Electronic Journal of Statistics
Vol. 6 (2012) 2309–2331
ISSN: 1935-7524
DOI: 10.1214/12-EJS746

Hierarchical Gaussian graphical models: Beyond reversible jump

Yuan Cheng

Institute of Mathematics
University of Potsdam, Germany
e-mail: yuan.cheng@uni-potsdam.de

and

Alex Lenkoski*

Norwegian Computing Center
Oslo, Norway
e-mail: alex@nr.no

Abstract: The Gaussian Graphical Model (GGM) is a popular tool for incorporating sparsity into joint multivariate distributions. The G-Wishart distribution, a conjugate prior for precision matrices satisfying general GGM constraints, has now been in existence for over a decade. However, due to the lack of a direct sampler, its use has been limited in hierarchical Bayesian contexts, relegating mixing over the class of GGMs mostly to situations involving standard Gaussian likelihoods. Recent work has developed methods that couple model and parameter moves, first through reversible jump methods and later by direct evaluation of conditional Bayes factors and subsequent resampling. Further, methods for avoiding prior normalizing constant calculations–a serious bottleneck and source of numerical instability–have been proposed. We review and clarify these developments and then propose a new methodology for GGM comparison that blends many recent themes. Theoretical developments and computational timing experiments reveal an algorithm that has limited computational demands and dramatically improves on computing times of existing methods. We conclude by developing a parsimonious multivariate stochastic volatility model that embeds GGM uncertainty in a larger hierarchical framework. The method is shown to be capable of adapting to swings in market volatility, offering improved calibration of predictive distributions.

Keywords and phrases: Gaussian graphical models, G-Wishart distribution, conditional Bayes factors, exchange algorithms.

Received May 2012.

Contents

*Corresponding Author

1. Introduction

The Gaussian graphical model (GGM) has received widespread consideration (see Jones et al., 2005) and estimators obeying graphical constraints in standard Gaussian sampling were proposed as early as Dempster (1972). Initial incorporation of GGMs in Bayesian estimation has largely focused on decomposable graphs (Dawid and Lauritzen, 1993), since prior distributions factorize into products of Wishart distributions. Roverato (2002) proposes a generalized extension of the Hyper-Inverse Wishart distribution for covariance matrices $\mathbf{\Sigma}$ over arbitrary graphs. Atay-Kayis and Massam (2005) turn this into a prior specified for precision matrices $\boldsymbol{K}$ and outline a Monte Carlo (MC) method that enables pairwise model comparisons. Following Letac and Massam (2007) and Rajaratnam et al. (2008), Lenkoski and Dobra (2011) term this distribution the G-Wishart, and propose computational improvements to direct model comparison and model search.

A number of samplers for precision matrices under a G-Wishart distribution have been proposed. These involve either block Gibbs sampling (Piccioni, 2000), Metropolis-Hastings (MH) moves (Mitsakakis et al., 2011; Dobra and Lenkoski, 2011; Dobra et al., 2011), or rejection sampling (Wang and Carvalho, 2010). Dobra et al. (2011) show that the rejection sampler of Wang and Carvalho (2010) suffers from extremely low acceptance probabilities in even moderate dimensions. Wang and Li (2012) conclusively show that block Gibbs sampling is both computationally more efficient and exhibits considerably less autocorrelation than the MH methods.

The block Gibbs sampler provides a Markov chain Monte Carlo (MCMC) sample. When the likelihood assumes standard Gaussian sampling, determining posterior expectations of $\boldsymbol{K}$ can technically be performed as in Lenkoski and Dobra (2011), whereby model probabilities are first directly assessed via stochastic search, and model averaged samples are then collected using block

Gibbs over each model. However, when the GGM is specified over latent data in a hierarchical Bayesian framework, such an approach is no longer valid. This is due to the use of the matrix $\boldsymbol{K}$ in updating other hyperparameters as well as its involvement in updating the latent Gaussian factors.

Dobra and Lenkoski (2011) propose a reversible jump MCMC (which for brevity we refer to as RJ) method (Green, 1995) that simultaneously updates the GGM and its associated $\boldsymbol{K}$, and embed the GGM in a semiparametric Gaussian copula. Dobra et al. (2011) expand the RJ algorithm and show how GGMs may be used to model dependent random effects in a generalized linear model context, focusing on lattice data. Wang and Li (2012) utilize conditional properties of G-Wishart variates that enables model moves through calculation of a conditional Bayes factor (CBF) (Dickey and Gunel, 1978) and subsequently update $\boldsymbol{K}$ through direct Gibbs sampling. Wang and Li (2012) also explore the use of a double MH algorithm (Liang, 2010) to avoid the computationally expensive and numerically unstable MC approximation of normalizing constants proposed by Atay-Kayis and Massam (2005).

We continue this departure from RJ and investigate an alternative method for simultaneously updating the GGM and associated $\boldsymbol{K}$ in hierarchical Bayesian settings. Our method is built on the framework outlined in Wang and Li (2012), but uses an alternative representation of the CBF with considerably less computational cost.

Simulation experiments compare our new algorithm to the algorithm of Wang and Li (2012) (which we refer to as WL). Both methods perform equally well at determining posterior distributions. However, we show that while the WL approach is theoretically appealing, it suffers significant computational overhead on account of many matrix inversions. By contrast, our new approach exhibits a dramatic improvement in computation time. Further links made to recent work in Gaussian Markov random fields (Rue, 2001) allow for additional improvement.

