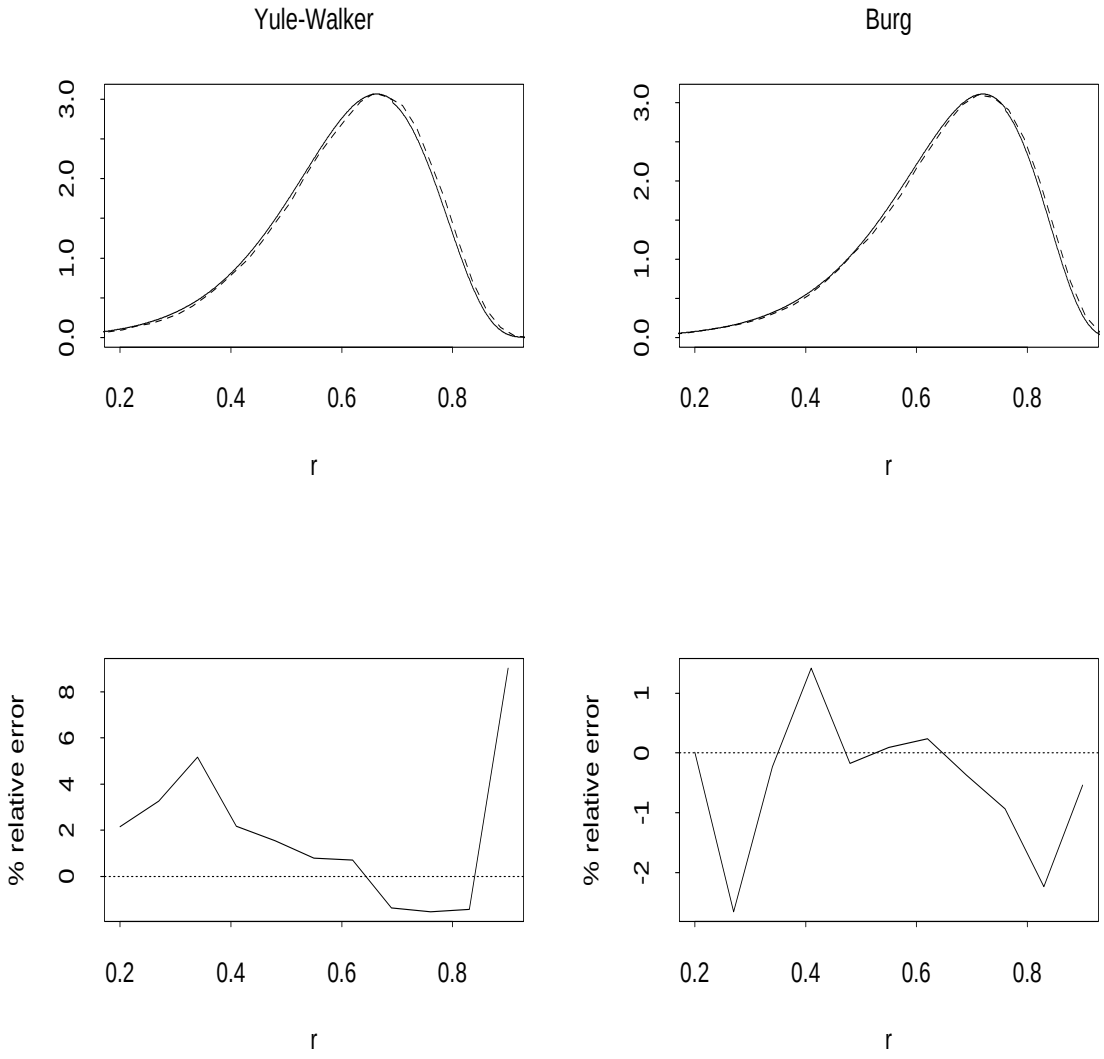


Figure 3.10: Comparisons of the saddlepoint approximations with simulations for the sampling distributions of the Yule-Walker ($\hat{\phi}(1,1)$, left) and Burg ($\hat{\phi}(\frac{1}{2},\frac{1}{2})$, right) estimators of the AR coefficient $\phi = 0.9$ of model (3.1), with $p = 2$, and sample size 30. The top figures show the saddlepoint pdfs (solid) and kernel density estimates of the simulated pdfs (dashed). The bottom figures show the corresponding PREs in comparing the saddlepoint to the simulated cdfs. The estimation is based on $100,000$ simulated realizations.

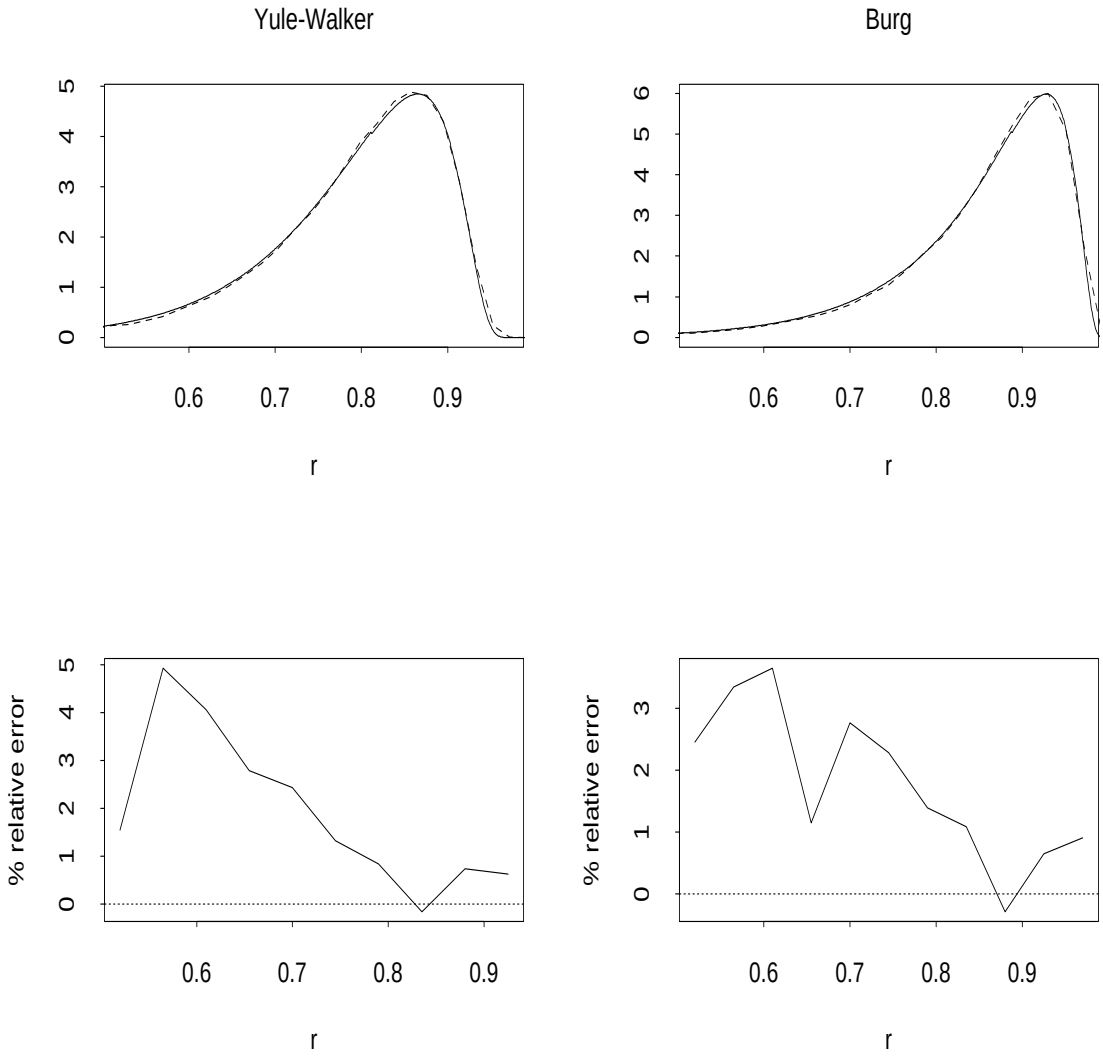


Figure 3.11: Comparisons of the saddlepoint approximations with simulations for the sampling distributions of the Yule-Walker ($\hat{\phi}(1,1)$, left) and Burg ($\hat{\phi}(\frac{1}{2},\frac{1}{2})$, right) estimators of the AR coefficient $\phi = 0.5$ of model (3.1), with $p = 2$, and sample size 100. The top figures show the saddlepoint pdfs (solid) and kernel density estimates of the simulated pdfs (dashed). The bottom figures show the corresponding PREs in comparing the saddlepoint to the simulated cdfs. The estimation is based on $100,000$ simulated realizations.

