Electronic Journal of Statistics
Vol. 6 (2012) 168–198
ISSN: 1935-7524
DOI: 10.1214/12-EJS669

Efficient Gaussian graphical model determination under *G*-Wishart prior distributions

Hao Wang

Department of Statistics, University of South Carolina,
Columbia, South Carolina 29208, U.S.A.
e-mail: haowang@sc.edu

and

Sophia Zhengzi Li

Department of Economics, Duke University,
Durham, North Carolina 27705, U.S.A.
e-mail: zhengzi.li@duke.edu

Abstract: This paper proposes a new algorithm for Bayesian model determination in Gaussian graphical models under *G*-Wishart prior distributions. We first review recent development in sampling from *G*-Wishart distributions for given graphs, with a particular interest in the efficiency of the block Gibbs samplers and other competing methods. We generalize the maximum clique block Gibbs samplers to a class of flexible block Gibbs samplers and prove its convergence. This class of block Gibbs samplers substantially outperforms its competitors along a variety of dimensions. We next develop the theory and computational details of a novel Markov chain Monte Carlo sampling scheme for Gaussian graphical model determination. Our method relies on the partial analytic structure of *G*-Wishart distributions integrated with the exchange algorithm. Unlike existing methods, the new method requires neither proposal tuning nor evaluation of normalizing constants of *G*-Wishart distributions.

Keywords and phrases: Exchange algorithms, Gaussian graphical models, *G*-Wishart, hyper-inverse Wishart, Gibbs sampler, non-decomposable graphs, partial analytic structure, posterior simulation.

Received August 2011.

Contents

1. Introduction

The purpose of this paper is to introduce a new algorithm for improving the efficiency of existing methods for Bayesian Gaussian graphical model determination under G-Wishart priors. Let $y = (y^{(1)}, y^{(2)}, \ldots, y^{(p)})'$ be a p-dimensional random vector having a multivariate normal distribution $\mathrm{N}(0, \Sigma)$ with mean zero and covariance matrix Σ. Let $\Omega = (\omega_{ij})_{p\times p} = \Sigma^{-1}$ be the inverse of the covariance matrix. Let $G = (V, E)$ be an undirected graph, where V is a non-empty set of vertices and E is a set of undirected edges. We apply G to Ω to represent strict conditional independencies. Specifically, each vertex $i \in V$ corresponds to $y^{(i)}$, and each edge $(i, j) \in E$ corresponds to $\omega_{ij} \neq 0$; $y^{(i)}$ and $y^{(j)}$ are conditionally independent if and only if $\omega_{ij} = 0$, or equivalently, $(i, j) \notin E$. The G-Wishart distribution [23, 1] is the conjugate prior for Ω when Ω is constrained by the graph G. A zero constrained random matrix Ω has the G-Wishart distribution $\mathrm{W}_G(b, D)$ if its density is

$$p(\Omega \mid G) = I_G(b, D)^{-1}|\Omega|^{(b-2)/2} \exp\left\{-\frac{1}{2}\mathrm{tr}(D\Omega)\right\} 1_{\{\Omega\in M^+(G)\}}, \tag{1.1}$$

where $b > 2$ is the degree of freedom parameter, D is a symmetric positive definite matrix, $I_G(b, D)$ is the normalizing constant, namely,

$$I_G(b, D) = \int |\Omega|^{(b-2)/2} \exp\left\{-\frac{1}{2}\mathrm{tr}(D\Omega)\right\} 1_{\{\Omega\in M^+(G)\}} \mathrm{d}\Omega,$$

and $M^+(G)$ is the cone of symmetric positive definite matrices with off-diagonal entries $\omega_{ij} = 0$ whenever $(i, j) \notin E$. For arbitrary graphs, the explicit formula for computing $I_G(b, D)$ is given in equation (3.1). The G-Wishart distribution is used extensively for analyzing covariance structures in models of increasing dimension and complexity in biology [11], finance [4, 29], economics [30], epidemiology [6] and other areas.

Conditional on a specific graph G and an observed dataset $Y = (y_1, \ldots, y_n)$ of sample size n, the posterior distribution of Ω is then

$$p(\Omega \mid Y, G) = I_G(b+n, D+S)^{-1}|\Omega|^{(b+n-2)/2} \exp\left[-\frac{1}{2}\mathrm{tr}\{(D+S)\Omega\}\right] 1_{\{\Omega \in M^+(G)\}}, \tag{1.2}$$

where $S = YY'$. To estimate Ω or any function of it, we need to sample from the G-Wishart distribution for any given graphs. For decomposable graphs, Carvalho, Massam and West [3] proposed a direct and efficient method based on the perfect ordering of the cliques. For arbitrary graphs, Piccioni [19] developed distributional theory for the block Gibbs sampler using Bayesian iterative proportional scaling. Implementation of this theory has been focused on a way that requires maximum clique decomposition and large matrix inversion [13, 15], leading to the conclusion that the Bayesian iterative proportional scaling is not good for large problems because enumerating all cliques is NP-hard and inverting large matrix is computationally expensive. Motivated by these limitations, several other methods [28, 15, 6] took a different approach that used theoretical innovations for non-decomposable graphical models developed in Atay-Kayis and Massam [1]. However, one key computational bottleneck of these methods is the matrix completion step for every update of Ω. The matrix completion is conducted iteratively with time complexity $O(p^2)$ for completing one non-free element. With increasingly large problems, each update becomes increasingly burdensome. In Section 2.4, we revisit the block Gibbs sampler from a different yet more straightforward perspective that relies on the theory of a non-ordinary Gibbs sampler. We show that the class of block Gibbs samplers is indeed very broad. It not only includes the previously proposed approach based on maximum cliques as a special case, but also motivates a simple implementation that uses individual edges as components in the Gibbs sampler to avoid maximum clique enumeration. Through simulation experiments, we illustrate the flexibility and efficiency of the class of block Gibbs samplers as compared with existing methods.

When G is unknown, most of the methods for determining graphical structures operate directly on the graphical model space by treating Ω as a nuisance parameter and computing the marginal likelihood function over graphs (e.g. [11, 24, 13]). Specifically, the marginal likelihood function for any graph G is computed by integrating out Ω with respect to its prior (1.1),

$$p(Y \mid G) = \int p(Y \mid \Omega, G)p(\Omega \mid G) = (2\pi)^{-np/2}\frac{I_G(b+n, D+S)}{I_G(b, D)}. \tag{1.3}$$

The ability to focus on the graph G alone allows for the development of various search algorithms to visit high probability region of graph space. Markov chain Monte Carlo (MCMC) methods are often outperformed by other stochastic search approaches [11, 24, 13]. The primary challenge in these approaches based on the marginal likelihood function is that computing $I_G(b, D)$ and $I_G(b+n, D+S)$ for non-decomposable graphs requires approximation. Two popular

approximations are the Monte Carlo integration of Atay-Kayis and Massam [1] and the Laplace approximation of Lenkoski and Dobra [13]. Neither approximation has theoretical results for the variance estimation, though they were empirically proven to be successful in guiding the graphical model search when carefully implemented (e.g. [11], [13]).

Alternatively, there are a number of carefully designed MCMC methods for sampling over the joint space of graphs and precision matrices [9, 6]. A salient feature of these joint space methods is that they do not need posterior normalizing constants whose approximation tends to be more numerically unstable than prior normalizing constants. However, all existing joint space samplers require the tuning of proposals for both across- and within-graph moves. Moreover, they do not remove the need for evaluating prior normalizing constants.

We argue that there are important situations where avoiding the approximation of $p(Y \mid G)$ is preferred. First, when the size of the prime component is not restricted to be small or when the graphical decomposition is not conducted, the accuracy of the approximation methods can be hard to access even empirically; see one example in Section 6. Second, graphical models are often embedded within a larger and more complicated class of models such as those developments in the seemingly unrelated regression (SUR) models [27], the conditionally autoregressive (CAR) models [6], the mixture models [22] and the copula models [5]. In these models, $p(Y \mid G)$ is typically unavailable in closed form even given the normalizing constant I_G. MCMC is routinely used for posterior computation, in which, the step of normalizing constant approximation often takes a substantial part of the run-time. Hence, a sampling method without evaluating I_G can facilitate efficient posterior computation. In Section 5, we introduce one such method. Two key features make our algorithm efficient. The first feature is that we use the partial analytic structure [10] of G-Wishart distributions to automatically choose proposals for the reversible jump algorithm, yielding essentially Gibbs steps for both across- and within-graph moves. The other feature is that we use an exchange algorithm [17, 14] to remove the need for evaluating prior normalizing constants in a carefully designed MCMC sampling scheme. Through simulation experiments, we illustrate the accuracy of the proposed algorithm, as well as highlighting its scalability to large graphs. Through a real-world example, we further illustrate that the algorithm can be embedded in a larger MCMC sampler for fitting broader classes of multivariate models.

2. Sampling from the G-Wishart distribution on given graphs

2.1. Accept-reject algorithm

Wang and Carvalho [28] proposed an accept-reject algorithm for sampling from the G-Wishart distribution (1.1). Write $D^{-1} = T'T$ and $\Omega = \Phi'\Phi$ as Cholesky decompositions and define $\Psi = \Phi T^{-1}$. Following the nomenclature of Atay-Kayis and Massam [1], the free elements of Φ are those ϕ_{ij} such that $(i, j) \in E$ or $i = j$. We let $\Psi^{\mathcal{V}} = [\psi_{11}^2, \ldots, \psi_{pp}^2, \{\psi_{ij}\}_{(i,j)\in E, i<j}]$. From Theorem 1 and

equation (38) of Atay-Kayis and Massam [1], these free elements have density defined by

$$p(\Psi^{\mathcal{V}}) \quad \propto \quad \prod_{i=1}^{p}(\psi_{ii}^2)^{(b+\nu_i)/2-1}\exp\Big\{-\frac{1}{2}\sum_{1\le i\le j\le p}\psi_{ij}^2\Big\}, \tag{2.1}$$

where $\nu_i = |\{j : j > i, (i,j) \in E\}|$, and the non-free elements $\{\psi_{rs} : (r,s) \notin E, r < s\}$ are uniquely defined functions of the free elements, namely:

$$\psi_{rs} = \sum_{j=r}^{s-1}(-\psi_{rj}t_{<js]}) - \sum_{i=1}^{r-1}\Big(\frac{\psi_{ir} + \sum_{j=i}^{r-1}\psi_{ij}t_{<jr]}}{\psi_{rr}}\Big)\Big(\psi_{is} + \sum_{j=i}^{s-1}\psi_{ij}t_{<js]}\Big), \tag{2.2}$$

with $t_{<ij]} = t_{ij}/t_{jj}$. Following the notation in Dobra, Lenkoski and Rodriguez [6], we rewrite (2.1) as

$$p(\Psi^{\mathcal{V}}) \quad \propto \quad f(\Psi^{\mathcal{V}})h(\Psi^{\mathcal{V}}) \tag{2.3}$$

where

$$\log f(\Psi^{\mathcal{V}}) = -\frac{1}{2}\sum_{(i,j)\notin E, i<j}\psi_{ij}^2$$

is a function of the non-free elements of Ψ which is uniquely determined by $\Psi^{\mathcal{V}}$ according to (2.2) and $h(\Psi^{\mathcal{V}})$ is the density of the product of mutually independent chi-square and standard normal distributions. Based on the expression (2.3), Wang and Carvalho [28] suggested the following rejection sampling algorithm [21]:

1. Sample $\Psi^{\mathcal{V}}$ following Step 1 and 2 in Section 4.3 of Atay-Kayis and Massam [1], and $u \sim \mathrm{U}[0,1]$.
2. Check whether $u < f(\Psi^{\mathcal{V}})$. If this holds, accept $\Psi^{\mathcal{V}}$ as a sample from (2.1); if not, reject the value of $\Psi^{\mathcal{V}}$ and repeat the sampling step.
3. Construct a sample of Ω following Step 3 and 4 in Section 4.3 of Atay-Kayis and Massam [1].

Clearly, the acceptance rate depends on the triple (b, D, G). Dobra, Lenkoski and Rodriguez [6] showed that the acceptance rate can be as low as 10^{-8} for some (b, D, G).

