https://doi.org/10.1016/j.csda.2018.01.019

On constrained estimation of graphical time series models

T.P. Yuen[a], H. Wong[a], K.F.C. Yiu[a,*]

[a]*Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China*

Abstract

Graphical time series models encode the conditional independence among the variables of a multivariate time series. An iterative method is proposed to estimate a graphical time series model based on a sparse vector autoregressive process. The method estimates both the autoregressive coefficients and the inverse of noise covariance matrix under sparsity constraints on both the coefficients and the inverse covariance matrix. This iterative method estimates a sparse vector autoregressive model by considering maximum likelihood estimation with the sparsity constraints as a biconcave problem, where the optimization problem becomes concave when either the autoregressive coefficients or the inverse noise covariance matrix is fixed. The method also imposes fewer restrictions in the estimation comparing to the use of a structural vector autoregressive model to study the dynamic interdependencies between time series variables.

Keywords: Graphical models, Time series, Estimation, Optimization, Air pollution

1. Introduction

Graphical models represent the conditional independencies among random variables in multivariate data. These independence relationships can be visualized by an undirected graph where vertices represent the variables and edges between vertices illustrate that the corresponding variables of the connected vertices are conditionally dependent. Since the introduction of log-linear models on discrete data, researchers have attempted to link up graphical models with log-linear models (Darroch et al., 1980). By analogy with the log-linear models for contingency tables, models based on the multivariate normal distribution have been introduced. Edwards (1995) and Lauritzen (1996) give good introduction to graphical modelling.

Consider a K-dimensional random variable $X \sim N(0,\Sigma)$; a Gaussian graphical model can be established by calculating the precision matrix, $\Theta = \Sigma^{-1}$. With the precision matrix, the conditional independence be-

[*] Corresponding author. Tel.: +852 3400 8981; fax: +852 2362 9045.
Email address: cedric.yiu@polyu.edu.hk (K.F.C. Yiu)

tween variables is determined. For example, two components of X are independent conditioning on the remaining components if and only if the corresponding entry in the precision matrix is zero, i.e., X_i and X_j are conditionally independent if and only if $\Theta_{ij} = 0$. With prior information on the conditional independence between variables, the estimation of a Gaussian graphical model can be formulated as the covariance selection problem (Dempster, 1972), as formulated in (1).

$$\begin{aligned} &\text{Maximize} \quad \log\det(\Theta) - \operatorname{tr}(\mathbf{S}\Theta) \\ &\text{subject to} \quad \Theta_{ij} = 0, \quad (i,j) \in \Omega, \end{aligned} \tag{1}$$

where $\mathbf{S}$ is the sample covariance matrix, Ω is a set consisting of pairs of known conditionally independent nodes.

The increasing interest in data science has heightened the need for the development of Gaussian graphical models with sparse coefficients on high dimension data (see Banerjee et al. (2008); Dahl et al. (2008); Friedman et al. (2008)). To achieve sparsity, researchers have considered the penalized likelihood methods shown below.

$$\begin{aligned} &\text{Maximize} \quad \log\det(\Theta) - \operatorname{tr}(\mathbf{S}\Theta) \\ &\text{subject to} \quad \rho(\Theta) \le k, \end{aligned} \tag{2}$$

where $\mathbf{S}$ is the sample covariance matrix, $\rho(\cdot)$ is a regularization term, and k is a tuning parameter.

Brillinger (1996) and Dahlhaus (2000) extended the use of graphical models to multivariate time series to explore the interrelationship between variables of a multivariate time series process. A recent summary of graphical time series models can be found in Tunnicliffe Wilson et al. (2015). The partial correlation structure of the components of the process given the remaining components can be identified by the partial spectral coherence or the inverse of the spectral density matrix. These frequency domain statistics measure the linear association of two components of a process given the linear effects of the remaining components. Similar to Gaussian graphical models, two components of a multiple time series is conditionally uncorrelated given the other components if and only if the corresponding partial spectral coherence is zero at all frequencies (Brillinger, 1981; Dahlhaus, 2000). The interrelationships are visualized by an undirected partial correlation graph, where each vertex represents a component of the process, and the edges are characterized by the partial spectral coherences. In particular, the partial correlation graph exploits the conditional dependence structure of the components if the time series is Gaussian.

With the partial correlation graph, the complexity of fitting a time series model with large dimension can be reduced by imposing sparsity constraints on the VAR model based on the partial correlation graph.

Songsiri et al. (2009) discussed the VAR model estimation problem, subject to conditional independence constraints based on the inverse of the spectral density matrix, using convex optimization methods. Davis et al. (2016) proposed a two-stage approach for fitting sparse VAR models in which non-zero autoregressive coefficients are selected according to the partial spectral coherence together with the Bayesian information criterion (BIC) (Schwarz, 1978). The model is then refined in the second stage to reduce the number of parameters further by using the t-ratios of the estimated autoregressive coefficients. These two articles (Davis et al., 2016; Songsiri et al., 2009) also investigated the penalized likelihood methods in the maximum likelihood estimation by imposing regularization term, like L_1 regularization, to achieve sparsity. Other related research (Hsu et al. (2008); Ren et al. (2013); Songsiri (2013); Jung et al. (2015)) also discussed penalized regression methods for VAR modelling. The penalized regression approaches for VAR modelling ignore the contemporaneous dependence in the time series (Song & Bickel, 2011) since the noise covariance matrix is not taken into account when a loss function of the sum of squared residuals is used.

To our knowledge, very few research have been done on fitting a VAR model with sparsity constraints on both the autoregressive coefficients and the inverse of the noise covariance matrix. These constraints become important when a graphical VAR model is constructed. Similar to the partial correlation graph, each vertex of a graphical VAR model represents a component of the multivariate time series. The autoregressive coefficients characterize the directed edges and the undirected edges are determined by the non-diagonal entries of the inverse of the noise covariance matrix, see Eichler (2012). An alternative to study the dynamic interdependencies among the components of a multivariate time series is to consider a structural vector autoregressive (SVAR) model and represent them by a directed acyclic graph (DAG) (Oxley et al., 2004). The estimation of an SVAR model, however, requires restrictions, so that it is identifiable and a DAG can be built to represent the model.

To avoid such restrictions, and impose sparsity on both the autoregressive coefficients and the inverse of innovation covariance matrix, we propose an iterative algorithm to estimate a sparse VAR model. The algorithm considers the maximum likelihood estimation with the sparsity constraints as a “biconcave” problem in the sense that the optimization problem becomes concave when either the autoregressive coefficients or the inverse of noise covariance matrix is fixed (Gorski et al., 2007). We solve the alternating maximization problem, assuming the sparsity structure is known, using the alternating convex search (ACS) method and compare with the interior point method (fmincon in Matlab) and the direct search method (patternsearch in Matlab). To identify the structure, we present two methods, namely the time domain and frequency domain methods, and compare these two methods by simulation experiments.

Section 2 presents the sparse vector autoregressive model. Section 3 provides the proposed algorithm and visualization method on the fitted VAR model. Section 4 gives simulation studies. Section 5 exemplifies the proposed method by real data applications. Section 6 concludes.

2. Sparse vector autoregressive model

2.1. Vector autoregressive model

Consider a K-dimensional VAR(p) process:

$$y_t = \nu + \mathbf{A}_1 y_{t-1} + \cdots + \mathbf{A}_p y_{t-p} + u_t, \tag{3}$$

where $y_t = (y_{1,t}, \cdots, y_{K,t})'$ is a $(K \times 1)$ vector, $\mathbf{A}_l$, $l = 1, \cdots, p$, are $(K \times K)$ autoregressive coefficient matrices, ν is a $(K \times 1)$ vector of intercepts, $u_t = (u_{1,t}, \cdots, u_{K,t})'$ is a K-dimensional Gaussian noise vector, with mean $\mathbf{0}$ and a $(K \times K)$ nonsingular covariance matrix, Σ_u, and $t = 1, \cdots, T$. We further assume that the process is stable, i.e., $\det\left(\mathbf{I}_K - \sum_{l=1}^{p} \mathbf{A}_l z^l\right) \neq 0$, for $z \in \mathbb{C}$, $|z| \leq 1$, and p pre-sample values, $y_{-p+1}, \cdots, y_0$, are available. The compact form of (3) is

$$\mathbf{Y} = \mathbf{B}\mathbf{Z} + \mathbf{U}, \tag{4}$$

where $\mathbf{Y} = (y_1, \cdots, y_T)$, $\mathbf{B} = (\nu, \mathbf{A}_1, \cdots, \mathbf{A}_p)$ is a $(K \times Kp + 1)$ coefficient matrix including the intercept terms, $\mathbf{Z} = (\mathbf{Z}_0, \cdots, \mathbf{Z}_{T-1})$ is a $(Kp + 1 \times T)$ matrix, $\mathbf{Z}_t = (1, {y_t}', \cdots, {y_{t-p+1}}')'$ is a $(Kp + 1 \times 1)$ vector, and $\mathbf{U} = (u_1, \cdots, u_T)$. The log-likelihood function for the conditional maximum likelihood estimation, assuming the VAR(p) process is Gaussian, is

$$l(\mathbf{B}, \Sigma_u) = -\frac{KT}{2}\log 2\pi - \frac{T}{2}\log\det(\Sigma_u) - \frac{1}{2}\mathrm{tr}\left((\mathbf{Y} - \mathbf{B}\mathbf{Z})'\Sigma_u^{-1}(\mathbf{Y} - \mathbf{B}\mathbf{Z})\right). \tag{5}$$

From (5), we can obtain the conditional maximum likelihood estimator (MLE) of $\mathbf{B}$ and Σ_u, which are

$$\hat{\mathbf{B}} = \mathbf{Y}\mathbf{Z}'\left(\mathbf{Z}\mathbf{Z}'\right)^{-1} \text{ and } \quad \hat{\Sigma}_u = \frac{1}{T}\left(\mathbf{Y} - \hat{\mathbf{B}}\mathbf{Z}\right)\left(\mathbf{Y} - \hat{\mathbf{B}}\mathbf{Z}\right)', \tag{6}$$

respectively, see Chapter 3.4 of Lütkepohl (2005). The log-likelihood function in (5) can also be rewritten as:

$$\begin{aligned} l(\beta, \Sigma_u) = &-\frac{KT}{2}\log 2\pi - \frac{T}{2}\log\det(\Sigma_u) \\ &-\frac{1}{2}\left[\mathbf{y} - \left(\mathbf{Z}' \otimes \mathbf{I}_K\right)\beta\right]'\left(\mathbf{I}_T \otimes \Sigma_u^{-1}\right)\left[\mathbf{y} - \left(\mathbf{Z}' \otimes \mathbf{I}_K\right)\beta\right], \end{aligned} \tag{7}$$

where $\beta = \text{vec}(\mathbf{B})$ is a $(K(Kp+1) \times 1)$ vector by stacking the coefficient matrix $\mathbf{B}$, $\mathbf{y} = \text{vec}(\mathbf{Y})$ is a $(KT \times 1)$ vector, $\mathbf{I}_K$ is a $(K \times K)$ identity matrix, and $\otimes$ denotes the Kronecker product. Suppose there are linear constraints on β which are in the form

$$\mathbf{C}\beta = \mathbf{c}, \tag{8}$$

where $\mathbf{C}$ is an $\left(N \times K^2 p + K\right)$ matrix of known constants with rank N, and $\mathbf{c}$ is an $(N \times 1)$ vector of known constants. Then, the constrained maximum likelihood estimation for β and Σ_u are

$$\begin{aligned} \hat{\beta} &= \tilde{\beta} + \left((\mathbf{Z}\mathbf{Z}')^{-1} \otimes \hat{\Sigma}_u\right) \mathbf{C}' \left[\mathbf{C}\left((\mathbf{Z}\mathbf{Z}')^{-1} \otimes \hat{\Sigma}_u\right) \mathbf{C}'\right]^{-1} \left(\mathbf{c} - \mathbf{C}\tilde{\beta}\right) \text{ and} \\ \hat{\Sigma}_u &= \frac{1}{T}\left(\mathbf{Y} - \hat{\mathbf{B}}\mathbf{Z}\right)\left(\mathbf{Y} - \hat{\mathbf{B}}\mathbf{Z}\right)', \end{aligned} \tag{9}$$

respectively, where $\tilde{\beta} = \left((\mathbf{Z}\mathbf{Z}')^{-1}\mathbf{Z} \otimes \mathbf{I}_K\right)\mathbf{y}$.

A fully parametrized K-dimensional VAR(p) model contains $K(Kp+1)$ parameters or $K^2 p$ parameters when the intercept terms are excluded, which means it is an over-parametrization problem to fit a VAR model when the dimension K is large relative to the sample size. To overcome this limitation, researchers have explored various methods to identify the zero autoregressive coefficients. The penalized regression methods for VAR modelling are some of the possible ways to determine the non-zero autoregressive coefficients. These methods consider a penalized regression problem when the VAR model is reformulated as a linear regression and use the popular Lasso penalty proposed by Tibshirani (1996) and its variants to shrink the values of the autoregressive coefficients (Haufe et al., 2009; Song & Bickel, 2011). In particular, Song & Bickel (2011) considered the use of group Lasso (Yuan & Lin, 2006) to identify sparsity and structural pattern in the model. Nicholson et al. (2017) also considered the utilization of group Lasso penalty to achieve lag sparsity. In the penalized regression methods, the contemporaneous dependence in the time series is ignored, since the loss function used is the sum of squared residuals which does not take the innovation covariance into account in the estimation. Song & Bickel (2011) discussed the possible impact when the contemporaneous dependence is not considered in fitting a VAR model.

Davis et al. (2016) proposed a two-stage approach for fitting a sparse VAR model. The partial spectral coherences are first calculated to identify the possible non-zero autoregressive coefficients. With the sparsity constraints on the autoregressive coefficients, the parameters are estimated by using the constrained maximum likelihood estimation. The number of pairs of non-zero autoregressive parameters M and the lag order p are chosen by minimizing the BIC of fitted VAR models over pre-specified ranges of M and p. The selected model is then refined to shrink the non-zero autoregressive coefficients further by comparing the t-statistics of the autoregressive coefficients.

2.2. Partial correlation graph

We provide a brief introduction to the partial spectral coherence and partial correlation graph below, based on Hu et al. (2016). See Dahlhaus (2000) for details. Suppose $\mathbf{X}_V(t) = (X_1(t), X_2(t), ..., X_n(t))'$, $t \in \mathbb{Z}$, is a multivariate weakly stationary time series and $\mathbf{Y}_{ab}(\cdot) = (X_j(\cdot),\ j \neq a,b)$. Let $G = (V,E)$ denote a graph, where $V = \{1,2,...,n\}$ is the set of vertices and $E = \{(a,b) \in V \times V\}$ is the set of edges. Thus, each node corresponds to one of the time series in $\mathbf{X}_V(t)$. The edge between node a and node b is characterized as follows.

In theory, we remove the linear effect of $Y_{ab}(t)$ from $X_a(t)$ by determining the optimal filters $g_j(u)$ which minimize $E\left[X_a(t) - \sum_{j \in V \backslash \{a,b\}} \sum_{u=-\infty}^{\infty} g_{a,j}(u) X_j(t-u)\right]^2$. Denote the optimal filters by $\hat{g}_{a,j}(u)$ for $j \in V \backslash \{a,b\}$ and $u \in \mathbb{Z}$, the remainders after removing the linear effect of $X_{V \backslash \{a,b\}}(t)$ from $X_a(t)$ and $X_b(t)$ are

$$\varepsilon_{a|V\backslash\{a,b\}}(t) = X_a(t) - \sum_{j \in V\backslash\{a,b\}} \sum_{u=-\infty}^{\infty} \hat{g}_{a,j}(u) X_j(t-u) \quad \text{and}$$

$$\varepsilon_{b|V\backslash\{a,b\}}(t) = X_b(t) - \sum_{j \in V\backslash\{a,b\}} \sum_{u=-\infty}^{\infty} \hat{g}_{b,j}(u) X_j(t-u),$$

respectively. Then, the two series are conditionally uncorrelated if and only if $\operatorname{cov}\left(\varepsilon_{a|V\backslash\{a,b\}}(t), \varepsilon_{b|V\backslash\{a,b\}}(t+u)\right) = 0$ for all $t, u \in \mathbb{Z}$ and hence $\mathscr{X}_a \perp\!\!\!\perp \mathscr{X}_b | \mathscr{Y}_{ab}$. Equivalently $(a,b) \notin E \iff \mathscr{X}_a \perp\!\!\!\perp \mathscr{X}_b | \mathscr{Y}_{ab}$, $G = (V,E)$ is the partial correlation graph for the time series of interest. The edges of the graph can be determined by two approaches, namely the time domain approach and the frequency domain approach.

Time domain approach

The partial correlation graph represents the linear association between two component series after removing the linear effect of all other components by two-sided filters. Similarly, we consider a bivariate VAR model to estimate the cross-correlation of the residuals, $\varepsilon_{a|V\backslash\{a,b\}}(t)$ and $\varepsilon_{b|V\backslash\{a,b\}}(t)$, following Hu et al. (2016). To illustrate the method, suppose $\mathbf{X}_V(t) = (X_1(t), X_2(t), X_3(t), X_4(t))'$, the VAR model to determine the partial cross-correlations of X_1 and X_2 is given by

$$\begin{pmatrix} X_1(t) \\ X_2(t) \end{pmatrix} = \begin{pmatrix} \mu_1(t) \\ \mu_2(t) \end{pmatrix} + \sum_{u=0}^{q} \mathbf{F}_u \begin{pmatrix} X_3(t-u) \\ X_4(t-u) \end{pmatrix} + \sum_{u=1}^{q} \Phi_u \begin{pmatrix} X_1(t-u) \\ X_2(t-u) \end{pmatrix} + \begin{pmatrix} e_{1|\{3,4\}}(t) \\ e_{2|\{3,4\}}(t) \end{pmatrix}.$$

Then, the partial cross-correlations of X_1 and X_2 given the remaining processes, denoted by $\rho_{X_1 X_2 | Y_{12}}(u)$, are the cross-correlations of $e_{1|\{3,4\}}(t)$ and $e_{2|\{3,4\}}(t)$, and other partial cross-correlations are computed similarly. The lag order q is determined by choosing a model that possesses the minimum BIC value among the bivariate models with various lag order over a pre-specified range of q, and the approximate 5% error bound of $\pm 2/\sqrt{T}$ is adopted for testing the partial cross-correlations. We note that the time domain approach in identifying a partial correlation graph of time series filters out the linear effect of the remaining components by one-sided filters, which only consider the past and present observations in the filtering. An alternative to determine a partial correlation graph of time series is the frequency domain method, which adopts two-sided filtering in the identification, and is introduced in the next part.

Frequency domain approach

The partial correlation graph of a time series can be determined from the partial spectral coherences, which is computed by first determining the cross-spectral density of the series. The cross-spectral density is defined by

$$f_{X_aX_b}(\lambda) = \frac{1}{2\pi}\sum_{u=-\infty}^{\infty} C_{ab}(u)e^{-i\lambda u}, \tag{10}$$

where $C_{ab}(u)$ is the cross-covariance function of $X_a(t)$ and $X_b(t)$ at lag u. Then, the partial spectral density is given by

$$f_{X_aX_b|Y_{ab}}(\lambda) = f_{X_aX_b}(\lambda) - \mathbf{f}_{X_aY}(\lambda)\mathbf{f}_{YY}(\lambda)^{-1}\mathbf{f}_{YX_b}(\lambda)^{*}, \tag{11}$$

where $\mathbf{A}^*$ is the conjugate transpose of matrix $\mathbf{A}$; $\mathbf{f}_{X_aY}(\lambda)$, $\mathbf{f}_{YY}(\lambda)$ and $\mathbf{f}_{YX_b}(\lambda)$ are some partitions of the spectral density matrix, see Hu et al. (2016) for details. The cross-spectral density $f_{X_aX_b}$ measures the degree of linear association between the two components, and the partial cross-spectral density $f_{X_aX_b|Y_{ab}}$ measures the degree of linear association of $X_a(t)$ and $X_b(t)$, after removing the influence of the remaining components.