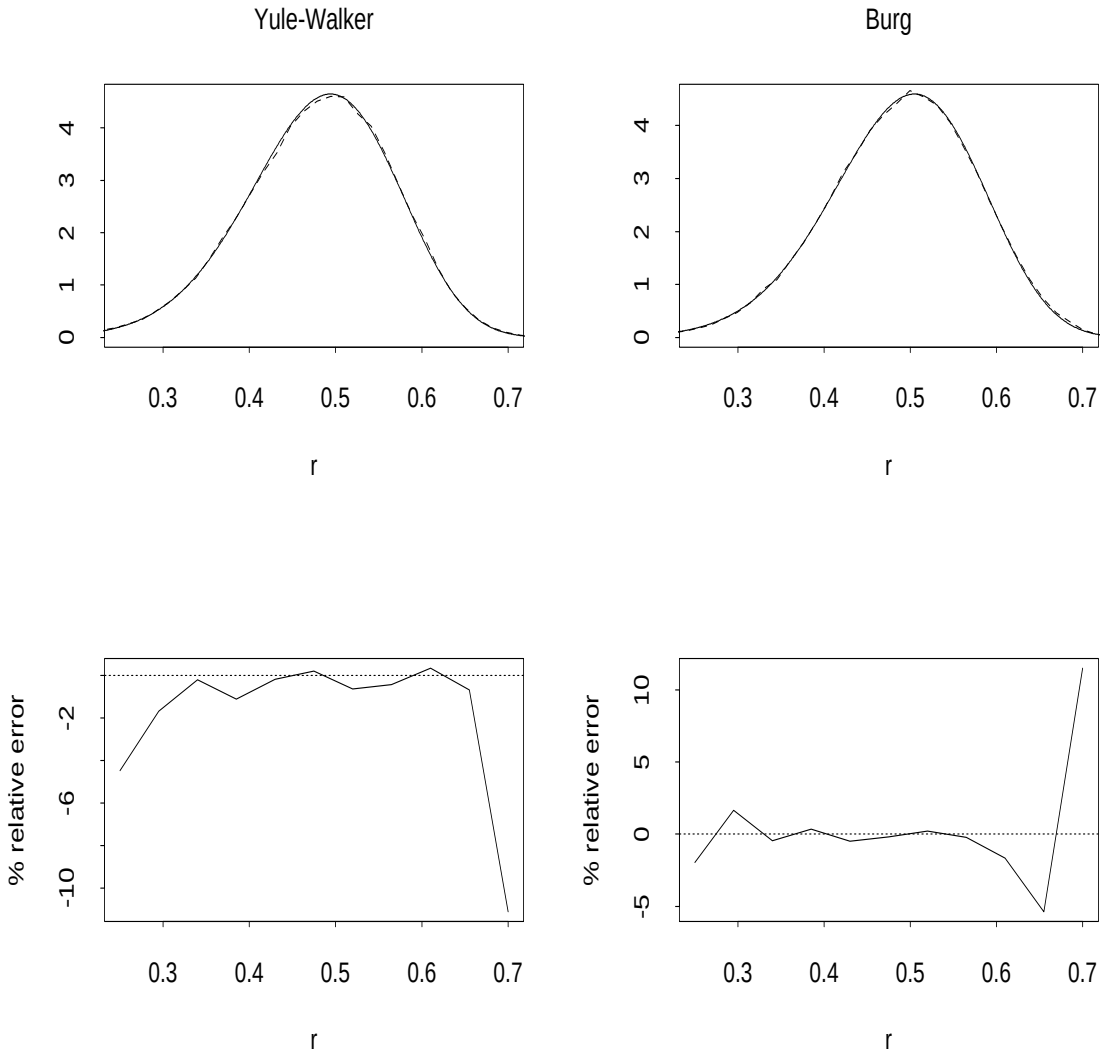


Figure 3.12: Comparisons of the saddlepoint approximations with simulations for the sampling distributions of the Yule-Walker ($\hat{\phi}(1,1)$, left) and Burg ($\hat{\phi}(\frac{1}{2},\frac{1}{2})$, right) estimators of the AR coefficient $\phi = 0.7$ of model (3.1), with $p = 2$, and sample size 100. The top figures show the saddlepoint pdfs (solid) and kernel density estimates of the simulated pdfs (dashed). The bottom figures show the corresponding PREs in comparing the saddlepoint to the simulated cdfs. The estimation is based on $100,000$ simulated realizations.

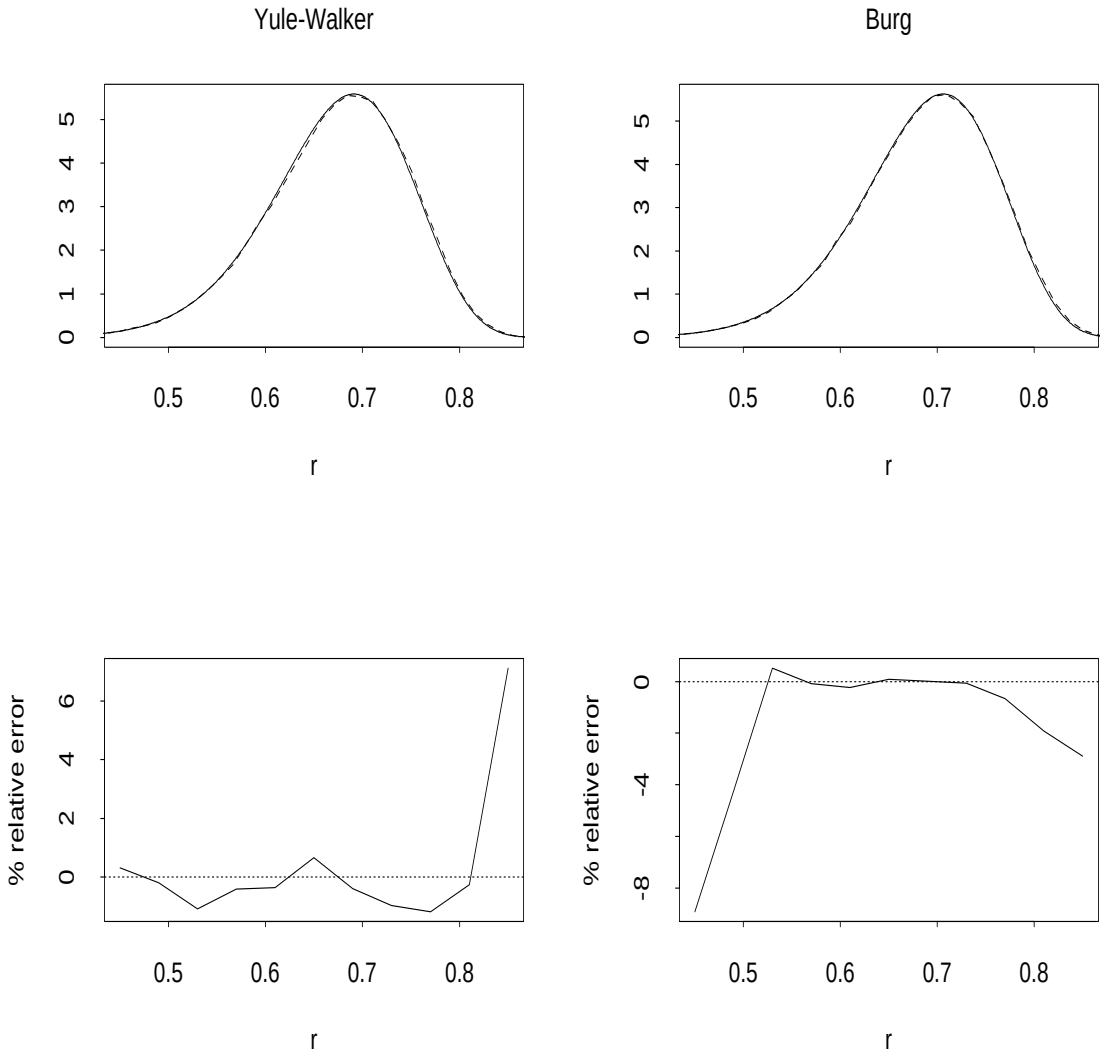
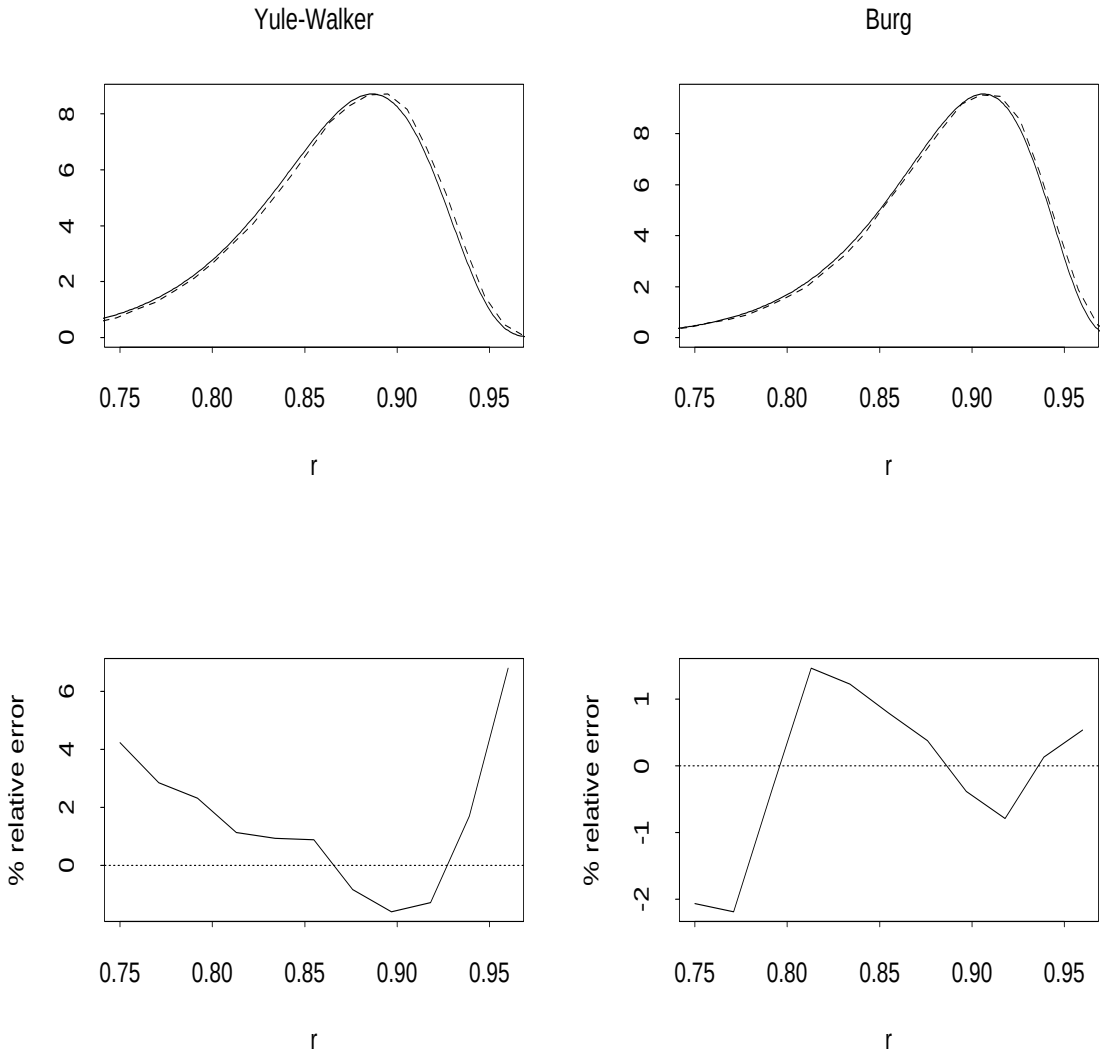


Figure 3.13: Comparisons of the saddlepoint approximations with simulations for the sampling distributions of the Yule-Walker ($\hat{\phi}(1,1)$, left) and Burg ($\hat{\phi}(\frac{1}{2},\frac{1}{2})$, right) estimators of the AR coefficient $\phi = 0.9$ of model (3.1), with $p = 2$, and sample size 100. The top figures show the saddlepoint pdfs (solid) and kernel density estimates of the simulated pdfs (dashed). The bottom figures show the corresponding PREs in comparing the saddlepoint to the simulated cdfs. The estimation is based on $100,000$ simulated realizations.



but tend to be larger than their counterparts in the tails. Overall the saddlepoint approximation is fairly robust in this particular scenario, but this is perhaps not too surprising given that the sampling distributions of the estimators under the Gaussian and Laplace noise models are similar.