We conclude with an example of how GGMs may be embedded in hierarchical Bayesian models. GGMs have been shown to yield parsimonious joint distributions useful in financial applications (Carvalho and West, 2007; Rodriguez et al., 2011; Wang et al., 2011) and we develop a multivariate analogue of a common stochastic volatility model (Jacquier et al., 1994) that embeds GGM uncertainty. Our proposal differs from the dynamic linear model context in that we model an exponential term that represents market volatility. This latent Gaussian factor requires a strategy for hierarchically estimating the GGM and associated precision term, which are used to subsequently update the volatility process. We show that our method is able to obtain sharper posterior distributions than a commonly employed alternative and remains robust throughout the financial crash associated with the collapse of Lehman Brothers.

The article is organized as follows. In Section 2 we review the G-Wishart distribution, establish results necessary for CBF calculations and describe the block Gibbs sampler. Section 3 conducts two simulation studies showing the computational advantage gained by our new algorithm. In Section 4 we describe our multivariate graphical stochastic volatility model and give results over the data mentioned above. We conclude in Section 5.

2. The G-Wishart distribution

2.1. Review of basic G-Wishart properties

Suppose that we collect data $\mathcal{D} = \{\boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(n)}\}$ such that $\boldsymbol{Z}^{(j)} \sim \mathcal{N}_p(0, \boldsymbol{K}^{-1})$ independently for $j \in \{1, \ldots, n\}$, where $\boldsymbol{K} \in \mathbb{P}$, the space of $p \times p$ positive definite matrices. This sample has likelihood

$$pr(\mathcal{D}|\boldsymbol{K}) = (2\pi)^{-np/2}|\boldsymbol{K}|^{n/2} \exp\left(-\frac{1}{2}\langle \boldsymbol{K}, \boldsymbol{U}\rangle\right),$$

where $\langle A, B\rangle = tr(A'B)$ denotes the trace inner product and $\boldsymbol{U} = \sum_{i=1}^{n} \boldsymbol{Z}^{(i)}\boldsymbol{Z}^{(i)'}$.

Further suppose that $G = (V, E)$ is a GGM where $V = \{1, \ldots, p\}$ and $E \subset V \times V$. We will slightly abuse notation throughout, by writing $(i, j) \in G$ to indicate that the edge (i, j) is in the edge set E. Associated with G is a subspace $\mathbb{P}_G \subset \mathbb{P}$ such that $\boldsymbol{K} \in \mathbb{P}_G$ implies that $\boldsymbol{K} \in \mathbb{P}$ and $K_{ij} = 0$ whenever $(i, j) \notin G$. The G-Wishart distribution (Roverato, 2002; Atay-Kayis and Massam, 2005) $\mathcal{W}_G(\delta, \boldsymbol{D})$ assigns prior probability to $\boldsymbol{K} \in \mathbb{P}_G$ as

$$pr(\boldsymbol{K}|\delta, \boldsymbol{D}, G) = \frac{1}{I_G(\delta, \boldsymbol{D})}|\boldsymbol{K}|^{(\delta-2)/2} \left(-\frac{1}{2}\langle \boldsymbol{K}, \boldsymbol{D}\rangle\right) \mathbf{1}_{\boldsymbol{K}\in\mathbb{P}_G}.$$

The normalizing constant I_G is in general not known to have an explicit form, and Atay-Kayis and Massam (2005) propose an MC approximation for this factor. Furthermore, the G-Wishart is conjugate and thus $pr(\boldsymbol{K}|\mathcal{D}, G) = \mathcal{W}_G(\delta + n, \boldsymbol{D}^*)$ where $\boldsymbol{D}^* = \boldsymbol{D} + \boldsymbol{U}$.

Let $\boldsymbol{\Phi}$ be the upper triangular matrix such that $\boldsymbol{\Phi}'\boldsymbol{\Phi} = \boldsymbol{K}$, the Cholesky decomposition. Rue (2001) notes that we may associate with G another graph F, called the *fill-in* graph, such that $G \subset F$, $\Phi_{ij} = 0$ when $(i, j) \notin F$ and

$$\Phi_{ij} = -\frac{1}{\Phi_{ii}} \sum_{l=1}^{i-1} \Phi_{li}\Phi_{lj} \tag{1}$$

when $i < j$ and $(i, j) \in F \setminus G$. Rue (2001) outlines a straightforward method for constructing a graph F from G, which is referred to as a symbolic Cholesky factorization. As noted by Rue (2001), use of node reordering software, which aims to reduce $F \setminus G$, can lead to additional reduction in computing time. In what follows, we use the `C` library `AMD` (Amestoy et al., 2004) to perform symbolic Cholesky factorizations and node reorderings.

Roverato (2002) shows that if $K \sim \mathcal{W}_G(\delta, \boldsymbol{D})$ then

$$pr(\boldsymbol{\Phi}|\delta, \boldsymbol{D}, G) \propto \prod_{i=1}^{p} \Phi_{ii}^{\delta+\nu_i^G-1} \exp\left(-\frac{1}{2}\langle \boldsymbol{\Phi}'\boldsymbol{\Phi}, \boldsymbol{D}\rangle\right), \tag{2}$$

where ν_i^G is the number of nodes in $\{i + 1, \ldots, p\}$ that are connected to node i in G.