The spectral coherence $R_{X_aX_b}(\lambda)$ and the partial spectral coherence $R_{X_aX_b|Y_{ab}}(\lambda)$, are defined by

$$R_{X_aX_b}(\lambda) = \frac{f_{X_aX_b}(\lambda)}{\left[f_{X_aX_a}(\lambda)f_{X_bX_b}(\lambda)\right]^{1/2}} \quad \text{and} \quad R_{X_aX_b|Y_{ab}}(\lambda) = \frac{f_{X_aX_b|Y_{ab}}(\lambda)}{\left[f_{X_aX_a|Y_{ab}}(\lambda)f_{X_bX_b|Y_{ab}}(\lambda)\right]^{1/2}},$$

respectively. Since the coherence is the normalization of the cross-spectral density, the links of the partial correlation graph can hence be characterized by the partial spectral coherences. Under the hypothesis of $R_{X_aX_b|Y_{ab}}(\lambda) = 0$, the following test statistic,

$$\frac{(n-q)\hat{R}^2_{X_aX_b|Y_{ab}}(\lambda)}{1-\hat{R}^2_{X_aX_b|Y_{ab}}(\lambda)}$$

follows the F distribution with 2 and $2(n-q)$ degrees of freedom at each frequency λ. Here, n is the equivalent degrees of freedom and q is the number of components other than component a and b. Similarly,

$$\frac{(n-1)\hat{R}^2_{X_aX_b}(\lambda)}{1-\hat{R}^2_{X_aX_b}(\lambda)}$$

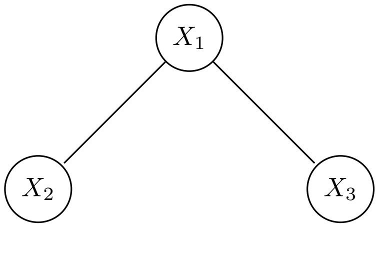
is a test statistic for testing zero coherence, which follows the F distribution with 2 and $2(n-1)$ degrees of freedom at each frequency λ. Thus, the edges in the partial correlation graph can be determined using the test statistics. The estimation of spectral density matrix is required to compute the statistics, we refer the reader to Brillinger (1981) and Koopmans (1974) for details. We next provides an example of the partial correlation graph.

Consider the following 3-dimensional VAR(1) process:

$$\begin{pmatrix} x_{1,t} \\ x_{2,t} \\ x_{3,t} \end{pmatrix} = \begin{pmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & 0 & a_{33} \end{pmatrix} \begin{pmatrix} x_{1,t-1} \\ x_{2,t-1} \\ x_{3,t-1} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \\ \varepsilon_{3,t} \end{pmatrix}, \tag{12}$$

where $\varepsilon = (\varepsilon_{1,t}, \varepsilon_{2,t}, \varepsilon_{3,t})' \sim N(\mathbf{0}, \Sigma)$ with $\Sigma^{-1} = \begin{pmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{12} & \theta_{22} & 0 \\ \theta_{13} & 0 & \theta_{33} \end{pmatrix}$. Figure 1 illustrates the partial correlation graph of this series, where components X_2 and X_3 are conditionally uncorrelated.

Figure 1: The partial correlation graph of the 3-dimensional VAR(1) process in (12), where components X_2 and X_3 are conditionally uncorrelated.



From the partial correlation graph, it is defined that the two components are partially uncorrelated at all lags, including lag zero. Therefore, the corresponding entries of the inverse of innovation covariance matrix Σ_u^{-1} should be insignificant. To impose the sparsity constraints on Σ_u^{-1}, we introduce an iterative algorithm to estimate a sparse VAR model, which will be discussed in the next section.

3. Model estimation and visualization

Most studies have been confined to impose sparsity on the autoregressive coefficients, rather than on the inverse of noise covariance matrix Σ_u^{-1}. We propose an iterative method to impose the sparsity constraints on both the autoregressive coefficients and the inverse of innovation covariance matrix in the estimation of a sparse VAR model.

Consider the following problem:

$$\begin{aligned}
&\underset{\mathbf{B},\Sigma_u}{\text{maximize}} \quad -\frac{KT}{2}\log 2\pi - \frac{T}{2}\log\det\left(\Sigma_u\right) - \frac{1}{2}\operatorname{tr}\left((\mathbf{Y}-\mathbf{BZ})'\Sigma_u^{-1}\left(\mathbf{Y}-\mathbf{BZ}\right)\right) \\
&\text{subject to} \quad \begin{cases} \left(\mathbf{A}_l\right)_{ij} = \left(\mathbf{A}_l\right)_{ji} = 0, \quad \text{for } l = 1,\cdots,p \text{ and } (i,j)\in\Omega, \\ \left(\Sigma_u^{-1}\right)_{ij} = 0, \quad (i,j)\in\Omega, \\ \Sigma_u^{-1} \succ 0, \end{cases}
\end{aligned} \tag{13}$$

where p is a pre-determined lag order and Ω contains the indices of the pairs of components that are conditionally uncorrelated, assuming that $(i,j)\in\Omega$ for $i<j$. This set is determined based on the identified partial correlation graph mentioned in Section 2.2 and will be discussed in Section 3.1. We rewrite the problem in (13), by using the relation: $-\log\det\left(\Sigma_u\right) = \log\det\left(\Sigma_u^{-1}\right)$ and incorporating the zero constraints of the autoregressive coefficients through $\mathbf{C}\beta = \mathbf{0}$, as:

$$\begin{aligned}
&\underset{\mathbf{B},\Sigma_u^{-1}}{\text{maximize}} \quad -\frac{KT}{2}\log 2\pi + \frac{T}{2}\log\det\left(\Sigma_u^{-1}\right) - \frac{1}{2}\operatorname{tr}\left((\mathbf{Y}-\mathbf{BZ})'\Sigma_u^{-1}\left(\mathbf{Y}-\mathbf{BZ}\right)\right) \\
&\text{subject to} \quad \begin{cases} \mathbf{C}\beta = \mathbf{0}, \\ \left(\Sigma_u^{-1}\right)_{ij} = 0, \quad (i,j)\in\Omega, \\ \Sigma_u^{-1} \succ 0, \end{cases}
\end{aligned} \tag{14}$$

where $\beta = \text{vec}(\mathbf{B})$, $\mathbf{C}$ is a matrix of known constants with full row rank, and $\mathbf{0}$ is a vector of zeros.

Theorem 1. *The optimization problem in* (14) *with respect to* $\mathbf{B}$ *and* Σ_u^{-1} *is biconcave.*

PROOF. See Appendix.

Theorem 1 shows that the optimization problem is "biconcave" in the sense that it is concave for either fixed $\mathbf{B}$ or Σ_u^{-1} (Gorski et al., 2007). Thus, we can adopt an iterative algorithm, called Alternate Convex Search (Gorski et al., 2007; Hastie et al., 2015), by first estimating $\mathbf{B}$ followed by estimating Σ_u^{-1}, until a stopping criterion is satisfied. The objective function of the problem for fixed Σ_u^{-1} is strictly concave and is strictly concave on the set of positive definite matrices for fixed $\mathbf{B}$. Therefore, a unique maximizer in each subproblem is obtained. Gorski et al. (2007) stated that each accumulation point generated by the Alternate Convex Search (ACS) algorithm is a stationary point of the objective function under the assumptions of the set of all accumulation points generated by the ACS algorithm form a connected, compact set. Note that the solution obtained using the ACS method is not guaranteed to be the global optimum of the problem.

The solution of Σ_u^{-1} for fixed $\mathbf{B}$ at each iteration guarantees the positive definiteness of the inverse of innovation covariance matrix. This positive definiteness is not ensured when the problem is solved by traditional iterative numerical procedures like Newton-Raphson method. The suggested iterative method does not require the computation of the Hessian or information matrix comparing to the Newton-Raphson method.

We note that the zero constraints are chosen based on the identified partial correlation graph mentioned in Section 2.2. Before introducing the proposed alternating maximization method for the estimation, we discuss the possible methods in identifying the constraint structure in the next section.

3.1. Estimation of the structure

To identify the constraint structure, we first determine the partial correlation graph of a series by the frequency or time domain methods, introduced in Section 2.2. Suppose a partial correlation graph of Figure 1 is identified, the constraint structure for model estimation is

$$\left(\Sigma_u^{-1}\right)_{23} = \left(\Sigma_u^{-1}\right)_{32} = 0 \quad \text{and} \quad (\mathbf{A}_l)_{23} = (\mathbf{A}_l)_{32} = 0, \quad \text{for } l = 1, \cdots, p.$$

The lag order p is determined by standard information criteria, such as BIC or HQC (Hannan & Quinn, 1979), before applying the alternating maximization method. In the calculation of the information criteria, we count all the unconstrained autoregressive coefficients and the unconstrained inverse noise covariances at the upper triangular part of the matrix as the number of parameters m, i.e. $m = K^2 p + \frac{K(K+1)}{2} - (2p+1)|\Omega|$.

In practice, some of the partial cross-correlations (partial spectral coherences) are marginally significant at few lags (frequencies) leading to some weak links in the estimated partial correlation graph. We can therefore further reduce the number of parameters. For the marginal partial cross-correlations, we rank them by their absolute values, $\max_u |\hat{\rho}_{X_a X_b|Y_{ab}}(u)|$, (or the supremum of the test statistics of the partial spectral coherences, $\sup_\lambda \frac{(n-q)\hat{R}^2_{X_a X_b|Y_{ab}}(\lambda)}{1-\hat{R}^2_{X_a X_b|Y_{ab}}(\lambda)}$), in descending order. Then we can exclude some of the originally marginal partial cross-correlations in a forward stepwise regression manner. We finally select the model that possesses the minimum BIC value among the fitted models. We next present the iterative estimation algorithm.

3.2. Proposed iterative method

(i) Initialization: Set the initial estimates using the unconstrained maximum likelihood estimators in (6),

$$\hat{\mathbf{B}}_{(0)} = \mathbf{Y}\mathbf{Z}'\left(\mathbf{Z}\mathbf{Z}'\right)^{-1} \text{ and } \quad \hat{\Sigma}^{-1}_{u_{(0)}} = \left[\frac{1}{T}\left(\mathbf{Y}-\hat{\mathbf{B}}_{(0)}\mathbf{Z}\right)\left(\mathbf{Y}-\hat{\mathbf{B}}_{(0)}\mathbf{Z}\right)'\right]^{-1}.$$

Remark 1. Although the starting point is not in the feasible region, the initialization can be considered as a warm start, since the solution of next iteration is feasible.

(ii) **B** step: Given the estimate of Σ_u^{-1} at the $(k-1)$-th iteration, denoted by $\hat{\Sigma}^{-1}_{u_{(k-1)}}$, the estimate of **B** at the k-th iteration is

$$\hat{\beta}_{(k)} = \operatorname{vec}\left(\hat{\mathbf{B}}_{(k)}\right) = \tilde{\beta} - \left((\mathbf{Z}\mathbf{Z}')^{-1} \otimes \hat{\Sigma}_{u_{(k-1)}}\right)\mathbf{C}'\left[\mathbf{C}\left((\mathbf{Z}\mathbf{Z}')^{-1} \otimes \hat{\Sigma}_{u_{(k-1)}}\right)\mathbf{C}'\right]^{-1}\mathbf{C}\tilde{\beta}, \qquad (15)$$

where $\tilde{\beta} = \operatorname{vec}\left(\hat{\mathbf{B}}_{(0)}\right) = \left((\mathbf{Z}\mathbf{Z}')^{-1}\mathbf{Z} \otimes \mathbf{I}_K\right)\mathbf{y}$, which is computed in the initialization stage. That means a substantial part of (15) need not be recalculated at every step.

(iii) Σ_u^{-1} step: Given $\hat{\mathbf{B}}_{(k)}$, solve for Σ_u^{-1} using

$$\begin{aligned} \hat{\Sigma}^{-1}_{u_{(k)}} = \underset{\Sigma_u^{-1} \succ 0}{\arg\max} \quad & \log\det\left(\Sigma_u^{-1}\right) - \operatorname{tr}\left(\mathbf{S}_{(k)}\Sigma_u^{-1}\right) \\ \text{subject to} \quad & \left(\Sigma_u^{-1}\right)_{ij} = 0, \quad (i,j) \in \Omega, \end{aligned} \qquad (16)$$

where $\mathbf{S}_{(k)} = \frac{1}{T}\left(\mathbf{Y}-\hat{\mathbf{B}}_{(k)}\mathbf{Z}\right)\left(\mathbf{Y}-\hat{\mathbf{B}}_{(k)}\mathbf{Z}\right)'$.

(iv) Repeat step (ii) and (iii) until a stopping criterion is met, say $\|\hat{\mathbf{B}}_{(k+1)} - \hat{\mathbf{B}}_{(k)}\|_F < \varepsilon$ and $\|\hat{\Sigma}^{-1}_{u_{(k+1)}} - \hat{\Sigma}^{-1}_{u_{(k)}}\|_F < \varepsilon$, where $\|\cdot\|_F$ denotes the Frobenius norm, and ε is a small positive number, for example, $\varepsilon = 10^{-6}$.

The Lagrange dual function of the covariance selection problem in step (iii) is

$$\begin{aligned} g(\nu) &= \inf_{\Sigma_u^{-1} \succ 0}\left(\log\det\Sigma_u^{-1} - \operatorname{tr}\left(\Sigma_u^{-1}\mathbf{S}\right) - 2\sum_{(i,j)\notin\Omega}\nu_{ij}\left(\Sigma_u^{-1}\right)_{ij}\right) \\ &= -\log\det\left(\mathbf{S} + \sum_{(i,j)\notin\Omega}\nu_{ij}\left(\mathbf{e}_i\mathbf{e}_j{}' + \mathbf{e}_j\mathbf{e}_i{}'\right)\right) - K, \end{aligned}$$

where $\mathbf{e}_i$ is a $(K \times 1)$ i-th unit vector, and ν_{ij} are the Lagrange multipliers for the equality constraints. The dual problem is

$$\begin{aligned} \underset{\mathbf{G} \succ 0}{\text{minimize}} \quad & -\log\det\mathbf{G} \\ \text{subject to} \quad & \mathbf{G}_{ij} = \mathbf{S}_{ij}, \quad (i,j) \in \Omega, \end{aligned}$$

where $\mathbf{G} = \mathbf{S} + \sum_{(i,j)\notin\Omega} \nu_{ij} (\mathbf{e}_i\mathbf{e}_j{}' + \mathbf{e}_j\mathbf{e}_i{}')$, which is a determinant maximization problem (Vandenberghe et al., 1998). Therefore, it can be solved by semidefinite programming (SDP) solvers, such as SDPT3 (Tütüncü et al., 2003) or SeDuMi (Sturm, 1999).

3.3. Model visualization

The estimated model is visualized using a mixed graph (Eichler, 2012), which is defined as follows: Suppose X_V is a K-dimensional stationary Gaussian process with

$$X_V(t) = \sum_{l=1}^{p} \mathbf{A}_l X_V(t-l) + u_V(t), \quad u_V \sim N(\mathbf{0}, \Sigma),$$

where $\mathbf{A}_l$ are $(K \times K)$ matrices and Σ_u is non-singular. Then, a mixed graph $G = (V,\ E)$ is used to visualize the VAR model, in which

(a) $a \longrightarrow b\ \notin E$ whenever $(\mathbf{A}_l)_{ba} = 0 \quad \forall l = 1, \cdots, p$,

(b) $a \text{——} b\ \notin E$ whenever $\Sigma^{-1}_{ab} = \Sigma^{-1}_{ba} = 0$.

We compute the partial correlation coefficients using the formula $\rho_{ij} = -\frac{\Sigma^{-1}_{ij}}{\sqrt{\Sigma^{-1}_{ii}\Sigma^{-1}_{jj}}}$, where Σ^{-1}_{ij} is the (i, j) entry of the inverse of the innovation covariance matrix, for better interpretation.

This mixed graph reflects the dynamic interdependencies among the components of the multiple time series process. In contrast, Oxley et al. (2004) suggested the use of a directed acyclic graph (DAG) to describe a structural vector autoregressive (SVAR) model, which represents each component at each time point by a vertex and encodes both the intra and interdependencies among the variables of the process.

Both graphs show the conditional contemporaneous dependencies. The mixed graph characterizes the undirected edges by the inverse of innovation covariance matrix which exhibits the contemporaneous dependence structure, whereas the DAG identifies the existence of directed edges between current variables based on the corresponding autoregressive coefficients of the current variables in the SVAR model. Indeed, the determination of the conditional contemporaneous dependencies in both methods are similar. To explain this point, we consider an SVAR model and its reduced form. For more detailed exposition, see Tunnicliffe Wilson et al. (2015). Consider an SVAR model of the form

$$\Theta_0 x_t = \Theta_1 x_{t-1} + \Theta_2 x_{t-2} + \cdots + \Theta_p x_{t-p} + e_t,$$

where Θ_0 is non-singular, and the covariance matrix $\mathbf{D}$ of e_t is assumed to be diagonal. This model can be transformed to a VAR model, i.e.,

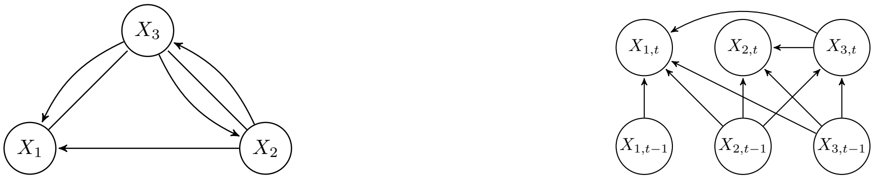
$$\begin{aligned} x_t &= \Theta_0^{-1}\Theta_1 x_{t-1} + \Theta_0^{-1}\Theta_2 x_{t-2} + \cdots + \Theta_0^{-1}\Theta_p x_{t-p} + \Theta_0^{-1} e_t \\ &= \Phi_1 x_{t-1} + \Phi_2 x_{t-2} + \cdots + \Phi_p x_{t-p} + u_t, \end{aligned}$$

where $\Phi_i = \Theta_0^{-1}\Theta_i$, $u_t = \Theta_0^{-1} e_t$, and the covariance matrix of u_t is Σ_u. Therefore, the relation between the residuals e_t of the SVAR and the innovation u_t of the transformed model is

$$\Sigma_u^{-1} = \Theta_0{}' \mathbf{D}^{-1} \Theta_0.$$

Thus, the inverse of innovation covariance matrix in the VAR reflects the conditional dependence between current variables given all the past variables. The inclusion of Θ_0 in SVAR, however, provides an alternative way to capture the conditional contemporaneous dependences. See Figure 2.

Figure 2: The graphical representation of VAR and SVAR model by mixed graph 2 a) and DAG 2 b).



(a)) An example of a mixed graph describing a VAR model.

(b)) An example of a DAG describing a SVAR model.