2.2. Independent Metropolis-Hastings algorithm

Mitsakakis, Massam and Escobar [15] proposed an independent Metropolis-Hastings algorithm for sampling from the G-Wishart distribution based on the density (2.3). Their method generates a candidate $(\Psi^*)^{\mathcal{V}}$ from $h(\Psi^{\mathcal{V}})$, then instead of accepting it by probability $f\{(\Psi^*)^{\mathcal{V}}\}$ as in Wang and Carvalho [28], they correct the sample using a Metropolis-Hastings step. This results in the following modification in Steps 2 above:

2. Check whether $u < \min[f\{(\Psi^*)^{\mathcal{V}}\}/f(\Psi^{\mathcal{V}}), 1]$. If this holds, accept $(\Psi^*)^{\mathcal{V}}$ as a sample from (2.1); if not, retain current $\Psi^{\mathcal{V}}$.

Although, this method improves the acceptance rate over Wang and Carvalho [28] considerably, it still suffers from the same problem of not accepting new samples frequently when graphs are large [6].

2.3. Random walk Metropolis-Hastings algorithm

The methods of Wang and Carvalho [28] and Mitsakakis, Massam and Escobar [15] both involve changes of all free elements $\Psi^{\mathcal{V}}$ in a single step. Furthermore, this change of $\Psi^{\mathcal{V}}$ does not depend on its current value. As a result, this may cause slow mixing and low acceptance rate. Dobra, Lenkoski and Rodriguez [6] proposed a random walk Metropolis-Hastings algorithm (RW) that perturbs only one element in $\Psi^{\mathcal{V}}$ in a single step by drawing a random value from a normal distribution with a mean equal to its current value and a pre-specified variance. The random walk algorithm improves the efficiency over the methods of Wang and Carvalho [28] and Mitsakakis, Massam and Escobar [15] significantly. Nevertheless, the three methods in Section (2.1)-(2.3) or any other methods that use the matrix completion (2.2) can be inefficient for large problems. To see this, note that the completion step (2.2) can only be conducted iteratively. To complete ψ_{rs}, it involves calculating terms such as $\sum_{i=1}^{r-1}\sum_{j=i}^{r-1}\psi_{ij}t_{<jr]}$ and $\sum_{i=1}^{r-1}\sum_{j=i}^{s-1}\psi_{ij}t_{<js]}$ at an estimated time complexity $O(rs)$. Although the exact computing time depends on (r, s), in general, it requires $O(p^2)$ calculations for completing a typical non-free element ψ_{rs} with $1 \leq r < s \leq p$. For a graph with the number of missing edges in the order of p^2, the matrix completion requires a time complexity $O(p^4)$ for one update, which makes these methods unacceptably slow in large graphs as shown in Section 2.5.

2.4. Block Gibbs sampler

Piccioni [19] presented a theoretical framework that allows the construction of a block Gibbs sampler for sampling from the natural conjugate prior for regular exponential families. When applied to graphical models, the block Gibbs sampler corresponds to the Bayesian iterative proportional scaling. Suppose $\{C_1, \ldots, C_K\}$ is the set of maximum cliques of an arbitrary graph $G = (V, E)$. The Bayesian iterative proportional scaling method can be summarized as follows:

Bayesian iterative proportional scaling [19]. Given the current value $\Omega \in M^+(G)$ and a set of maximum cliques $\{C_1, \ldots, C_K\}$, for each $j = 1, \ldots, K$:

1. Sample $A \sim W(b, D_{C_j})$.
2. Set $\Omega_{C_j,C_j} = A + \Omega_{C_j,V\setminus C_j}(\Omega_{V\setminus C_j,V\setminus C_j})^{-1}\Omega_{V\setminus C_j,C_j}$.

Lenkoski and Dobra [13] and Mitsakakis, Massam and Escobar [15] implemented this method using the output of maximum clique enumeration algorithms. They pointed out the Bayesian iterative proportional scaling has two limitations: It requires maximum clique enumeration which is NP-hard and lacks good algorithms; and it involves a series of large matrix inversions in Step 2 for small cliques which is computationally expensive. Although it is unclear whether the maximum condition is necessary in applying the general results of Piccioni [19] to G-Wishart distributions, we are able to provide and justify a class of block Gibbs samplers directly from a Gibbs theory.

We first review a non-ordinary but theoretically valid Gibbs sampler. Suppose θ is a random vector that can be partitioned into p subvectors, $\theta = (\theta_1, \ldots, \theta_p)$, and $p(\theta)$ is the target distribution. Consider a collection of index sets $\mathcal{I} = \{\mathcal{I}_1, \ldots, \mathcal{I}_K\}$, where $\mathcal{I}_k \subseteq \{1, \ldots, p\}$ for $k = 1, \ldots, K$ such that $\cup_{k=1}^K \mathcal{I}_k = \{1, \ldots, p\}$. Let $\theta_{\mathcal{I}_k}$ denote the subset of elements in θ corresponding to the index set $\theta_{\mathcal{I}_k}$. Step k of a Gibbs sampler then generates from

$$\theta_{\mathcal{I}_k} \sim p(\theta_{\mathcal{I}_k} \mid \theta \backslash \theta_{\mathcal{I}_k}).$$

Note that the elements of $\{\theta_{\mathcal{I}_k} : \mathcal{I}_k \in \mathcal{I}\}$ may not be disjoint; thus some components of θ may be updated in multiple steps. This is not an ordinary Gibbs sampler which updates each component of θ only once in one sweep. In our notation, an ordinary Gibbs sampler means $\mathcal{I}$ is a partition of $\{1, \ldots, p\}$. However, updating elements of θ multiple times in one sweep does not cause any theoretical problem – moving components in a step of a Gibbs sampler from being conditioned on to being sampled neither changes the invariant distribution of the chain nor destroys the compatibility of the conditional distributions. In fact, this technique can improve the convergence property of the Gibbs sampler [26].

The above discussion shows that a non-ordinary Gibbs sampler has its target distribution as the stationary distribution. The following proposition is useful in proving that a Gibbs sampler is irreducible and aperiodic and hence will converge to its stationary distribution.

Proposition 2.1 (See, for example, Proposition 5 of [19])**.** *Let $p(\theta)$ be the target distribution. Suppose θ can be partitioned into p subvectors, $\theta = (\theta_1, \ldots, \theta_p)$. Consider the collection of index sets $\mathcal{I} = \{\mathcal{I}_1, \ldots, \mathcal{I}_K\}$, where $\mathcal{I}_k \subseteq \{1, \ldots, p\}$ for $k = 1, \ldots, K$ such that $\cup_{k=1}^K \mathcal{I}_k = \{1, \ldots, p\}$. For the Gibbs sampler that simulates each component from $p(\theta_{\mathcal{I}_k} \mid \theta \backslash \theta_{\mathcal{I}_k})$, if the marginal density*

$$p(\theta \backslash \theta_{\mathcal{I}_k}) = \int p(\theta \backslash \theta_{\mathcal{I}_k}, \theta_{\mathcal{I}_k}) \mathrm{d}\theta_{\mathcal{I}_k},$$

is bounded in $\theta \backslash \theta_{\mathcal{I}_k}$ for each $k = 1 \ldots, K$, then it is irreducible and aperiodic.

Using the above general formulation of a Gibbs sampler, we can design a class of Gibbs samplers for simulating from G-Wishart distributions and prove its convergence. We start with the construction of the Gibbs samplers that have the stationary distribution (1.1). Let $\Omega^{\mathcal{V}} = [\omega_{11}^2, \ldots, \omega_{pp}^2, \{\omega_{ij}\}_{(i,j) \in E, i<j}]$ be the set of the free elements of Ω. Consider a sequence of index sets $\mathcal{I} = \{\mathcal{I}_1, \ldots, \mathcal{I}_K\}$,

where $\mathcal{I}_k \subseteq V = \{1, \ldots, p\}$ for $k = 1, \ldots, K$ such that: (i) the subset $\mathcal{I}_k$ is complete, and (ii) $\cup_{\mathcal{I}_k \in \mathcal{I}} \Omega_{\mathcal{I}_k,\mathcal{I}_k} = \Omega^{\mathcal{V}}$ where $\Omega_{\mathcal{I}_k,\mathcal{I}_k}$ is a submatrix of Ω corresponding to $\mathcal{I}_k$. The two conditions for the construction of $\mathcal{I}$ are important. Condition (i) ensures that updating $\Omega_{\mathcal{I}_k,\mathcal{I}_k}$ can be carried out with the Wishart distribution, and condition (ii) ensures that all free elements in Ω can be updated. The Gibbs sampler then cycles through the collection of submatrices $\{\Omega_{\mathcal{I}_k,\mathcal{I}_k} : \mathcal{I}_k \in \mathcal{I}\}$, drawing each submatrix $\Omega_{\mathcal{I}_k,\mathcal{I}_k}$ from its conditional distribution given all the other components in Ω:

$$p(\Omega_{\mathcal{I}_k,\mathcal{I}_k} \mid \Omega \backslash \Omega_{\mathcal{I}_k,\mathcal{I}_k}), \tag{2.4}$$

where $\Omega \backslash \Omega_{\mathcal{I}_k,\mathcal{I}_k}$ represents all the components of Ω except for $\Omega_{\mathcal{I}_k,\mathcal{I}_k}$.

For any complete subset $\mathcal{I}_k$ of V, simulating $\Omega_{\mathcal{I}_k,\mathcal{I}_k}$ from (2.4) can be carried out as follows. Lemma 1 of Roverato [23] shows that

$$\Omega_{\mathcal{I}_k,\mathcal{I}_k} - \Omega_{\mathcal{I}_k,V\backslash\mathcal{I}_k}(\Omega_{V\backslash\mathcal{I}_k,V\backslash\mathcal{I}_k})^{-1}\Omega_{V\backslash\mathcal{I}_k,\mathcal{I}_k} \mid \Omega \backslash \Omega_{\mathcal{I}_k,\mathcal{I}_k} \sim \mathrm{W}(b, D_{\mathcal{I}_k,\mathcal{I}_k}). \tag{2.5}$$

Thus, we can first generate a Wishart random matrix $A \sim \mathrm{W}(b, D_{\mathcal{I}_k,\mathcal{I}_k})$ and then set $\Omega_{\mathcal{I}_k,\mathcal{I}_k} = A + \Omega_{\mathcal{I}_k,V\backslash\mathcal{I}_k}(\Omega_{V\backslash\mathcal{I}_k,V\backslash\mathcal{I}_k})^{-1}\Omega_{V\backslash\mathcal{I}_k,\mathcal{I}_k}$. Note that we use a different notation for a Wishart distribution with density (1.1) than Roverato [23]. We write $\mathrm{W}(b, D)$ while Roverato [23] used $\mathrm{W}(b + p - 1, D^{-1})$.

We next examine the convergence property the above Gibbs samplers by considering the bound of its marginal distributions in order to apply Proposition 2.1.

Proposition 2.2. *Suppose $\Omega \sim \mathrm{W}_G(b, D)$ and $\mathcal{I}_k \subseteq V$ is a complete subset. Then the marginal density*

$$p(\Omega \backslash \Omega_{\mathcal{I}_k,\mathcal{I}_k})$$

is bounded in $\Omega \backslash \Omega_{\mathcal{I}_k,\mathcal{I}_k}$.