Figure 3.14: Comparisons of the saddlepoint approximations with simulations for the sampling distributions of the Yule-Walker ($\hat{\phi}(1,1)$, left) and Burg ($\hat{\phi}(\frac{1}{2},\frac{1}{2})$, right) estimators of the AR coefficient $\phi = 0.5$ of model (3.1), with $p = 2$, and sample size 30. The top figures show the saddlepoint pdfs (solid) and kernel density estimates of the simulated pdfs (dashed). The bottom figures show the corresponding PREs in comparing the saddlepoint to the simulated cdfs. The empirical pdfs and cdfs are based on $100,000$ realizations, simulated from a model driven by Laplace noise.

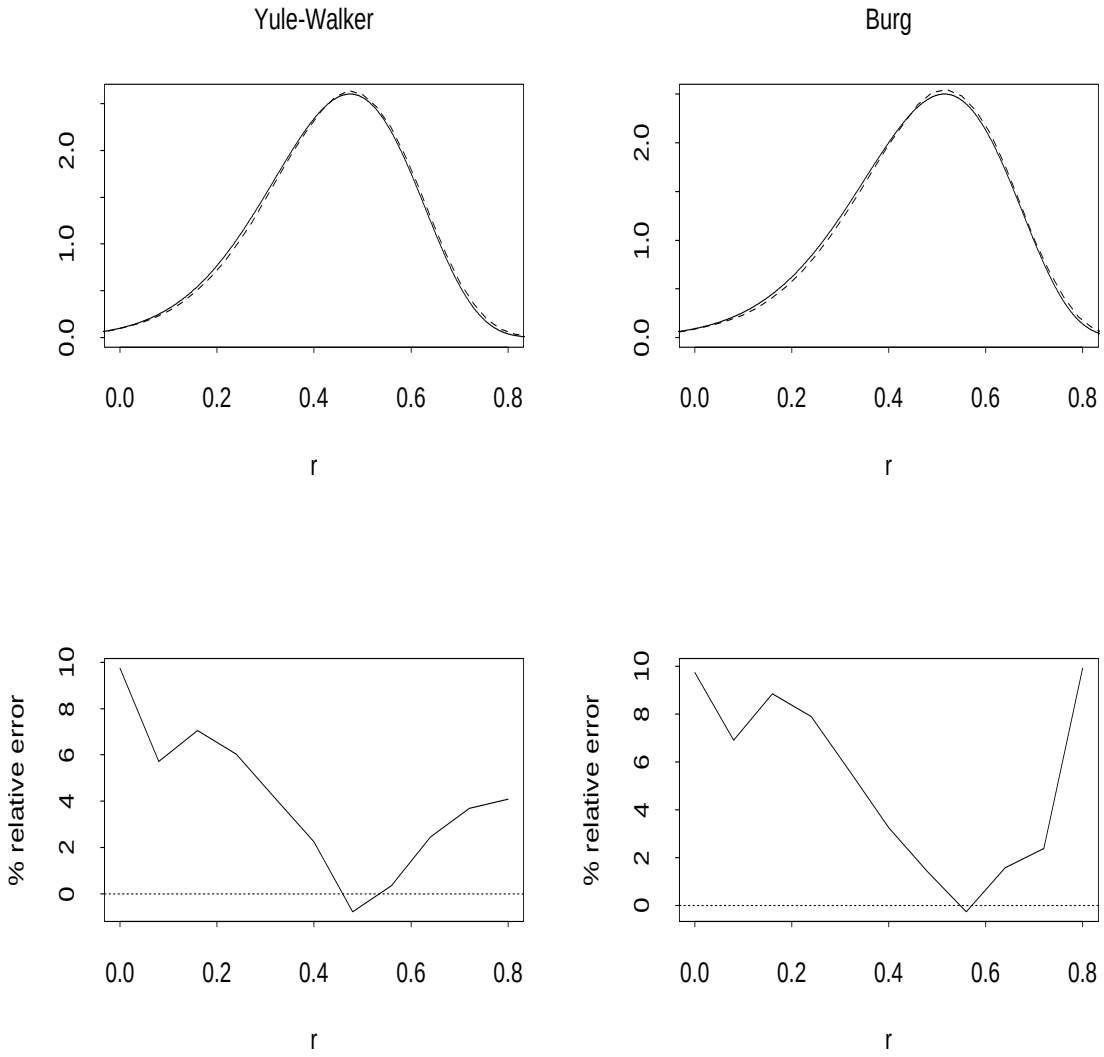


Figure 3.15: Comparisons of the saddlepoint approximations with simulations for the sampling distributions of the Yule-Walker ($\hat{\phi}(1,1)$, left) and Burg ($\hat{\phi}(\frac{1}{2},\frac{1}{2})$, right) estimators of the AR coefficient $\phi = 0.7$ of model (3.1), with $p = 2$, and sample size 30. The top figures show the saddlepoint pdfs (solid) and kernel density estimates of the simulated pdfs (dashed). The bottom figures show the corresponding PREs in comparing the saddlepoint to the simulated cdfs. The empirical pdfs and cdfs are based on $100,000$ realizations, simulated from a model driven by Laplace noise.

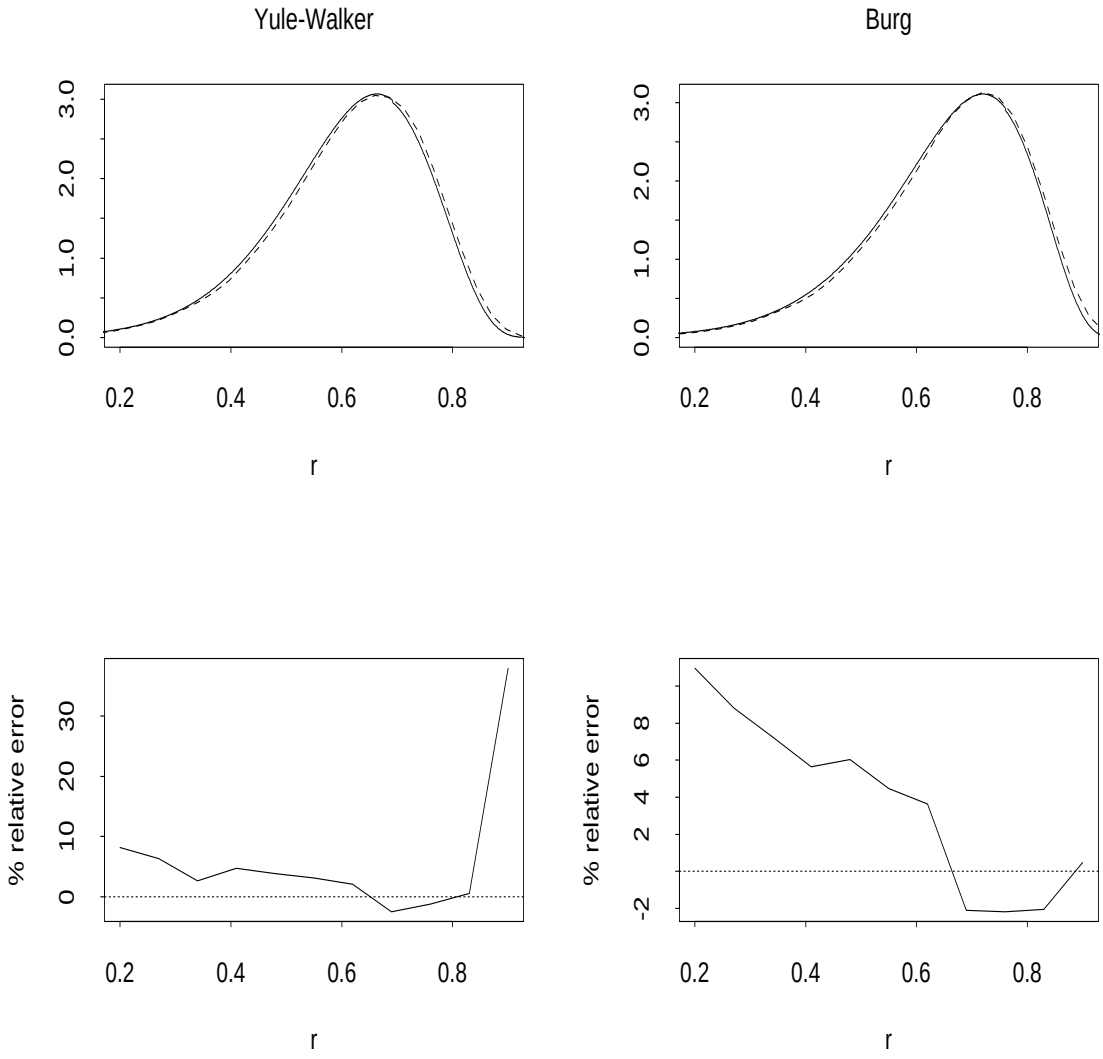


Figure 3.16: Comparisons of the saddlepoint approximations with simulations for the sampling distributions of the Yule-Walker ($\hat{\phi}(1,1)$, left) and Burg ($\hat{\phi}(\frac{1}{2},\frac{1}{2})$, right) estimators of the AR coefficient $\phi = 0.9$ of model (3.1), with $p = 2$, and sample size 30. The top figures show the saddlepoint pdfs (solid) and kernel density estimates of the simulated pdfs (dashed). The bottom figures show the corresponding PREs in comparing the saddlepoint to the simulated cdfs. The empirical pdfs and cdfs are based on $100,000$ realizations, simulated from a model driven by Laplace noise.