For the SVAR model, the covariance matrix $\mathbf{D}$ of e_t is assumed to be diagonal so that the model is identifiable. The dependence between current variables is also assumed being recursive and not cyclical so that the matrix Θ_0 is triangular with unit diagonal after reordering the variables. Note that these restrictions are not required when building a VAR model to study the dynamic interdependencies between variables of a multivariate time series process. To us, the VAR model is simpler and more natural.

4. Simulation

In the simulation study, we consider five different stable VAR models, in which the autoregressive coefficient matrix $\mathbf{A}_l$ and the inverse noise covariance matrix Σ_u^{-1} have the same structure (i.e. $(\mathbf{A}_l)_{ij} = (\mathbf{A}_l)_{ji} = \left(\Sigma_u^{-1}\right)_{ij} = \left(\Sigma_u^{-1}\right)_{ji} = 0$, $1 \le i < j \le K$, $l = 1, \cdots, p$), to measure the performance of the estimation method. The inverses of noise covariance matrices of each model are positive definite. We perform the experiments using the following models,

Model 1. $\mathbf{y}_t^{(1)} = \mathbf{A}_1^{(1)}\mathbf{y}_{t-1}^{(1)} + \mathbf{u}_t^{(1)}, \quad \mathbf{u}_t^{(1)} \sim N(\mathbf{0}, \Sigma_1),$

Model 2. $\mathbf{y}_t^{(2)} = \mathbf{A}_1^{(2)}\mathbf{y}_{t-1}^{(2)} + \mathbf{u}_t^{(2)}, \quad \mathbf{u}_t^{(2)} \sim N(\mathbf{0}, \Sigma_2),$

Model 3. $\mathbf{y}_t^{(3)} = \mathbf{A}_1^{(3)}\mathbf{y}_{t-1}^{(3)} + \mathbf{u}_t^{(3)}, \quad \mathbf{u}_t^{(3)} \sim N(\mathbf{0}, \Sigma_3),$

Model 4. $\mathbf{y}_t^{(4)} = \mathbf{A}_1^{(4)}\mathbf{y}_{t-1}^{(4)} + \mathbf{u}_t^{(4)}, \quad \mathbf{u}_t^{(4)} \sim N(\mathbf{0}, \Sigma_4),$

Model 5. $\mathbf{y}_t^{(5)} = \mathbf{A}_1^{(5)}\mathbf{y}_{t-1}^{(5)} + \mathbf{A}_2^{(5)}\mathbf{y}_{t-2}^{(5)} + \mathbf{u}_t^{(5)}, \quad \mathbf{u}_t^{(5)} \sim N(\mathbf{0}, \Sigma_5),$

where

$$\mathbf{A}_1^{(1)} = \begin{pmatrix} -0.7458 & 0.3938 & -0.9575 \\ -0.1824 & -0.6798 & 0 \\ -0.1779 & 0 & 0.4294 \end{pmatrix}, \; \Sigma_1^{-1} = \begin{pmatrix} 1.3030 & -1.0613 & 0.8662 \\ -1.0613 & 1.4196 & 0 \\ 0.8662 & 0 & 2.6625 \end{pmatrix},$$

$$\mathbf{A}_1^{(2)} = \begin{pmatrix} 0.9508 & 0 & 0.4352 & 0 \\ 0 & -0.8232 & 0.5138 & 0.0274 \\ -0.8592 & -0.8289 & 0.6247 & 0.5984 \\ 0 & -0.4878 & -0.1426 & -0.6542 \end{pmatrix}, \; \Sigma_2^{-1} = \begin{pmatrix} 1.7975 & 0 & 0.1025 & 0 \\ 0 & 3.1785 & 0.8908 & 0.5532 \\ 0.1025 & 0.8908 & 0.4838 & 0.0586 \\ 0 & 0.5532 & 0.0586 & 4.1300 \end{pmatrix},$$

$$\mathbf{A}_1^{(3)} = \begin{pmatrix} 0.4352 & -0.6552 & 0.4154 & 0.3930 & -0.5200 & 0.2256 \\ 0.1478 & -0.4932 & 0 & 0 & 0 & 0 \\ -0.7940 & 0 & -0.8933 & 0 & 0 & 0 \\ 0.5894 & 0 & 0 & -0.1478 & 0 & 0 \\ -0.8009 & 0 & 0 & 0 & -0.4169 & 0 \\ 0.4197 & 0 & 0 & 0 & 0 & -0.2439 \end{pmatrix}, \; \Sigma_3^{-1} = \begin{pmatrix} 1 & 0.4 & 0.4 & 0.4 & 0.4 & 0.4 \\ 0.4 & 1 & 0 & 0 & 0 & 0 \\ 0.4 & 0 & 1 & 0 & 0 & 0 \\ 0.4 & 0 & 0 & 1 & 0 & 0 \\ 0.4 & 0 & 0 & 0 & 1 & 0 \\ 0.4 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\mathbf{A}_1^{(4)} = \begin{pmatrix} 0.2177 & 0.3066 & 0 & 0 & 0 & 0.3775 \\ -0.6324 & -0.6650 & 0.0214 & 0 & 0 & 0 \\ 0 & -0.2749 & -0.7509 & 0.4482 & 0 & 0 \\ 0 & 0 & -0.3046 & -0.8066 & 0.9940 & 0 \\ 0 & 0 & 0 & -0.7313 & 0.5054 & 0.7959 \\ -0.0587 & 0 & 0 & 0 & -0.5140 & -0.9470 \end{pmatrix}, \; \Sigma_4^{-1} = \begin{pmatrix} 1 & 0.4 & 0 & 0 & 0 & 0.4 \\ 0.4 & 1 & 0.4 & 0 & 0 & 0 \\ 0 & 0.4 & 1 & 0.4 & 0 & 0 \\ 0 & 0 & 0.4 & 1 & 0.4 & 0 \\ 0 & 0 & 0 & 0.4 & 1 & 0.4 \\ 0.4 & 0 & 0 & 0 & 0.4 & 1 \end{pmatrix},$$

$$\mathbf{A}_1^{(5)} = \begin{pmatrix} -0.6 & 0.4 & 0 & 0 & 0 & 0.4 \\ 0.4 & -0.6 & 0.4 & 0 & 0 & 0 \\ 0 & 0.4 & -0.6 & 0.4 & 0 & 0 \\ 0 & 0 & 0.4 & -0.6 & 0.4 & 0 \\ 0 & 0 & 0 & 0.4 & -0.6 & 0.4 \\ 0.4 & 0 & 0 & 0 & 0.4 & -0.6 \end{pmatrix}, \; \mathbf{A}_2^{(5)} = \begin{pmatrix} -0.3 & 0.2 & 0 & 0 & 0 & 0.2 \\ 0.2 & -0.3 & 0.2 & 0 & 0 & 0 \\ 0 & 0.2 & -0.3 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & -0.3 & 0.2 & 0 \\ 0 & 0 & 0 & 0.2 & -0.3 & 0.2 \\ 0.2 & 0 & 0 & 0 & 0.2 & -0.3 \end{pmatrix} \text{ and}$$

$$\Sigma_5^{-1} = \begin{pmatrix} 1 & -0.3 & 0 & 0 & 0 & -0.3 \\ -0.3 & 1 & -0.3 & 0 & 0 & 0 \\ 0 & -0.3 & 1 & -0.3 & 0 & 0 \\ 0 & 0 & -0.3 & 1 & -0.3 & 0 \\ 0 & 0 & 0 & -0.3 & 1 & -0.3 \\ -0.3 & 0 & 0 & 0 & -0.3 & 1 \end{pmatrix}.$$

Model 1 is a stable VAR(1) model of dimension three, in which the second and the third components of the series are Granger non-causal and contemporaneously independent (i.e. $\mathbf{A}_{23} = \mathbf{A}_{32} = 0$ and $\left(\Sigma_1^{-1}\right)_{23} = \left(\Sigma_1^{-1}\right)_{32} = 0$). Model 2 is a 4-dimensional VAR(1) model, where the first and second; and the first and fourth components of the multivariate time series are Granger non-causal and contemporaneously independent. We further extend the investigation of the proposed method to some higher dimension stable VAR model. Model 3 is a 6-dimensional stable VAR(1) model with every node connected to the first node in the mixed graph. Model 4 is a stable VAR(1) model in which both the autoregressive coefficient matrix and the inverse noise covariance matrix have a Toeplitz structure. To explore the performance of the estimation method on VAR model with higher lag order p, we consider a 6-dimensional VAR(2) model with Toeplitz autoregressive coefficient matrices and Toeplitz inverse noise covariance matrix.

The experiments are carried out with sample size T of 100, 200, 500, 1000 over 500 replications using MATLAB R2016b on a Linux based workstation with two 2.1 GHz CPUs and 503 GB main memory. We use SDPT3 (Tütüncü et al., 2003) to estimate the inverse noise covariance matrix in the experiments. SDPT3 is a MATLAB based convex optimization tool for solving semidefinite programming. As a comparison to the alternating maximization method, we also solve the optimization problem, assuming the true sparsity structure is known, by two widely used algorithms in nonlinear optimization. They are the interior point algorithm and the direct search method by the MATLAB command 'fmincon' and 'patternsearch', respectively. We impose the positive definiteness constraint, in the two comparison methods, based on the fact that the leading principal minors of the inverse covariance matrix are positive. The following metrics are computed for the comparisons: the bias of the AR coefficient estimates,

$$\text{Bias} = \sum_{l=1}^{p} \sum_{i,j=1}^{K} \left| E\left((\hat{\mathbf{A}}_l)_{i,j} \right) - (\mathbf{A}_l)_{i,j} \right| ; \tag{17}$$

the variance of the AR coefficient estimates,

$$\text{Variance} = \sum_{l=1}^{p} \sum_{i,j=1}^{K} \text{Var}\left[(\hat{\mathbf{A}}_l)_{i,j} \right] ; \tag{18}$$

and the mean squared error (MSE) of the AR coefficient estimates,

$$\text{MSE} = \sum_{l=1}^{p} \sum_{i,j=1}^{K} \left\{ \left[E\left((\hat{\mathbf{A}}_l)_{i,j} \right) - (\mathbf{A}_l)_{i,j} \right]^2 + \text{Var}\left[(\hat{\mathbf{A}}_l)_{i,j} \right] \right\} . \tag{19}$$

For the inverse noise covariance estimates, the upper triangular part of the estimates is considered in the computation of these three metrics since the estimates are symmetric.

We also perform the simulation experiments with unknown structure (including the lag order) and estimate the structure using the frequency domain and time domain methods described in Section 3.1. The metrics are modified to account for the error incurred in selecting a wrong lag order (i.e. the p in the above formulas are changed to be the maximum between the determined lag order and the true lag order, and $(\mathbf{A}_l)_{i,j}$ is defined to be zero whenever $l > p$).

4.1. Simulation with known structure

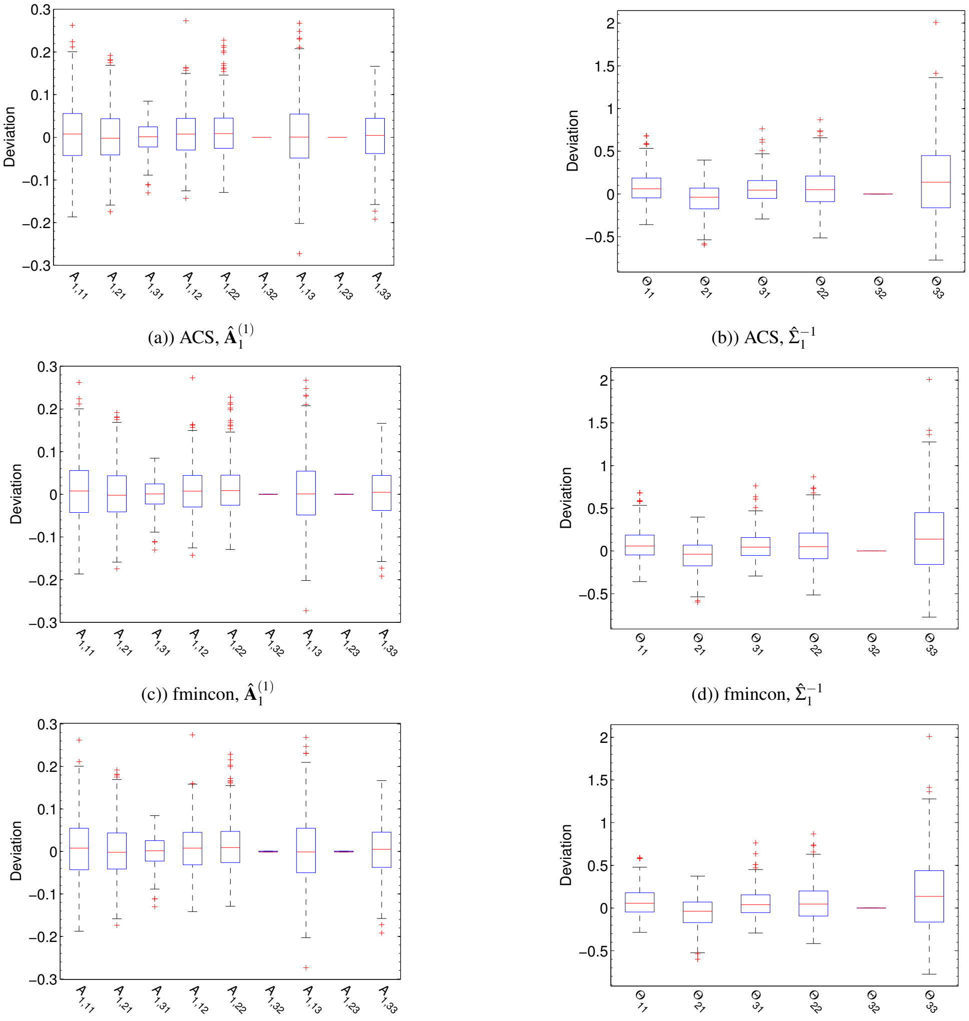
Tables 1 and 2 document the bias, variance and mean squared error (MSE) of the estimates using the three mentioned algorithms (the alternating maximization method, the interior-point method and the direct search method) for the studied models. These three metrics are compiled using the simulation results whenever the corresponding algorithm converges. The columns 'T', 'Method', 'NC', 'Cputime' and 'Iterations' are, respectively, the sample size, the optimization method used, the number of incomplete experiments, due to non-convergence, out of 500 replications, the average CPU time consumed in seconds and the average number of iterations involved in solving the problem. The value in parenthesis is the standard deviation of the corresponding measurement. We consider a completion of the $\mathbf{B}$ step followed by the Σ_u^{-1} step, as mentioned in Section 3.2, as one iteration of the alternating maximization method.

Table 1: Simulation results for Model 1 over 500 replications (The metrics are compiled using the results whenever the corresponding algorithm converges. NC indicates the number of incomplete experiments, Cputime is the average CPU time consumed in seconds, and Iteration is the average number of iterations involved. Standard deviation is in the parenthesis).

						$\hat{\mathbf{A}}$			$\hat{\Sigma}_u^{-1}$	
T	Method	NC	Cputime	Iterations	Bias	Variance	MSE	Bias	Variance	MSE
100	ACS	0	2.3934 (0.8676)	4.1700 (0.6147)	0.0387	0.0280	0.0284	0.4284	0.3047	0.3498
	fmincon	3	10.5573 (0.9191)	30.4064 (1.5358)	0.0392	0.0279	0.0283	0.4297	0.3046	0.3501
	patternsearch	22	1598.7132 (686.5228)	9.0732 (0.2687)	0.0382	0.0280	0.0284	0.4055	0.2937	0.3346
200	ACS	0	2.1998 (0.7500)	3.7740 (0.5133)	0.0229	0.0141	0.0142	0.2831	0.1323	0.1520
	fmincon	3	10.9860 (1.0686)	30.8008 (1.6346)	0.0223	0.0141	0.0142	0.2814	0.1325	0.1520
	patternsearch	4	1555.4650 (518.1913)	9.0444 (0.2061)	0.0227	0.0141	0.0142	0.2812	0.1320	0.1516
500	ACS	0	2.3436 (1.0875)	3.3520 (0.4781)	0.0152	0.0055	0.0056	0.0944	0.0471	0.0494
	fmincon	2	12.4463 (1.4386)	31.8514 (1.7028)	0.0151	0.0055	0.0055	0.0938	0.0472	0.0494
	patternsearch	0	1681.1368 (456.9749)	9.0320 (0.1762)	0.0154	0.0055	0.0056	0.0948	0.0471	0.0494
1000	ACS	0	1.8870 (0.5567)	3.1000 (0.3003)	0.0060	0.0026	0.0026	0.0455	0.0258	0.0262
	fmincon	6	12.9251 (1.5171)	32.6377 (1.9756)	0.0061	0.0026	0.0026	0.0443	0.0256	0.0260
	patternsearch	0	1493.6605 (280.7624)	9.0240 (0.1532)	0.0060	0.0026	0.0026	0.0456	0.0258	0.0262

Table 1 is the simulation results for Model 1 using the three studied methods, namely the alternating maximization method (denoted by 'ACS'), the interior-point algorithm (denoted by 'fmincon') and the direct search method (denoted by 'patternsearch'). Few simulation experiments using the interior-point method do not converge successfully. The direct search method terminates before obtaining a solution in some of the experiments, especially when the sample size is low. The alternating maximization method consumes less CPU time comparing to the two other methods, while the direct search method spends the most. The average number of iterations for the ACS and the direct search methods decrease as the sample size increases. The three metrics (bias, variance and MSE) for both the AR coefficient and the inverse covariance estimates drop steadily as the sample size increases for the three studied methods. We also generate boxplots of deviations of the estimates (i.e. $\hat{\theta} - \theta$) to gain a better insight into the dispersion of the estimates for each method.

Figure 3: Boxplot of deviations of the estimates for Model 1 when $T = 100$.



(a)) ACS, $\hat{\mathbf{A}}_1^{(1)}$

(b)) ACS, $\hat{\Sigma}_1^{-1}$

(c)) fmincon, $\hat{\mathbf{A}}_1^{(1)}$

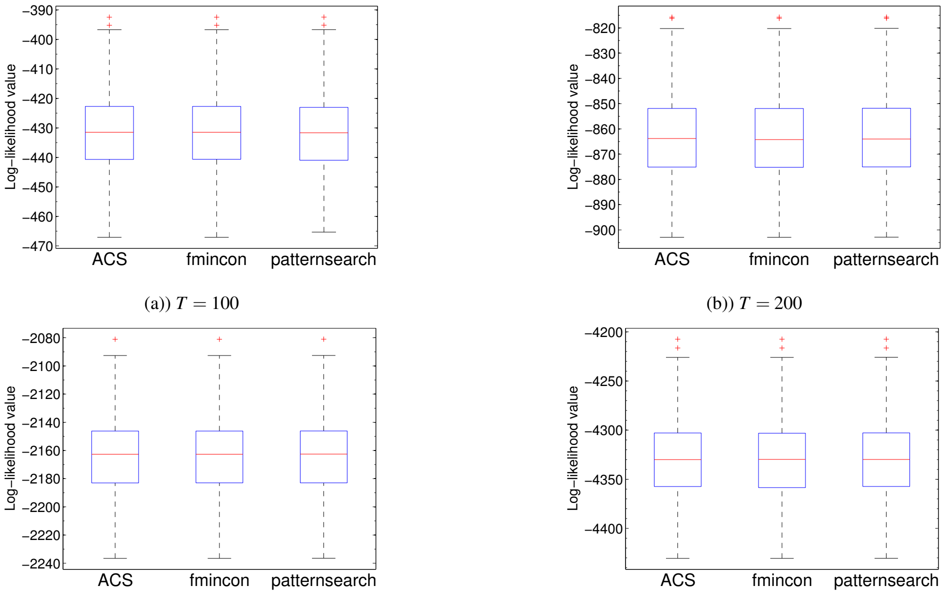
(d)) fmincon, $\hat{\Sigma}_1^{-1}$

(e)) patternsearch, $\hat{\mathbf{A}}_1^{(1)}$

(f)) patternsearch, $\hat{\Sigma}_1^{-1}$

Figure 3 depicts boxplots of deviations of the AR coefficient (on the left panel), and the inverse covariance (on the right panel) estimates for Model 1 when $T = 100$. The deviations are computed whenever the corresponding algorithm converges. We can observe from the boxplots that all algorithms restrict the corresponding AR coefficient and inverse covariance estimates to zero. We note that the 'patternsearch' solver returns a small value even the corresponding parameter is constrained to zero, which leads to a small deviation for the zero constrained estimates. For the unconstrained estimates, both three algorithms obtain estimates that possess similar dispersion. We next consider the log-likelihood values to investigate the convergence properties of the three studied algorithms.