Dobra et al. (2011) and Wang and Li (2012) contain detailed reviews of the many methods that have been proposed for sampling $\mathcal{W}_G(\delta, \boldsymbol{D})$ variates. We briefly outline the block Gibbs sampler, originally discussed by Piccioni (2000). Let $\mathcal{C}$ denote the cliques of G. In the following, we consider a clique to be a maximally complete subgraph, though Wang and Li (2012) note that this can be relaxed to any complete subgraph. Piccioni (2000) shows that for any $C \in \mathcal{C}$,

$$\boldsymbol{K}_C - \boldsymbol{K}_{C,V\setminus C}\boldsymbol{K}_{V\setminus C}^{-1}\boldsymbol{K}_{V\setminus C,C} \sim \mathcal{W}(\delta, \boldsymbol{D}_C), \tag{3}$$

where $\mathcal{W}$ denotes a standard Wishart variate. The expression (3) thereby gives the full conditionals of $\mathcal{W}_G(\delta, \boldsymbol{D})$. The block Gibbs sampler thus cycles through $\mathcal{C}$, updating each component using (3). Wang and Li (2012) convincingly show that for posterior inference of $\mathcal{W}_G(\delta + n, \boldsymbol{D}^*)$ the block Gibbs sampler outperforms all other proposed methods, both in terms of computing time and mixing. The authors also provide a useful review of the algorithm and indicate its broad flexibility. Throughout, we use the block Gibbs sampler for updating the matrix $\boldsymbol{K}$.

2.2. Conditional Bayes factors

Prior to Wang and Li (2012), model moves between two graphs G and G' focused on approximating the ratio

$$\frac{pr(G|\mathcal{D})}{pr(G'|\mathcal{D})} = \frac{pr(\mathcal{D}|G)}{pr(\mathcal{D}|G')} \times \frac{pr(G)}{pr(G')}, \tag{4}$$

first through MC (Atay-Kayis and Massam, 2005; Jones et al., 2005), then a combination of MC and Laplace approximation (Lenkoski and Dobra, 2011) and ultimately through RJ (Dobra and Lenkoski, 2011; Dobra et al., 2011).

Suppose that $G \subset G'$ which differ only by the edge $e = (i, j) \in G'$ and that $\boldsymbol{K} \in \mathbb{P}_G$. Let $\boldsymbol{K}^{-e} = \boldsymbol{K} \setminus \{K_{ij}, K_{ji}, K_{jj}\}$. In lieu of (4), Wang and Li (2012) consider ratios of the form

$$\frac{pr(G|\boldsymbol{K}^{-e}, \mathcal{D})}{pr(G'|\boldsymbol{K}^{-e}, \mathcal{D})} = \frac{pr(\mathcal{D}, \boldsymbol{K}^{-e}|G)}{pr(\mathcal{D}, \boldsymbol{K}^{-e}|G')} \times \frac{pr(G)}{pr(G')} \tag{5}$$

which are related to the conditional Bayes factors (CBFs) of Dickey and Gunel (1978).

Using properties related to the form (3) Wang and Li (2012) show that

$$\frac{pr(\mathcal{D}, \boldsymbol{K}^{-e}|G)}{pr(\mathcal{D}, \boldsymbol{K}^{-e}|G')} = H(\delta + n, e, \boldsymbol{K}^{-e}, \boldsymbol{D}^*)\frac{I_G(\delta, \boldsymbol{D})}{I_{G'}(\delta, \boldsymbol{D})} \tag{6}$$

where, in general

$$H(d, e, \boldsymbol{K}^{-e}, \boldsymbol{S}) = \frac{I(d, S_{jj})}{J(d, \boldsymbol{S}_{ee}, A_{11})}\left(\frac{|\boldsymbol{K}^0_{V\setminus j}|}{|\boldsymbol{K}^1_{V\setminus e}|}\right)^{(d-2)/2} \exp\left(-\frac{1}{2}\langle \boldsymbol{S}, \boldsymbol{K}^0 - \boldsymbol{K}^1\rangle\right)$$

where $I(b,c) = c^{-b/2}2^{b/2}\Gamma(b/2)$,

$$J(h,\boldsymbol{B},b) = \left(\frac{2\pi}{B_{22}}\right)^{1/2} b^{\frac{(h-1)}{2}} I(h,B_{22}) \exp\left(-\frac{b}{2}\left[B_{11} - \frac{B_{12}^2}{B_{22}}\right]\right),$$

such that $\boldsymbol{A} = \boldsymbol{K}_{ee} - \boldsymbol{K}_{e,V\setminus e}\boldsymbol{K}_{V\setminus e}^{-1}\boldsymbol{K}_{V\setminus e,e}$. The matrix $\boldsymbol{K}^0$ is equal to $\boldsymbol{K}$ except that $K^0_{jj} = \boldsymbol{K}_{j,V\setminus j}\boldsymbol{K}^{-1}_{V\setminus j}\boldsymbol{K}_{V\setminus j,j}$ and $K^0_{ij} = K^0_{ji} = 0$. Finally, the matrix $\boldsymbol{K}^1$ is equal to $\boldsymbol{K}$ except that $\boldsymbol{K}^1_e = \boldsymbol{K}_{e,V\setminus e}\boldsymbol{K}^{-1}_{V\setminus e}\boldsymbol{K}_{V\setminus e,e}$.