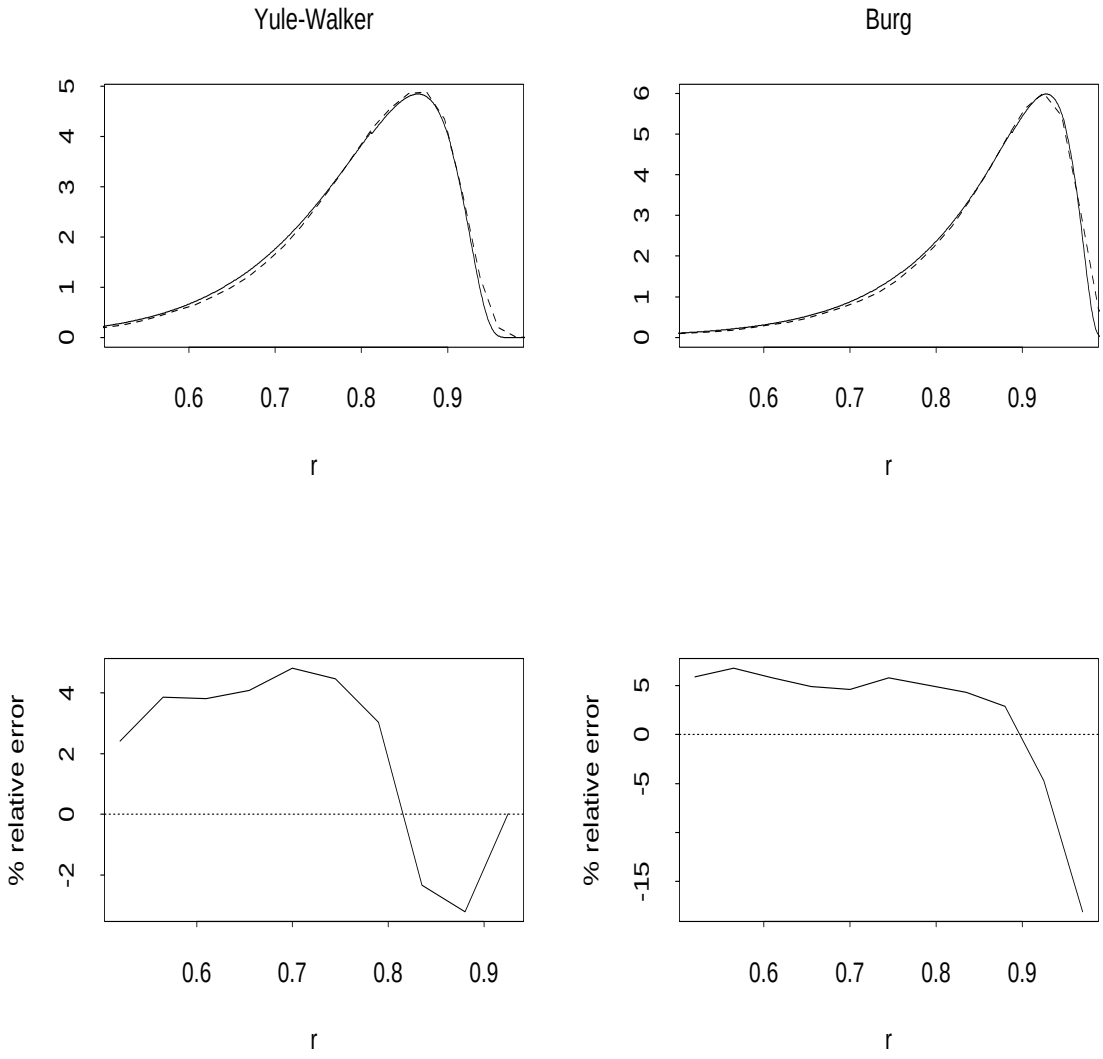


Figure 3.17: Comparisons of the saddlepoint approximations with simulations for the sampling distributions of the Yule-Walker ($\hat{\phi}(1,1)$, left) and Burg ($\hat{\phi}(\frac{1}{2},\frac{1}{2})$, right) estimators of the AR coefficient $\phi = 0.5$ of model (3.1), with $p = 2$, and sample size 100. The top figures show the saddlepoint pdfs (solid) and kernel density estimates of the simulated pdfs (dashed). The bottom figures show the corresponding PREs in comparing the saddlepoint to the simulated cdfs. The empirical pdfs and cdfs are based on $100,000$ realizations, simulated from a model driven by Laplace noise.

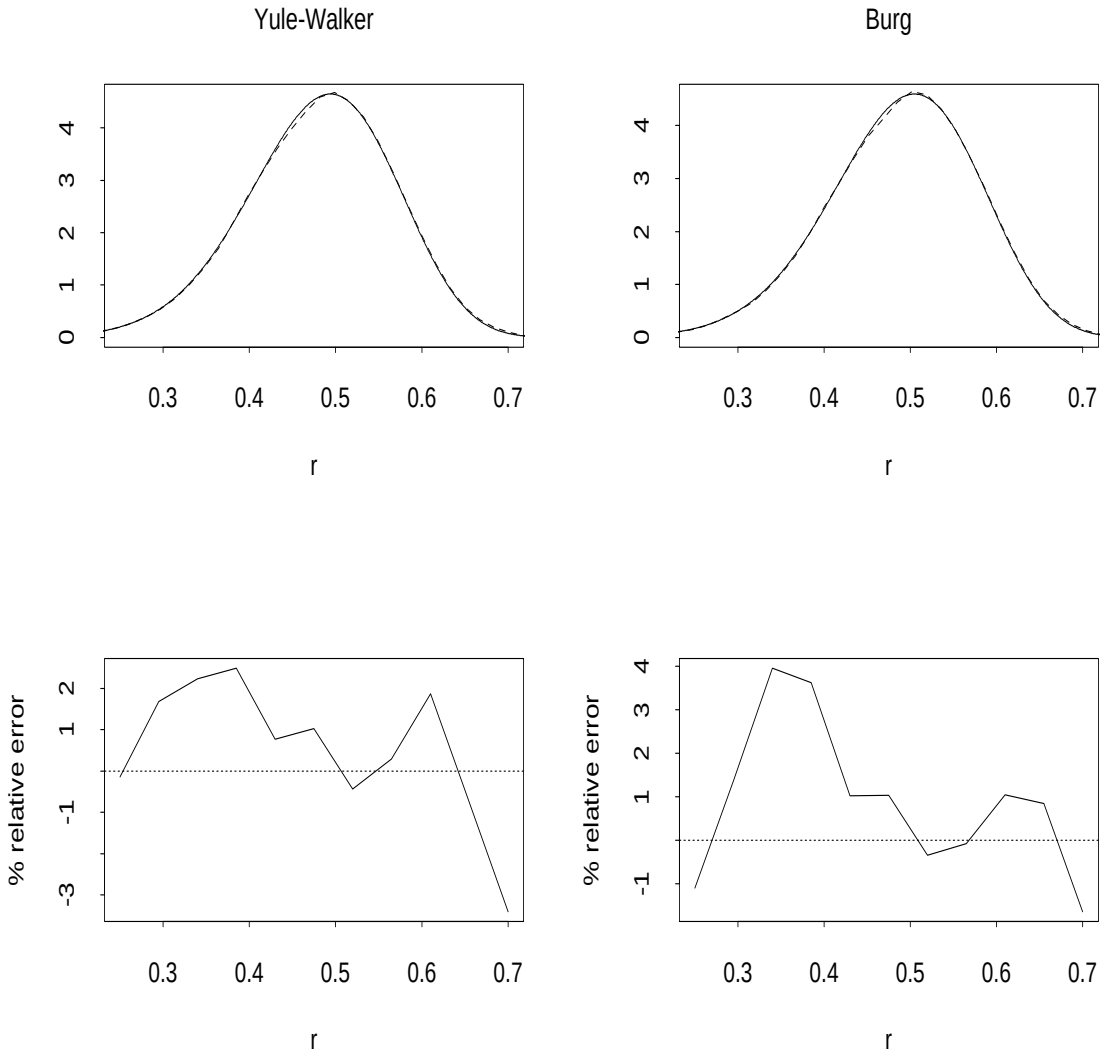


Figure 3.18: Comparisons of the saddlepoint approximations with simulations for the sampling distributions of the Yule-Walker ($\hat{\phi}(1,1)$, left) and Burg ($\hat{\phi}(\frac{1}{2},\frac{1}{2})$, right) estimators of the AR coefficient $\phi = 0.7$ of model (3.1), with $p = 2$, and sample size 100. The top figures show the saddlepoint pdfs (solid) and kernel density estimates of the simulated pdfs (dashed). The bottom figures show the corresponding PREs in comparing the saddlepoint to the simulated cdfs. The empirical pdfs and cdfs are based on $100,000$ realizations, simulated from a model driven by Laplace noise.