Figure 4: Boxplot of the loglikelihood values for Model 1.



(a)) $T = 100$

(b)) $T = 200$

(c)) $T = 500$

(d)) $T = 1000$

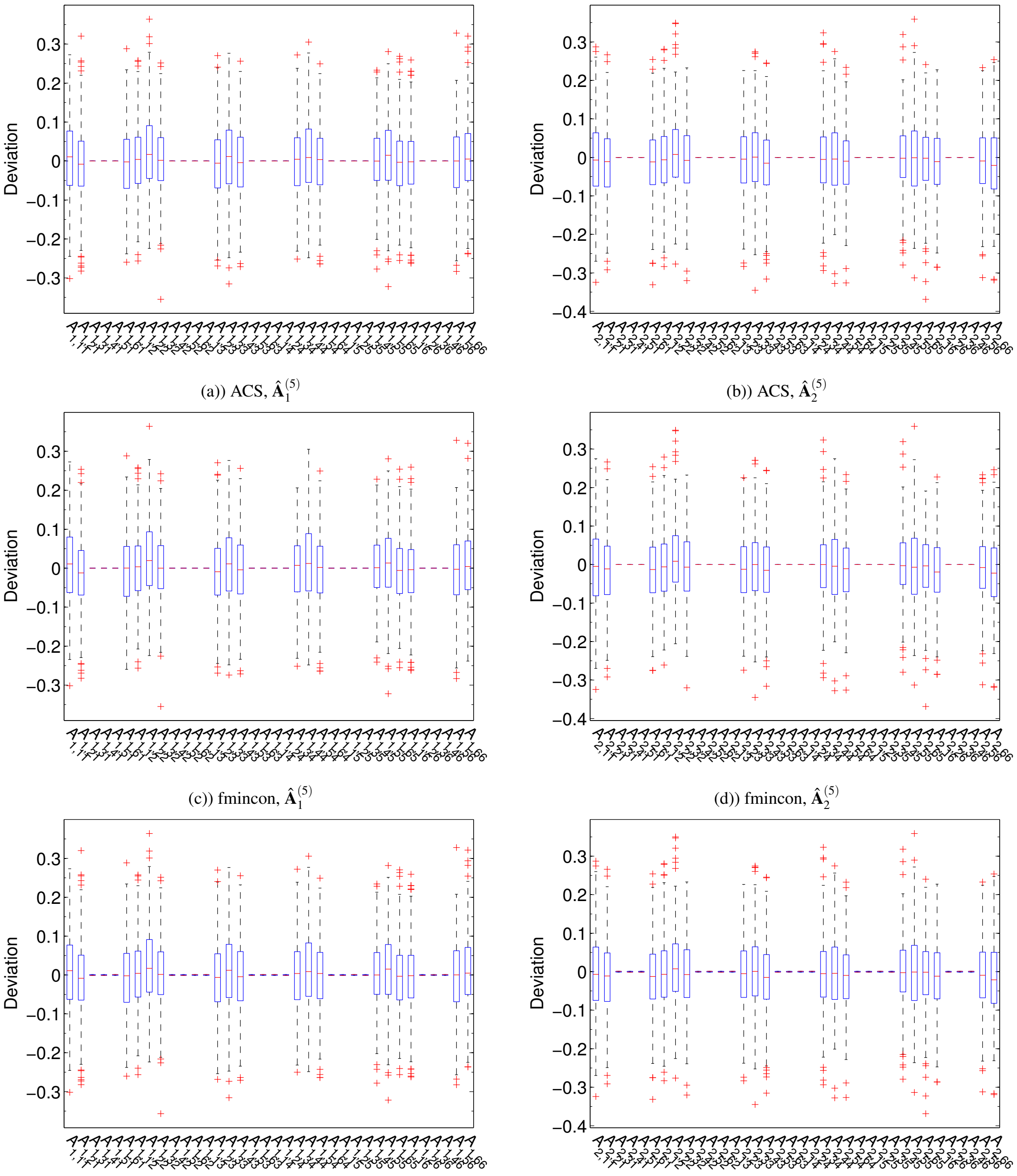
Figure 4 shows boxplots of the log-likelihood values, computed using the obtained AR coefficient and inverse noise covariance estimates, for the three investigated methods with different sample sizes. It is observed that the three algorithms obtain similar log-likelihood values, and the average log-likelihood values are less dispersed as the sample size increases. The results indicate that the log-likelihood values obtained from these three methods converge to some values that are close to each other, whenever the methods converge.

Table 2: Simulation results for Model 5 over 500 replications (The metrics are compiled using the results whenever the corresponding algorithm converges. NC indicates the number of incomplete experiments, Cputime is the average CPU time consumed in seconds, and Iteration is the average number of iterations involved. Standard deviation is in the parenthesis).

					$\hat{\mathbf{A}}$			$\hat{\Sigma}_u^{-1}$		
T	Method	NC	Cputime	Iterations	Bias	Variance	MSE	Bias	Variance	MSE
	ACS	0	2.8934 (0.2696)	6.2340 (0.6066)	0.2682	0.3019	0.3047	0.8525	0.2329	0.3094
100	fmincon	120	97.0799 (9.4558)	94.6684 (8.3720)	0.3044	0.3033	0.3068	0.8943	0.2338	0.3179
	patternsearch	0	4851.2951 (268.9346)	8.0000 (0.0000)	0.2722	0.3019	0.3048	0.8653	0.2338	0.3126
	ACS	0	2.6351 (0.2383)	5.1580 (0.4067)	0.1455	0.1461	0.1469	0.3766	0.0967	0.1122
200	fmincon	123	104.6940 (7.4198)	99.8806 (5.9679)	0.1694	0.1470	0.1481	0.3925	0.0982	0.1151
	patternsearch	0	4616.1554 (340.5639)	8.0000 (0.0000)	0.1478	0.1462	0.1470	0.3850	0.0970	0.1131
	ACS	0	2.3304 (0.2798)	4.2760 (0.4475)	0.0640	0.0585	0.0587	0.1597	0.0362	0.0389
500	fmincon	127	118.6068 (25.2486)	101.4236 (6.9447)	0.0732	0.0587	0.0590	0.1730	0.0359	0.0392
	patternsearch	0	4859.9765 (1098.6708)	8.0000 (0.0000)	0.0662	0.0585	0.0587	0.1648	0.0362	0.0392
	ACS	0	2.2658 (0.1815)	4.0000 (0.0000)	0.0390	0.0291	0.0291	0.0699	0.0174	0.0179
1000	fmincon	145	147.1842 (12.9014)	99.3380 (7.9624)	0.0355	0.0286	0.0286	0.0683	0.0173	0.0178
	patternsearch	0	5945.0423 (250.1966)	8.0000 (0.0000)	0.0409	0.0291	0.0292	0.0737	0.0174	0.0180

Table 2 documents the simulation results for Model 5 using the three studied algorithms, namely the alternating maximization method (denoted by ‘ACS’), the interior-point algorithm (denoted by ‘fmincon’) and the direct search method (denoted by ‘patternsearch’). Some simulation experiments using the interior-point method do not converge successfully. The direct search method obtains a solution in all simulation experiments. The alternating maximization method consumes less CPU time comparing to the two other methods, while the direct search method spends the most. The average computation time and the average number of iterations for the ACS method decrease as the sample size increases. The three metrics (bias, variance and MSE) for both the AR coefficient and the inverse covariance estimates decline gradually as the sample size raises for the three studied methods. We plot boxplots of deviations of the estimates to investigate the dispersion of the estimates for each method.

Figure 5: Boxplot of deviations of the estimates for Model 5 when $T = 100$.



(a)) ACS, $\hat{\mathbf{A}}_1^{(5)}$

(b)) ACS, $\hat{\mathbf{A}}_2^{(5)}$

(c)) fmincon, $\hat{\mathbf{A}}_1^{(5)}$

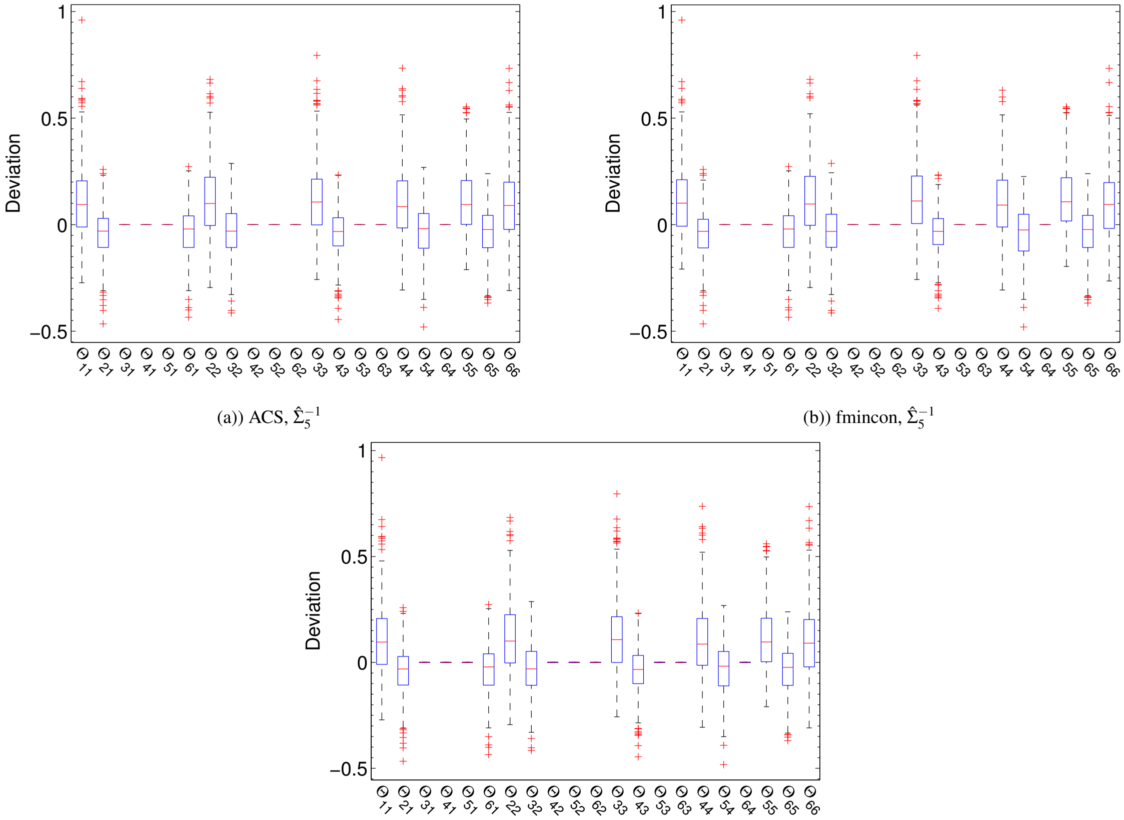
(d)) fmincon, $\hat{\mathbf{A}}_2^{(5)}$

(e)) patternsearch, $\hat{\mathbf{A}}_1^{(5)}$

(f)) patternsearch, $\hat{\mathbf{A}}_2^{(5)}$

Figure 5 displays boxplots of deviations of the AR coefficient of lag 1 (on the left panel), and the lag 2

Figure 6: Boxplot of deviations of the inverse covariance estimates for Model 5 when $T = 100$.



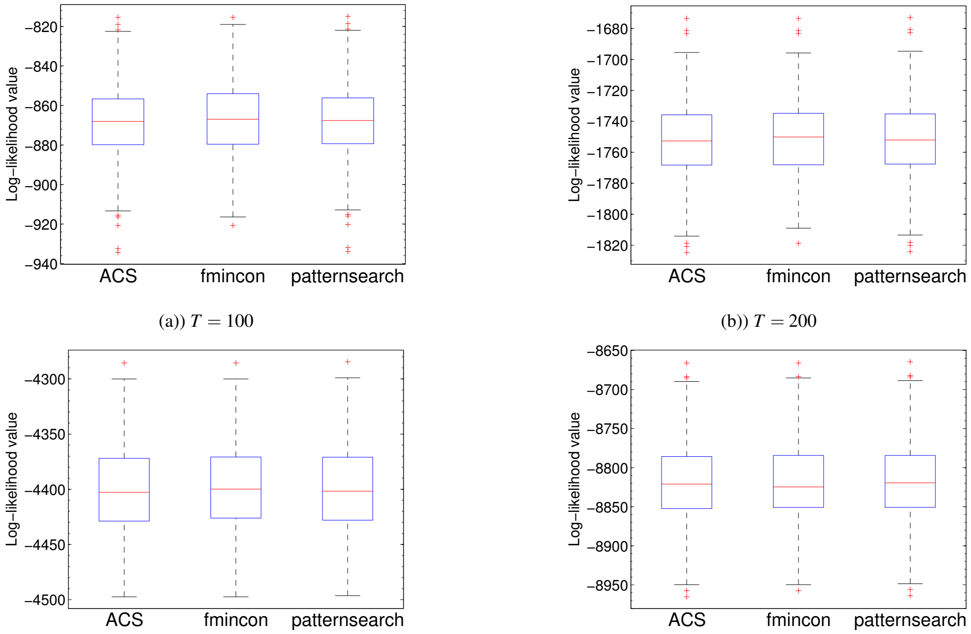
(a)) ACS, $\hat{\Sigma}_5^{-1}$

(b)) fmincon, $\hat{\Sigma}_5^{-1}$

(c)) patternsearch, $\hat{\Sigma}_5^{-1}$

AR coefficient (on the right panel) estimates for Model 5 when $T = 100$. Figure 6 is boxplots of deviations of the inverse covariance estimates when $T = 100$. We can observe from the two figures that all algorithms restrict the corresponding AR coefficient and inverse covariance estimates to zero. For the unconstrained estimates, both three methods obtain estimates that carry similar dispersion. Furthermore, the estimates with larger true parameter values are more dispersed. We next explore the convergence properties of the three studied algorithms by considering the log-likelihood values.

Figure 7: Boxplot of the log-likelihood values for Model 5.



(a)) $T = 100$

(b)) $T = 200$

(c)) $T = 500$

(d)) $T = 1000$

Figure 7 displays boxplots of the log-likelihood values, computed using the obtained AR coefficient and inverse noise covariance estimates, for the three investigated methods with various sample sizes. It is observed that the ACS and the direct search methods obtain similar log-likelihood values, while the interior point method is slightly different. This is because the interior point method does not obtain solution in many simulation experiments. We can also see from the figure that the variability of the average log-likelihood values decreases when the sample size raises. The results suggest that the log-likelihood values obtained from these three methods converge to some values that are close to each other, whenever the methods converge.

In summary, the simulation results reflect that the alternating maximization method is more robust and is rare to obtain a solution that has a significant deviation from the actual parameter, whereas the other two methods fail to converge in some cases, especially when the number of estimation parameters is large. It seems that the alternating method has an advantage that it always converges while the other two methods may not, and the alternating method consumes less CPU time to obtain a solution comparing to the other two algorithms. The results obtained using the alternating method are similar to that acquired by the other two methods whenever these methods converge.

4.2. Simulation with unknown structure

Tables 3 and 4 document the bias, variance and mean squared error (MSE) of the estimates using the frequency domain and the time domain methods introduced in Section 3.1, assuming the lag order and structure are unknown. These three metrics are compiled using all simulation results. The columns 'T', 'Method', 'Cputime' and '$\hat{p}$' are, respectively, the sample size, the algorithm applied, the average CPU time consumed in seconds and the average lag order determined. For the inverse noise covariance estimate, the upper triangular part of the estimate is considered in computing the three metrics (bias, variance and MSE). The value in parenthesis is the standard deviation of the corresponding measurement.

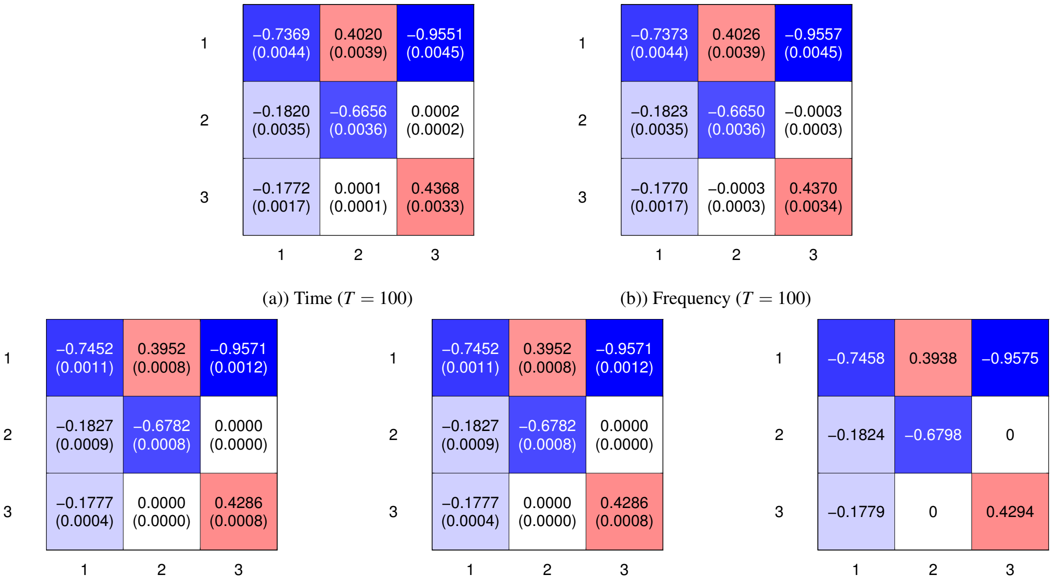
Table 3: Simulation results for Model 1 over 500 replications ($\hat{p}$ is the average lag order determined, and Cputime is the average CPU time consumed in seconds. Standard deviations are in parenthesis).