Proof. From (2.5), we have

$$\begin{aligned} p(\Omega \backslash \Omega_{\mathcal{I}_k,\mathcal{I}_k}) &= \int I_G(b, D)^{-1} |\Omega|^{\frac{b-2}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}(D\Omega)\right\} \mathrm{d}\Omega_{\mathcal{I}_k,\mathcal{I}_k} \\ &= \frac{I(b, D_{\mathcal{I}_k,\mathcal{I}_k})}{I_G(b, D)} |\Omega_{V\backslash\mathcal{I}_k,V\backslash\mathcal{I}_k}|^{\frac{b-2}{2}} \exp\Big[-\frac{1}{2}\mathrm{tr}\big\{D_{V\backslash\mathcal{I}_k,V\backslash\mathcal{I}_k}\Omega_{V\backslash\mathcal{I}_k,V\backslash\mathcal{I}_k} \\ &\quad + 2\Omega_{\mathcal{I}_k,V\backslash\mathcal{I}_k}D_{V\backslash\mathcal{I}_k,\mathcal{I}_k} + D_{\mathcal{I}_k,\mathcal{I}_k}\Omega_{\mathcal{I}_k,V\backslash\mathcal{I}_k}(\Omega_{V\backslash\mathcal{I}_k,V\backslash\mathcal{I}_k})^{-1}\Omega_{V\backslash\mathcal{I}_k,\mathcal{I}_k}\big\}\Big]. \end{aligned} \tag{2.6}$$

Let

$$\begin{aligned} X &= \Omega_{V\backslash\mathcal{I}_k,V\backslash\mathcal{I}_k}, \quad Y = \Omega_{V\backslash\mathcal{I}_k,\mathcal{I}_k}D_{\mathcal{I}_k,\mathcal{I}_k}^{\frac{1}{2}}, \\ A &= D_{V\backslash\mathcal{I}_k,V\backslash\mathcal{I}_k}, \quad B = D_{V\backslash\mathcal{I}_k,\mathcal{I}_k}D_{\mathcal{I}_k,\mathcal{I}_k}^{-\frac{1}{2}}, \end{aligned}$$

and notice that $I(b, D_{\mathcal{I}_k,\mathcal{I}_k})$ and $I_G(b, D)$ are two constants not involving $\Omega \backslash \Omega_{\mathcal{I}_k,\mathcal{I}_k}$. Then, to show that (2.6) is bounded, it will suffice to show that

$$f(X, Y) = |X|^{\frac{b-2}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}(AX + 2Y'B + Y'X^{-1}Y)\right\},$$

is bounded in (X, Y). Taking the derivative of $f(X, Y)$ with respect to Y and then solving the first order condition that $\partial f(X, Y)/\partial Y = 0$, we obtain $Y = -XB$ and

$$f(X, Y) \leq f(X, -XB) = |X|^{\frac{b-2}{2}} \exp\left[-\frac{1}{2}\mathrm{tr}\{(A - BB')X\}\right].$$

Note that $A - BB' = D_{V\setminus\mathcal{I}_k, V\setminus\mathcal{I}_k} - D_{V\setminus\mathcal{I}_k,\mathcal{I}_k} D^{-1}_{\mathcal{I}_k,\mathcal{I}_k} D_{\mathcal{I}_k, V\setminus\mathcal{I}_k}$ is positive definite since D is positive definite. Thus, $f(X, -XB)$ is the density kernel of a G-Wishart distribution:

$$X \sim \mathrm{W}_{G_{V\setminus\mathcal{I}_k}}(b, A - BB'),$$

whose density function is bounded by the property of G-Wishart distributions. □

Coupled with Proposition 2.1, Proposition 2.2 shows that the class of block Gibbs samplers is indeed irreducible and aperiodic. With this elaboration, we may summarize the class of block Gibbs samplers as follows:

Block Gibbs sampler. Construct a sequence of index sets $\mathcal{I} = \{\mathcal{I}_1, \ldots, \mathcal{I}_K\}$, where $\mathcal{I}_k \subseteq V = \{1, \ldots, p\}$ for $k = 1, \ldots, K$ such that: (i) the subset $\mathcal{I}_k$ is complete, and (ii) $\cup_{\mathcal{I}_k \in \mathcal{I}} \Omega_{\mathcal{I}_k,\mathcal{I}_k} = \Omega^{\mathcal{V}}$. Given the current value $\Omega \in M^+(G)$, for $i = 1, \ldots, K$:

1. Sample $A \sim \mathrm{W}(b, D_{\mathcal{I}_k,\mathcal{I}_k})$.
2. Set $\Omega_{\mathcal{I}_k,\mathcal{I}_k} = A + \Omega_{\mathcal{I}_k, V\setminus\mathcal{I}_k}(\Omega_{V\setminus\mathcal{I}_k, V\setminus\mathcal{I}_k})^{-1}\Omega_{V\setminus\mathcal{I}_k,\mathcal{I}_k}$.

For an arbitrary graph G, the choice of the collection of the index sets $\mathcal{I}$ may not be unique. For example, consider a 3-node complete graph with $V = \{1, 2, 3\}$ and $E = \{(1, 2), (1, 3), (2, 3)\}$. Then two collections $\{(1, 2, 3)\}$ and $\{(1, 2), (1, 3), (2, 3)\}$ can both be used as $\mathcal{I}$. However, different choices of $\mathcal{I}$ lead to different configurations of the Gibbs sampler and hence different efficiency. In the 3-node complete graph example, the choice $\mathcal{I} = \{(1, 2, 3)\}$ leads to a 1-step sampler that directly generates Ω from a Wishart distribution, while the choice $\mathcal{I} = \{(1, 2), (1, 3), (2, 3)\}$ implies a 3-step Gibbs sampler. Intuitively, to choose $\mathcal{I}$, one would like K (the number of complete subsets) to be small and $|\mathcal{I}_k|$ (the dimension of the complete subset) to be large in order to reduce the number of steps of the Gibbs sampler as well as the correlation between $\Omega_{\mathcal{I}_k,\mathcal{I}_k}$'s. The extreme case where $\mathcal{I}$ is a collection of the maximum cliques of G offers one such choice. This corresponds to the algorithm implemented by Lenkoski and Dobra [13], which requires an algorithm for maximum clique decomposition.

In the other extreme, one can choose $\mathcal{I}$ to be the edge set E and the isolated node set I, that is, $\mathcal{I} = E \cup I$, and we might call it edgewise block Gibbs sampler. This choice of $\mathcal{I} = E \cup I$ does not require any clique decomposition and can be directly read from the graph. Our simulation studies show that the edgewise block Gibbs sampler is easy to implement and converges rapidly for sparse graphs. In cases where p is large, we obtain a fast updating scheme that avoids inverting a $(p - 2) \times (p - 2)$ matrix at Step 2 of the above algorithm as follows. For any $e = (i, j) \in E$, we aim to fast compute Ω^{-1}_{ee}. Note that

$\Omega^{-1}_{V\setminus e,V\setminus e} = \Sigma_{V\setminus e,V\setminus e} - \Sigma_{V\setminus e,e}\Sigma^{-1}_{ee}\Sigma_{e,V\setminus e}$ and Σ_{ee} is only 2×2. To rapidly compute $\Omega^{-1}_{V\setminus e,V\setminus e}$ using Σ, we only have to show that we can fast update Σ after Ω is updated at Step 2.

Suppose (Ω,Σ) is the present value and $\Omega_{ee,new}$ is a new value drawn by Step 2 in the block Gibbs sampler for edge e given (Ω,Σ). To update Ω, we do

$$\Omega_{new} = \Omega - U\Delta U',$$

where $\Delta = (\Omega_{ee} - \Omega_{ee,new})$ and U is a $p\times 2$ matrix with the identity matrix in rows i and j and zeros in the remaining entries. To update Σ, we apply the identity

$$(\Omega - U\Delta U')^{-1} = \Omega^{-1} + \Omega^{-1}U(\Delta^{-1} - U'\Omega^{-1}U)^{-1}U'\Omega^{-1},$$

and update Σ as follows

$$\Sigma_{new} = (\Omega - U\Delta^{-1}U')^{-1} = \Sigma + \Sigma_{\cdot e}(\Delta^{-1} - \Sigma_{ee})^{-1}\Sigma_{e\cdot}.$$

All matrix inversions in the edgewise block Gibbs sampler are inversions of 2×2 matrices only which can be done very efficiently.

The two extreme block Gibbs samplers of using maximum cliques or edges illustrate a tradeoff between the efficiency and the ease of implementation. In practice, a more useful choice of $\mathcal{I}$ is, perhaps, a mix of large complete components and edges. For example, starting from $\mathcal{I} = E\cup I$, one can merge any number of $\mathcal{I}_k$ such that the union of those $\mathcal{I}_k$ forms a complete component to improve the efficiency over the edgewise samplers. The two general conditions for the construction of $\mathcal{I}$ makes the implementation of our block Gibbs sampler indeed flexible. Finally, we emphasize that the G-Wishart distribution is unimodal for any G. This important property ensures that the block Gibbs samplers typically converge rapidly and give reliable estimates.

2.5. Simulated experiments comparing samplers

To illustrate the computational aspects of the block Gibbs samplers, we compare both the edgewise and the maximum clique block Gibbs samplers with the random walk Metropolis-Hastings algorithm (RW) of Dobra, Lenkoski and Rodriguez [6], where RW is shown to be dominating other methods. We consider three types of graphs:

1. A sparse circle graph. The edge set is $E = \{(i,i+1) : 1\le i\le p-1\}\cup(1,p)$. The G-Wishart parameters are $b = 103$ and $D = I_p + 100A^{-1}$ where $A_{ii} = 1$ for $i\in V$, $A_{ij} = 0.5$ for $|i-j| = 1$, $A_{1p} = A_{p1} = 0.4$, and $A_{ij} = 0$ for $(i,j)\notin E$.
2. A random graph. The edge set E is randomly generated from independent Bernoulli distributions with probability 0.3. The G-Wishart parameters are $b = 103$ and $D = I + 100J^{-1}$ where $J = B + \delta I_p$ and $B_{ij} = 0.5$ if $(i,j)\in E$. B has zeros on the diagonal, and δ is chosen so that the condition number of J is p. Here the condition number is defined as $\max(\lambda)/\min(\lambda)$ where λ is the eigenvalues of the matrix J.

3. A two-clique graph. The graph has two maximum cliques: $C_1 = \{1, \ldots, p/2+2\}$ and $C_2 = \{p/2-2, \ldots, p\}$. The G-Wishart parameters are $b = 103$ and $D = I + 100J^{-1}$ where $J = B + \delta I_p$ and $B_{ij} = 0.5$ if $(i,j) \in E$. B has zeros on the diagonal, and δ is chosen so that the condition number of J is p.

The sparse circle graph was used in Dobra, Lenkoski and Rodriguez [6] under a set of different values of p for $p \leq 20$. For the RW approach, let σ_m be the standard deviation of the normal proposal. Several initial runs under different combinations of $(\sigma_m, p) \in \{0.1, 0.5, 1, 2, 3\} \times \{10, 20, 30\}$ suggested that $\sigma_m = 2$ gives the best mixing results for all cases. Hence we used $\sigma_m = 2$ in this simulation study. When updating one free element of $\Psi^{\mathcal{V}}$, we completed only the non-free elements coming after the free element which we perturbed using (2.2). One iteration entails updating all free elements $\Psi^{\mathcal{V}}$ once. For the maximum clique Gibbs samplers, we used the algorithm of Bron and Kerbosch [2] to produce all maximum cliques. For the two block Gibbs samplers, one iteration entails updating all components in the set $\mathcal{I}$ once.

We saved 5000 iterations after discarding 2000 burn-in iterations for all three samplers. To measure efficiency, we recorded the total CPU run time and the lag of iterations required to obtain samples that could be practically treated as independent, measured as

$$L_{ij} = \text{argmin}_k\{\rho_{ij}(k) < 2/\sqrt{M}, k \geq 1\}, \quad i = j \text{ or } (i,j) \in E,$$

where $\rho_{ij}(k)$ is the autocorrelation at lag k for ω_{ij} and M is the total number of saved iterations. We also calculated the effective sample size:

$$ESS_{ij} = M/\{1 + 2\sum_{k=1}^{\infty} \rho_{ij}(k)\}, \quad i = j \text{ or } (i,j) \in E$$

where we cut off the sum at lag $(L_{ij} - 1)$ to reduce noise from higher order lags [12].

Finally, we computed the percent error

$$PE_{ij} = \frac{|\hat{\sigma}_{ij} - E(\sigma_{ij})|}{E(\sigma_{ij})} \times 100\%, \quad i = j \text{ or } (i,j) \in E,$$

where $E(\sigma_{ij})$ is the theoretical expectation available in closed form (Corollary 2 of [23])

$$E(\sigma_{ij}) = D_{ij}/(b-2) \quad i = j \text{ or } (i,j) \in E,$$

and $\hat{\sigma}_{ij}$ is the posterior mean estimates of $E(\sigma_{ij})$. We used $E(\sigma_{ij})$ not $E(\omega_{ij})$ because only $E(\sigma_{ij})$ is analytically available for non-decomposable graphs. To summarize these different L_{ij}, ESS_{ij} and PE_{ij} for different entries, we used their corresponding medians.