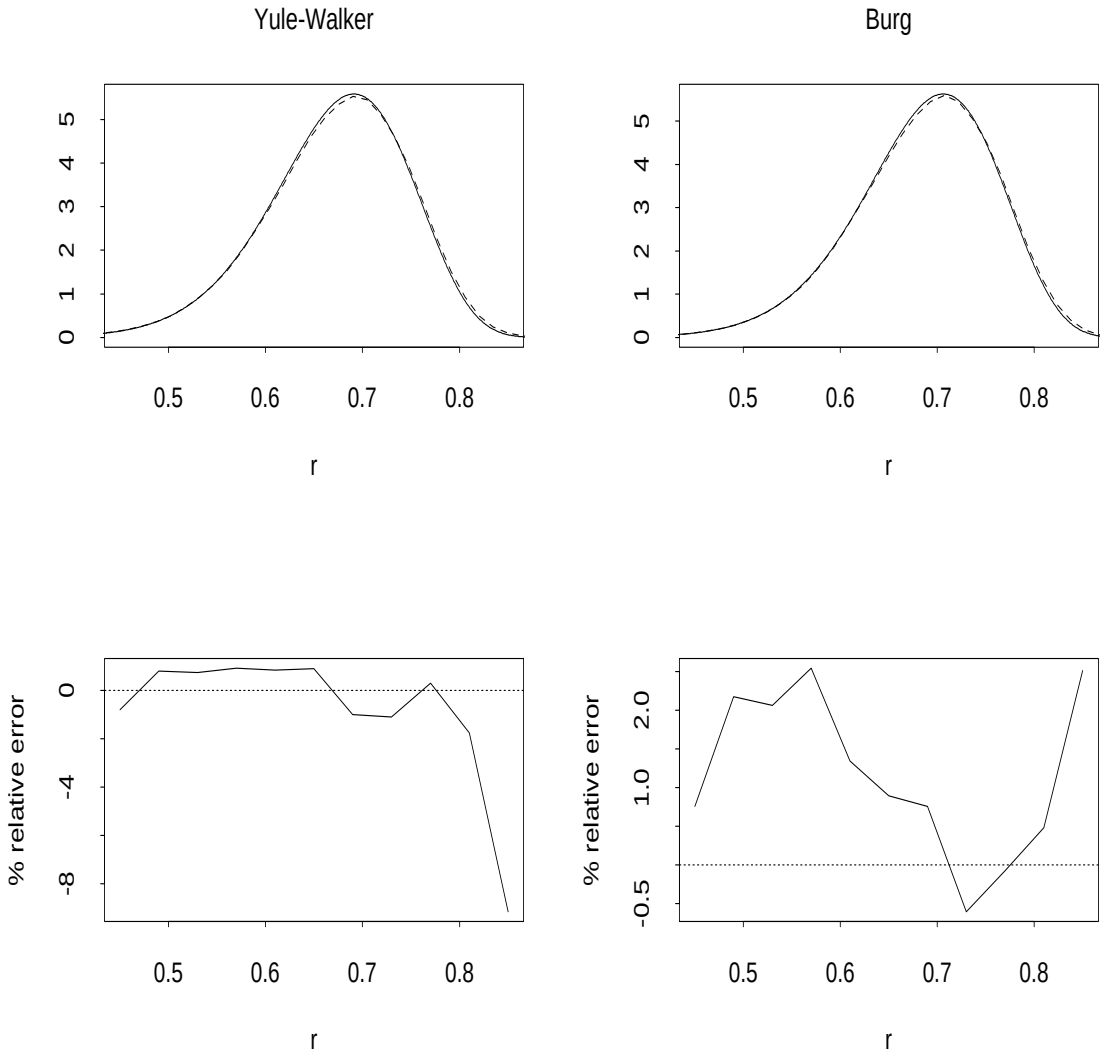
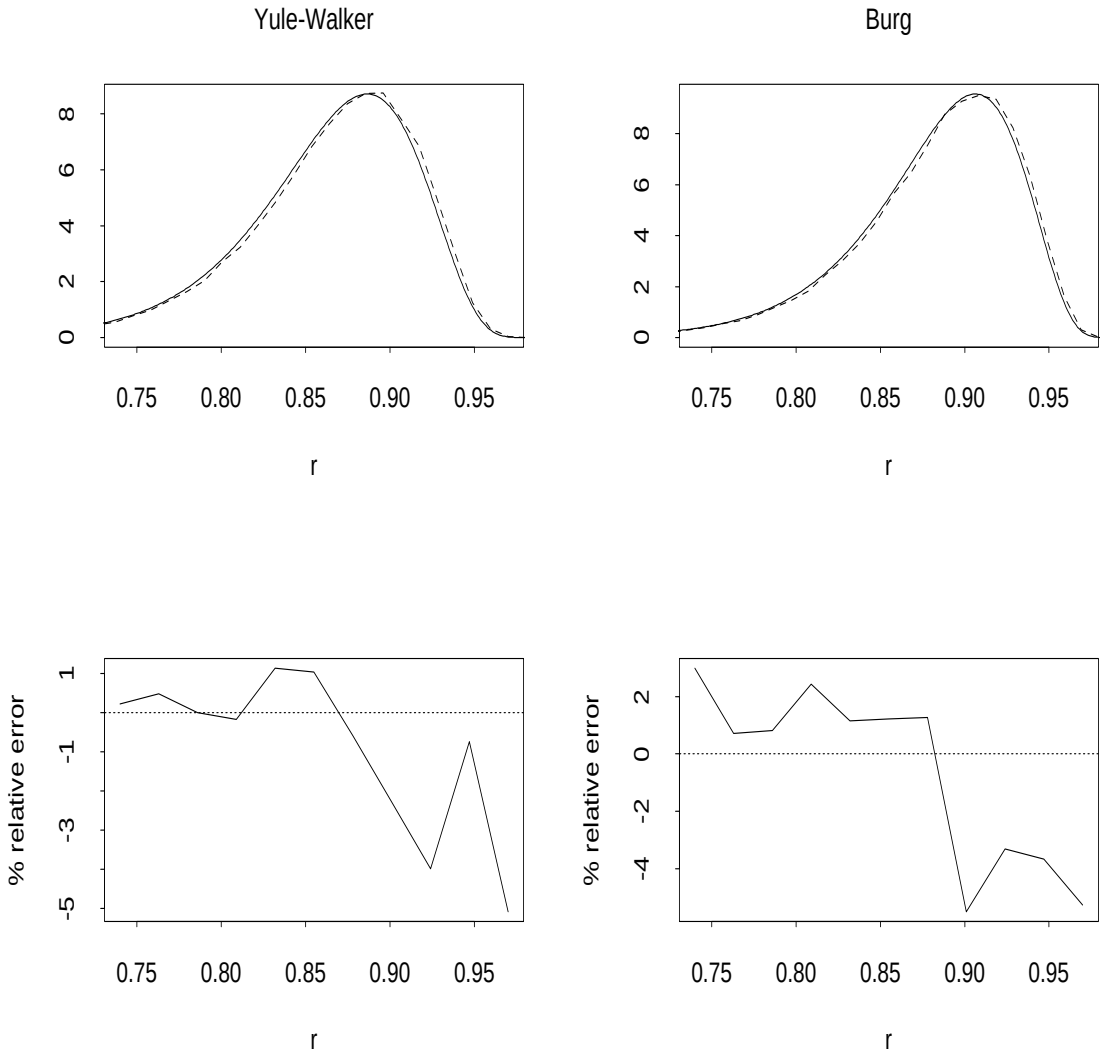


Figure 3.19: Comparisons of the saddlepoint approximations with simulations for the sampling distributions of the Yule-Walker ($\hat{\phi}(1,1)$, left) and Burg ($\hat{\phi}(\frac{1}{2},\frac{1}{2})$, right) estimators of the AR coefficient $\phi = 0.9$ of model (3.1), with $p = 2$, and sample size 100. The top figures show the saddlepoint pdfs (solid) and kernel density estimates of the simulated pdfs (dashed). The bottom figures show the corresponding PREs in comparing the saddlepoint to the simulated cdfs. The empirical pdfs and cdfs are based on $100,000$ realizations, simulated from a model driven by Laplace noise.



Appendix A

SOME MATRIX RESULTS AND IDENTITIES

A.1 Matrix calculus

In this section we state some matrix calculus results that will enable us to carry out the minimization of (1.22) with respect to X. All results are adapted from Magnus and Neudecker (1999).

A.1.1 Results

Definition A.1.1 *Given the matrix function $F(X)$, the* Jacobian Matrix *of F at X is defined to be the matrix*

$$\frac{\partial\ vec\ F(X)}{\partial (vec\ X)'}$$

Theorem A.1.1

(First identification theorem of matrix differential calculus)

Let $F(X)$ be a scalar-valued differentiable function of the $(d \times d)$ matrix X. Then the Jacobian matrix of F at X is the $(1 \times d^2)$ matrix $A(X)$ in

$$\partial F(X) = A(X)\ vec\ \partial X, \tag{A.1}$$

where $A(X)$ may depend on X but not on ∂X.

Remark A.1.1 *To find the critical points of $F(X)$, we seek to write its differential in the form of (A.1). Its critical points are then found by equating the Jacobian matrix to zero and solving for X.*

Definition A.1.2 *The* Hessian Matrix *of F at X is defined to be the matrix $H(X)$ as follows*

$$H(X) = \frac{\partial^2 F(X)}{\partial (vec\ X) \partial (vec\ X)'}$$

Theorem A.1.2

(Second identification theorem of matrix differential calculus)

Let $F(X)$ be a scalar-valued twice differentiable function of the $(d \times d)$ matrix X. Suppose the second differential of $F(X)$ can be expressed in the form

$$\partial^2 F(X) = (vec\ \partial X)' B(X) (vec\ \partial X). \tag{A.2}$$

Then, the Hessian matrix of F at X is just $B(X)$ itself if it is symmetric, otherwise take

$$H(X) = \frac{1}{2} \left(B(X) + B(X)' \right),$$

where $B(X)$ may depend on X but not on ∂X.

Theorem A.1.3 *If the Hessian $H(X)$ of $F(X)$ is positive semi definite (psd) for all X, then $F(X)$ is a convex function. In addition, if $H(X)$ is positive definite (pd), then $F(X)$ is a strictly convex function.*

Theorem A.1.4 *If $F(X)$ is a (strictly) convex function with X_c as a critical point, then $F(X_c)$ is a (unique) global minimum.*

Definition A.1.3 *For the $(m \times n)$ matrix A, and the $(p \times q)$ matrix B, define the* Kronecker product *$A \otimes B$ as the $(mn \times pq)$ matrix*

$$A \otimes B = [a_{ij} B]$$

A.1.2 Identities

In the following, let A and B be constant matrices, and X be the variable matrix with respect to which differentiation is sought.