By using the CBF in (6), Wang and Li (2012) propose model moves that do not rely on RJ methods, and after assessing which graph to move to, the parameter K_{jj}, as well as K_{ij} if e is in the accepted graph, are resampled according to their conditional distributions given $\boldsymbol{K}^{-e}$. This method is appealing, as it offers an automatic manner of moving between graphs and does not rely on the tuning parameters used in the RJ methods of Dobra and Lenkoski (2011) and Dobra et al. (2011).

While the result has significant theoretical appeal we show that computation of the factor $H(\delta + n, e, \boldsymbol{K}^{-e}, \boldsymbol{D}^*)$ is extremely costly, even in low dimensions. This is due to the formation of the matrices $\boldsymbol{K}^0$ and $\boldsymbol{K}^1$, which require the solution of systems involving large matrices, in particular, $\boldsymbol{K}_{V\setminus j}$ and $\boldsymbol{K}_{V\setminus e}$.

Suppose now that G and G' differ only by the edge $f = (p-1,p)$ again with $G \subset G'$. We consider the CBF

$$\frac{pr(\mathcal{D},\boldsymbol{\Phi}^{-f}|G')}{pr(\mathcal{D},\boldsymbol{\Phi}^{-f}|G)},$$

where $\boldsymbol{\Phi}^{-f} = \boldsymbol{\Phi} \setminus \{\Phi_{p-1,p}, \Phi_{pp}\}$. In Appendix A we show that

$$\frac{pr(\mathcal{D}|\boldsymbol{\Phi}^{-f},G')}{pr(\mathcal{D}|\boldsymbol{\Phi}^{-f},G)} = N(\boldsymbol{\Phi}^{-f},\boldsymbol{D}^*)\frac{I_G(\delta,\boldsymbol{D})}{I_{G'}(\delta,\boldsymbol{D})}, \tag{7}$$

with, in general

$$N(\boldsymbol{\Phi}^{-f},\boldsymbol{S}) = \Phi_{p-1,p-1}\left(\frac{2\pi}{S_{pp}}\right)^{1/2}\exp\left(\frac{1}{2}S_{pp}(\phi_0-\mu)^2\right)$$

where $\mu = \Phi_{p-1,p-1}S_{p-1,p}/S_{pp}$, and $\phi_0 = -\Phi^{-1}_{p-1,p-1}\sum_{l=1}^{p-2}\Phi_{lp-1}\Phi_{lp}$.

This result originally appeared in an early version of Wang and Li (2012). In order to update a general edge e, we propose determining a permutation ς of V such that the nodes of $V\setminus e$ are reordered to reduce the fill-in of the graph $G_{V\setminus e}$ and finally, the edge e is placed in the $(p-1,p)$ position. Equation (7) is then calculated after permuting $\boldsymbol{K}$ and $\boldsymbol{D}^*$ according to ς.

The benefit of this method is the reduced computational overhead required to compute (7). The method requires merely relabeling the matrices $\boldsymbol{K}$ and $\boldsymbol{D}^*$ and determining the Cholesky decomposition of the permuted version of $\boldsymbol{K}$.

2.3. Avoiding normalizing constant calculation

Both (6) and (7) require determination of the prior normalizing constants I_G and $I_{G'}$. While the MC method of Atay-Kayis and Massam (2005) enables these factors to be approximated, the routine can be subject to numerical instability (Lenkoski and Dobra, 2011; Wang and Li, 2012) and involves significant computational effort.

Wang and Li (2012) propose a method for avoiding the use of the MC approximation for prior normalizing constants. Their method employs the double MH algorithm of Liang (2010), which is an extension of the exchange algorithm developed by Murray et al. (2006).

We briefly review the implementation of the double MH algorithm in Wang and Li (2012). Suppose that $(\boldsymbol{K}, G)$ is the current state of the MCMC chain and we propose to move to G' by adding the edge e to G. The double MH algorithm then forms a copy $\tilde{\boldsymbol{K}}$ of $\boldsymbol{K}$, resamples $\tilde{K}_{ij}$ and $\tilde{K}_{jj}$ according to G'. It then updates $\tilde{\boldsymbol{K}}$ via block Gibbs according to $\mathcal{W}_{G'}(\delta, \boldsymbol{D})$. Equation (6) is then replaced with

$$\frac{H(\delta + n, e, \boldsymbol{K}^{-e}, \boldsymbol{D}^*)}{H(\delta, e, \tilde{\boldsymbol{K}}^{-e}, \boldsymbol{D})} \tag{8}$$

We see that the expression (8) has replaced the prior normalizing constants with an evaluation of H in the prior, evaluated at $\tilde{\boldsymbol{K}}$ (see Murray et al., 2006; Liang, 2010, for theoretical justifications of this procedure). This is clearly beneficial, as it avoids the need for involved MC approximation.

We follow this procedure with our revised version of the CBF calculation. Again suppose that $(\boldsymbol{K}, G)$ is the current state and we propose to move to G' by adding the edge $f = (p-1, p)$. We first determine $\boldsymbol{\Phi}$ from $\boldsymbol{K}$. We also run an iteration of the block Gibbs sampler relative to $\mathcal{W}_{G'}(\delta, \boldsymbol{D})$ starting from $\boldsymbol{K}$, with the cliques of G ordered in such a manner that the $\boldsymbol{K}_f$ subblock is updated first. We then extract the matrix $\tilde{\boldsymbol{\Phi}}$ from this update. Equation (7) is then replaced with

$$\frac{N(\boldsymbol{\Phi}^{-f}, \boldsymbol{D}^*)}{N(\tilde{\boldsymbol{\Phi}}^{-f}, \boldsymbol{D})} \tag{9}$$

The expression (9) requires less computation than (8) as only Cholesky decompositions are used.