					$\hat{\mathbf{A}}$			$\hat{\Sigma}_u^{-1}$	
T	Method	Cputime	$\hat{p}$	Bias	Variance	MSE	Bias	Variance	MSE
100	Time	1.2625 (0.3147)	1.1140 (0.4398)	0.0799	0.0771	0.0776	0.4904	0.3221	0.3809
	Frequency	2.5579 (0.1399)	1.1140 (0.4398)	0.0806	0.0775	0.0780	0.4903	0.3214	0.3804
200	Time	1.2214 (0.2728)	1.0200 (0.1538)	0.0260	0.0188	0.0189	0.2889	0.1334	0.1540
	Frequency	2.6691 (0.1126)	1.0200 (0.1538)	0.0260	0.0188	0.0189	0.2889	0.1334	0.1540
500	Time	1.2549 (0.2333)	1.0140 (0.1176)	0.0166	0.0063	0.0063	0.0960	0.0471	0.0495
	Frequency	3.1467 (0.1365)	1.0140 (0.1176)	0.0166	0.0063	0.0063	0.0960	0.0471	0.0495
1000	Time	1.8200 (0.6775)	1.0040 (0.0632)	0.0063	0.0030	0.0030	0.0458	0.0258	0.0263
	Frequency	5.1397 (0.1897)	1.0040 (0.0632)	0.0063	0.0030	0.0030	0.0458	0.0258	0.0263

Table 3 is the simulation results for Model 1 using the methods as mentioned earlier, namely the frequency domain method (denoted by 'Frequency') and the time domain approach (denoted by 'Time'). The frequency domain method consumes more CPU time comparing to the time domain approach. Both methods select a lag order of 2 or above in few experiments when the sample size is 100, and the over-selection of lag order is alleviated as the sample size raises.

It is observed from the column $\hat{\mathbf{A}}$ of Table 3 that the frequency and time domain methods perform similarly, concerning the three metrics, in the estimation of AR coefficients. Figure 8 depicts the average AR estimates using the Frequency method and the Time method when $T = 100$ and $T = 1000$, together with th actual parameter value. From this figure, both methods obtain similar estimates at the same sample size, and the AR coefficient estimates at the positions (2,3) and (3,2) are, in particular, close to zero. The AR coefficient estimates deviate less from the true parameter value and possess less variability when the sample size increases.

Figure 8: Average values of the AR coefficient estimates for Model 1, $\hat{\mathbf{A}}_1^{(1)}$. Standard errors are in parenthesis.



	1	2	3
1	−0.7369 (0.0044)	0.4020 (0.0039)	−0.9551 (0.0045)
2	−0.1820 (0.0035)	−0.6656 (0.0036)	0.0002 (0.0002)
3	−0.1772 (0.0017)	0.0001 (0.0001)	0.4368 (0.0033)

(a)) Time ($T = 100$)

	1	2	3
1	−0.7373 (0.0044)	0.4026 (0.0039)	−0.9557 (0.0045)
2	−0.1823 (0.0035)	−0.6650 (0.0036)	−0.0003 (0.0003)
3	−0.1770 (0.0017)	−0.0003 (0.0003)	0.4370 (0.0034)

(b)) Frequency ($T = 100$)

	1	2	3
1	−0.7452 (0.0011)	0.3952 (0.0008)	−0.9571 (0.0012)
2	−0.1827 (0.0009)	−0.6782 (0.0008)	0.0000 (0.0000)
3	−0.1777 (0.0004)	0.0000 (0.0000)	0.4286 (0.0008)

(c)) Time ($T = 1000$)

	1	2	3
1	−0.7452 (0.0011)	0.3952 (0.0008)	−0.9571 (0.0012)
2	−0.1827 (0.0009)	−0.6782 (0.0008)	0.0000 (0.0000)
3	−0.1777 (0.0004)	0.0000 (0.0000)	0.4286 (0.0008)

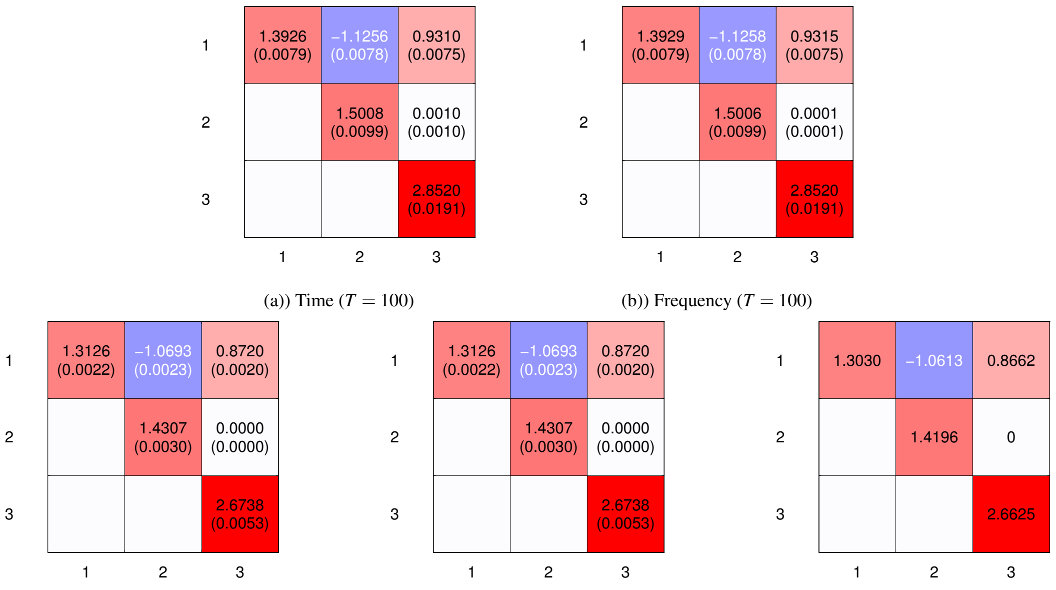
(d)) Frequency ($T = 1000$)

	1	2	3
1	−0.7458	0.3938	−0.9575
2	−0.1824	−0.6798	0
3	−0.1779	0	0.4294

(e)) True

As shown in the column $\hat{\Sigma}_u^{-1}$ of Table 3 that the inverse covariance estimates obtained by the frequency and time domain methods possess bias, variance and MSE that are close in value. Figure 9 plots the average inverse covariance estimates using the two methods when $T = 100$ and $T = 1000$. We can observe from the figure that both methods perform similarly in the estimation of inverse covariances at the same sample size, and the inverse covariance estimates at the positions (2,3) and (3,2) are in particular close to zero. The inverse covariance estimates deviate less from the true parameter value and are less disperse when the sample size raises.

Figure 9: Average values of the inverse covariance estimates for Model 1, $\hat{\Sigma}_1^{-1}$. Standard errors are in parenthesis.



(a)) Time ($T = 100$)

(b)) Frequency ($T = 100$)

(c)) Time ($T = 1000$)

(d)) Frequency ($T = 1000$)

(e)) True

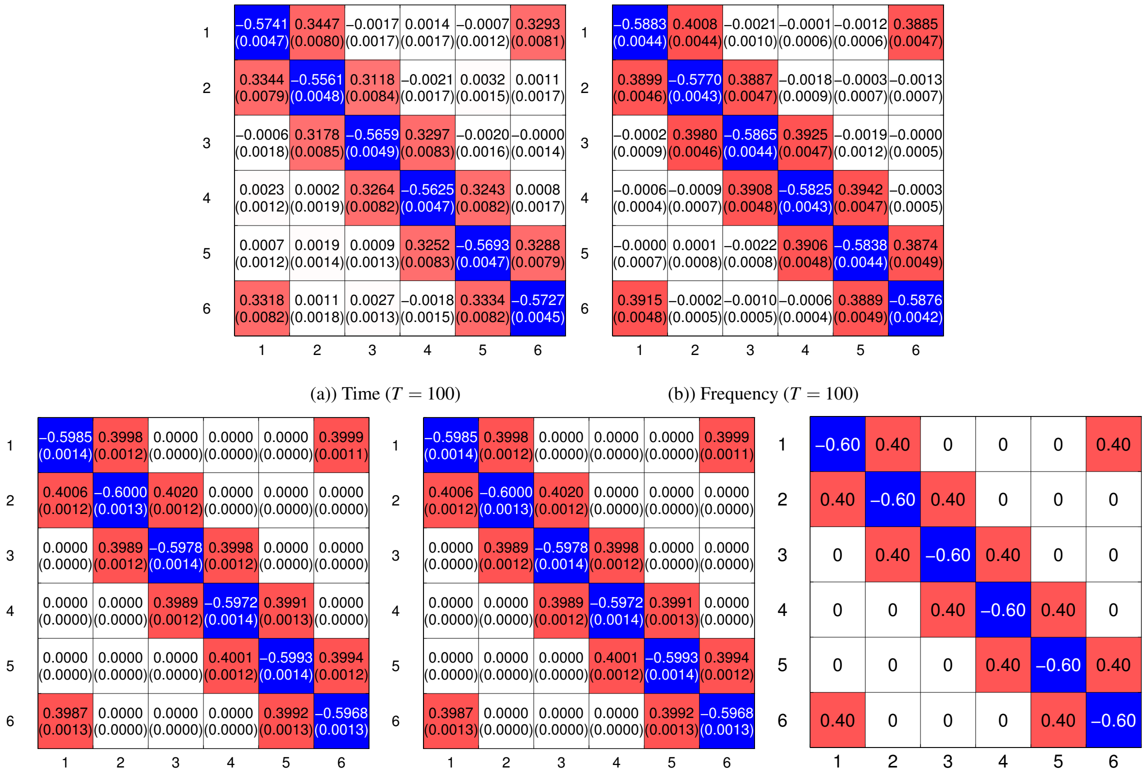
Table 4: Simulation results for Model 5 over 500 replications ($\hat{p}$ is the average lag order determined, and Cputime is the average CPU time consumed in seconds. Standard deviation is in the parenthesis).

				$\hat{\mathbf{A}}$			$\hat{\Sigma}_u^{-1}$		
T	Method	Cputime	$\hat{p}$	Bias	Variance	MSE	Bias	Variance	MSE
100	Time	5.6652 (3.5030)	2.0000 (0.0000)	1.8869	0.7734	0.8776	0.3095	0.4717	0.4786
	Frequency	10.5889 (13.4758)	2.0000 (0.0000)	0.3906	0.3612	0.3658	0.8073	0.2689	0.3381
200	Time	4.4266 (2.0759)	2.0000 (0.0000)	0.2231	0.1816	0.1832	0.3374	0.1148	0.1276
	Frequency	11.6884 (8.3678)	2.0000 (0.0000)	0.1517	0.1471	0.1479	0.3808	0.0971	0.1129
500	Time	4.5902 (1.9192)	2.0000 (0.0000)	0.0640	0.0585	0.0587	0.1597	0.0362	0.0389
	Frequency	17.2995 (10.9602)	2.0000 (0.0000)	0.0640	0.0585	0.0587	0.1597	0.0362	0.0389
1000	Time	7.1894 (2.3372)	2.0000 (0.0000)	0.0390	0.0291	0.0291	0.0699	0.0174	0.0179
	Frequency	21.9524 (3.7277)	2.0000 (0.0000)	0.0390	0.0291	0.0291	0.0699	0.0174	0.0179

Table 4 reports the simulation results for Model 5 using the frequency domain method (denoted by ‘Frequency’) and the time domain approach (denoted by ‘Time’). The frequency domain method consumes more CPU time comparing to the time domain approach. Both the methods choose the lag order correctly in all experiments.

The column $\hat{\mathbf{A}}$ of Table 4 shows that the frequency domain method performs better, regarding the three metrics, in the estimation of AR coefficients when the sample size is 200 or below. Both methods behave similarly in the estimation of AR coefficients when the sample size is 500 or above. Figure 10 (11) depicts the average AR estimates of lag 1 (lag 2) using the two methods when $T = 100$ and $T = 1000$. As shown in the figures, the AR estimates obtained using the Frequency method deviate less from the actual value when the sample size is 100, especially for the non-zero AR coefficients, comparing to the Time method. Both methods obtain similar AR coefficient estimates when the sample size is larger ($T = 1000$).

Figure 10: Average values of the AR coefficient of lag 1 estimates for Model 5, $\hat{\mathbf{A}}_1^{(5)}$. Standard errors are in parenthesis.



(a)) Time ($T = 100$)

	1	2	3	4	5	6
1	–0.5741 (0.0047)	0.3447 (0.0080)	–0.0017 (0.0017)	0.0014 (0.0017)	–0.0007 (0.0012)	0.3293 (0.0081)
2	0.3344 (0.0079)	–0.5561 (0.0048)	0.3118 (0.0084)	–0.0021 (0.0017)	0.0032 (0.0015)	0.0011 (0.0017)
3	–0.0006 (0.0018)	0.3178 (0.0085)	–0.5659 (0.0049)	0.3297 (0.0083)	–0.0020 (0.0016)	–0.0000 (0.0014)
4	0.0023 (0.0012)	0.0002 (0.0019)	0.3264 (0.0082)	–0.5625 (0.0047)	0.3243 (0.0082)	0.0008 (0.0017)
5	0.0007 (0.0012)	0.0019 (0.0014)	0.0009 (0.0013)	0.3252 (0.0083)	–0.5693 (0.0047)	0.3288 (0.0079)
6	0.3318 (0.0082)	0.0011 (0.0018)	0.0027 (0.0013)	–0.0018 (0.0015)	0.3334 (0.0082)	–0.5727 (0.0045)

(b)) Frequency ($T = 100$)

	1	2	3	4	5	6
1	–0.5883 (0.0044)	0.4008 (0.0044)	–0.0021 (0.0010)	–0.0001 (0.0006)	–0.0012 (0.0006)	0.3885 (0.0047)
2	0.3899 (0.0046)	–0.5770 (0.0043)	0.3887 (0.0047)	–0.0018 (0.0009)	–0.0003 (0.0007)	–0.0013 (0.0007)
3	–0.0002 (0.0009)	0.3980 (0.0046)	–0.5865 (0.0044)	0.3925 (0.0047)	–0.0019 (0.0012)	–0.0000 (0.0005)
4	–0.0006 (0.0004)	–0.0009 (0.0007)	0.3908 (0.0048)	–0.5825 (0.0043)	0.3942 (0.0047)	–0.0003 (0.0005)
5	–0.0000 (0.0007)	0.0001 (0.0008)	–0.0022 (0.0008)	0.3906 (0.0048)	–0.5838 (0.0044)	0.3874 (0.0049)
6	0.3915 (0.0048)	–0.0002 (0.0005)	–0.0010 (0.0005)	–0.0006 (0.0004)	0.3889 (0.0049)	–0.5876 (0.0042)

(c)) Time ($T = 1000$)

	1	2	3	4	5	6
1	–0.5985 (0.0014)	0.3998 (0.0012)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	0.3999 (0.0011)
2	0.4006 (0.0012)	–0.6000 (0.0013)	0.4020 (0.0012)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
3	0.0000 (0.0000)	0.3989 (0.0012)	–0.5978 (0.0014)	0.3998 (0.0012)	0.0000 (0.0000)	0.0000 (0.0000)
4	0.0000 (0.0000)	0.0000 (0.0000)	0.3989 (0.0012)	–0.5972 (0.0014)	0.3991 (0.0013)	0.0000 (0.0000)
5	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	0.4001 (0.0012)	–0.5993 (0.0014)	0.3994 (0.0012)
6	0.3987 (0.0013)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	0.3992 (0.0013)	–0.5968 (0.0013)

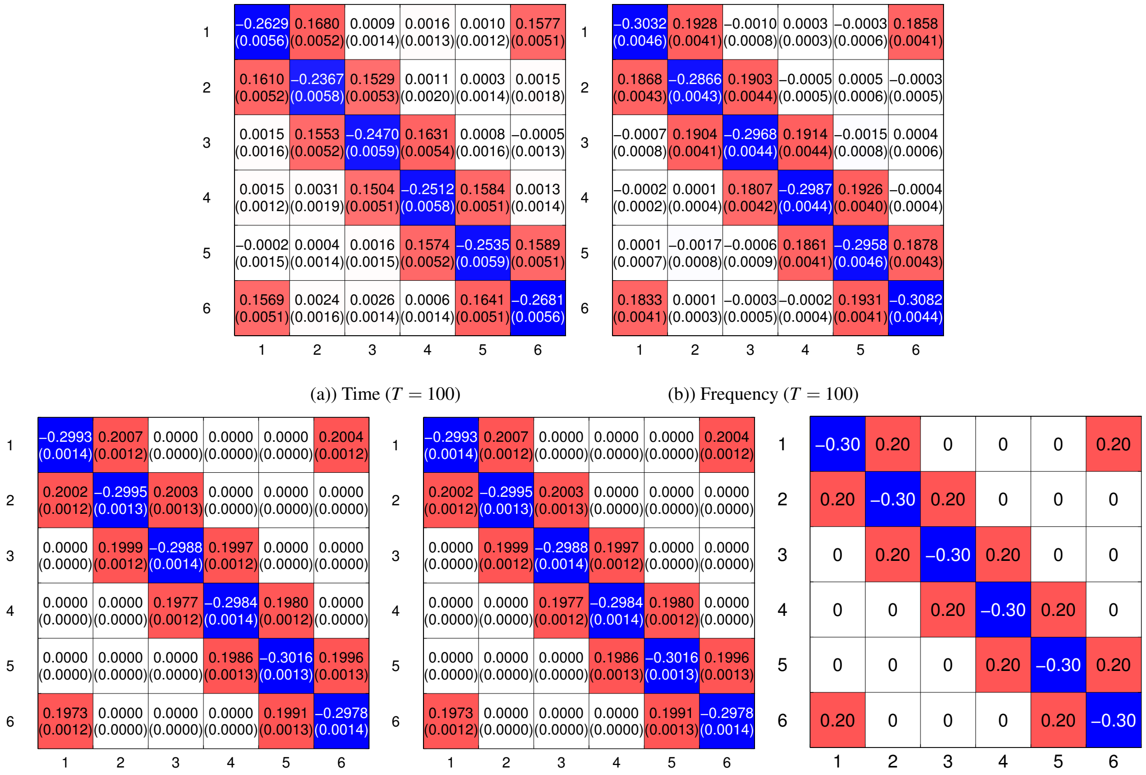
(d)) Frequency ($T = 1000$)

	1	2	3	4	5	6
1	–0.5985 (0.0014)	0.3998 (0.0012)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	0.3999 (0.0011)
2	0.4006 (0.0012)	–0.6000 (0.0013)	0.4020 (0.0012)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
3	0.0000 (0.0000)	0.3989 (0.0012)	–0.5978 (0.0014)	0.3998 (0.0012)	0.0000 (0.0000)	0.0000 (0.0000)
4	0.0000 (0.0000)	0.0000 (0.0000)	0.3989 (0.0012)	–0.5972 (0.0014)	0.3991 (0.0013)	0.0000 (0.0000)
5	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	0.4001 (0.0012)	–0.5993 (0.0014)	0.3994 (0.0012)
6	0.3987 (0.0013)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	0.3992 (0.0013)	–0.5968 (0.0013)

(e)) True

	1	2	3	4	5	6
1	–0.60	0.40	0	0	0	0.40
2	0.40	–0.60	0.40	0	0	0
3	0	0.40	–0.60	0.40	0	0
4	0	0	0.40	–0.60	0.40	0
5	0	0	0	0.40	–0.60	0.40
6	0.40	0	0	0	0.40	–0.60

Figure 11: Average values of the AR coefficient of lag 2 estimates for Model 5, $\hat{\mathbf{A}}_2^{(5)}$. Standard errors are in parenthesis.