[D-1] $\partial A = 0$

[D-2] $\partial(AX) = A\,\partial X$

[D-3] $\partial(XB) = (\partial X)B$

[D-4] $\partial(X') = (\partial X)'$

[D-5] $\partial(AXB) = A(\partial X)B$

[D-6] $\partial(vec\ X) = vec(\partial X)$

[D-7] $\partial(tr(AX)) = tr(A\,\partial X) = (vec\ A')'vec(\partial X)$

[D-8] $\partial(G(X) + F(X)) = \partial G(X) + \partial F(X)$

[D-9] $\partial(G(X)F(X)) = (\partial G(X))F(X) + G(X)(\partial F(X))$

[D-10] $\partial\left((vec\ X)'A(vec\ \partial X)\right) = (vec\ \partial X)'A(vec\ \partial X)$

A.2 Vec and Kronecker product

Let A, B, C, D be conformable matrices; $\boldsymbol{a}$, $\boldsymbol{b}$ vectors.

[K-1] $vec(\alpha A + \beta B) = \alpha\ vec\ A + \beta\ vec\ B$, where $\alpha,\ \beta$ are scalars

[K-2] $tr(AB) = tr(BA)$, provided the product BA makes sense

[K-3] $tr(A) = tr(A')$

[K-4] $tr(A'B) = (vec\ A)'vec\ B$

[K-5] $(A \otimes B)' = A' \otimes B'$

[K-6] $vec(ABC) = (C' \otimes A)vec\ B$

[K-7] $tr(A + B) = tr(A) + tr(B)$

[K-8] $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$, for generalized inverses

[K-9] $(A \otimes B)(C \otimes D) = (AC \otimes BD)$

[K-10] $vec(\boldsymbol{ab'}) = \boldsymbol{b} \otimes \boldsymbol{a}$

[K-11] $(A + B) \otimes C = A \otimes C + B \otimes C$

A.3 Positive definite (pd) and positive semi-definite (psd) symmetric matrices

These can be found in say, Graybill (1983) and Lutkepohl (1996).

[M-1] *Characterization 1 of pd and psd matrices*: a square matrix A is psd iff it can be written as $A = B'B$, for some square matrix B. A is pd iff B is of full rank.

[M-2] *Characterization 2 of pd and psd matrices*: a square matrix A is psd iff its eigenvalues are all non negative. A is pd iff all its eigenvalues are positive.

[M-3] The sum of psd matrices is again psd. If at least one of the matrices is pd, then the whole sum is pd.

[M-4] The Kronecker product of pd matrices is again pd. If at least one of the matrices is psd, then the whole Kronecker product is psd.

Proof: From Magnus and Neudecker (1999) theorem 1 page 28, if A and B (both $d \times d$) have eigenvalues $\lambda_1, \ldots, \lambda_d$ and $\mu_1, \ldots, \mu_d$, respectively, then the

d^2 eigenvalues of their Kronecker product are precisely $\lambda_i \mu_j,\ i, j = 1, \ldots, d$. The result then follows by M-2.

[M-5] If A is pd, then so are A^{-1} and A^2.

[M-6] If A and B are psd, then so is ABA.

Proof: Let $A = D'D$, $B = E'E$. Then,

$$ABA = D'DE'ED'D = (ED'D)'(ED'D),$$

which by M-1 has the prerequisite psd form.

Appendix B

RELATING THE CHARACTERISTIC POLYNOMIAL OF BIVARIATE VAR MODELS TO THE COEFFICIENTS

In the multivariate full set VAR(p) setting, the vector autoregressive characteristic polynomial of

$$\mathbf{X}_t - \Phi_1 \mathbf{X}_{t-1} - \cdots - \Phi_p \mathbf{X}_{t-p} = \mathbf{Z}_t$$

is defined to be the polynomial of degree dp given by

$$|\Phi(z)| = |I_d - \Phi_1 z - \cdots - \Phi_p z^p|.$$

Causality in the multivariate setting requires all roots of $|\Phi(z)|$ to be greater than 1 in magnitude, i.e. all roots must lie outside the unit circle in the complex plane. Restricting ourselves to bivariate models,

$$\begin{aligned} |\Phi(z)| &= \left(1 - \Phi_1^{11} z - \cdots - \Phi_p^{11} z^p\right)\left(1 - \Phi_1^{22} z - \cdots - \Phi_p^{22} z^p\right) \\ &\quad - \left(\Phi_1^{12} z + \cdots + \Phi_p^{12} z^p\right)\left(\Phi_1^{21} z + \cdots + \Phi_p^{21} z^p\right), \end{aligned} \tag{B.1}$$

where

$$\Phi_t \equiv \begin{bmatrix} \Phi_t^{11} & \Phi_t^{12} \\ \Phi_t^{21} & \Phi_t^{22} \end{bmatrix}, \quad t = 1, \ldots, p.$$

Using the notation

$$\Phi_{s,t} \equiv \begin{bmatrix} \Phi_s^{11} & \Phi_s^{12} \\ \Phi_t^{21} & \Phi_t^{22} \end{bmatrix} \equiv \begin{bmatrix} 1^{\text{st}} \text{ row of } \Phi_s \\ 2^{\text{nd}} \text{ row of } \Phi_t \end{bmatrix}, \quad s, t = 1, \ldots, p$$

(note that $\Phi_{t,t} \equiv \Phi_t$), and expanding and combining terms of powers of z in (B.1), we can write the characteristic polynomial for a bivariate VAR(p) as

$$|\Phi(z)| = 1 + \alpha_1 z + \cdots + \alpha_{2p} z^{2p},$$

where

$$\alpha_k = \begin{cases} -\mathrm{Tr}(\Phi_1), & k = 1 \\ \sum_{t=1}^{k-1} |\Phi_{t,k-t}| - \mathrm{Tr}(\Phi_k), & 2 \leq k \leq p \\ \sum_{t=k-p}^{p} \Phi_t^{11} \Phi_{k-t}^{22} - \left(\Phi_{k-p}^{12} \Phi_p^{21} + \Phi_{k-p}^{21} \Phi_p^{12} \right), & p+1 \leq k \leq 2p-1 \\ |\Phi_p|, & k = 2p. \end{cases}$$

If we specify the characteristic polynomial and attempt to find a set of corresponding VAR coefficients, the resulting system will have $2p$ equations in twice as many unknowns (4 for each $\Phi_t,\ t = 1, \ldots, p$). In the examples of section 1.9.2, we approach this problem by fixing some of the elements of the coefficient matrices, thus obtaining a system in as many equations as unknowns. For low order models, this can easily be solved by an efficient non-linear system of equations solver.

Appendix C

DESCRIPTION OF THE AR/VAR MODELING PROGRAMS

C.1 Introduction

The four VAR modeling algorithms presented in chapter 1, are all based on Algorithm 1.4.1, which gives the Yule-Walker solution. The Burg and Vieira-Morf solutions are obtained by modifying only the manner in which the *forward reflection coefficients* are computed, (1.16); while the Nuttall-Strand solution requires modifications in both the forward and *backward reflection coefficients*, (1.16) and (1.17).

Although typically not of interest, the backward coefficients must of necessity be computed at every iteration. At the very least, $\hat{\Psi}_{K^*}(k_m)$ and $\hat{V}_{K^*}$ should be evaluated in any given iteration, since the coefficients and white noise variance of the forward modeling problem depend on them ($\hat{\Psi}_{K^*}(k_m)$ in the current iteration, and $\hat{V}_{K^*}$ possibly in a subsequent one). Defining the *level* of an iteration to be the cardinality of the set K (i.e. m), a computational savings can sometimes be obtained in the univariate case by evaluating the complete sets of both forward and backward coefficients in key iterations. This happens for example when K^* in one iteration coincides with K of another at the same level.

Example C.1.1 *As an illustration of the application of this type of algorithm in the univariate setting, suppose for example that modeling on the subset of lags* $K =$

$\{1,3,4,5\}$ *is desired. We can maximize the computational efficiency of the algorithm by proceeding as follows:*

Level	K	K^*	J	J^*	*Compute ... coefficients*
1	$\{1\}$	$\{1\}$	$\emptyset$	$\emptyset$	*forward*
1	$\{2\}$	$\{2\}$	$\emptyset$	$\emptyset$	*forward*
2	$\{1,2\}$	$\{1,2\}$	$\{1\}$	$\{1\}$	*forward*
2	$\{2,3\}$	$\{1,3\}$	$\{2\}$	$\{1\}$	*forward and backward*
3	$\{1,3,4\}$	$\{1,3,4\}$	$\{1,3\}$	$\{1,3\}$	*forward*
3	$\{1,2,4\}$	$\{2,3,4\}$	$\{1,2\}$	$\{2,3\}$	*forward*
4	$\{1,3,4,5\}$	$\{1,2,4,5\}$	$\{1,3,4\}$	$\{1,2,4\}$	*forward*

Note that the full application of the algorithm to compute both forward and backward coefficients at the 4th iteration ($K = \{2,3\}$), enables us to bypass an extra iteration at level 2 with $K = \{1,3\}$.

The "efficient" rendering of this type of algorithm in a programming language, is in itself a substantial problem. Although far from efficient, we have succeeded in implementing it in FORTRAN 90, using complex programming structures such as recursive pointers, recursive subroutines, and data types that incorporate recursive definitions. In this chapter we will give an overview of the programming logic and layout employed in the development of these algorithms for univariate and bivariate modeling problems. The FORTRAN 90 programs themselves are appended at the end.

C.2 Building a Tree of Nodes

Consider modeling on the set of lags $K = \{1,3,7\}$ as an example. In order to determine where application of the algorithm should begin, we first need to work down to level 1 by successively forming the J and J^* sets of lags for all *parent* sets of lags K, as shown in figure C.1. We can therefore begin the algorithm by

computing the coefficients on the sets of lags $\{1\}$, $\{2\}$, and $\{4\}$. With these, we can now compute the coefficients on the sets of lags $\{1,3\}$ and $\{4,6\}$, at level 2. Finally, regarding these last two sets as our J and J^*, we can compute the coefficients and corresponding white noise variance on the set $K = \{1,3,7\}$.

Figure C.1: Recursive modeling on the set $K = \{1,3,7\}$.