2.4. Algorithms for full posterior determination

In this section we outline the two algorithms we will consider for full posterior determination. Both algorithms create a sequence $\{(\boldsymbol{K}^{[1]}, G^{[1]}), \ldots, (\boldsymbol{K}^{[S]}, G^{[S]})\}$ where $\boldsymbol{K}^{[s]} \in \mathbb{P}_{G^{[s]}}$. Given the current state $(\boldsymbol{K}^{[s]}, G^{[s]})$ the WL algorithm proceeds as follows

0. Set $\boldsymbol{K} = K^{[s]}$ and $G = G^{[s]}$

1. For each edge e, do:
   a. if $e \notin G$ attempt to update G to $G' = G \cup e$ with probability
   $$\frac{q(G'|\boldsymbol{K}^{-e}, \mathcal{D})}{q(G|\boldsymbol{K}^{-e}, \mathcal{D})} = \frac{pr(G')H(\delta + n, e, \boldsymbol{K}^{-e}, D^*)}{pr(G)}$$
   if $e \in G$ the ratio is flipped. If G is not to be updated, skip to step c.
   b. If attempting to update G to G', sample $\tilde{\boldsymbol{K}}$ as discussed in Section 2.3 and calculate
   $$\alpha = \min\{1, H^{-1}(\delta, e, \tilde{\boldsymbol{K}}, \boldsymbol{D})\}$$
   if $e \in G'$, otherwise calculate
   $$\alpha = \min\{1, H(\delta, e, \tilde{\boldsymbol{K}}, \boldsymbol{D})\}$$
   and with probability α set $G = G'$, otherwise leave it unchanged.
   c. Resample K_{ij}, K_{jj} according to G.

   After attempting to update each edge, set $G^{[s+1]} = G$.
2. Resample $\boldsymbol{K}^{[s+1]}$ using the block Gibbs sampler relative to $G^{[s+1]}$ and the current state of $\boldsymbol{K}$.

We see that in one iteration of the WL algorithm, each edge is potentially updated in the graph. Our new algorithm (which we call CL) also follows this idea, and proceeds as follows

0. Set $\boldsymbol{K} = \boldsymbol{K}^{[s]}$ and $G = G^{[s]}$
1. For each edge e, do:
   a. Randomly selection a permutation ς of V_p, which places the edge e in the $(p-1, p) = f$ position, and likewise permute $\boldsymbol{K}$, G, $\boldsymbol{D}$ and $\boldsymbol{D}^*$. Let G^ς denote the permuted version of G and $\boldsymbol{\Phi}$ be the Cholesky decomposition of the permuted version of $\boldsymbol{K}$. If $f \notin G^\varsigma$ attempt to update G^ς to $G' = G^\varsigma \cup f$ with probability
   $$\frac{q(G'|\boldsymbol{\Phi}^{-f}, \mathcal{D})}{q(G^\varsigma|\boldsymbol{\Phi}^{-f}, \mathcal{D})} = \frac{pr(G')N(\boldsymbol{\Phi}^{-f}, D^*)}{pr(G^\varsigma)}$$
   if $f \in G^\varsigma$ the ratio is flipped. If G^ς is not to be updated then, skip to step c.
   b. If attempting to update G^ς to G', run the block Gibbs sampler over the permuted $\boldsymbol{K}$ relative to $\mathcal{W}_{G'}(\delta, \boldsymbol{D})$ and form $\tilde{\Phi}^{-f}$. Then calculate
   $$\alpha = \min\{1, N^{-1}(\tilde{\boldsymbol{\Phi}}^{-f}, \boldsymbol{D})\}$$
   if $f \in G'$, otherwise calculate
   $$\alpha = \min\{1, N(\tilde{\boldsymbol{\Phi}}^{-f}, \boldsymbol{D})\}$$
   and with probability α set $G^\varsigma = G'$, otherwise leave it unchanged.

c. Resample $\Phi_{p-1,p}, \Phi_{pp}$ according to G^{ς}. Then reform $\boldsymbol{K}$ and G by unpermuting the system.

After attempting to update each edge, set $G^{[s+1]} = G$.

2. Resample $\boldsymbol{K}^{[s+1]}$ using the block Gibbs sampler relative to $G^{[s+1]}$ and the current state of $\boldsymbol{K}$.

As we can see, there is somewhat more bookkeeping involved in the implementation of the CL algorithm, as the system is constantly being permuted. However, the reduction in computation time by the use of expression (9) and requiring only the calculation of the factors $N(\boldsymbol{\Phi}^{-f}, \boldsymbol{D}^*)$ and $N(\tilde{\boldsymbol{\Phi}}^{-f}, \boldsymbol{D})$ is considerable, as we show below.

3. Simulation study

In this section we conduct two simulation studies. The first shows, in a relatively small example, that both the WL and CL algorithms yield correct answers, however the CL algorithm takes less computing time. The second example shows how these approaches scale with dimension, both for sparse and dense true underlying graphs.