	1	2	3	4	5	6
1	–0.2629 (0.0056)	0.1680 (0.0052)	0.0009 (0.0014)	0.0016 (0.0013)	0.0010 (0.0012)	0.1577 (0.0051)
2	0.1610 (0.0052)	–0.2367 (0.0058)	0.1529 (0.0053)	0.0011 (0.0020)	0.0003 (0.0014)	0.0015 (0.0018)
3	0.0015 (0.0016)	0.1553 (0.0052)	–0.2470 (0.0059)	0.1631 (0.0054)	0.0008 (0.0016)	–0.0005 (0.0013)
4	0.0015 (0.0012)	0.0031 (0.0019)	0.1504 (0.0051)	–0.2512 (0.0058)	0.1584 (0.0051)	0.0013 (0.0014)
5	–0.0002 (0.0015)	0.0004 (0.0014)	0.0016 (0.0015)	0.1574 (0.0052)	–0.2535 (0.0059)	0.1589 (0.0051)
6	0.1569 (0.0051)	0.0024 (0.0016)	0.0026 (0.0014)	0.0006 (0.0014)	0.1641 (0.0051)	–0.2681 (0.0056)

(a)) Time ($T = 100$)

	1	2	3	4	5	6
1	–0.3032 (0.0046)	0.1928 (0.0041)	–0.0010 (0.0008)	0.0003 (0.0003)	–0.0003 (0.0006)	0.1858 (0.0041)
2	0.1868 (0.0043)	–0.2866 (0.0043)	0.1903 (0.0044)	–0.0005 (0.0005)	0.0005 (0.0006)	–0.0003 (0.0005)
3	–0.0007 (0.0008)	0.1904 (0.0041)	–0.2968 (0.0044)	0.1914 (0.0044)	–0.0015 (0.0008)	0.0004 (0.0006)
4	–0.0002 (0.0002)	0.0001 (0.0004)	0.1807 (0.0042)	–0.2987 (0.0044)	0.1926 (0.0040)	–0.0004 (0.0004)
5	0.0001 (0.0007)	–0.0017 (0.0008)	–0.0006 (0.0009)	0.1861 (0.0041)	–0.2958 (0.0046)	0.1878 (0.0043)
6	0.1833 (0.0041)	0.0001 (0.0003)	–0.0003 (0.0005)	–0.0002 (0.0004)	0.1931 (0.0041)	–0.3082 (0.0044)

(b)) Frequency ($T = 100$)

	1	2	3	4	5	6
1	–0.2993 (0.0014)	0.2007 (0.0012)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	0.2004 (0.0012)
2	0.2002 (0.0012)	–0.2995 (0.0013)	0.2003 (0.0013)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
3	0.0000 (0.0000)	0.1999 (0.0012)	–0.2988 (0.0014)	0.1997 (0.0012)	0.0000 (0.0000)	0.0000 (0.0000)
4	0.0000 (0.0000)	0.0000 (0.0000)	0.1977 (0.0012)	–0.2984 (0.0014)	0.1980 (0.0012)	0.0000 (0.0000)
5	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	0.1986 (0.0013)	–0.3016 (0.0013)	0.1996 (0.0013)
6	0.1973 (0.0012)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	0.1991 (0.0013)	–0.2978 (0.0014)

(c)) Time ($T = 1000$)

	1	2	3	4	5	6
1	–0.2993 (0.0014)	0.2007 (0.0012)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	0.2004 (0.0012)
2	0.2002 (0.0012)	–0.2995 (0.0013)	0.2003 (0.0013)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
3	0.0000 (0.0000)	0.1999 (0.0012)	–0.2988 (0.0014)	0.1997 (0.0012)	0.0000 (0.0000)	0.0000 (0.0000)
4	0.0000 (0.0000)	0.0000 (0.0000)	0.1977 (0.0012)	–0.2984 (0.0014)	0.1980 (0.0012)	0.0000 (0.0000)
5	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	0.1986 (0.0013)	–0.3016 (0.0013)	0.1996 (0.0013)
6	0.1973 (0.0012)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	0.1991 (0.0013)	–0.2978 (0.0014)

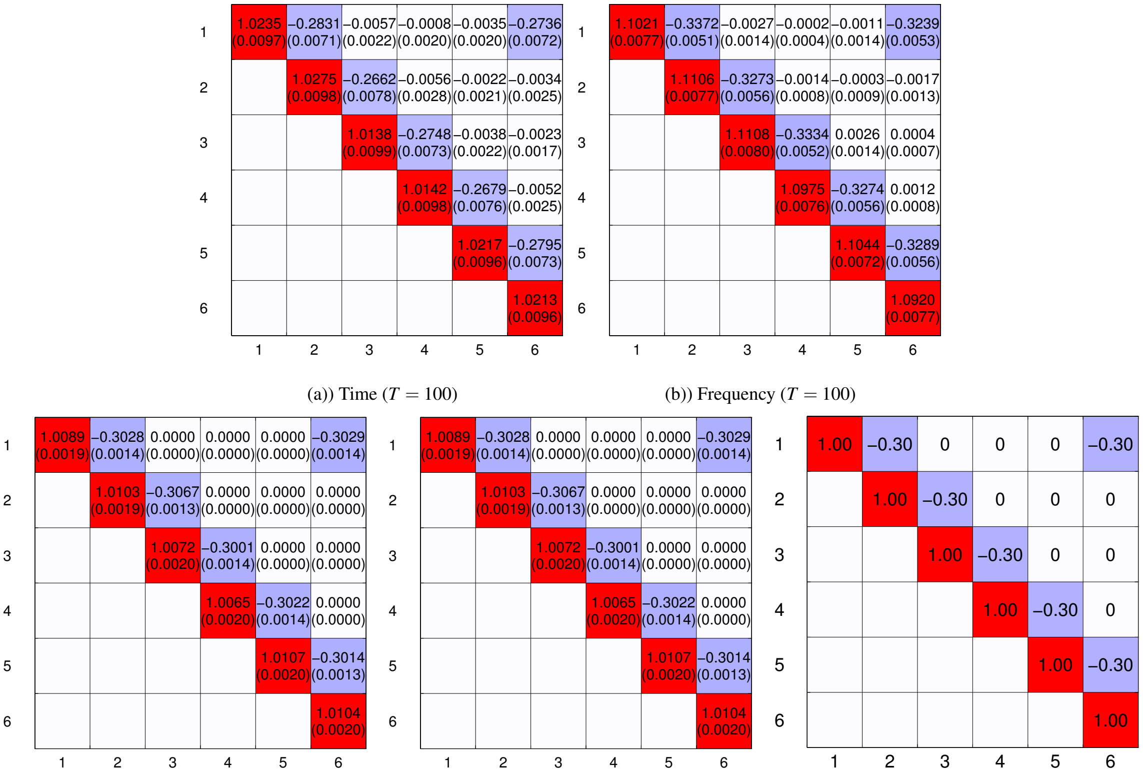
(d)) Frequency ($T = 1000$)

	1	2	3	4	5	6
1	–0.30	0.20	0	0	0	0.20
2	0.20	–0.30	0.20	0	0	0
3	0	0.20	–0.30	0.20	0	0
4	0	0	0.20	–0.30	0.20	0
5	0	0	0	0.20	–0.30	0.20
6	0.20	0	0	0	0.20	–0.30

(e)) True

We can observe from the column $\hat{\Sigma}_u^{-1}$ of Table 4 that the frequency domain method outperforms the time domain method, concerning the variance and MSE of the estimates, in the estimation of inverse covariances when the sample size of 200 or below, although the bias of the estimates is higher. The two methods perform similarly in the estimation when the sample size is 500 or above. Figure 12 shows the average inverse covariance estimates using the Frequency and Time methods when $T = 100$ and $T = 1000$. From this figure, the inverse covariance estimates obtained by the Frequency method deviate slightly higher form the actual values than that obtained by the Time method when $T = 100$, especially for the non-zero inverse covariances. Both methods obtain similar inverse covariance estimates as the sample size raises to 1000.

Figure 12: Average values of the inverse covariance estimates for Model 5, $\hat{\Sigma}_5^{-1}$. Standard errors are in parenthesis.



(a)) Time ($T = 100$)

	1	2	3	4	5	6
1	1.0235 (0.0097)	−0.2831 (0.0071)	−0.0057 (0.0022)	−0.0008 (0.0020)	−0.0035 (0.0020)	−0.2736 (0.0072)
2		1.0275 (0.0098)	−0.2662 (0.0078)	−0.0056 (0.0028)	−0.0022 (0.0021)	−0.0034 (0.0025)
3			1.0138 (0.0099)	−0.2748 (0.0073)	−0.0038 (0.0022)	−0.0023 (0.0017)
4				1.0142 (0.0098)	−0.2679 (0.0076)	−0.0052 (0.0025)
5					1.0217 (0.0096)	−0.2795 (0.0073)
6						1.0213 (0.0096)

(b)) Frequency ($T = 100$)

	1	2	3	4	5	6
1	1.1021 (0.0077)	−0.3372 (0.0051)	−0.0027 (0.0014)	−0.0002 (0.0004)	−0.0011 (0.0014)	−0.3239 (0.0053)
2		1.1106 (0.0077)	−0.3273 (0.0056)	−0.0014 (0.0008)	−0.0003 (0.0009)	−0.0017 (0.0013)
3			1.1108 (0.0080)	−0.3334 (0.0052)	0.0026 (0.0014)	0.0004 (0.0007)
4				1.0975 (0.0076)	−0.3274 (0.0056)	0.0012 (0.0008)
5					1.1044 (0.0072)	−0.3289 (0.0056)
6						1.0920 (0.0077)

(c)) Time ($T = 1000$)

	1	2	3	4	5	6
1	1.0089 (0.0019)	−0.3028 (0.0014)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	−0.3029 (0.0014)
2		1.0103 (0.0019)	−0.3067 (0.0013)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
3			1.0072 (0.0020)	−0.3001 (0.0014)	0.0000 (0.0000)	0.0000 (0.0000)
4				1.0065 (0.0020)	−0.3022 (0.0014)	0.0000 (0.0000)
5					1.0107 (0.0020)	−0.3014 (0.0013)
6						1.0104 (0.0020)

(d)) Frequency ($T = 1000$)

	1	2	3	4	5	6
1	1.0089 (0.0019)	−0.3028 (0.0014)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)	−0.3029 (0.0014)
2		1.0103 (0.0019)	−0.3067 (0.0013)	0.0000 (0.0000)	0.0000 (0.0000)	0.0000 (0.0000)
3			1.0072 (0.0020)	−0.3001 (0.0014)	0.0000 (0.0000)	0.0000 (0.0000)
4				1.0065 (0.0020)	−0.3022 (0.0014)	0.0000 (0.0000)
5					1.0107 (0.0020)	−0.3014 (0.0013)
6						1.0104 (0.0020)

(e)) True

	1	2	3	4	5	6
1	1.00	−0.30	0	0	0	−0.30
2		1.00	−0.30	0	0	0
3			1.00	−0.30	0	0
4				1.00	−0.30	0
5					1.00	−0.30
6						1.00

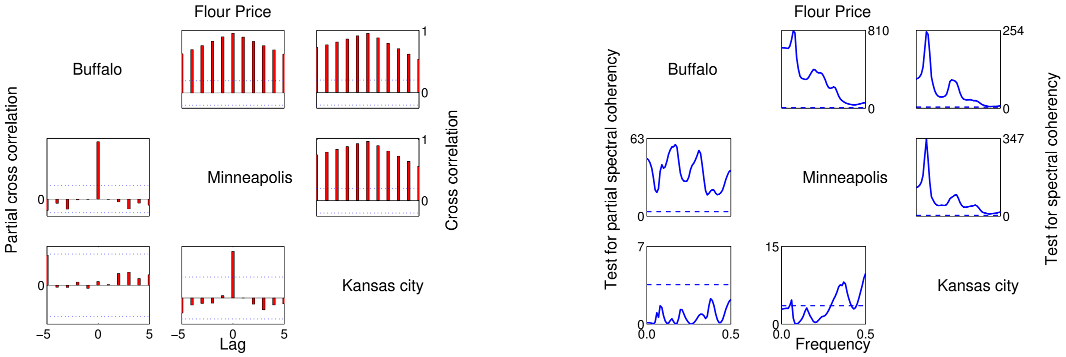
In summary, the time domain and the frequency domain methods perform similarly in the estimation of AR coefficients and inverse covariances, especially when the sample size is large. Their estimates have similar biases and sample variances. The time domain method consumes less CPU time in the estimation comparing to the frequency domain method. This is natural as the latter approach needs the evaluation of Fourier transforms.

5. Applications

5.1. Flour price indices

We employ the introduced method to a monthly flour price indices data in Buffalo, Minneapolis and Kansas city, over the period from August 1972 to November 1980, with a length of 100 months. This dataset has been studied by Tunnicliffe Wilson et al. (2015) to investigate the dynamic interdependencies among the indices by fitting a parsimonious structural vector autoregressive model. We utilize the time domain and frequency domain methods, described in Section 2.2, to identify partial correlation graphs of the three price series. With the determined partial correlation graph, we fit a sparse VAR model to the series using the procedure introduced in Section 3.1 to explore the dynamic interdependencies between the flour price indices. The 2-Stage approach (Davis et al., 2016) is also adopted as a comparison.

Figure 13: Partial cross-correlations and Test statistics of spectral and partial spectral coherences for the flour prices data.



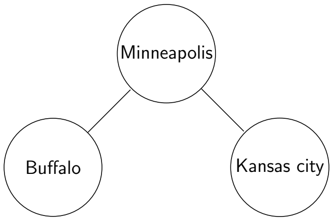
(a)) Partial cross-correlations for the flour prices data (the blue dotted line represents an approximate 5% error bound of $\pm 2/\sqrt{T}$).

(b)) Test statistics of spectral coherences (above diagonal) and partial spectral coherences (below diagonal) for the flour prices data.

Figure 13 a) shows the cross-correlations (upper triangular part) and partial cross-correlations (lower triangular part) for the flour prices data. The blue dotted line represents an approximate 5% error bound of $\pm 2/\sqrt{T}$). This figure suggests the partial cross-correlation of the Buffalo and the Kansas city series is insignificant at all lags. Figure 13 b) depicts the test statistics of spectral (upper triangular part) and partial spectral (lower triangular part) coherences for the flour prices data. The blue dotted line in each subplot is the corresponding critical value of the F distribution at 5% level of significance. This figure shows all coherences are significant, while the partial coherence of the Buffalo and the Kansas city series is insignificant at all frequencies. Based on this result, both methods identify the same partial correlation graph for the flour price indices (Figure 14).

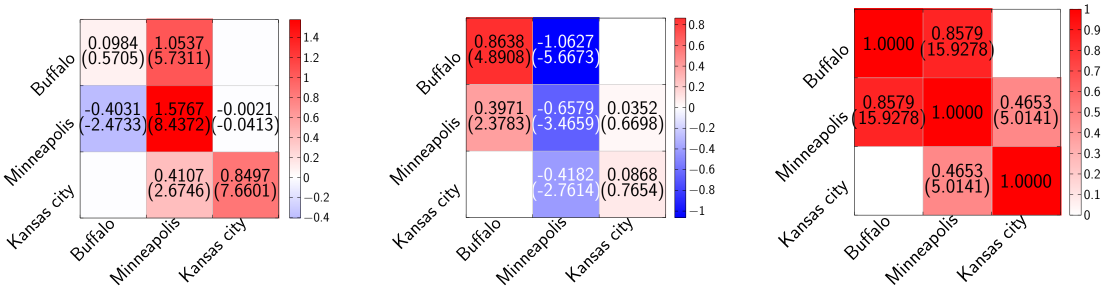
With the identified partial correlation graph, we can determine the sparsity constraints on the autoregressive coefficients and the inverse noise covariances, and estimate a VAR model using the procedure described in Section 3. A model of lag order 2 is identified, and both time and frequency domain methods do not select more autoregressive coefficient pairs and inverse covariances to be zero (i.e. only the autoregressive coefficients and inverse innovation covariance of the case: Buffalo / Kansas city are restricted to zero). The

Figure 14: Partial correlations graph for the flour prices data.



estimated autoregressive coefficients and partial correlations of innovations are visualized by the heatmaps in Figure 15. Figure 16 a) renders the determined VAR model by the mixed graph presented in Section 3.3. Figure 16 b) plots the directed acyclic graph representing the structural VAR model for the flour prices series suggested by Tunnicliffe Wilson et al. (2015), where $X_{1,t}$, $X_{2,t}$ and $X_{3,t}$ represents the flour price indices at time t at Buffalo, Minneapolis and Kansas city, respectively.

Figure 15: The autoregressive coefficient estimates and the estimated partial correlations of innovations using the time and frequency domain methods for the flour prices data (t-values are in parentheses).



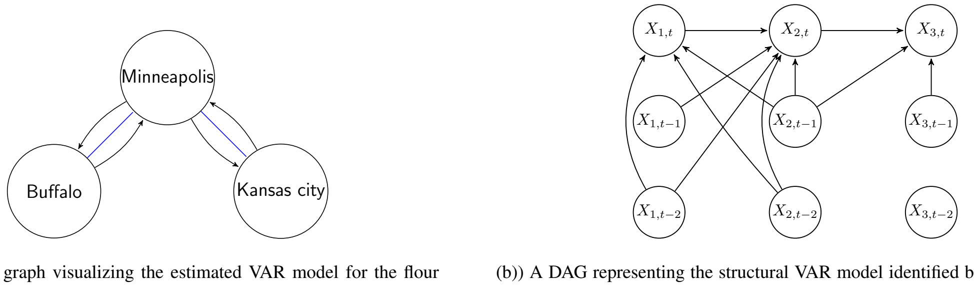
(a)) AR coefficients of lag order 1

(b)) AR coefficients of lag order 2

(c)) Partial correlations of innovations

As shown in Figure 15, some of the autoregressive coefficients are insignificant and all partial correlations of innovations are significant. We note that the estimate VAR model possesses similar dynamic inter-relation structure comparing to the structural VAR model identified by Tunnicliffe Wilson et al. (2015). Both models suggest that the Buffalo and the Kansas city flour price indices are contemporaneously conditionally independent. For the dependence between current and lagged variables, we can observe from the DAG in Figure 16 b) that there are no links in the directions $X_{1,t-1} \to X_{1,t}$, $X_{3,t-1} \to X_{2,t}$, $X_{3,t-2} \to X_{2,t}$, and $X_{3,t-2} \to X_{3,t}$. The autoregressive coefficients, determined by the introduced method, of the corresponding directions are also insignificant, for example, $X_{3,t-1} \to X_{2,t}$ corresponds to the autoregressive coefficient estimates with value $-0.0021(-0.0413)$ in the estimated VAR model (see Figure 15 a)).

Figure 16: Graphs for the flour prices data.



(a)) A mixed graph visualizing the estimated VAR model for the flour prices data.

(b)) A DAG representing the structural VAR model identified by Tunnicliffe Wilson et al. (2015) for the flour prices data.

The 2-Stage method (Davis et al., 2016) obtains a sparse VAR model of order 6, which perhaps is because the method over-selects the autoregressive coefficients to be zero in the first stage of the 2-Stage approach. The method, in particular, constrains the autoregressive coefficient matrices to be diagonal for all lag order, leading to a less interpretable VAR model regarding the dynamic interdependencies structure. Heatmaps of the estimated autoregressive coefficients and partial correlations of innovations by the 2-Stage method are shown on page 50 in the online appendix.

5.2. Air pollution data in the Pearl River Delta region

We apply the proposed estimation method to an air pollution time series data in the Pearl River Delta region (PRDR)[1]. The government authorities have published the monthly time series data on the pollutants, including sulphur dioxide (SO_2), nitrogen dioxide (NO_2), ozone (O_3), and respirable suspended particles (RSP), in a number of air quality monitoring stations across the PRDR on a quarterly basis. During the past decade, some of the monitoring stations in the region were under maintenance for an extended period, some were closed and replaced by new ones, we thus select seven locations that have full data to study the interaction of RSP between the stations. Note that RSP is equivalent to particulate matter with particle size less than 10 microns (PM_{10}).