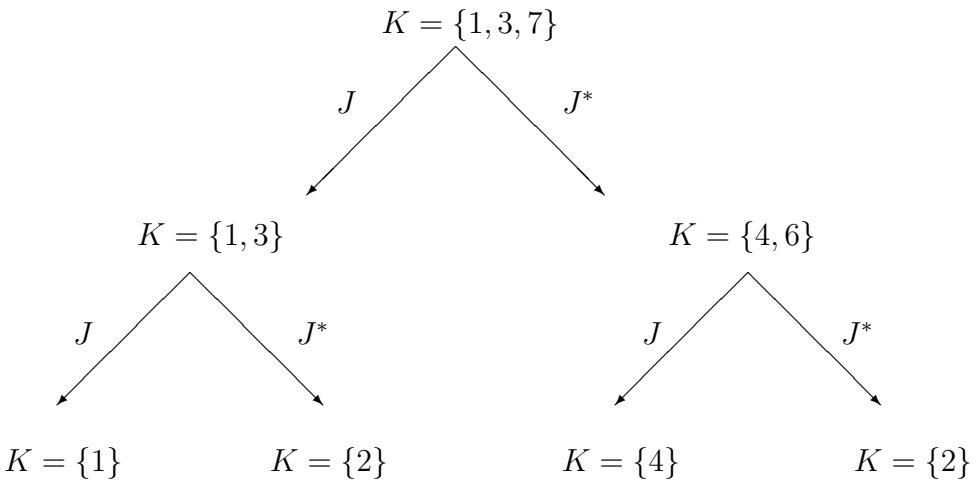


Figure C.1 exemplifies the logic that will be employed in the program. The first step is to create a *tree of nodes*. A `node` will be a FORTRAN 90 user-defined data type, containing the following components:

`level` - Level of the node in the tree, ranging from top (m), to bottom (1).

`lags` - Integer vector containing the current subset of lags on which modeling is desired.

`phi/A, B` - Real vector containing the coefficients of the AR/VAR being modeled, corresponding to each of the lags in `lags`. In the bivariate case, `A` and `B`, the forward and backward coefficient matrices, respectively, will be of type `matrix` (`%mat`), a user-defined data type consisting of a real (2×2) matrix.

`v/vf, vb` - The estimate of the white noise variance for the AR/VAR model. In the bivariate case, we need to keep track of both the forward (`vf`), and backward (`vb`) white noise covariance matrices (which will be of type `matrix`).

`eps` - Real vector of forward prediction residuals for the current model. In the bivariate case, each component of this vector will be of type `vector` (`%vec`), a user-defined data type consisting of a real vector of dimension 2.

`eta` - Real vector of backward prediction residuals for the current model. In the bivariate case, each component of this vector will be of type `vector`.

`reg, star` - Pointers to the J and J^* subnodes one level down from the current one. These are themselves of type `node`, and are therefore defined recursively.

Starting at level m, we successively compute the subnodes J and J^* for each set of lags K, directing pointers to each of them as shown in figure C.1. When level 1 is reached, the tree of nodes will be initialized. With this framework, we can now begin at level 1, filling each node with its constituents outlined in the above description. With all nodes in level 1 filled, we move up the tree to level 2. At each of these nodes, we will retrieve subnode information (J and J^*), by following the appropriate pointers that were allocated by the tree building routine. With this information, we can now fill each of the nodes at this level. We continue in this fashion, gradually migrating up the tree until the unique node at level m is filled. Its now an easy matter to retrieve these coefficients, report them, and calculate the -2 log likelihood ($\mathcal{L}$) for the attained model.

The bulk of the program is defined within `MODULE Tree`, which is made known to the main driving program `Burg` by the call: `Use Tree`. Upon execution, the user will be prompted for the following inputs:

- The file name containing the data to be modeled.
- The number of lags to be modeled: m.
- The specific lags on which subset AR/VAR modeling is desired: K.
- The modeling method: one of Yule-Walker, Burg, Vieira-Morf, and Nuttall-Strand.

Program `Burg` itself only reads in these inputs and calls the subroutine `Make_Tree`, the latter being the driving subroutine in module `Tree`. The corresponding bivariate modeling program is called `Burg2`.

C.3 Description of Principal Program Subroutines

As already stated, the core of the subset modeling programs `Burg` and `Burg2` is the globally visible `MODULE Tree`, with `SUBROUTINE Make_Tree` its driving subroutine. In this section, we will provide a brief description of the essential functions of each of its constituent subroutines.

C.3.1 `Build_Node_Tree`

This is a `RECURSIVE SUBROUTINE` that initializes the tree of nodes by allocating pointers to and from nodes. It takes on the `level`, `lags`, and a `pointer` of type `node` as arguments. It begins execution at the unique node of level m (`top_node`), creating pointers to the J and J^* subnodes (`this_node%reg` and `this_node%star`, respectively). Following these pointers to level $m-1$, `Build_Node_Tree` subsequently allocates pointers to the subnodes in level $m-2$. It achieves this by calling itself with the appropriate arguments:`level` should be the current level minus one, and pointers `this_node%reg` and `this_node%star`. The procedure is repeated, always

following pointer `this_node%reg` before `this_node%star`, until level 1 is reached. At this point, the two pointers are initialized and made to point nowhere (`NULLIFIED`). By the order of precedence inherent in it, the routine then backs up one level and proceeds to follow pointer `this_node%star` to the "dead end" at level 1.

In this fashion, the tree is initialized from left (J) to right (J^*), with the pointer to the subnode J^* of the rightmost node being allocated last. If we refer back to figure C.1, the nodes for the tree of this example will be initialized in the following order:

$$\{1,3,7\} \rightarrow \{1,3\} \rightarrow \{1\} \rightarrow \{2\} \rightarrow \{4,6\} \rightarrow \{4\} \rightarrow \{2\}.$$

Note that identical copies of nodes will sometimes be created ($\{2\}$ in the above). For small m, this is a minor inefficiency in the program that can be improved by a more competent programmer!

In order for subsequent routines to identify an initialized but unfilled (constituents of `node` empty) node, `Build_Node_Tree` will set `this_node%v` (`this_node%vf%mat(1,1)` in `Burg2`) to zero, upon allocation of pointers.

C.3.2 `Fill_Tree`

A `RECURSIVE SUBROUTINE`, taking on a `pointer` of type `node` as argument. Its function is to traverse the now initialized tree, and using the flag for an unfilled node, fill it by calling `Fill_Node`.

C.3.3 `Fill_Node`

A `RECURSIVE SUBROUTINE`, called by `Fill_Tree`, whose function is to fill the particular node that its `pointer` argument points to. It is in this routine that the

various AR/VAR modeling algorithms proper are applied. After first initializing some variables, the routine essentially applies algorithm 1.4.1, but modifying the reflection coefficient calculation according to the modeling method selected. This method is indicated by the global variable `method`, set in `Burg/Burg2`. Care must be taken when calculating the forward and backward prediction errors, $\boldsymbol{\varepsilon}_K(t)$ and $\boldsymbol{\eta}_K(t)$, before termination of the routine. We must ensure that each is calculated over a sufficiently large range of t values that will span that required by any subsequent nodes that may use them. A look at chapter 1 will verify that it will be sufficient to take these ranges of definition to be $t \in \{1, \ldots, n+k_m\}$ for Yule-Walker, and $t \in \{1 + k_m, \ldots, n\}$ for the remaining three methods.

C.3.4 `Print_Node_Tree`

With its `pointer` argument, the `RECURSIVE SUBROUTINE Print_Node_Tree` will traverse the now completed tree of nodes, and proceed to print the estimated coefficients and white noise variance stored in each node. The addition of an appropriate `IF` statement between recursive calls to itself, ensures that it will only print this information for the `top_node`.

C.3.5 `Undo_Node_Tree`

The `RECURSIVE SUBROUTINE Undo_Node_Tree` will undo the pointer allocation put in place by `Build_Node_Tree`. As designed, the modeling programs themselves don't need this routine. It becomes desirable only if an extra loop is added to do repeated modeling, such as in large scale simulations. In this way, repeated memory allocation when initalizing pointers is avoided every time a new tree is built. This becomes increasingly important as the number of modeled lags (m) grows, since the number of nodes created by the programs in any one run is exactly 2^m. In repeated modeling,

large values of m will quickly exhaust the memory capacity of the average computer. Again, there is room for improvement here for an astute programmer, in that it would be more efficient to only initialize the tree of nodes once before plunging into the `DO` loop of the repeated modeling scenario.

C.3.6 `Causal_Check`

This routine is needed in the bivariate program only, in order to ensure the obtained VAR model is causal before proceeding with the likelihood calculations. In the univariate program, this function is performed within the likelihood calculation routine itself. The strategy is to use the state space representation to write a VAR(p) as a VAR(1), as follows:

Random vectors $\{\mathbf{X}_t, \ldots, \mathbf{X}_{t-k_m}\}$ from model (2.1), will satisfy the relationships

$$\begin{bmatrix} \mathbf{X}_t \\ \mathbf{X}_{t-1} \\ \mathbf{X}_{t-2} \\ \\ \vdots \\ \\ \\ \mathbf{X}_{t-k_m+1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & k_1 & \cdots & k_m \\ I_d & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & I_d & & & & & 0 \\ & & \ddots & & & & \\ \vdots & & & \ddots & & & \vdots \\ & & & & \ddots & & \\ 0 & & & \cdots & & I_d & 0 \end{bmatrix} \begin{bmatrix} \mathbf{X}_{t-1} \\ \mathbf{X}_{t-2} \\ \mathbf{X}_{t-3} \\ \\ \vdots \\ \\ \\ \mathbf{X}_{t-k_m} \end{bmatrix} + \begin{bmatrix} \mathbf{Z}_t \\ \mathbf{0} \\ \mathbf{0} \\ \\ \vdots \\ \\ \\ \mathbf{0} \end{bmatrix},$$

which can be written in the compact form

$$\underbrace{\mathbf{Y}_t}_{(dk_m \times 1)} = \underbrace{A}_{(dk_m \times dk_m)} \underbrace{\mathbf{Y}_{t-1}}_{(dk_m \times 1)} + \underbrace{\mathbf{W}_t}_{(dk_m \times 1)} . \tag{C.1}$$

In block matrix form, vectors $\mathbf{Y}_t$ and $\mathbf{W}_t$ have length k_m, while the square matrix A has dimension k_m. Note that the only nonzero entries of the first block matrix row of A are $\{\Phi_K(k_1), \Phi_K(k_2), \ldots, \Phi_K(k_{m-1}), \Phi_K(k_m)\}$, occurring at block matrix column numbers $\{k_1, k_2, \ldots, k_{m-1}, k_m\}$, respectively. The covariance matrix of $\mathbf{W}_t$

is

$$\Sigma_W = \mathbb{E}\begin{bmatrix} \mathbf{Z}_t \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix} [\mathbf{Z}_t', \mathbf{0}', \ldots, \mathbf{0}'] = \begin{bmatrix} \Sigma & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix}.$$

(C.1) is now a VAR(1) of dimension dk_m, and its causality (and thus that of the original process) can be assessed by determining if all eigenvalues of A are less than 1 in absolute value.