3.1. First simulation study

We conduct a simulation study that compares the method we have developed to the WL algorithm. Our example comes directly from Wang and Li (2012). We consider a situation in which $p = 6$ and let $\boldsymbol{U} = \boldsymbol{Y}\boldsymbol{Y}' = nA^{-1}$ where $n = 18$ and $A_{ii} = 1$ for $i = 1, \ldots, 6; A_{i,i+1} = A_{i+1,i} = .5$ for $i = 1, \ldots, 5$ and $A_{16} = A_{61} = .4$. We finally assume the prior $\boldsymbol{K} \sim \mathcal{W}_G(3, \mathbb{I}_6)$. Using exhaustive MC approximation of the entire graph space, Wang and Li (2012) show that the posterior probability of each edge is

$$(p_{ij}|A) = \begin{pmatrix} 1 & 0.969 & 0.106 & 0.085 & 0.113 & 0.85 \\ 0.969 & 1 & 0.98 & 0.098 & 0.081 & 0.115 \\ 0.106 & 0.98 & 1 & 0.982 & 0.0098 & 0.086 \\ 0.085 & 0.098 & 0.982 & 1 & 0.98 & 0.106 \\ 0.113 & 0.081 & 0.098 & 0.98 & 1 & 0.97 \\ 0.85 & 0.115 & 0.086 & 0.106 & 0.97 & 1 \end{pmatrix}$$

We use this example and compare the CL algorithm to the WL algorithm. We run two different versions of the CL algorithm: CL simple is a straightforward implementation that does not reorder nodes when computing Cholesky decompositions, beyond moving the edge to be updated to the end of the graph, nor does this compute the fill-in graph F, and therefore relies only on the full Cholesky decomposition. CL Fill-in, does compute node reorderings and the fill-in graph F and uses this as guidance when computing Cholesky factorizations.

TABLE 1
Comparison of CL and WL algorithms for the six dimensional example

	Time (sec)		MSE	
	Mean	SD	Mean	SD
CL Simple	200.5	(5.1)	0.0088	(6e-04)
CL Fill-in	202.1	(5.2)	0.0088	(6e-04)
WL	818.4	(19.2)	0.0349	(0.0025)

Following Wang and Li (2012) we run both the WL and the two CL algorithms as described in Section 2.4 for $60,000$ iterations and discard the first $10,000$ iterations as burn-in. All algorithms were implemented in `R`, though `C++` was used for block-Gibbs updates. The `C` library `AMD` of Amestoy et al. (2004) is called (from `R`) to perform reorderings.

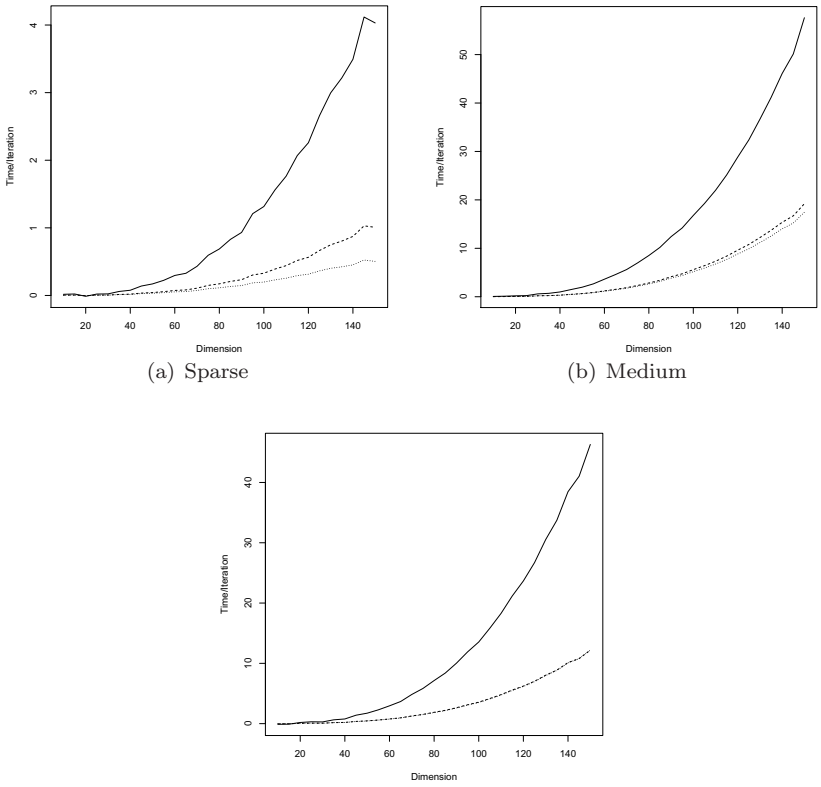
We record the total computing time and looked at the mean squared errors of the posterior inclusion probabilities from these runs compared with the true values given above. We repeated the entire process 100 times, each time starting both WL and CL from the same random starting point. Table 1 shows the average computing time in seconds (on a 2.8 GHz desktop computer with 4GB of RAM running Linux), average MSE and standard deviations across the 100 runs. The first column shows the expected result: even in six dimensions the WL algorithm takes more than 4 times as long to perform the same number of iterations as the CL algorithms. This shows the improved efficiency of the proposed method. Furthermore, we actually see that in this low dimensional example, the additional work put into forming sparse Cholesky decompositions is largely unhelpful. Indeed, computing times are slightly higher, due to the overhead involved in using these additional routines.

We found the results in the third column surprising, but do not draw broad conclusions from it. It appears that in this example, using $60,000$ iterations, the CL algorithm approaches the true posterior edge expectation more quickly than the WL algorithm. Since both algorithms are correct theoretically, we choose not to emphasize this result. Furthermore, we have determined that by doubling the number of iterations, both approaches yield essentially the exact posterior distribution, though again the WL algorithm takes almost 4 times as long to run.