The data from January 2006 to December 2015, with a length of 120 months, are analyzed. Box-Cox transformations on each RSP series are first performed to stabilize the variance. Some of the transformed series possess decreasing trend, and all series possess seasonal pattern. Therefore, we detrend the transformed series that have a decreasing trend, and then deseasonalize all the RSP series after treatments using har-

[1] `http://www.epd.gov.hk/epd/english/resources_pub/publications/m_report.html`

monic regression described in McLeod & Gweon (2012). The time domain and frequency domain methods are applied to determine a partial correlation graph of the seven RSP series, and hence determine a sparse VAR model to further investigate the inter-relationship of the RSP between monitoring stations. We also implement the 2-Stage method (Davis et al., 2016) to the series as a comparison.

Figure 17: Partial cross-correlations for the PRDR air pollution data (the blue dotted line represents an approximate 5% error bound of $\pm 2/\sqrt{T}$).

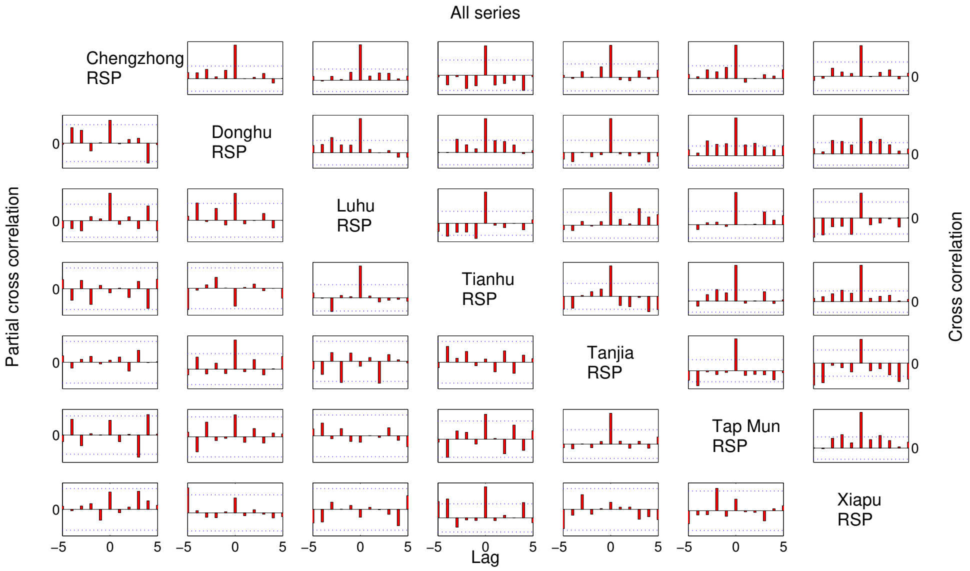
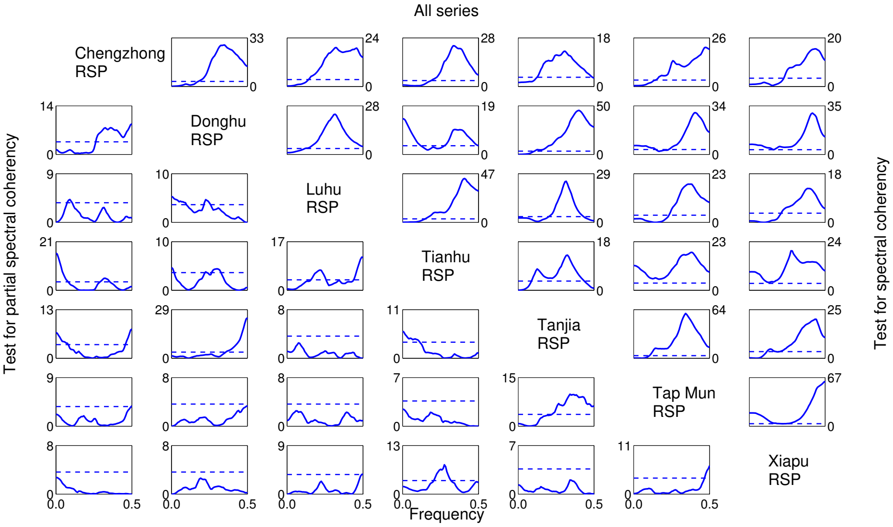


Figure 17 plots the partial cross-correlations for the air pollution data. The upper (lower) triangular part of the diagram shows the cross-correlations (partial cross-correlations) described in Section 2.2. The blue dotted line in each subplot is the approximate 5% error bound of $\pm 2/\sqrt{T}$. Based on the significance of the partial cross-correlations, the partial correlation graph is identified and is shown in Figure 19 a). The nodes in the partial correlation graph are connected if the corresponding partial cross-correlation is significant. A blue dotted (bold red) line in the graph represents the corresponding partial cross-correlation is significant at non-zero (zero) lags.

Figure 18 depicts the test statistics of spectral and partial spectral coherences. The upper (lower) triangular part of the plot shows the test statistics of spectral (partial spectral) coherences described in Section 2.2. The blue dotted line in each subplot is the corresponding critical value of the F distribution at 5% significance level. We then determine a partial correlation graph based on the significance of the partial spectral

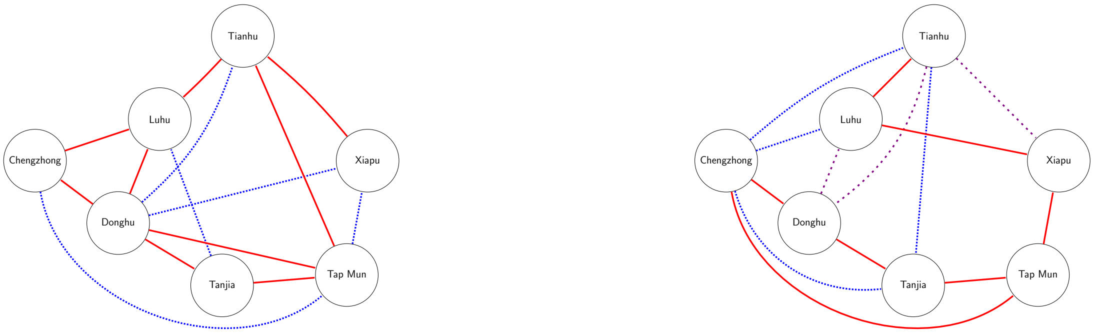
Figure 18: Test statistics of spectral coherences (above diagonal, the blue dotted line represents a 95% quantile of the $F(2,20)$ distribution) and partial spectral coherences (below diagonal, the blue dotted line represents a 95% quantile of the $F(2,12)$ distribution) for the PRDR air pollution data.



coherences and the graph is displayed in Figure 19 b). The nodes in the identified partial correlation graph are linked when the corresponding partial spectral coherence is significant. A blue dotted (purple dashed, bold red) line in the graph represents the corresponding partial coherency is significant at low (mid, high) frequencies.

As shown in Figure 19, the time domain and frequency domain methods identify similar, though not identical partial correlation graphs. The two methods agree on 10 out of 14 edges. The nodes between the cases: Luhu / Tianhu, Chengzhong / Donghu, Donghu / Tanjia and Tanjia / Tap Mun are, in particular, connected by red edges, indicated the corresponding partial cross-correlation is significant at zero lags or the corresponding partial spectral coherence is significant at high frequencies. These observations seems to echo with the flight distances between the monitoring stations, for example the flight distances between the stations of the cases Luhu / Tianhu, Donghu / Tianhu and Tanjia / Tap Mun are 67 km, 61 km and 73 km, respectively. Some of the partial cross-correlations and the partial spectral coherences are marginally significant at few lags or few frequencies, such as the case Donghu / Tianhu. Both the time and frequency domain methods in the estimation of a sparse VAR model on the RSP series identifies and restricts the corresponding autoregressive coefficients and inverse covariances to be zero.

Figure 19: Partial correlation graph for the PRDR air pollution data. The figure displays the approximate geographical location and is not drawn to scale.



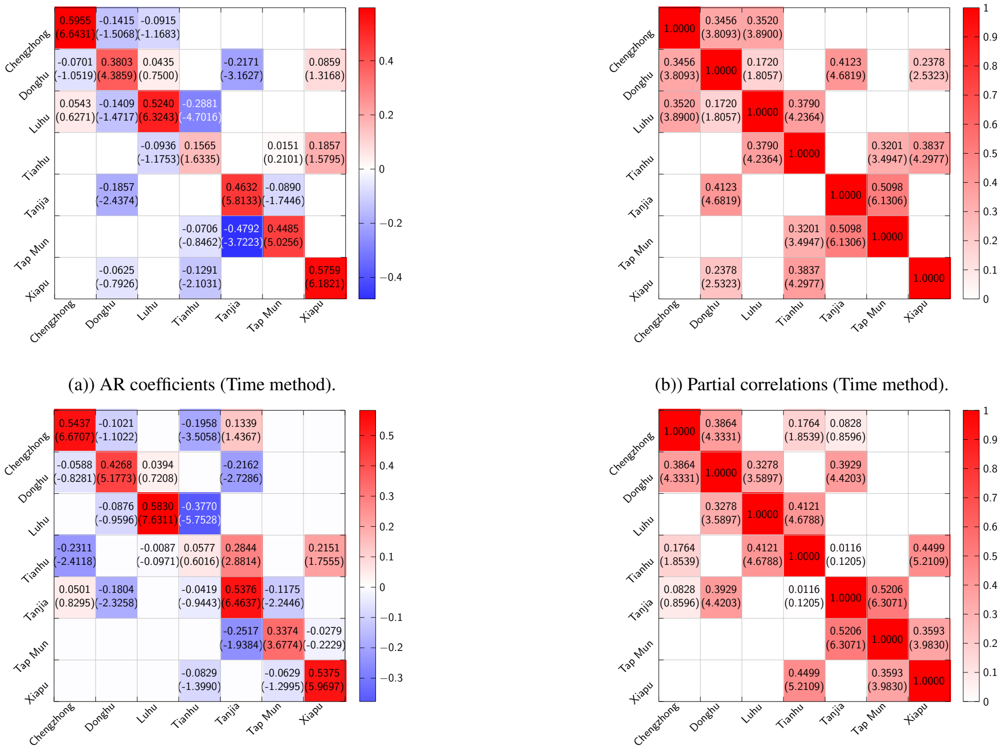
(a)) Time domain method (the blue dotted (bold red) line indicates that the corresponding partial cross-correlation is significant at non-zero (zero) lags).

(b)) Frequency domain method (the blue dotted (purple dashed, bold red) line indicates that the corresponding partial coherence is significant at low (mid, high) frequencies).

Both the time and frequency domain methods determine a model of order 1, and the corresponding autoregressive coefficient and partial correlation of innovations estimates are visualized by the heatmaps in Figures 20 a) and 20 b) for the time domain method and Figures 20 c) and 20 d) for the frequency domain method. Figure 21 depicts the estimated VAR models, using the frequency domain and time domain methods, by the mixed graph presented in Section 3.3. Each node represents the RSP series at a specific location among the seven monitoring stations. The directed and undirected edges are drawn based on the definitions (a) and (b) in Section 3.3, respectively. The estimated autoregressive coefficients and the partial correlation coefficients of the noise terms are printed next to the edges. The value in parenthesis is the t-value of the estimate of autoregressive coefficient or partial correlation. Some of the partial correlation coefficients are relatively small in magnitude, for example the partial correlation coefficient of the case Tianhu / Tanjia using the frequency domain method.

Following Tunnicliffe Wilson et al. (2015), we fit a structural VAR (SVAR) model for the PRDR series. Figure 22 plots the directed acyclic graph (DAG) representing a SVAR model for the PRDR series, where $X_{1,t}$ represents the RSP series at time t at Chengzhong and others correspond to the labels at the bottom of the graph. We note that the estimated VAR model by the time domain method possesses similar dynamic inter-relation structure comparing to the SVAR model, especially for the contemporaneous dependence part. For the nodes that are connected by undirected edges in the mixed graph identified by the time domain method (Figure 21 a)), there are directed edges connecting the corresponding nodes of the current variables

Figure 20: The autoregressive coefficient estimates and the estimated partial correlations of innovations for the PRDR air pollution data (t-values are in parentheses).



	Chengzhong	Donghu	Luhu	Tianhu	Tanjia	Tap Mun	Xiapu
Chengzhong	0.5955 (6.6431)	-0.1415 (-1.5068)	-0.0915 (-1.1683)				
Donghu	-0.0701 (-1.0519)	0.3803 (4.3859)	0.0435 (0.7500)		-0.2171 (-3.1627)		0.0859 (1.3168)
Luhu	0.0543 (0.6271)	-0.1409 (-1.4717)	0.5240 (6.3243)	-0.2881 (-4.7016)			
Tianhu			-0.0936 (-1.1753)	0.1565 (1.6335)		0.0151 (0.2101)	0.1857 (1.5795)
Tanjia		-0.1857 (-2.4374)			0.4632 (5.8133)	-0.0890 (-1.7446)	
Tap Mun				-0.0706 (-0.8462)	-0.4792 (-3.7223)	0.4485 (5.0256)	
Xiapu		-0.0625 (-0.7926)		-0.1291 (-2.1031)			0.5759 (6.1821)

(a)) AR coefficients (Time method).

	Chengzhong	Donghu	Luhu	Tianhu	Tanjia	Tap Mun	Xiapu
Chengzhong	1.0000	0.3456 (3.8093)	0.3520 (3.8900)				
Donghu	0.3456 (3.8093)	1.0000	0.1720 (1.8057)		0.4123 (4.6819)		0.2378 (2.5323)
Luhu	0.3520 (3.8900)	0.1720 (1.8057)	1.0000	0.3790 (4.2364)			
Tianhu			0.3790 (4.2364)	1.0000		0.3201 (3.4947)	0.3837 (4.2977)
Tanjia		0.4123 (4.6819)			1.0000	0.5098 (6.1306)	
Tap Mun				0.3201 (3.4947)	0.5098 (6.1306)	1.0000	
Xiapu		0.2378 (2.5323)		0.3837 (4.2977)			1.0000

(b)) Partial correlations (Time method).

	Chengzhong	Donghu	Luhu	Tianhu	Tanjia	Tap Mun	Xiapu
Chengzhong	0.5437 (6.6707)	-0.1021 (-1.1022)		-0.1958 (-3.5058)	0.1339 (1.4367)		
Donghu	-0.0588 (-0.8281)	0.4268 (5.1773)	0.0394 (0.7208)		-0.2162 (-2.7286)		
Luhu		-0.0876 (-0.9596)	0.5830 (7.6311)	-0.3770 (-5.7528)			
Tianhu	-0.2311 (-2.4118)		-0.0087 (-0.0971)	0.0577 (0.6016)	0.2844 (2.8814)		0.2151 (1.7555)
Tanjia	0.0501 (0.8295)	-0.1804 (-2.3258)		-0.0419 (-0.9443)	0.5376 (6.4637)	-0.1175 (-2.2446)	
Tap Mun					-0.2517 (-1.9384)	0.3374 (3.6774)	-0.0279 (-0.2229)
Xiapu				-0.0829 (-1.3990)		-0.0629 (-1.2995)	0.5375 (5.9697)

(c)) AR coefficients (Frequency method).

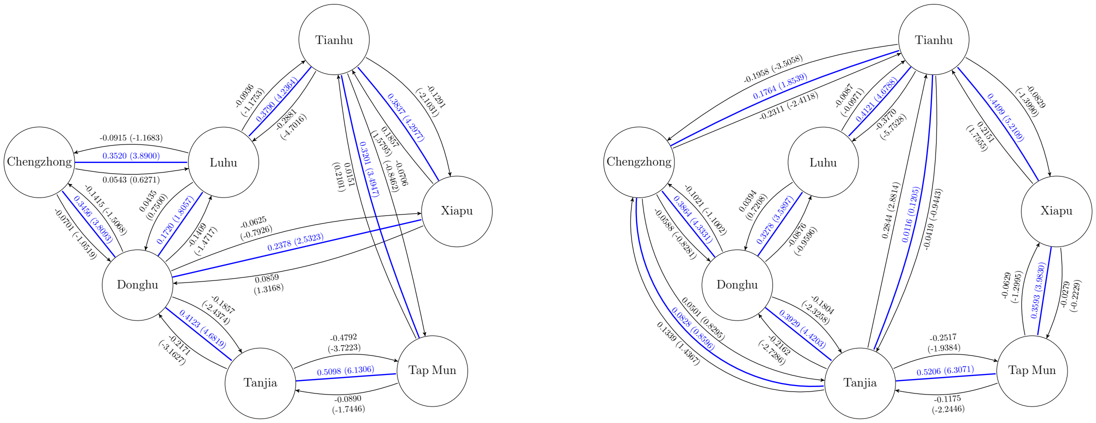
	Chengzhong	Donghu	Luhu	Tianhu	Tanjia	Tap Mun	Xiapu
Chengzhong	1.0000	0.3864 (4.3331)		0.1764 (1.8539)	0.0828 (0.8596)		
Donghu	0.3864 (4.3331)	1.0000	0.3278 (3.5897)		0.3929 (4.4203)		
Luhu		0.3278 (3.5897)	1.0000	0.4121 (4.6788)			
Tianhu	0.1764 (1.8539)		0.4121 (4.6788)	1.0000	0.0116 (0.1205)		0.4499 (5.2109)
Tanjia	0.0828 (0.8596)	0.3929 (4.4203)		0.0116 (0.1205)	1.0000	0.5206 (6.3071)	
Tap Mun					0.5206 (6.3071)	1.0000	0.3593 (3.9830)
Xiapu				0.4499 (5.2109)		0.3593 (3.9830)	1.0000

(d)) Partial correlations (Frequency method).

in the DAG (Figure 22), such as the cases Chengzhong ($X_{1,t}$) / Donghu ($X_{2,t}$), Luhu ($X_{3,t}$) / Donghu ($X_{2,t}$) and Tap Mun ($X_{6,t}$) / Tanjia ($X_{5,t}$).

Figure 23 plots the autoregressive coefficient and the partial correlation of noise estimates obtained by the 2-Stage method (Davis et al., 2016). Comparing to the autoregressive coefficient estimates obtained by the introduced ACS method, the 2-Stage approach achieves higher sparsity in the estimation of the autoregressive coefficients, especially for the autoregressive coefficient estimates determined by the ACS method which are insignificant. For the partial correlations, the 2-Stage method does not impose sparsity constraints on the inverse covariance matrix. Therefore, the partial correlations estimated by the 2-Stage approach are all non-zero. We note that there are some partial correlations, obtained by the 2-Stage method, are insignificant using a 5% level of significant; or a critical value of $t_{0.025;\nu=107} = -1.9824$, namely Tanjia / Luhu,

Figure 21: A mixed graph visualizing the estimated VAR model for the PRDR air pollution data (the bold blue line represents the undirected edge determined by the inverse of noise covariance matrix, the black arrow is the directed edge characterized by the AR coefficient, and t-values in parentheses). The figure displays the approximate geographical location and is not drawn to scale.



(a)) Time domain method.

(b)) Frequency domain method.

Tanjia / Tianhu, Tap Mun / Luhu, Xiapu / Luhu, and Xiapu / Tanjia. The partial correlations of innovations for these cases obtained by the time and frequency domain methods are zero or insignificant.