The results based on 10 repetitions are given in Table 1. We report the mean among the 10 runs. The standard deviations around the mean are less than 5% for CPU, L and ESS, and less than 40% for PE. The programs are written in Matlab and run on a quad-cpu 3.33GHz desktop running CentOS 5.0

TABLE 1
Summary of performance measures in Section 2.5 for different graph and size combinations, comparing the random walk Metropolis-Hastings (RW) algorithm, the edgewise block Gibbs algorithm (Edgewise), and the maximum clique block Gibbs sampler (MaxC) for sampling from the G-Wishart distribution

		Circle			Random			Two-clique		
		p=10	p=20	p=30	p=10	p=20	p=30	p=10	p=20	p=30
CPU	RW	82	1549	8496	92	2521	19080	183	650	5980
	Edgewise	16	35	56	18	93	210	62	215	467
	MaxC	16	35	56	14	52	129	3	4	4
ESS	RW	854	964	1003	1098	1002	1026	1241	1109	1113
	Edgewise	5000	5000	5000	5000	2782	2618	1360	1083	890
	MaxC	5000	5000	5000	5000	4529	4625	5000	5000	5000
Lag	RW	11	11	11	9	10	9	9	9	9
	Edgewise	1	1	1	1	4	4	11	15	19
	MaxC	1	1	1	1	2	2	1	1	1
PE	RW	0.27	0.29	0.29	0.31	0.53	0.68	0.6	0.8	1.5
	Edgewise	0.17	0.17	0.17	0.17	0.26	0.34	0.6	0.8	1.5
	MaxC	0.17	0.17	0.17	0.17	0.25	0.33	0.5	0.8	1.4
Relative ESS	RW	1	1	1	1	1	1	1	1	1
	Edgewise	28	228	720	23	75	233	0.35	3	10
	MaxC	28	228	720	28	218	675	28	841	6317

unix. The last two rows in Table 1 record the relative ESS of the two Gibbs samplers over RW after standardizing for the CPU run time. As expected, the maximum clique block Gibbs sampler performs best in all scenarios in terms of computing time and mixing. For the circle and the random graphs, the edgewise sampler dominates RW in every dimension. Depending on p and G, the edgewise sampler gives a 5- to 150-fold reduction in run-time and around a 2- to 5-fold improvement in the effective sample size, and producing substantially smaller percentage errors. For the two clique graph, the edgewise sampler mixes less well than the RW sampler; however, it is significantly faster and overall more efficient as measured by relative ESS for large problems. In summary, the RW sampler does not scale well as the block Gibbs samplers as the dimension p grows. The maximum clique sampler is the most efficient sampler for G-Wishart distributions given the availability of maximum cliques. The edgewise sampler has excellent performances for sparse graphs and is easy to use.

3. Existing methods for normalizing constant approximation

3.1. Monte Carlo integration

Atay-Kayis and Massam [1] developed a Monte Carlo method to approximate the normalizing constant of the G-Wishart distribution based the decomposition described in Section 2.1. For a G-Wishart distribution $\mathrm{W}_G(b, D)$, its normalizing

constant can be expressed as

$$I_G(b,D) = \left\{\prod_{i=1}^{p} 2^{\frac{b+\nu_i}{2}}(2\pi)^{\frac{\nu_i}{2}}\Gamma\left(\frac{b+\nu_i}{2}\right)T_{ii}^{\frac{b+h_i-1}{2}}\right\}E_\Psi\left\{f(\Psi^{\mathcal{V}})\right\} \quad (3.1)$$

where ν_i is the number of neighbors of node i subsequent to it in the ordering of vertices, h_i is the total number of neighbors of node i plus 1, and $f(\Psi^{\mathcal{V}})$ is defined in (2.3). Because the distribution of $\Psi^{\mathcal{V}}$ can be easily sampled from, it is straightforward to estimate the expectation part of (3.1) by Monte Carlo.

The Monte Carlo integration can be computationally expensive when the non-complete prime component is large or when it is used without graph decomposition. This is because it relies on the matrix completion (2.2) to evaluate the function $f(\Psi^{\mathcal{V}})$. Moreover, the variance of the Monte Carlo estimator depends on the data, the graph and the order of nodes, making it difficult to evaluate [11].

3.2. Laplace approximation

Lenkoski and Dobra [13] proposed a Laplace approximation to $I_G(b,D)$, namely,

$$\widehat{I_G(b,D)} = \exp\left\{l(\hat{\Omega})\right\}(2\pi)^{|\mathcal{V}|/2}|H(\hat{\Omega})|^{-1/2}, \quad (3.2)$$

where

$$l = \frac{b-2}{2}\log|\Omega| - \frac{1}{2}\mathrm{tr}(D\Omega),$$

$\hat{\Omega}$ is the mode of $\mathrm{W}_G(b,D)$, and H is the Hessian matrix associated with l.

Theoretical evaluation of the Laplace approximation in Gaussian graphical models has yet to be investigated, though Lenkoski and Dobra [13] empirically demonstrated its potential to facilitate computation in problems where cliques are restricted to be small, e.g. $p \leq 5$. Intuitively, the accuracy of the Laplace approximation depends on the degree to which the density of $\Omega_{\mathcal{V}}$ resembles a multivariate normal distribution. By comparing the margins of Ω to a normal distribution, Lenkoski and Dobra [13] empirically showed that the closeness increases as d increases. Hence, they suggested to use the computationally fast but less accurate Laplace approximation for the posterior normalizing constant and the computationally expensive but more accurate Monte Carlo integration for the prior normalizing constant.

4. Existing reversible jump samplers for graphical model determination

The normalizing constant approximation methods in Section 3 allow us to marginalize over Ω and work directly on the graphical model space. However, these approximations may be numerically unstable especially for posterior normalizing constants. Alternatively, there are several special samplers that can

explore the joint space of graphs and precision matrices without integrating out Ω. Giudici and Green [9] proposed a reversible jump sampler for jointly simulating (G, Ω) for decomposable graphs. For both across- and within-graph moves, they update one entry in Σ at a time by proposing from an independent normal distribution with mean zero and the variance appropriately tuned. These types of moves require the check of the positive definite constraint of Ω for each update. Dobra, Lenkoski and Rodriguez [6] designed another reversible jump sampler based on the re-parameterization (G, Ψ) where $\Psi = \Phi T^{-1}$ with Φ and T the Cholesky decompositions of Ω and $(D + Y'Y)^{-1}$ respectively. This sampler ensures the positive definitiveness of Ω automatically; however, it requires the computationally intensive matrix completion and the tuning of proposals for both across- and within-graph moves. Furthermore, both Giudici and Green [9] and Dobra, Lenkoski and Rodriguez [6] still require the evaluation of prior normalizing constants.

In the following section, we improve the efficiency of the reversible jump sampler by first eliminating the need of proposal tuning and pilot run, and then eliminating the need for evaluating prior normalizing constants.

5. Proposed algorithms for graphical model determination

5.1. Eliminating proposal tuning

We first present a new algorithm that requires neither the matrix completion nor the proposal tuning for both across- and within-graph moves. The central idea of our sampler is to make use of the *partial analytic structure* (PAS) of G-Wishart distributions for stochastic model selection. We briefly summarize the main feature of a reversible jump algorithm that uses the partial analytic structure [10]. Suppose there are K candidate models $\{M_k\}_{k=1}^K$, each of which is associated with a likelihood as $p(y \mid \theta_k, M_k)$ that depends upon a set of unknown parameter θ_k. Consider a move from the current model M_k to a new model $M_{k'}$. Suppose there exists a subvector $(\theta_{k'})_{\mathcal{U}}$ of the parameter $\theta_{k'}$ for a new model $M_{k'}$ such that $p\{(\theta_{k'})_{\mathcal{U}} \mid (\theta_{k'})_{-\mathcal{U}}, M_{k'}, y\}$ is available in closed form, and in the current model M_k, there exists an equivalent subset of parameters $(\theta_k)_{-\mathcal{U}}$ with the same dimension as $(\theta_{k'})_{-\mathcal{U}}$. The PAS reversible jump algorithm uses a proposal distribution that sets $(\theta_{k'})_{-\mathcal{U}} = (\theta_k)_{-\mathcal{U}}$, draws $M_{k'} \sim q(M_{k'} \mid M_k)$ and $(\theta_{k'})_{\mathcal{U}} \sim p\{(\theta_{k'})_{\mathcal{U}} \mid (\theta_{k'})_{-\mathcal{U}}, M_{k'}, y\}$. The reverse move then sets $(\theta_k)_{-\mathcal{U}} = (\theta_{k'})_{-\mathcal{U}}$, draws $M_k \sim q(M_k \mid M_{k'})$ and $(\theta_k)_{\mathcal{U}} \sim p\{(\theta_k)_{\mathcal{U}} \mid (\theta_k)_{-\mathcal{U}}, M_k, y\}$. In summary, the PAS algorithm proceeds as follows:

PAS algorithm [10]. Given the current state (M_k, θ_k):

1. Update M_k. Note that this move also involves making changes of $(\theta_k)_{\mathcal{U}}$.
   (a) Propose a new model $M_{k'}$ from the proposal distribution $q(M_{k'} \mid M_k)$; set $(\theta_{k'})_{-\mathcal{U}} = (\theta_k)_{-\mathcal{U}}$.

(b) Accept the proposed model $M_{k'}$ with probability:

$$\alpha = \min\left[1, \frac{p\{M_{k'} \mid (\theta_{k'})_{-\mathcal{U}}, y\} q(M_k \mid M_{k'})}{p\{M_k \mid (\theta_k)_{-\mathcal{U}}, y\} q(M_{k'} \mid M_k)}\right], \tag{5.1}$$

where $p\{M_{k'} \mid (\theta_{k'})_{-\mathcal{U}}, y\} = \int p\{M_{k'}, (\theta_{k'})_{\mathcal{U}} \mid (\theta_{k'})_{-\mathcal{U}}, y\} \mathrm{d}(\theta_{k'})_{\mathcal{U}}$.

(c) If $M_{k'}$ is accepted, generate $(\theta_{k'})_{\mathcal{U}} \sim p\{(\theta_{k'})_{\mathcal{U}} \mid (\theta_{k'})_{-\mathcal{U}}, M_{k'}, y\}$. Otherwise, generate $(\theta_k)_{\mathcal{U}} \sim p\{(\theta_k)_{\mathcal{U}} \mid (\theta_k)_{-\mathcal{U}}, M_k, y\}$.

2. Update θ_k.

(a) Update the parameters $\theta_{k'}$ if $M_{k'}$ is accepted using standard MCMC steps. Otherwise, update the parameters θ_k using standard MCMC steps.

We now detail the PAS algorithm for sampling (G, Ω) from the full posterior distribution

$$p(\Omega, G \mid Y) \propto I_G^{-1}(b, D) |\Omega|^{\frac{n+b-2}{2}} \exp\left[-\frac{1}{2}\mathrm{tr}\{(S+D)\Omega\}\right] p(G) 1_{\Omega \in M^+(G)}. \tag{5.2}$$

Consider two graphs $G = (V, E)$ and $G' = (V, E')$ that differ by one edge (i, j) only. With no loss of generality, suppose edge $(i, j) \in E$ and $E' = E \backslash (i, j)$. In the notation of PAS algorithm, we set $M_k = G, M_{k'} = G', (\theta_{k'})_{-\mathcal{U}} = (\theta_k)_{-\mathcal{U}} = \Omega \backslash (\omega_{ij}, \omega_{jj}), (\theta_k)_{\mathcal{U}} = \{\omega_{jj}, \omega_{ij}\}$ and $(\theta_{k'})_{\mathcal{U}} = \omega_{jj}$ to make use of the analytical structure of $p(\omega_{jj} \mid \Omega \backslash (\omega_{ij}, \omega_{jj}), G')$ and $p(\omega_{ij}, \omega_{jj} \mid \Omega \backslash (\omega_{ij}, \omega_{jj}), G)$. From (5.1), the acceptance probability for a move G to G' from a proposal $q(G' \mid G)$ is then

$$\alpha(G \to G') = \min\left[1, \frac{p\{G' \mid \Omega \backslash (\omega_{ij}, \omega_{jj}), Y\} q(G \mid G')}{p\{G \mid \Omega \backslash (\omega_{ij}, \omega_{jj}), Y\} q(G' \mid G)}\right], \tag{5.3}$$

where the conditional posterior odds against the edge (i, j) is given by:

$$\frac{p\{G' \mid \Omega \backslash (\omega_{ij}, \omega_{jj}), Y\}}{p\{G \mid \Omega \backslash (\omega_{ij}, \omega_{jj}), Y\}} = \frac{p\{Y, \Omega \backslash (\omega_{ij}, \omega_{jj}) \mid G'\} p(G')}{p\{Y, \Omega \backslash (\omega_{ij}, \omega_{jj}) \mid G\} p(G)}.$$

For the first numerator term, since $\omega_{ij} = 0$ under G', it is defined as

$$p\{Y, \Omega \backslash (\omega_{ij}, \omega_{jj}) \mid G'\} = \int_{\omega_{jj}} p\{Y \mid \Omega, G'\} p(\Omega \mid G') \mathrm{d}\omega_{jj}. \tag{5.4}$$