This example was chosen as it appears in Wang and Li (2012) and has an exact answer. The fact that the CL algorithms are faster than the WL approach even in 6-dimensions indicates the broader appeal for searching truly high dimensional spaces.

3.2. Second simulation study

This study is conducted to determine how the computing time for the CL and WL algorithms scales as dimension increases. To do this, we consider the following framework. For a fixed dimension p, we first sample a graph G where the probability of an edge being in the graph is a given θ. We then form a matrix $\Phi_{ii} = 1/i$ for $i \in \{1, \ldots, p\}$ and $\Phi_{ij} = -0.5$ for those $(i, j) \in G, i < j$. The matrix $\boldsymbol{\Phi}$ is then completed so that $\boldsymbol{K} = \boldsymbol{\Phi}'\boldsymbol{\Phi} \in \mathbb{P}_G$. We finally sample



(a) Sparse (b) Medium

(c) Dense

FIG 1. *Timing comparison of WL algorithm (solid line), CL simple (dashed line) and CL fill-in (dotted line) for varying degrees of sparsity.*

$\mathcal{D} = \{\boldsymbol{Z}^{(1)}, \ldots, \boldsymbol{Z}^{(3p)}\}$ independently from a $\mathcal{N}_p(\boldsymbol{0}, \boldsymbol{K}^{-1})$. The WL and the two CL algorithms are then run for $50,000$ iterations using these data.

We consider three scenarios, when $\theta = .05$ (Sparse), $\theta = .5$ (Medium) and $\theta = .9$ (Dense) and consider p in 5 unit increments from 10 to 150. For each setting of p and θ, the process is repeated 25 times. Computation was performed in parallel on a 400 core server with 3.2 gHz Xeon chips and 96 GB of RAM, running Linux. The entire experiment took two weeks to conduct. Figure 1 shows the average computing time (in seconds) per iteration of the WL and the two CL algorithms under these settings.

The key feature of these results is that the WL appears to be consistently 4 times as slow as the CL alternatives. It seems reasonable to have a linear improvement, as all algorithms should technically be $O(p^3)$ in computational complexity. The difference is therefore largely the additional work involved in computing the CBF in the WL setting.

Further, for dense graphs, there is no discernable difference between the different CL algorithms. However, in the Medium and Sparse cases, we begin to

see the more involved CL algorithm contributing some additional improvement over the CL full algorithm. Indeed, in 150 dimensions, in the sparse graphs case, the CL full method is twice as fast as the CL simple.

The main purpose of this section is to highlight that the CBF calculation involved in the CL algorithm yields the majority of computational improvement over the WL algorithm and that this improvement holds as dimension increases and regardless of graph density. Further, when the graphs under consideration are very sparse, use of the fill-in graph F and node reordering software offer additional increase in performance, while the computational overhead is negligible when the graph is dense.

While the overall trends are the same in the three density scenarios, we note that that the per-iteration computing time is longest in the Medium density case. This is because of the block Gibbs sampler which operates on the cliques $\mathcal{C}$ of the graph. Medium sized graphs tend to have more cliques than dense or sparse graphs. Since this feature is shared by both the WL and CL algorithms, it does not affect their relative performance.

4. A multivariate graphical stochastic volatility model

Modeling the joint distribution of returns for a large number of assets is an important component of portfolio allocation and risk management. Carvalho and West (2007), Rodriguez et al. (2011) and Wang et al. (2011) all show that the use of GGMs can substantially improve modeling of joint asset returns. In each of these studies heterogeneity in asset returns was addressed either through the use of dynamic linear models with variance discounting (Carvalho and West, 2007; Wang et al., 2011) or infinite hidden Markov models (Rodriguez et al., 2011).

We consider an alternative approach, that models market volatility in a manner analogous to the stochastic volatility models discussed in Jacquier et al. (1994) which jointly determines the general graphical model along with all other parameters.

4.1. The stochastic volatility model

Let $\boldsymbol{Y}_t$ be the log-returns of p correlated assets. We specify the following hierarchical model for these returns:

$$\begin{aligned}
\boldsymbol{Y}_t|\boldsymbol{K}, X_t &\sim \mathcal{N}_p\left(\mathbf{0}, [\exp(X_t)\boldsymbol{K}]^{-1}\right) \\
X_t|\alpha, \phi, X_{t-1} &\sim \mathcal{N}(\alpha + \phi X_{t-1}, 1) \\
X_1 &= 0 \\
(\alpha\ \phi)' &\sim \mathcal{N}_2(0, \mathbb{I}_2)\mathbf{1}_{\phi\in(-1,1)} \\
\boldsymbol{K}|G &\sim \mathcal{W}_G(3, \mathbb{I}_p) \\
pr(G) &\propto 1.
\end{aligned} \tag{10}$$

In the likelihood (10) we see that asset returns are assumed to be mean-zero. Such an assumption is common in the absence of a detailed forecasting model. The X_t terms then dictate an overall level of market volatility, while a constant precision parameter $\boldsymbol{K}$ dictates the degree to which asset returns are correlated.

While this model is parsimonious, it serves as a useful first departure from previous studies as it explicitly incorporates notions of stochastic volatility and can be seen as a multivariate extension of the models outlined in Jacquier et al. (1994) and elsewhere. For purposes of model identification, we set $X_1 = 0$ as well as assume that the variance of the X_t is equal to 1. In the conclusions section we discuss further possible generalizations to this framework.