C.3.7 `Likelihood/Approx_Likelihood`

In the univariate program, we compute the exact likelihood in `SUBROUTINE Likelihood`. The only sizeable difficulty is in evaluating the model autocovariances $\gamma(0), \ldots, \gamma(k_m)$, accomplished by inverting the Yule-Walker equations. The -2 log likelihood, $\mathcal{L}(\boldsymbol{\phi}_K, \sigma^2)$, for the data $\mathbf{x}_1, \ldots, \mathbf{x}_n$, is then evaluated via the Innovations Algorithm (Brockwell and Davis (1991), proposition 5.2.2, and equation (8.7.4)):

$$\mathcal{L}(\boldsymbol{\phi}_K, \sigma^2) = n\log(2\pi\sigma^2) + \sum_{t=1}^{n}\log(r_{t-1}) + \frac{1}{\sigma^2}\sum_{t=1}^{n}(\mathbf{x}_t - \hat{\mathbf{x}}_t)^2/r_{t-1}.$$

In the bivariate program, `SUBROUTINE Likelihood` uses the same approach to compute the likelihood, i.e. the Multivariate Innovations Algorithm (Brockwell and Davis (1991), proposition 11.4.2, and equation (11.5.5)):

$$\mathcal{L}(\Phi_K, \Sigma) = nd\log(2\pi) + \sum_{t=1}^{n}\log|V_{t-1}| + \sum_{t=1}^{n}(\mathbf{X}_t - \hat{\mathbf{X}}_t)'V_{t-1}^{-1}(\mathbf{X}_t - \hat{\mathbf{X}}_t).$$

Computing the model autocovariance matrices, $\Gamma(1-k_m), \ldots, \Gamma(0), \ldots, \Gamma(k_m-1)$, is a much more formidable task here, but this can be accomplished via the state space formulation of the previous subsection. Transforming the SVAR(K) to the VAR(1) of equation (C.1), gives the following solution for the autocovariances $\Gamma_Y(\cdot)$ of the process $\{\mathbf{Y}_t\}$:

$$\Gamma_Y(h) = \begin{cases} A\Gamma_Y(h)A' + \Sigma_W, & h = 0 \\ A\Gamma_Y(h-1), & h > 0 \end{cases},$$

whence we obtain

$$vec\left(\Gamma_Y(0)\right) = \left[I_{d^2k_m^2} - A \otimes A\right]^{-1} vec\left(\Sigma_W\right).$$

The required autocovariance matrices can be found in the first block row and column of the $(k_m \times k_m)$ block matrix $\Gamma_Y(0)$, since

$$\underbrace{\Gamma_Y(0)}_{(dk_m \times dk_m)} = \begin{bmatrix} \Gamma(0) & \Gamma(1) & \cdots & \Gamma(k_m - 1) \\ \Gamma(-1) & \Gamma(0) & \cdots & \Gamma(k_m - 2) \\ \vdots & & \ddots & \vdots \\ \Gamma(1 - k_m) & \cdots & \Gamma(-1) & \Gamma(0) \end{bmatrix}.$$

Due to the computational intensity involved in finding $\Gamma_Y(\cdot)$ however, the bivariate routine `Likelihood` is extremely slow. We opt instead to approximate the autocovariances via the causal representation

$$\Gamma(h) = \sum_{j=0}^{\infty} \Psi_{h+j} \Sigma \Psi_j',$$

truncating the summation at 100 terms, and computing the likelihood via (1.31). This "approximate likelihood", is computed in `SUBROUTINE Obj_Fun`. `SUBROUTINE Approx_Likelihood` not only calls `Obj_Fun` in order to compute this approximate likelihood for Σ_{AL}, but also searches for the white noise covariance matrix that maximizes the likelihood for the given VAR coefficient matrices (Σ_{ML}). It does so by using Σ_{AL} as an initial guess, and by repeated calls to `SUBROUTINE Hooke`, which employs a direct search algorithm to locate the global minimum of an objective function of several variables (Hooke and Jeeves (1961)).

C.3.8 `Simulate/Simulate2`

These appear in the modeling programs of appendix **??** only for completeness. They are not called by the modeling programs themselves, but were used extensively to

simulate realizations from causal subset univariate and bivariate VAR models with Gaussian noise:

$$\mathbf{X}_t = \sum_{i \in K} \Phi_K(i) \mathbf{X}_{t-i} + \mathbf{Z}_t, \quad \{\mathbf{Z}_t\} \sim \text{IID N}(\mathbf{0}, \Sigma). \tag{C.2}$$

The logic in both programs is identical:

- Obtain $n + 500$ observations from the noise process $\{\mathbf{Z}_t\}$.
- Setting $\mathbf{X}_1, \ldots, \mathbf{X}_{k_m}$ equal to zero, use (C.2) to obtain $\mathbf{X}_{k_m+1}, \ldots, \mathbf{X}_{n+500}$.
- Select only the last n of these $\mathbf{X}_t$'s as a *bona fide* sample from (C.2).

Bibliography

Brockwell, P. and Dahlhaus, R. (1998). Generalized Durbin-Levinson and Burg algorithms. Technical Report 98/3, Department of Statistics, Colorado State University, Fort Collins, Colorado.

Brockwell, P. and Davis, R. (1991). *Time Series: Theory and Methods*. Springer-Verlag, New York, second edition.

Burg, J. (1978). A new analysis technique for time series data, (1968). In Childers, D., editor, *Modern Spectrum Analysis*, New York. NATO Advanced Study Institute of Signal Processing with emphasis on Underwater Acoustics, IEEE Press.

Butler, R. and Paolella, M. (1998). Saddlepoint approximations to the density and distribution of ratios of quadratic forms in normal variables with application to the sample autocorrelation function. Technical Report 98/16, Department of Statistics, Colorado State University, Fort Collins, Colorado.

Daniels, H. (1956). The approximate distribution of serial correlation coefficients. *Biometrika*, 43:169–185.

Duong, Q. (1984). On the choice of the order of autoregressive models: A ranking and selection approach. *Journal of Time Series Analysis*, 5:145–157.

Durbin, J. (1980). The approximate distribution of partial serial correlation coefficients calculated from residuals from regression on fourier series. *Biometrika*, 67:335–349.

Fuller, W. (1996). *Introduction to Statistical Time Series*. Wiley, New York, second edition.

Graybill, F. (1983). *Matrices with Applications in Statistics*. Wadsworth, Belmont, California, second edition.

Hainz, G. (1994). The asymptotic properties of Burg estimators. Beltrage zur Statistik 18, Institut fur Angewandte Mathematik, Universitat Heidelberg, Germany.

Hall, P. and Heyde, C. (1980). *Martingale limit theory and its applications*. Academic Press, New York.

Hannan, E. (1970). *Multiple time series*. Wiley, New York.

Hooke, R. and Jeeves, T. (1961). A direct search solution of numerical and statistical problems. *Journal of Association for Computing Machinery*, 8:212–229.

Jones, R. (1978). Multivariate autoregression estimation using residuals. In Findley, D., editor, *Applied Time Series Analysis*, pages 139–162, New York. Proceedings of the First Applied Time Series Symposium, Tulsa, Okla., 1976, Academic Press.

Lutkepohl, H. (1993). *Introduction to Multiple Time Series Analysis*. Springer-Verlag, Berlin; New York, second edition.

Lutkepohl, H. (1996). *Handbook of Matrices*. Wiley, Chichester.

Magnus, J. and Neudecker, H. (1999). *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley, Chichester, revised edition.

McClave, J. (1978). Estimating the order of autoregressive models: the max χ^2 method. *Journal of the American Statistical Association*, 73:122–128.

Morf, M., Vieira, A., Lee, D., and Kailath, T. (1978). Recursive multichannel maximum entropy spectral estimation. *IEEE Trans. Geosci. Electron.*, GE-16:85–94.

Ochi, Y. (1983). Asymptotic expansions for the distribution of an estimator in the first order autoregressive process. *Journal of Time Series Analysis*, 4:57–67.

Penm, J. and Terrell, R. (1982). On the recursive fitting of subset autoregressions. *Journal of Time Series Analysis*, 3:43–59.

Phillips, P. (1978). Edgeworth and saddlepoint approximations to the first-order noncircular autoregression. *Biometrika*, 65:91–98.

Sarkar, A. and Kanjilal, P. (1995). On a method of identification of best subset model from full ar model. *Communications in Statistics, Part A – Theory and Methods*, 24:1551–1567.

Sarkar, A. and Sharma, K. (1997). An approach to direct selection of best subset ar model. *Journal of Statistical Computation and Simulation*, 56:273–291.

Strand, O. (1977). Multichannel complex maximum entropy (autoregressive) spectral analysis. *IEEE Trans. Automat. Control.*, 22:634–640.

Yu, G. and Lin, Y. (1991). A methodology for selecting subset autoregressive time series. *Journal of Time Series Analysis*, 12:363–373.

Zhang, X. and Terrell, R. (1997). Projection modulus: A new direction for selecting best subset autoregressive models. *Journal of Time Series Analysis*, 18:195–212.