Note that, from (2.5), the full conditional posterior for ω_{jj} is

$$\omega_{jj} - c \mid (\Omega \backslash \omega_{jj}, Y) \sim \mathrm{W}(b + n, D_{jj} + S_{jj}), \tag{5.5}$$

where $c = \Omega_{j, V\backslash j} (\Omega_{V\backslash j, V\backslash j})^{-1} \Omega_{V\backslash j, j}$. Let $\Omega^0 = \Omega$ except for an entry 0 in the positions (i, j) and (j, i), and an entry c in the position (j, j). Then (5.4) can be analytically evaluated as

$$\begin{aligned} p\{Y, \Omega \backslash (\omega_{ij}, \omega_{jj}) \mid G'\} = (2\pi)^{-\frac{np}{2}} \frac{I(b+n, D_{jj}+S_{jj})}{I_{G'}(b, D)} |\Omega^0_{V\backslash j, V\backslash j}|^{\frac{n+b-2}{2}} \\ \times \exp\left[-\frac{1}{2}\mathrm{tr}\{(S+D)\Omega^0\}\right]. \end{aligned} \tag{5.6}$$

Where $I(b, D)$ is the normalizing constant of a scalar G-Wishart distribution $\mathrm{W}_G(b, D)$ with $p = 1$. For the first denominator term, since $\omega_{ij} \neq 0$ under G, it is defined as

$$p\{Y, \Omega\backslash(\omega_{ij}, \omega_{jj}) \mid G\} \quad = \quad \int_{(\omega_{ij},\omega_{jj})} p\{Y \mid \Omega, G\} p(\Omega \mid G') \mathrm{d}\omega_{ij} \mathrm{d}\omega_{jj}. \quad (5.7)$$

To evaluate (5.7), we need the full conditional distribution of $(\omega_{ij}, \omega_{jj})$. Let $e = (i, j)$ and write $\Omega_{ee|V\backslash e} = \Omega_{ee} - \Omega_{e,V\backslash e}(\Omega_{V\backslash e,V\backslash e})^{-1}\Omega_{V\backslash e,e}$. From (2.5), the conditional posterior of $(\omega_{ii}, \omega_{ij}, \omega_{jj})$ is

$$\Omega_{ee|V\backslash e} \mid (\Omega\backslash\Omega_{ee}, Y) \sim \mathrm{W}(b + n, D_{ee} + S_{ee}).$$

Further conditioned on ω_{ii}, the full conditional distribution of $(\omega_{ij}, \omega_{jj})$ can then be obtained by applying the standard Wishart theory in the following Proposition 3 to the Wishart matrix $\Omega_{ee|V\backslash e}$.

Proposition 5.1. *Suppose a* 2×2 *random matrix* A *follows a Wishart distribution* $\mathrm{W}(h, B)$ *with density*

$$p(A) = I(h, B)^{-1}|A|^{\frac{h-2}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}(BA)\right\}.$$

Write

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}.$$

Then,

(i) $a_{12} \mid a_{11} \sim \mathrm{N}(-B_{22}^{-1}B_{12}a_{11}, B_{22}^{-1}a_{11})$ *and* $a_{22} - a_{11}^{-1}a_{12}^2 \mid a_{11}, a_{12} \sim \mathrm{W}(h, B_{22})$.
(ii) $p(a_{12}, a_{22} \mid a_{11}) = \{J(h, B, a_{11})\}^{-1}|A|^{\frac{h-2}{2}} \exp\{-\frac{1}{2}\mathrm{tr}(BA)\}$, *where*

$$\begin{aligned} J(h, B, a_{11}) \quad &= \quad \int |A|^{\frac{h-2}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}(BA)\right\} \mathrm{d}a_{12}\mathrm{d}a_{22} \\ &= \quad (2\pi B_{22}^{-1})^{\frac{1}{2}} a_{11}^{\frac{h-1}{2}} I(h, B_{22}) \exp\left\{-\frac{1}{2}(B_{11} - B_{22}^{-1}B_{12}^2)a_{11}\right\}. \end{aligned}$$

Let Ω^1 be equal to Ω except for entries of $\Omega_{e,V\backslash e}(\Omega_{V\backslash e,V\backslash e})^{-1}\Omega_{V\backslash e,e}$ in the positions corresponding to e. Applying Proposition 5.1 by letting $A = \Omega_{ee|V\backslash e}$, $h = b + n$ and $B = D_{ee} + S_{ee}$ allows us to evaluate the density (5.7) analytically:

$$\begin{aligned} p\{Y, \Omega\backslash(\omega_{ij}, \omega_{jj}) \mid G\} = (2\pi)^{-\frac{np}{2}} \frac{J(b + n, D_{ee} + S_{ee}, a_{11})}{I_G(b, D)} |\Omega^1_{V\backslash e,V\backslash e}|^{\frac{n+b-2}{2}} \\ \times \exp\left[-\frac{1}{2}\mathrm{tr}\{(S + D)\Omega^1\}\right], \end{aligned} \quad (5.8)$$

where a_{11} is the first element of $A = \Omega_{ee|V\backslash e}$.

We plug-in (5.6) and (5.8) to (5.3) to provide the acceptance rate for a move from G to G':

$$\alpha(G \to G') \quad = \quad \min\left\{1, \frac{p(G')q(G \mid G')I_G(b, D)}{p(G)q(G' \mid G)I_{G'}(b, D)} H(e, \Omega)\right\}, \quad (5.9)$$

where

$$H(e,\Omega) = \frac{I(b+n, D_{jj}+S_{jj})}{J(b+n, D_{ee}+S_{ee}, a_{11})}\left(\frac{|\Omega^0_{V\setminus j, V\setminus j}|}{|\Omega^1_{V\setminus e, V\setminus e}|}\right)^{\frac{n+b-2}{2}} \times \exp\left[-\frac{1}{2}\mathrm{tr}\{(S+D)(\Omega^0-\Omega^1)\}\right],$$

can be analytically evaluated.

We can summarize the PAS sampler for graphical model determination as follows:

Algorithm 1. Given the current state (G, Ω):

1. Update G. Note that this move also involves making changes of $(\omega_{ij}, \omega_{jj})$.
   (a) Propose a new graph $G' = (V, E')$ differing only one edge from $G = (V, E)$ from the proposal distribution $q(G' \mid G)$. Without loss of generality, assume edge $e = (i, j) \in E$ and $E' = E\setminus e$.
   (b) Accept G' with probability α in (5.9).
   (c) If G' is accepted, set $\omega_{ij} = 0$, update the parameters ω_{jj} from (5.5). If G' is rejected, update the parameters $(\omega_{ij}, \omega_{jj})$ from its full conditional distribution using Proposition 2.2 (i). Specifically, in the notation of Proposition 2.2, let $A = (a_{ij}) = \Omega_{ee|V\setminus e}$, $h = b+n$ and $B = (B_{ij}) = D_{ee} + S_{ee}$. In addition, let $F = (f_{ij}) = \Omega_{e,V\setminus e}(\Omega_{V\setminus e, V\setminus e})^{-1}\Omega_{V\setminus e, e}$, then $(\omega_{ij}, \omega_{jj})$ is generated as follows:
      (i) Generate $u \mid a_{11} \sim \mathrm{N}(-B_{22}^{-1}B_{12}a_{11}, B_{22}^{-1}a_{11})$ and $v \mid a_{11} \sim \mathrm{W}(h, B_{22})$.
      (ii) Set $\omega_{ij} = u + f_{12}$ and $\omega_{jj} = v + a_{11}^{-1}u^2 + f_{22}$.
2. Update Ω conditional on the most recent G using the block Gibbs sampler in Section 2.4.

In Step 1(a) of Algorithm 1, instead of randomly picking up an edge and then correcting it by a Metropolis-Hastings step, we can often scan through all (i, j) for $i < j$ according to various deterministic or random schedules and update edge (i, j) as a Bernoulli random variable with the following conditional posterior odds

$$\frac{p\{G' \mid Y, \Omega\setminus(\omega_{ij}, \omega_{jj})\}}{p\{G \mid Y, \Omega\setminus(\omega_{ij}, \omega_{jj})\}} = \frac{p(G')I_G(b, D)H(e, \Omega)}{p(G)I_{G'}(b, D)},$$

which lead to a Gibbs sampler with the acceptance rate (5.9) uniformly equal to 1.

The main benefit of the above PAS algorithm is that the acceptance rate (5.9), with $(\omega_{ij}, \omega_{jj})$ integrated out, eliminates the need of across-graph proposal tuning. The new algorithm also uses the block Gibbs sampler for simulating from G-Wishart distributions at given graphs, eliminating the need of matrix completion and within-graph proposal tuning. However, it still requires the prior normalizing constant approximation.

5.2. Eliminating evaluation of prior normalizing constants

This section aims to circumvent the remaining computational bottleneck arising from the intractable prior normalizing constants in Algorithm 1. Our main tool is the double Metropolis-Hastings algorithm [14], which is an extension of the exchange algorithm [17] for simulating from distributions with intractable normalizing constants.

We start with a brief review of the exchange algorithm proposed by Murray, Ghahramani and MacKay [17]. Suppose data y are generated from the density $p(y \mid \theta) = \mathcal{Z}(\theta)^{-1} f(y \mid \theta)$ where θ is the parameter and $\mathcal{Z}(\theta) = \int f(y \mid \theta) \mathrm{d}y$ is the normalizing constant that depends on θ and is not analytically available. Suppose the prior for θ is $p(\theta)$. A standard Metropolis-Hastings (M-H) algorithm simulates from the posterior of θ: $p(\theta \mid y) \propto p(\theta) f(y \mid \theta)/\mathcal{Z}(\theta)$ by proposing θ' from a proposal $q(\theta' \mid \theta)$ and then accepting it with probability

$$\alpha = \min\left\{1, \frac{p(\theta') f(y \mid \theta') \mathcal{Z}(\theta) q(\theta \mid \theta')}{p(\theta) f(y \mid \theta) \mathcal{Z}(\theta') q(\theta' \mid \theta)}\right\},$$

which depends on the ratio of two intractable normalizing constants. The exchange algorithm removes the need to evaluate $\mathcal{Z}$ by considering an augmented distribution

$$p(\theta, \theta', x \mid y) = p(\theta) \frac{f(y \mid \theta)}{\mathcal{Z}(\theta)} q(\theta' \mid \theta, y) \frac{f(x \mid \theta')}{\mathcal{Z}(\theta')},$$

where $q(\theta' \mid \theta, y)$ is an arbitrary distribution and x is an auxiliary variable. Marginally, the original distribution $p(\theta \mid y)$ is maintained. The exchange algorithm samples (θ, θ', x) from the augmented distribution using a standard Metropolis-Hastings sampler. Operationally,

The exchange algorithm [17]. Given the current state (θ, θ', x):

1. Update (θ', x) using a block Gibbs step.
   (a) Generate $(\theta', x) \sim q(\theta' \mid \theta, y) f(x \mid \theta')$ by first drawing $\theta' \sim q(\theta' \mid \theta, y)$ and then drawing an auxiliary variable $x \sim f(x \mid \theta')$ using an exact sampler.
2. Update θ using a Metropolis step.
   (a) Propose θ' by exchanging θ and θ'. Note that this is a symmetric proposal.
   (b) Accept θ' with probability

$$\alpha = \min\left\{1, \frac{p(\theta') f(y \mid \theta') f(x \mid \theta) q(\theta \mid \theta', y)}{p(\theta) f(y \mid \theta) f(x \mid \theta') q(\theta' \mid \theta, y)}\right\}.$$

Comparing the acceptance rate of the exchange algorithm with that of the traditional M-H algorithm, we see that the exchange algorithm replaces the

intractable normalizing constant ratio with an estimate from a single sample at each parameter setting:

$$\mathcal{Z}(\theta)/\mathcal{Z}(\theta') \approx f(x \mid \theta)/f(x \mid \theta'), \quad x \sim p(x \mid \theta'), \tag{5.10}$$

which provides some insight about why the exchange algorithm works. The use of the auxiliary variable x removes $\mathcal{Z}(\theta)$ from the joint distribution; however it requires an exact sampler for $p(x \mid \theta')$, which is not practical in many applications. Liang [14] proposed a double Metropolis-Hastings algorithms to avoid the need of exact samplers. Their approach generates x from $p(x \mid \theta')$ using a product of Metropolis-Hastings updates starting at y:

$$P_{\theta'}^{(m)}(x \mid y) = K_{\theta'}(y \to y_1) \dots K_{\theta'}(y_{m-1} \to x),$$

where $K(\cdot \to \cdot)$ is the M-H transition kernel of $p(x \mid \theta')$. They derived the following extension of the exchange algorithm that does not require an exact sampler for $x \sim p(x \mid \theta')$:

Double M-H algorithm [14]. Given the current state (θ, θ', x)

1. Update (θ', x)
   (a) Generate $\theta' \sim q(\theta' \mid \theta, y)$ and then $x \sim P_{\theta'}^{(m)}(x \mid y)$ where $P_{\theta'}^{(m)}(x \mid y)$ is a sequence of M-H kernels of the target distribution $p(x \mid \theta')$ initialized at y.
2. Same as Step 2 in the exchange algorithm.