6. Conclusion

The estimation problem of high-dimensional graphical time series models is considered in this paper. We study an alternating maximization method to estimate a sparse vector autoregressive model with sparsity constraints on both the autoregressive coefficients and the inverse of noise covariance matrix. To our knowledge, this is a new research direction as existing research focuses on constraint or regularization of the parameters in the autoregressive matrices. The algorithm introduced considers the conditional maximum likelihood estimation with the sparsity constraints as a "biconcave" problem, such that the problem becomes concave when either the autoregressive coefficients or the inverse of noise covariance matrix is fixed. This "biconcave" approach is also new in the study of graphical time series models. The sparsity constraints imposed can help reduce the number of parameters when modelling a high-dimensional time series. We also investigate the performance of the proposed method with simulation studies, which is satisfactory under the three metrics discussed. Further, in comparing with two popular Newton-type methods, the proposed has much less nonconvergent cases. We apply the method to an air pollution time series dataset in the Pearl River

Figure 22: The DAG representing a SVAR for the PRDR series.

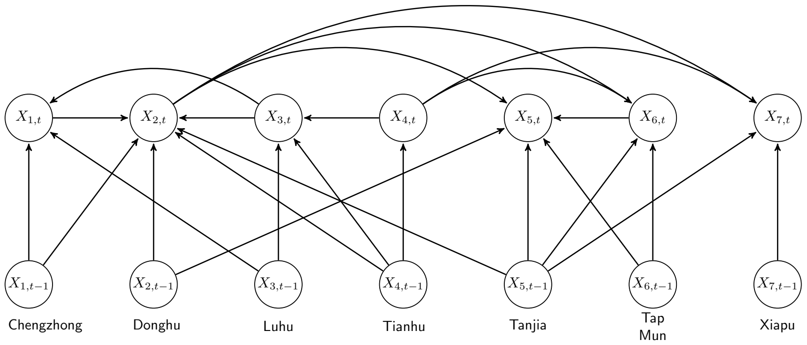
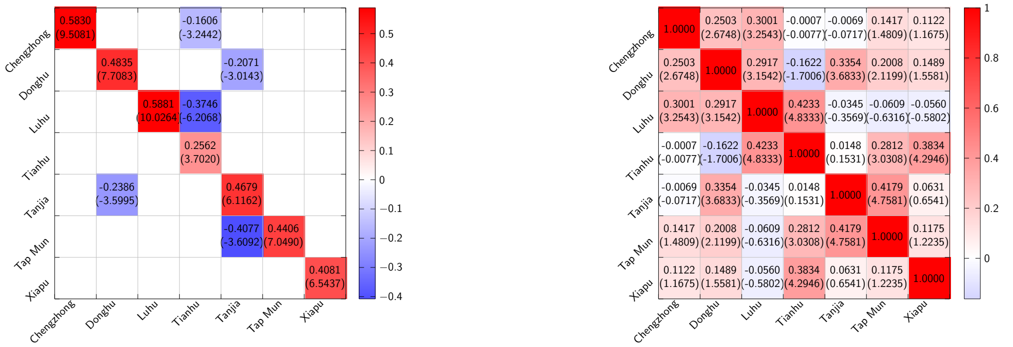


Figure 23: The autoregressive coefficient estimates and the estimated partial correlations of innovations using the 2-Stage method for the PRDR air pollution data (t-values are in parentheses).



	Chengzhong	Donghu	Luhu	Tianhu	Tanjia	Tap Mun	Xiapu
Chengzhong	0.5830 (9.5081)			-0.1606 (-3.2442)			
Donghu		0.4835 (7.7083)			-0.2071 (-3.0143)		
Luhu			0.5881 (10.0264)	-0.3746 (-6.2068)			
Tianhu				0.2562 (3.7020)			
Tanjia		-0.2386 (-3.5995)			0.4679 (6.1162)		
Tap Mun					-0.4077 (-3.6092)	0.4406 (7.0490)	
Xiapu							0.4081 (6.5437)

(a)) AR coefficients.

	Chengzhong	Donghu	Luhu	Tianhu	Tanjia	Tap Mun	Xiapu
Chengzhong	1.0000	0.2503 (2.6748)	0.3001 (3.2543)	-0.0007 (-0.0077)	-0.0069 (-0.0717)	0.1417 (1.4809)	0.1122 (1.1675)
Donghu	0.2503 (2.6748)	1.0000	0.2917 (3.1542)	-0.1622 (-1.7006)	0.3354 (3.6833)	0.2008 (2.1199)	0.1489 (1.5581)
Luhu	0.3001 (3.2543)	0.2917 (3.1542)	1.0000	0.4233 (4.8333)	-0.0345 (-0.3569)	-0.0609 (-0.6316)	-0.0560 (-0.5802)
Tianhu	-0.0007 (-0.0077)	-0.1622 (-1.7006)	0.4233 (4.8333)	1.0000	0.0148 (0.1531)	0.2812 (3.0308)	0.3834 (4.2946)
Tanjia	-0.0069 (-0.0717)	0.3354 (3.6833)	-0.0345 (-0.3569)	0.0148 (0.1531)	1.0000	0.4179 (4.7581)	0.0631 (0.6541)
Tap Mun	0.1417 (1.4809)	0.2008 (2.1199)	-0.0609 (-0.6316)	0.2812 (3.0308)	0.4179 (4.7581)	1.0000	0.1175 (1.2235)
Xiapu	0.1122 (1.1675)	0.1489 (1.5581)	-0.0560 (-0.5802)	0.3834 (4.2946)	0.0631 (0.6541)	0.1175 (1.2235)	1.0000

(b)) Partial correlations of innovations.

Delta region to explore the inter-relationship of respirable suspended particles between the seven selected air quality monitoring stations across the area and a flour price indices dataset to investigate the dynamic interdependence structure of the flour price series. The results are consistent with another graphical time series approach based on structural vector autoregressive models.

Appendix

Lemma 1. *For any positive semidefinite matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *and* $0 < \alpha < 1$, $\det(\alpha\mathbf{A} + (1-\alpha)\mathbf{B}) \geq \det(\mathbf{A})^{\alpha}\det(\mathbf{B})^{1-\alpha}$ *with equality if and only if* $\mathbf{A} = \mathbf{B}$ *or* $\det(\alpha\mathbf{A} + (1-\alpha)\mathbf{B}) = 0$.

PROOF. See Magnus & Neudecker (1999, Chapter 11).

PROOF OF THEOREM 1. To prove the problem in (14) is biconcave, we first show the feasible set D is biconvex followed by showing the objective function of the problem is biconcave. Let $\Omega = \Sigma_u^{-1}$, and S be the set of lower triangular positions of Ω that are not constrained to be zero having q elements (i.e. $S = \{(i_1, j_1), \cdots, (i_k, j_k), \cdots, (i_q, j_q)\}$ with $1 \leq i_k \leq j_k \leq K$ for $k = 1, \cdots, q$). Define two $(K \times q)$ matrices

$$E_1 = \begin{pmatrix} e_{i_1} & e_{i_2} & \cdots & e_{i_q} \end{pmatrix} \quad \text{and} \quad E_2 = \begin{pmatrix} e_{j_1} & e_{j_2} & \cdots & e_{j_q} \end{pmatrix},$$

where e_{i_k} is a vector of zeros except the i_k-th entry being one, for $k = 1, \cdots, q$. We express the optimization problem (14) as an unconstrained problem, following Dahl et al. (2005), with objective function $f(\beta, \omega)$ given by

$$\begin{aligned} f(\beta, \omega) &= \frac{T}{2} \log\det\left[\Omega(\omega)\right] - \frac{1}{2} \mathrm{tr}\big((\mathbf{Y} - \mathbf{BZ})' \Omega(\omega) (\mathbf{Y} - \mathbf{BZ})\big) \\ &= \frac{T}{2} \log\det\left[\Omega(\omega)\right] - \frac{1}{2} \left[\mathbf{y} - \left(\mathbf{Z}' \otimes \mathbf{I}_K\right) \beta\right]' \left[\mathbf{I}_T \otimes \Omega(\omega)\right] \left[\mathbf{y} - \left(\mathbf{Z}' \otimes \mathbf{I}_K\right) \beta\right]. \end{aligned}$$

Here, the constant term is omitted and the inverse of innovation covariance matrix Ω is parameterized as

$$\Omega(\omega) = E_1 \mathbf{diag}(\omega) E_2{}' + E_2 \mathbf{diag}(\omega) E_1{}',$$

where $\omega \in \mathbb{R}^q$ contains the non-zero element in the strict lower triangular part of Ω, and the non-zero elements on the diagonal are divided by 2, i.e.

$$\omega_k = \begin{cases} \Omega_{i_k j_k}, & i_k \neq j_k \\ \frac{1}{2} \Omega_{i_k j_k}, & i_k = j_k \end{cases} \quad \text{for } k = 1, \cdots, q.$$

We now prove that the feasible set D is biconvex based on the definition in Gorski et al. (2007). Let $B = \{\beta \in \mathbb{R}^{K(Kp+1)} \mid \mathbf{C}\beta = \mathbf{0}\} \subseteq \mathbb{R}^{K(Kp+1)}$ and $W = \{\omega \in \mathbb{R}^q \mid \Omega(\omega) \succ 0\} \subseteq \mathbb{R}^q$ which are non-empty and convex. Let $D = \{(\beta, \omega) \in \mathbb{R}^{K(Kp+1)} \times \mathbb{R}^q \mid \mathbf{C}\beta = \mathbf{0},\ \Omega(\omega) \succ 0\} \subseteq B \times W$. Define $D_\omega = \{\beta \in B \mid (\beta, \omega) \in D\}$ and $D_\beta = \{\omega \in W \mid (\beta, \omega) \in D\}$.

For any $\beta_1, \beta_2 \in D_\omega$ and any $\lambda \in [0, 1]$, $\lambda \mathbf{C}\beta_1 + (1 - \lambda)\mathbf{C}\beta_2 \in D_\omega$ for every $\omega \in W$, since

$$\mathbf{C}(\lambda \beta_1 + (1 - \lambda)\beta_2) = \lambda \mathbf{C}\beta_1 + (1 - \lambda)\mathbf{C}\beta_2 = 0.$$

Therefore, D_ω is convex for every $\omega \in W$.

For any $\omega_1, \omega_2 \in D_\beta$ and any $\lambda \in [0, 1]$, $\lambda \omega_1 + (1 - \lambda)\omega_2 \in D_\beta$ for every $\beta \in B$, since

$$\Omega(\lambda \omega_1 + (1 - \lambda)\omega_2) = \lambda \Omega(\omega_1) + (1 - \lambda)\Omega(\omega_2) \succ 0.$$

Therefore, D_β is convex for every $\beta \in B$. Hence, the set $D \subseteq B \times W$ is biconvex.

We next show that the objective function is biconcave. Define $f_\omega(\cdot) = f(\cdot, \omega)\colon D_\omega \to \mathbb{R}$ and $f_\beta(\cdot) = f(\beta, \cdot)\colon D_\beta \to \mathbb{R}$.

For every fixed $\omega \in W$,

$$\begin{aligned} f_\omega(\beta) &= \frac{T}{2}\log\det\left[\Omega(\omega)\right] - \frac{1}{2}\left[\mathbf{y} - \left(\mathbf{Z}' \otimes \mathbf{I}_K\right)\beta\right]'\left[\mathbf{I}_T \otimes \Omega(\omega)\right]\left[\mathbf{y} - \left(\mathbf{Z}' \otimes \mathbf{I}_K\right)\beta\right] \\ &= -\frac{1}{2}\beta'\left[\mathbf{Z}\mathbf{Z}' \otimes \Omega(\omega)\right]\beta + \beta'\left[\mathbf{Z} \otimes \Omega(\omega)\right]\mathbf{y} - \frac{1}{2}\mathbf{y}'\left[\mathbf{I}_T \otimes \Omega(\omega)\right]\mathbf{y} + \frac{T}{2}\log\det\left[\Omega(\omega)\right] \end{aligned}$$

is a quadratic function of β and is strictly concave since $\mathbf{Z}\mathbf{Z}' \otimes \Omega(\omega) \succ 0$. Therefore, $f_\omega(\cdot)$ is a concave function on D_ω for every fixed $\omega \in W$.

Denote $\mathbf{S} = (\mathbf{Y} - \mathbf{B}\mathbf{Z})(\mathbf{Y} - \mathbf{B}\mathbf{Z})'$. For all $\omega_1, \omega_2 \in D_\beta$ with $\omega_1 \neq \omega_2$, $\lambda \in (0,1)$ and for every fixed $\beta \in B$,

$$\begin{aligned} f_\beta(\lambda\omega_1 + (1-\lambda)\omega_2) &= \frac{T}{2}\log\det\left[\Omega(\lambda\omega_1 + (1-\lambda)\omega_2)\right] - \frac{1}{2}\mathrm{tr}(\mathbf{S}\Omega(\lambda\omega_1 + (1-\lambda)\omega_2)) \\ &= \frac{T}{2}\log\det\left[\lambda\Omega(\omega_1) + (1-\lambda)\Omega(\omega_2)\right] - \frac{1}{2}\left\{\lambda\,\mathrm{tr}(\mathbf{S}\Omega(\omega_1)) + (1-\lambda)\mathrm{tr}(\mathbf{S}\Omega(\omega_2))\right\} \\ &> \frac{T}{2}\log\left\{\left[\det\Omega(\omega_1)\right]^{\lambda}\left[\det\Omega(\omega_2)\right]^{1-\lambda}\right\} - \frac{1}{2}\left\{\lambda\,\mathrm{tr}(\mathbf{S}\Omega(\omega_1)) + (1-\lambda)\mathrm{tr}(\mathbf{S}\Omega(\omega_2))\right\} \\ &= \lambda f_\beta(\omega_1) + (1-\lambda)f_\beta(\omega_2). \end{aligned}$$

Here, the inequality 2 lines above follows from Lemma 1. Therefore, $f_\beta(\cdot)$ is a strictly concave function on D_β for every fixed $\beta \in B$. Hence, the optimization problem in (14) is biconcave.

References

References

Banerjee, O., El Ghaoui, L., & d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, *9*, 485–516.

Brillinger, D. R. (1981). *Time series: data analysis and theory*. San Francisco: Holden-Day.

Brillinger, D. R. (1996). Remarks concerning graphical models for time series and point processes. *Revista de Econometria*, *16*, 1–23.

Dahl, J., Roychowdhury, V., & Vandenberghe, L. (2005). *Maximum likelihood estimation of gaussian graphical models: Numerical implementation and topology selection*. Technical Report Electrical Engineering Department, UCLA.

Dahl, J., Vandenberghe, L., & Roychowdhury, V. (2008). Covariance selection for nonchordal graphs via chordal embedding. *Optimization Methods and Software*, *23*, 501–520. doi:10.1080/10556780802102693.

Dahlhaus, R. (2000). Graphical interaction models for multivariate time series. *Metrika*, *51*, 157–172. doi:10.1007/s001840000055.

Darroch, J. N., Lauritzen, S. L., & Speed, T. P. (1980). Markov fields and log-linear interaction models for contingency tables. *The Annals of Statistics*, *8*, 522–539.

Davis, R. A., Zang, P., & Zheng, T. (2016). Sparse vector autoregressive modeling. *Journal of Computational and Graphical Statistics*, *25*, 1077–1096. doi:10.1080/10618600.2015.1092978.

Dempster, A. P. (1972). Covariance selection. *Biometrics*, *28*, 157–175. doi:10.2307/2528966.

Edwards, D. (1995). *Introduction to graphical modelling*. New York, N.Y: New York : Springer-Verlag.

Eichler, M. (2012). Graphical modelling of multivariate time series. *Probability Theory and Related Fields*, *153*, 233–268. doi:10.1007/s00440-011-0345-8.

Friedman, J., Hastie, T., & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, *9*, 432–441. doi:10.1093/biostatistics/kxm045.

Gorski, J., Pfeuffer, F., & Klamroth, K. (2007). Biconvex sets and optimization with biconvex functions: a survey and extensions. *Mathematical Methods of Operations Research*, *66*, 373–407. doi:10.1007/s00186-007-0161-1.

Hannan, E. J., & Quinn, B. G. (1979). The determination of the order of an autoregression. *Journal of the Royal Statistical Society. Series B (Methodological)*, *41*, 190–195.

Hastie, T., Tibshirani, R., & Wainwright, M. (2015). *Statistical learning with sparsity: the Lasso and generalizations*. Boca Raton, FL : CRC Press.

Haufe, S., Müller, K.-R., Nolte, G., & Kräemer, N. (2009). Sparse causal discovery in multivariate time series. In I. Guyon, D. Janzing, & B. Schölkopf (Eds.), *Proceedings of the NIPS 2008 workshop on Causality* (pp. 97–106).

Hsu, N.-J., Hung, H.-L., & Chang, Y.-M. (2008). Subset selection for vector autoregressive processes using lasso. *Computational Statistics and Data Analysis*, *52*, 3645–3657. doi:10.1016/j.csda.2007.12.004.

Hu, F., Lu, Z., Wong, H., & Yuen, T. P. (2016). Analysis of air quality time series of hong kong with graphical modeling. *Environmetrics*, *27*, 169–181. doi:10.1002/env.2386.

Jung, A., Hannak, G., & Goertz, N. (2015). Graphical lasso based model selection for time series. *Signal Processing Letters, IEEE*, *22*, 1781–1785. doi:10.1109/LSP.2015.2425434.

Koopmans, L. H. (1974). *The spectral analysis of time series*. New York : Academic Press.

Lauritzen, S. L. (1996). *Graphical models*. Oxford England : Clarendon Press.

Lütkepohl, H. (2005). *New introduction to multiple time series analysis*. New York : Springer-Verlag.

Magnus, J., & Neudecker, H. (1999). *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Chichester ; New York : John Wiley.

McLeod, A. I., & Gweon, H. (2012). Optimal deseasonalization for monthly and daily geophysical time series. *Journal of Environmental Statistics*, *4*.

Nicholson, W. B., Matteson, D. S., & Bien, J. (2017). VARX-L: Structured regularization for large vector autoregressions with exogenous variables. *International Journal of Forecasting*, *33*, 627–651. doi:10.1016/j.ijforecast.2017.01.003.

Oxley, L., Reale, M., & Tunnicliffe Wilson, G. (2004). Finding directed acyclic graphs for vector autoregressions. *Proceedings in computational statistics 2004*, (pp. 1621–1628).

Ren, Y., Xiao, Z., & Zhang, X. (2013). Two-step adaptive model selection for vector autoregressive processes. *Journal of Multivariate Analysis*, *116*, 349–364. doi:10.1016/j.jmva.2013.01.004.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*, 461–464.

Song, S., & Bickel, P. J. (2011). Large vector auto regressions. *Arxiv preprint arXiv:1106.3915v1*, .

Songsiri, J. (2013). Sparse autoregressive model estimation for learning granger causality in time series. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 3198–3202). doi:10.1109/ICASSP.2013.6638248.

Songsiri, J., Dahl, J., & Vandenberghe, L. (2009). Graphical models of autoregressive processes. In Y. E. Palomar, & D. (Eds.), *Convex Optimization in Signal Processing and Communications* (pp. 89–116). Cambridge: Cambridge University Press.

Sturm, J. F. (1999). Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, *11*, 625–653. doi:10.1080/10556789908805766.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, *58*, 267–288.

Tunnicliffe Wilson, G., Reale, M., & Haywood, J. (2015). *Models for dependent time series*. Boca Raton, FL : CRC Press LLC.

Tütüncü, R. H., Toh, K. C., & Todd, M. J. (2003). Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, *95*, 189–217. doi:10.1007/s10107-002-0347-5.

Vandenberghe, L., Boyd, S., & Wu, S.-P. (1998). Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis & Applications*, *19*, 499. doi:10.1137/S0895479896303430.

Yuan, M., & Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *68*, 49–67. doi:10.1111/j.1467-9868.2005.00532.x.