Since two types of Metropolis-Hastings moves are performed for updating θ: One for generating the auxiliary variable x in Step 1(a) and the other for accepting θ in Step 2. The algorithm is called a double Metropolis-Hastings algorithm by Liang [14]. When x is generated exactly from $f(x \mid \theta')$ by an exact sampler, the double Metropolis-Hastings reduces to the exchange algorithm. When x is generated approximately by M-H methods, the double Metropolis-Hastings can be viewed as an approximated exchange algorithm. In such cases, caution must be made about the convergence of the double M-H algorithm. Since the relationship of (5.10) suggests that we use one sample of x to provide the information about the global normalizing constant $\mathcal{Z}(\theta)$, this sample must be generated in a way that considers the entire space $f(x \mid \theta)$. An auxiliary variable x generated by an exact sampler considers the entire space of $f(x \mid \theta)$; however, an auxiliary variable x generated by a Markov chain will be biased towards a local mode near the starting point with only a few M-H steps. Choosing the M-H kernel $K(\cdot \to \cdot)$ for a MCMC to rapidly explore the global auxiliary variable space without being trapped by local modes is the key. We refer to Chapter 5 of Murray [16] for a discussion about using MCMC to generate the auxiliary variable x. Now, we extend the PAS algorithm in Section 5.1 by applying the double M-H algorithm to remove the need of prior normalizing constants. In the notation of the double M-H algorithm, let $\theta = G$ and $y = \Omega$ and consider the augmented joint distribution

$$p\{\Omega, G, G', \Omega' \mid Y\} = p(\Omega, G \mid Y) q(G' \mid \Omega, G, Y) p(\Omega' \mid G'),$$

where $p(\Omega, G \mid Y)$ is the original target distribution (5.2), $q(G' \mid \Omega, G, Y)$ is any distribution that proposes a graph G' that differs by one edge (i,j) from G with $(i,j) \in E$ and $(i,j) \notin E'$, and $p(\Omega' \mid G')$ is the density function of $\Omega' \sim \mathrm{W}_{G'}(b, D)$. Marginally, the original posterior (5.2) is unaffected. We consider the following move types.

(1) Update (G', Ω').
(2) Update G. Note that this also involves updating $(\omega_{ij}, \omega_{jj})$.
(3) Update Ω.

Move (1) generates G' directly from $q(G' \mid \Omega, G, Y)$ and Ω' from $p(\Omega' \mid G')$ using a sequence of M-H steps starting from the current Ω. Notice that $\omega_{ij} \neq 0$ and $\omega'_{ij} = 0$. Thus, starting at $\Omega \backslash (\omega_{ij}, \omega_{jj})$, we first update $(\omega_{ij}, \omega_{jj})$ from their conditional prior distributions under G' and then use m steps of the block Gibbs sampler to generate the auxiliary Ω'. Hence the product of M-H updates $P^{(m)}_{G'}(\Omega' \mid \Omega)$ consists of m steps of the block Gibbs sampler applied to $\mathrm{W}_{G'}(b, D)$. Thanks to the unimodal property of the G-Wishart distribution, the Gibbs kernels will properly consider the entire auxiliary data space $\Omega' \sim \mathrm{W}_{G'}(b, D)$ without being biased towards a local mode near the starting point. As for the choice of m, Liang [14] suggested only a small m (e.g. $m = 1$) is needed for obtaining a good sample of $x \sim p(x \mid \theta')$. In the examples analyzed in this paper, we found that one iteration of a block Gibbs sampler is sufficient to provide good mixing results.

For Move (2), this is essentially a PAS step that proposes G by swapping G and G', with $(\omega_{ij}, \omega_{jj})$ and $(\omega'_{ij}, \omega'_{jj})$ integrated out. The acceptance rate is then

$$\alpha = \min\left[1, \frac{p\{G' \mid Y, \Omega \backslash (\omega_{ij}, \omega_{jj})\} q(G \mid \Omega, G', Y)}{p\{G \mid Y, \Omega \backslash (\omega_{ij}, \omega_{jj})\} q(G' \mid \Omega, G, Y)} \times \frac{p\{\Omega' \backslash (\omega'_{ij}, \omega'_{jj}) \mid G\}}{p\{\Omega' \backslash (\omega'_{ij}, \omega'_{jj}) \mid G'\}}\right],$$

where the first part is exactly equal to the original acceptance rate (5.3) which is expressed in (5.9) as

$$\frac{p\{G' \mid Y, \Omega \backslash (\omega_{ij}, \omega_{jj})\} q(G \mid \Omega, G', Y)}{p\{G \mid Y, \Omega \backslash (\omega_{ij}, \omega_{jj})\} q(G' \mid \Omega, G, Y)} = \frac{p(G') q(G \mid \Omega, G', Y) I_G(b, D)}{p(G) q(G' \mid \Omega, G, Y) I_{G'}(b, D)} H(e, \Omega),$$

and the second part $p\{\Omega' \backslash (\omega'_{ij}, \omega'_{jj}) \mid G\} / p\{\Omega' \backslash (\omega'_{ij}, \omega'_{jj}) \mid G'\}$ can be evaluated by making use of the full conditional distributions of $(\omega'_{ij}, \omega'_{jj})$ under G' and G respectively. Let $\Omega'^0 = \Omega'$ except for an entry 0 in the positions (i,j) and (j,i) and an entry $\Omega'_{j,V\backslash j}(\Omega'_{V\backslash j, V\backslash j})^{-1}\Omega'_{V\backslash j, j}$ in the position (j,j); let $\Omega'^1 = \Omega'$ except for $\Omega'_{e,V\backslash e}(\Omega'_{V\backslash e, V\backslash e})^{-1}\Omega'_{V\backslash e, e}$ in the positions corresponding to the edge $e = (i,j)$. It is apparent to show the following:

$$p\{\Omega' \backslash (\omega'_{ij}, \omega'_{jj}) \mid G'\} = \frac{f(\Omega' \backslash (\omega'_{ij}, \omega'_{jj}) \mid G')}{I_{G'}(b, D)},$$
$$p\{\Omega' \backslash (\omega'_{ij}, \omega'_{jj}) \mid G\} = \frac{f(\Omega' \backslash (\omega'_{ij}, \omega'_{jj}) \mid G)}{I_G(b, D)},$$

where $f(\Omega'\backslash(\omega'_{ij},\omega'_{jj}) \mid G')$ and $f(\Omega'\backslash(\omega'_{ij},\omega'_{jj}) \mid G)$ are analytically evaluated as

$$f(\Omega'\backslash(\omega'_{ij},\omega'_{jj}) \mid G') = I(b, D_{jj})|\Omega'^0_{V\backslash j,V\backslash j}|^{\frac{b-2}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}(D\Omega'^0)\right\},$$

and

$$f(\Omega'\backslash(\omega'_{ij},\omega'_{jj}) \mid G) = J(b, D_{ee}, a_{11})|\Omega'^1_{V\backslash e,V\backslash e}|^{\frac{b-2}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}(D\Omega'^1)\right\}.$$

Collecting all terms together, we have the acceptance rate of a move from G to G' as

$$\alpha(G \to G') = \min\left\{1, \frac{p(G')q(G \mid \Omega, Y, G')f(\Omega'\backslash(\omega'_{ij},\omega'_{jj}) \mid G)}{p(G)q(G' \mid \Omega, Y, G)f(\Omega'\backslash(\omega'_{ij},\omega'_{jj}) \mid G')}H(e,\Omega)\right\}, \tag{5.11}$$

where all terms are analytically available. Comparing the acceptance rate (5.11) of the double M-H algorithm to the acceptance rate (5.9) of the original PAS sampler, we see that the double M-H algorithm replaces the intractable prior normalizing constant with the unbiased estimate based on a single sample from the prior:

$$\frac{I_G(b,D)}{I_{G'}(b,D)} \approx \frac{f(\Omega'\backslash(\omega'_{ij},\omega'_{jj}) \mid G)}{f(\Omega'\backslash(\omega'_{ij},\omega'_{jj}) \mid G')}.$$

This gives an interpretation on why the double M-H algorithm works.

Finally, Move (3) generates Ω conditional on the graph from Move (2) using the block Gibbs sampler. We may summarize the algorithm as follows:

Algorithm 2. Given the current state $\{G, \Omega, G', \Omega'\backslash(\omega'_{ij},\omega'_{jj})\}$:

1. Update $\{G', \Omega'\backslash(\omega'_{ij},\omega'_{jj})\}$
   (a) Propose a new graph G' differing only one edge from G from the proposal distribution $q(G' \mid \Omega, G, Y)$. Without loss of generality, assume that edge $(i,j) \in G$ and $(i,j) \notin G'$.
   (b) Generate the auxiliary variable $\Omega'\backslash(\omega'_{ij},\omega'_{jj}) \sim P^{(m)}_{G'}\{\Omega'\backslash(\omega'_{ij},\omega'_{jj}) \mid \Omega\backslash(\omega_{ij},\omega_{jj})\}$ using the block Gibbs sampler with initial value $\Omega\backslash(\omega_{ij},\omega_{jj})$.
2. Update G
   (a) Exchange G and G'.
   (b) Accept G' with probability (5.11).
   (c) According to whether G' is accepted or not, update $(\omega_{ij},\omega_{jj})$ from their conditional distributions as in Step 1(c) in Algorithm 1.
3. Update Ω conditional on the most recent G using the block Gibbs sampler.

In step 1(a), we can also systematically scan through all (i,j) for $i < j$ and update edge (i,j) using a Bernoulli proposal with the following odds

$$\frac{q(G' \mid \Omega, Y)}{q(G \mid \Omega, Y)} = \frac{p(G')H(e,\Omega)}{p(G)},$$

which simplifies the acceptance rate (5.11) as

$$\alpha = \min\left\{1, \frac{f(\Omega'\backslash(\omega'_{ij}, \omega'_{jj}) \mid G)}{f(\Omega'\backslash(\omega'_{ij}, \omega'_{jj}) \mid G')}\right\}.$$

6. Simulation experiment

6.1. A 6 node example

To investigate the accuracy of Algorithm 2, we consider a case with $p = 6$, yielding a graphical model space of size 32768, which is small enough to be enumerated, yet large enough to be interesting and to have a significant proportion of non-decomposable graphs that are about 45% of all graphs. We let $S = YY' = nA^{-1}$ where $n = 18$ and

$$A = \begin{pmatrix} 1 & 0.5 & 0 & 0 & 0 & 0.4 \\ & 1 & 0.5 & 0 & 0 & 0 \\ & & 1 & 0.5 & 0 & 0 \\ & & & 1 & 0.5 & 0 \\ & & & & 1 & 0.5 \\ & & & & & 1 \end{pmatrix}.$$

This choice of (S, n) represents 18 samples of Y from $\mathrm{N}(0, A^{-1})$. We placed the G-Wishart prior $\mathrm{W}_G(3, I_6)$ on Ω and the uniform prior $p(G) \propto 1$ on G.

We calculated the theoretical posterior edge inclusion probabilities, denoted by $(p_{ij})_{1\le i<j\le p} \mid Y$, and the theoretical posterior expectations of Σ and Ω, denoted by $E(\Sigma \mid Y)$ and $E(\Omega \mid Y)$ respectively as follows. For each $G \in \mathcal{G}$ where $\mathcal{G}$ is the space of all 32768 graphs, we calculated its posterior probability as

$$p(G \mid Y) \quad = \quad \frac{p(G) I_G(b+n, D+S)/I_G(b, D)}{\sum_{G\in\mathcal{G}}\{p(G) I_G(b+n, D+S)/I_G(b, D)\}}, \tag{6.1}$$

using the Monte Carlo integration of Section 3.1 for I_G when G is non-decomposable and the closed-form of I_G when G is decomposable. We then calculated the theoretical posterior edge inclusion probabilities as

$$p_{ij} \mid Y = \sum_{G=(V,E)\in\mathcal{G}} 1_{\{(i,j)\in E\}}\, p(G \mid Y) \quad 1 \le i < j \le p,$$

and the theoretical posterior expectations of Σ and Ω as

$$\begin{aligned} E(\Sigma \mid Y) \quad &= \quad \sum_{G\in\mathcal{G}} E(\Sigma \mid Y, G)\, p(G \mid Y), \\ E(\Omega \mid Y) \quad &= \quad \sum_{G\in\mathcal{G}} E(\Omega \mid Y, G)\, p(G \mid Y), \end{aligned} \tag{6.2}$$

respectively. In (6.2), $E(\Sigma \mid Y, G)$ and $E(\Omega \mid Y, G)$ are analytically available only for decomposable graphs [20]. For non-decomposable graphs, we estimated $E(\Sigma \mid$

$Y, G)$ and $E(\Omega \mid Y, G)$ based on their corresponding posterior sample means calculated from the output of the Gibbs sampler of Section 2.4. We shall report the Monte Carlo sample sizes we used: In (6.1), sample sizes 1000 and 50000 were used for the prior and the posterior normalizing constants, respectively; in (6.2), a MCMC sample of 10000 iterations after an initial 5000 runs as burn-in was used. These sample sizes allow the Monte Carlo estimation to be performed for each of the non-decomposable graphs and also yield an agreement of about 2 decimal places for almost all elements in $(p_{ij}) \mid Y, E(\Sigma \mid Y)$ and $E(\Omega \mid Y)$ when we repeated the entire process two more times. The final results of the theoretical posterior edge inclusion probabilities and the theoretical posterior expectations of Σ and Ω are:

$$(p_{ij})_{1\le i<j\le p} \mid Y = \begin{pmatrix} 1 & 0.969 & 0.106 & 0.085 & 0.113 & 0.850 \\ & 1 & 0.980 & 0.098 & 0.081 & 0.115 \\ & & 1 & 0.982 & 0.098 & 0.086 \\ & & & 1 & 0.980 & 0.106 \\ & & & & 1 & 0.970 \\ & & & & & 1 \end{pmatrix},$$

$$E(\Sigma \mid Y) = \begin{pmatrix} 5.211 & -4.953 & 4.746 & -4.544 & 4.338 & -4.131 \\ & 6.461 & -5.897 & 5.378 & -4.863 & 4.345 \\ & & 7.072 & -6.204 & 5.372 & -4.547 \\ & & & 7.074 & -5.890 & 4.748 \\ & & & & 6.452 & -4.951 \\ & & & & & 5.214 \end{pmatrix},$$

and

$$E(\Omega \mid Y) = \begin{pmatrix} 1.139 & 0.569 & -0.011 & 0.006 & -0.013 & 0.403 \\ & 1.175 & 0.574 & -0.008 & 0.005 & -0.014 \\ & & 1.176 & 0.574 & -0.008 & 0.006 \\ & & & 1.175 & 0.573 & -0.011 \\ & & & & 1.175 & 0.569 \\ & & & & & 1.138 \end{pmatrix}.$$

Now, we compare the results obtained from Algorithm 2 to the above theoretical values. We applied Algorithm 2 with a systematic scan for 60000 sweeps and discarded the first 10000 as burn-in. Each sweep entails updating all possible edges and all elements in Ω once. Two chains were run: One starting at the identity matrix for Ω and one at the sample precision matrix. The results were essentially the same for both chains. The posterior mean estimates of (p_{ij}), Σ and Ω are

$$(\hat{p}_{ij}) = \begin{pmatrix} 1 & 0.969 & 0.106 & 0.087 & 0.116 & 0.854 \\ & 1 & 0.983 & 0.096 & 0.083 & 0.113 \\ & & 1 & 0.980 & 0.103 & 0.087 \\ & & & 1 & 0.978 & 0.110 \\ & & & & 1 & 0.963 \\ & & & & & 1 \end{pmatrix},$$

$$\hat{\Sigma} = \begin{pmatrix} 5.217 & -4.952 & 4.749 & -4.545 & 4.339 & -4.135 \\ & 6.452 & -5.896 & 5.373 & -4.858 & 4.343 \\ & & 7.074 & -6.198 & 5.367 & -4.544 \\ & & & 7.065 & -5.880 & 4.739 \\ & & & & 6.443 & -4.936 \\ & & & & & 5.211 \end{pmatrix},$$

and

$$\hat{\Omega} = \begin{pmatrix} 1.139 & 0.570 & -0.010 & 0.006 & -0.014 & 0.404 \\ & 1.179 & 0.575 & -0.008 & 0.006 & -0.014 \\ & & 1.174 & 0.571 & -0.009 & 0.006 \\ & & & 1.174 & 0.572 & -0.013 \\ & & & & 1.173 & 0.564 \\ & & & & & 1.135 \end{pmatrix}.$$

Comparing these MCMC estimates with the theoretical values computed above, we see that Algorithm 2 is able to produce accurate estimates. As for the computing time, under Matlab implementation, Algorithms 2 took about 15 minutes to complete 60000 sweeps, while the Monte Carlo integration method took about 16 hours to evaluate all 32768 graphs.

6.2. A 100 node circle graph example

The second example is more challenging as it has a large non-complete prime components of size 100. We simulated a sample of size $n = 150$ from the model $N(0, A^{-1})$ where A is defined in Section 2.5. The prior parameters were $b = 3$, $D = I_{100}$ and independent edge inclusion probabilities $2/(p-1)$. We ran the systematic scan version of Algorithm 2 for 30000 sweeps while discarding the first 30000 warm-up iterations. Two chains were run: One starting at the identity matrix and one at the sample covariance matrix. The results from these two runs were similar. The median effective sample size of the free elements of Ω corresponding to the posterior mean graph was 30000 for a sample of size 30000. The posterior mean graph which includes only edges having posterior inclusion probability exceeding 0.5 is the true underlying circle graph. The highest probability excluded edge has probability 0.08 while the lowest probability included edge has probability 1.

As for comparison, we used the Monte Carlo integration of Section 3.1 to approximate the marginal likelihood of the true underlying graph. Under a C++ implementation, it took about 2 minutes to calculate the prior normalizing constant using 1000 Monte Carlos iterations. For the posterior normalizing constant, the computing time is about the same. However, the algorithm seems to underflow in a standard implementation. That is, the true value of the function $f(\Psi^{\mathcal{V}})$ tends to be smaller than the computer's smallest positive floating point number. Figure 1 displays the boxplot of values of the function $\log f(\Psi^{\mathcal{V}})$ evaluated at $M = 1000$ samples of $\Psi^{\mathcal{V}}$ adjusted by an offset:

$$\log f(\Psi_i^{\mathcal{V}}) - \text{offset}, i = 1, \ldots, M$$

where offset $= \max\{\log f(\Psi_i^{\mathcal{V}}) : i = 1, \ldots, M\}$. The majority of these values are less than -2000, while the smallest positive floating point number in double precision is about -709 in a natural logarithm scale. Recall that $\log I_G$ is estimated by,

$$\widehat{\log I_G} = \text{offset} + \log\left(\sum_{i=1}^{M} \exp\{\log f(\Psi_i^{\mathcal{V}}) - \text{offset}\}\right) - \log M,$$

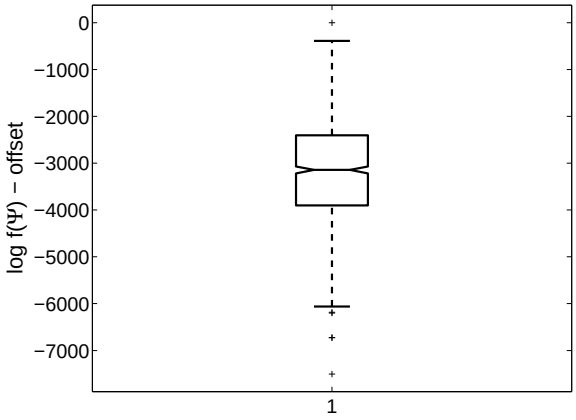

FIG 1. *Box plot showing the distribution of* $(\log f(\Psi^{\mathcal{V}}) - \textit{offset})$ *in the 100 node cycle graph example.*

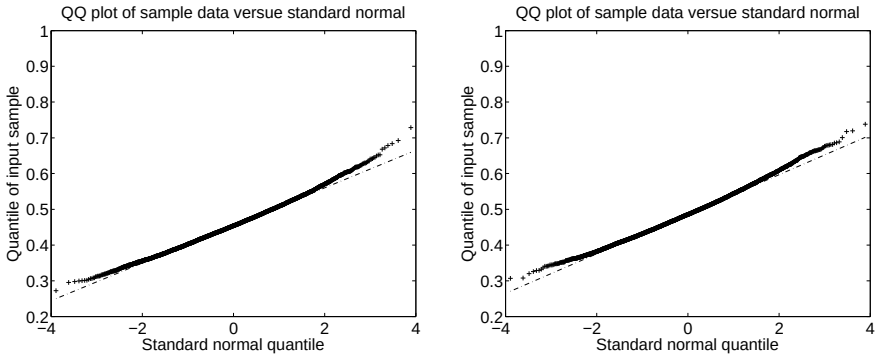

FIG 2. *Q-Q plots comparing the marginal distributions of two entries of* $\Omega \sim \mathrm{W}_G(b+n, D+S)$ *with the normal distribution in the 100 node cycle graph example.*

so the summation is taken over zeros most of the time. This example illustrates that the Monte Carlo integration may require a high precision arithmetic library so that it can precisely give results of exponential functions of $-10^4 \sim -10^3$ or so. To our knowledge, current software for Gaussian graphical models has yet supported this level of precision. Even if the underflow problem is addressed, the computation time can be unacceptable when we increase the number of Monte Carlo sample size until the variance falls below a fixed level. For example, using the default sample size $1.5p^3$ suggested by Jones et al. [11] will cost $1.5 \times 100^3 \times 2/1000 = 3 \times 10^3$ minutes to evaluate this graph. Without a good estimate of the normalizing constant, we were unable to evaluate the accuracy of the Laplace approximation. However, note that the accuracy largely depends on the similarity between $\mathrm{W}_G(b+n, D+S)$ and the normal distribution. Using the edgewise sampler, we simulated 10000 samples from $\mathrm{W}_G(b+n, D+S)$ under the true graph. Figure 2 illustrates the Q-Q plots for two univariate margins of $\Omega^{\mathcal{V}}$. These margins are clearly different from the normal distribution.

7. Mutual fund performance evaluation

In this section, we illustrate the extension of Gaussian graphical models to a class of sparse seemingly unrelated regression models. We show that how graphs can be useful in modeling real problems and how Algorithm 2 can be used as a key component of a larger sampling scheme.

The historical performance of a mutual fund can be summarized by estimating its alpha. This term is defined as the intercept in a regression of the excess return of the fund on the excess return of one or more passive benchmarks. This is usually estimated by applying an ordinary least square analysis (OLS) to the regression

$$y_{0,t} = \alpha_0 + x_t'\beta_0 + e_{0,t}, \quad t = 1:T$$

where $y_{0,t}$ is the fund return at time t, x_t is a $k \times 1$ vector of benchmark returns at time t, and α_0 is the fund alpha. The choice of benchmarks is often guided by a pricing model, such as the capital asset pricing model (CAPM) [25] and the Fama-French three factor model [7]. The work of Pástor and Stambaugh [18] has explored the role of nonbenchmark passive assets in estimating a fund's alpha using a seemingly unrelated regression (SUR) model. Suppose there are p nonbenchmark passive returns $y_{1:p,t}$ besides the k benchmark returns x_t. Further suppose returns on passive assets including benchmark or nonbenchmark assets are constructed for the period from 1 to T and a mutual fund only has a history from t_0 to T where $t_0 \geq 1$. Then the SUR model used to estimate the mutual fund α_0 is written as

$$\begin{aligned} y_{0,t} &= \alpha_0 + x_t'\beta_0 + e_{0,t}, \qquad t = 1:T, \\ y_{i,t} &= \alpha_i + x_t'\beta_i + e_{i,t}, \qquad i = 1:p;\ t = 1:T, \end{aligned} \tag{7.1}$$

where $y_{0,t} = y_{0,t}^\star$ is missing for $t < t_0$ and the error vector $e_{0:p,t}$ is distributed as $N(0, \Sigma)$. The basic idea is that a more precise estimate of α_0 is provided through a more precise estimate of $\alpha_{1:p}$ when $e_{0,t}$ is correlated with the $e_{1:p,t}$. Note that many mutual funds have relatively short histories as compared with passive assets. Given the more accurate estimate of $\alpha_{1:p}$ computed from a longer sample period, the α_0 estimated from SUR is more precise than the α_0 estimated solely based on OLS.

Some interesting questions arise in evaluating mutual fund performance using SUR of (7.1). First, as is observed by Pástor and Stambaugh [18], the assumption of pricing power of benchmark assets on nonbenchmark assets, i.e. $\alpha_i = 0$ or not for $i = 1:p$, is critical in estimating a fund's alpha in a SUR model. The second question concerns the strictness of the SUR model assumption, that is, returns are assumed to be contemporaneously correlated with all nonbenchmark returns given the benchmark returns. For some managed funds, perhaps only the errors from a subset of nonbenchmark assets are relevant in explaining returns of the fund. Including too many correlated nonbenchmark assets to estimate alpha will mean a potentially high misspecification risk.

Motivated by these practically important considerations, we consider the following sparse seemingly unrelated regression (SSUR) models that extend SUR

to address the two questions above. We use the hierarchical mixture prior for each of $\alpha_{0:p}$:

$$\alpha_i \sim (1 - z_i)N(0, \nu_{i,0}^2) + z_i N(0, \nu_{i,1}^2),$$

where $z_i = 0$ or 1 according to whether the benchmark assets have the pricing power or not respectively and ν_{i0} and ν_{i1} are set to be small and large respectively [8]. We next apply the Gaussian graphical model to the residual covariance matrix Σ to naturally model the contemporaneous dependence among mutual fund and nonbenchmark returns. Algorithm 2 developed in Section 5 will then extend to include components to sample $(\alpha_{0:p}, z_{0:p}, \beta_{0:p})$ and $y^{\star}_{0,1:t_0-1}$ at each iteration, using the efficient stochastic search variable selection (SSVS) procedure and conventional imputation approach.

To evaluate the efficacy of the model, we applied it to a collection of 15 actively managed Vanguard mutual funds, using monthly returns through December 2008 available from the Center for Research in Security Prices (CRSP) mutual fund database. The set of benchmark and nonbenchmark assets consists of eleven portfolios constructed passively. Monthly returns on these passive assets are available from January 1927 through December 2008. The sample period for any given mutual fund is a much shorter subset of this overall period. We specify the benchmark series as the excess market returns (MKT), and so the alpha is exclusively defined with respect to just MKT. The first two of nonbenchmark passive portfolios are the Fama-French factors, which are the payoffs on long-short spreads constructed by sorting stocks according to the market capitalization and the book-to-market ratio. The third, fourth and fifth nonbenchmark series are the momentum, short term and long term reversal factors respectively. The remaining five nonbenchmark assets are the value-weighted returns for five industrial portfolios.

We choose $\nu_{i,0} = 0.025$ and $\nu_{i,1} = 0.5$ for monthly α_i. This choice of hyperparameters is in line with the view that a yearly return of $0.025 \times 2 \times 12 = 0.6\%$ in excess of the compensation for the risk borne may possibly be ignored and the maximum plausible yearly return for α_i is about $0.5 \times 2 \times 12 = 12\%$. We assume a uniform prior for z_i. We compare three methods for estimating α_0: OLS, SUR and SSUR. Table 2 reports the estimated α_0, the standard error and the posterior probability of the event $\{z_0 = 1\}$ within each fund based on the three methods for the period since a fund's inception. The SSUR estimates are nontrivially different from their OLS and SUR counterparts. In particular, the α_0's tend towards zeros under SSUR. This is not surprising since SSUR assumes a positive probability for small values of α_0. One important issue in fund performance evaluation is whether the managed fund adds value beyond the standard passive benchmarks. We address this issue by computing the estimated probability of the event $\{z_0 = 1\}$ in the last column. Only a few funds have the estimated probability exceeding 0.5. This suggests that most of the 15 mutual funds do not generate excess returns beyond the passive benchmark assets. Furthermore, the SUR standard errors are generally smaller than their OLS counterparts. This observation is compatible with that in Pástor and Stambaugh [18]. With few exceptions, SSUR seems to reduce the standard er-

TABLE 2
Summary of the estimated monthly α for three different models: OLS, SUR and SSUR. For OLS and SUR, point estimates and standard errors are reported. For SSUR, posterior mean, standard deviation and probability of $\alpha \neq 0$ are reported.

	OLS		SUR		SSUR		
Fund name	$\hat{\alpha}$	$s.e.(\hat{\alpha})$	$\hat{\alpha}$	$s.e.(\hat{\alpha})$	$\hat{\alpha}$	$s.e.(\hat{\alpha})$	$P(\alpha \neq 0)$
Cap Opp	0.34	0.26	0.43	0.13	0.45	0.15	0.98
Dividend Growth	0.05	0.18	0.05	0.08	0.01	0.04	0.11
Equity-Income	0.14	0.12	0.16	0.08	0.04	0.07	0.25
Explorer	-0.05	0.14	0.07	0.15	0.02	0.06	0.17
Growth& Income	0.02	0.06	0.08	0.10	0.01	0.05	0.13
Growth Equity	-0.20	0.16	-0.14	0.12	-0.03	0.08	0.23
Mid Cap Growth	0.55	0.38	0.47	0.16	0.51	0.16	0.99
Morgan Growth	0.04	0.07	0.14	0.12	0.05	0.09	0.28
PRIMECAP	0.23	0.12	0.33	0.11	0.30	0.14	0.90
Selected Value	0.09	0.28	0.10	0.11	0.03	0.07	0.19
Strategic Equity	0.14	0.17	0.19	0.11	0.09	0.12	0.42
US Growth	0.31	0.26	0.39	0.20	0.24	0.19	0.62
US Value	0.31	0.17	0.33	0.09	0.31	0.13	0.92
Windsor	0.14	0.09	0.19	0.12	0.10	0.12	0.47
Windsor II	0.13	0.12	0.14	0.10	0.03	0.07	0.22

ror even more than SUR. Recall that the standard error of the SSUR estimates takes into account of structure uncertainty. The reduced standard errors seem to suggest that there is a great deal of sparsity within SUR and that identifying this sparsity can help provide more precise estimates of α_0's. Finally, we note that the estimated graphs representing a fund's contemporaneously dependencies on nonbenchmark assets seem to reflect a fund's portfolio composition. For example, the fund Explorer seeks small US companies with growth potential and has top two holdings on the information technology and health care sectors as of May, 2008. The error of this fund is related to the error of nonbenchmark assets representing market capitalization, and high technology, and health care.

8. Discussion

We have described a sampling algorithm for Bayesian model determination in Gaussian graphical models. Our method has three ingredients: A block Gibbs sampler for within-graph moves, a reversible jump sampler using partial analytic structure for across-graph moves, and an exchange algorithm for avoiding the evaluation of prior normalizing constants.

For the covariance selection problem, a possible disadvantage of not approximating the marginal likelihood is that this does not allow for more flexible search algorithms for rapid traversal of the graph space. However, the subsequence of graphs from the auxiliary chain we developed will in many cases have the property that high probability graphs will appear more quickly than low ones, providing useful guidelines for setting Monte Carlo sample size or starting graphs using the more computationally intensive methods based on the normalizing constant approximation.

For problems where graphical models are only components of larger models, search algorithm does not apply and MCMC is routinely used for posterior computation with graphs either restricted to be decomposable or determined by approximating normalizing constants conditional on other parameters. The approximation step often costs substantial computational burden. Our method then has an advantage of being able to be easily embedded within a large MCMC algorithm to accelerate posterior computation.

Finally, we note that software implementing all analyses discussed in the paper is freely available from the first author's the web site of the paper.

Acknowledgements

The authors are grateful for the constructive comments of the editor, associate editor, and two anonymous referees on the original version of this manuscript. Support was provided by China National Social Science Foundation grant 11CJY096.

References

[1] Atay-Kayis, A. and Massam, H. (2005). The marginal likelihood for decomposable and non-decomposable graphical Gaussian models. *Biometrika* **92** 317-35. MR2201362

[2] Bron, C. and Kerbosch, J. (1973). Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM* **16** 575–577.

[3] Carvalho, C., Massam, H. and West, M. (2007). Simulation of hyper-inverse Wishart distributions in graphical models. *Biometrika* **94** 647-659. MR2410014

[4] Carvalho, C. M. and West, M. (2007). Dynamic matrix-variate graphical models. *Bayesian Analysis* **2** 69-98. MR2289924

[5] Dobra, A. and Lenkoski, A. (2011). Copula Gaussian Graphical Models. *Annals of Applied Statistics* **5** 969-993. MR2840183

[6] Dobra, A., Lenkoski, A. and Rodriguez, A. (2011). Bayesian inference for general Gaussian graphical models with application to multivariate lattice data. *Journal of the American Statistical Association (to appear)*.

[7] Fama, E. F. and French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics* **33** 3-56.

[8] George, E. I. and McCulloch, R. E. (1997). Approaches for Bayesian variable selection. *Statistica Sinica* **7** 339–373.

[9] Giudici, P. and Green, P. J. (1999). Decomposable graphical Gaussian model determination. *Biometrika* **86** 785-801. MR1741977

[10] Godsill, S. J. (2001). On the Relationship Between Markov chain Monte Carlo Methods for Model Uncertainty. *Journal of Computational and Graphical Statistics* **10** 230-248. MR1939699

[11] Jones, B., Carvalho, C., Dobra, A., Hans, C., Carter, C. and West, M. (2005). Experiments in stochastic computation for high-dimensional graphical models. *Statistical Science* **20** 388-400. MR2210226

[12] Kass, R. E., Carlin, B. P., Gelman, A. and Neal, R. M. (1998). Markov Chain Monte Carlo in Practice: A Roundtable Discussion. *The American Statistician* **52** 93-100. MR1628427

[13] Lenkoski, A. and Dobra, A. (2011). Computational Aspects Related to Inference in Gaussian Graphical Models With the G-Wishart Prior. *Journal of Computational and Graphical Statistics* **20** 140-157. MR2816542

[14] Liang, F. (2010). A double Metropolis-Hastings sampler for spatial models with intractable normalizing constants. *Journal of Statistical Computing and Simulation* **80** 1007-1022. MR2742519

[15] Mitsakakis, N., Massam, H. and Escobar, M. (2010). A Metropolis-Hastings based method for sampling from G-Wishart distribution in Gaussian graphical Models. *Electronic Journal of Statistics* **5** 18-31. MR2763796

[16] Murray, I. (2007). Advances in Markov chain Monte Carlo methods PhD Thesis, Gatsby computational neuroscience unit,University College London.

[17] Murray, I., Ghahramani, Z. and MacKay, D. (2006). MCMC for doubly-intractable distributions. In *(Proceedings) Uncertainty in Artificial Intelligence* (R. Dechter and T. Richardson, eds.) 359-366. AUAI Press.

[18] Pástor, L. and Stambaugh, R. F. (2002). Mutual fund performance and seemingly unrelated assets. *Journal of Financial Economics* **63** 315-349.

[19] Piccioni, M. (2000). Independence Structure of Natural Conjugate Densities to Exponential Families and the Gibbs' Sampler. *Scandinavian Journal of Statistics* **27** 111-127. MR1774047

[20] Rajaratnam, B., Massam, H. and Carvalho, C. M. (2008). Flexible Covariance Estimation in Graphical Gaussian Models. *Annals of Statistics* **36** 2818–49. MR2485014

[21] Robert, C. and Casella, G. (2010). *Monte Carlo Statistical Methods*, 2 ed. Springer-Verlag, New York. MR2080278

[22] Rodriguez, A., Lenkoski, A. and Dobra, A. (2011). Sparse covariance estimation in heterogeneous samples. *Electronic Journal of Statistics (forthcoming)*.

[23] Roverato, A. (2002). Hyper-Inverse Wishart Distribution for Non-decomposable Graphs and its Application to Bayesian Inference for Gaussian Graphical Models. *Scandinavian Journal of Statistics* **29** 391-411. MR1925566

[24] Scott, J. G. and Carvalho, C. M. (2008). Feature-Inclusion Stochastic Search for Gaussian Graphical Models. *Journal of Computational and Graphical Statistics* **17** 790-808. MR2649067

[25] Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *Journal of Finance* **19** 425–442.

[26] van Dyk, D. A. and Park, T. (2008). Partially Collapsed Gibbs Samplers. *Journal of the American Statistical Association* **103** 790-796. MR2524010

[27] Wang, H. (2010). Sparse seemingly unrelated regression modelling: Applications in finance and econometrics. *Computational Statistics & Data Analysis* **54** 2866-2877. MR2720481

[28] Wang, H. and Carvalho, C. M. (2010). Simulation of hyper-inverse Wishart distributions for non-decomposable graphs. *Electronic Journal of Statistics* **4** 1470-1475. MR2741209

[29] Wang, H., Reeson, C. and Carvalho, C. M. (2011). Dynamic Financial Index Models: Modeling Conditional Dependencies via Graphs. *Bayesian Analysis* **6** 639-664.

[30] Wang, H. and West, M. (2009). Bayesian analysis of matrix normal graphical models. *Biometrika* **96** 821-834. MR2564493