After collecting a time-series of returns $\boldsymbol{Y}^{(1:T)}$, we then aim to determine the posterior distribution

$$pr(\boldsymbol{K}, G, \alpha, \phi, \boldsymbol{X}|\boldsymbol{Y}^{(1:T)})$$

where $\boldsymbol{X} = (X_1, \ldots, X_T)$. Furthermore, we may be interested in the posterior predictive distribution $pr(\boldsymbol{Y}^{(T+1)}|\boldsymbol{Y}^{(1:T)})$. The full detail of our MCMC algorithm is outlined in Appendix B, but largely follows the guidelines discussed in Rue and Held (2005). The posterior predictive distribution of $\boldsymbol{Y}^{(T+1)}$ is easily formed from these parameters. However, we note in particular that

$$\boldsymbol{K}|G, \boldsymbol{X}, \boldsymbol{Y}^{(1:T)} \sim \mathcal{W}_G \left(\delta + T, \boldsymbol{D} + \sum_{t=1}^{T} \exp(X_t) \boldsymbol{Y}^{(t)} \boldsymbol{Y}^{(t)'} \right). \tag{11}$$

From (11) we see why the developments in Section 2 prove useful. We may update $\boldsymbol{K}$ and G jointly using the CL algorithm discussed in Section 2.4 simply by setting $\boldsymbol{D}^* = \boldsymbol{D} + \sum_{t=1}^{T} \exp(X_t) \boldsymbol{Y}^{(t)} \boldsymbol{Y}^{(t)'}$. This allows us to easily embed a sparse precision matrix $\boldsymbol{K}$ and mix over the class of GGMs in any hierarchical Bayesian model that involves a standard Wishart distribution.

In what follows we will compare our methodology to an alternative, discussed in Carvalho and West (2007) and extended using our methodology. The variance discounting approach assumes that for the log-returns $\boldsymbol{Y}^{(t+1)}$ the following evolution model holds

$$\begin{aligned} \boldsymbol{Y}^{(t+1)} &\sim \mathcal{N}_p(0, \boldsymbol{K}_{t+1}^{-1}) \\ \boldsymbol{K}_{t+1} &\sim \mathcal{W}_G(\vartheta \delta_t, \vartheta \boldsymbol{D}_t), \end{aligned}$$

where,

$$\begin{aligned} \delta_t &= \vartheta \delta_{t-1} + 1 \\ \boldsymbol{D}_t &= \vartheta \boldsymbol{D}_{t-1} + \boldsymbol{Y}_t \boldsymbol{Y}_t' \end{aligned}$$

for $0 << \vartheta < 1$ with $\delta_0 = 3$ and $\boldsymbol{D}_0 = \mathbb{I}_p$. We use this framework and mix over the graph space, the details of which are provided in Appendix C.

The variance discounting model falls into the category of dynamic linear models (West and Harrison, 1997) and works by rapidly discounting previous observations, thereby adapting quickly to swings in market behavior.

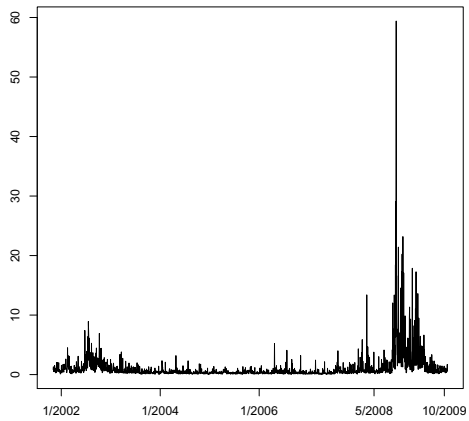


FIG 2. *Mean of the squared returns taken over all 20 stocks during the entire time period from October 31, 2001 to October 23, 2009.*

4.2. Description of data

To apply our model and algorithm we randomly chose 20 stocks from the S&P 500. Similar to the analysis of Wang et al. (2011) the stock were chosen at random to reduce the possibility of sampling bias in our results. These stocks were: Aetna Inc. (AET), CA Inc. (CA), Campbell Soup (CPB), CVS Caremark Corp. (CVS), Family Dollar Stores (FDO), Honeywell Int'l Inc. (HON), Hudson City Bancorp (HCBK), JDS Uniphase Corp. (JDSU), Johnson Controls (JCI), Morgan Stanley (MS), PPG Industries (PPG), Principal Financial Group (PFG), Sara Lee Corp. (SLE), Sempra Energy (SRE), Southern Co. (SO), Supervalu Inc. (SVU), Thermo Fisher Scientific (TMO), Wal-Mart Stores (WMT), Walt Disney Co. (DIS), Wellpoint Inc. (WLP).

We the collected data from time period between October 31, 2001 to October 23, 2009. Figure 2 shows the mean of the squared returns for the these 20 securities over the entire dataset. The extreme volatility present in the markets after the collapse of Lehman brothers in September 2008 is readily evident, showing that a homoskedasticity assumption is untenable for these data.

4.3. Model validation

In order to compare our stochastic volatility approach to that of variance discounting, we take the perspective verifying distributional forecasts (Gneiting and Raftery, 2007). This, in part, has to do with the challenging dynamics of the period that we consider. Previously, Carvalho and West (2007), Rodriguez et al. (2011) and Wang et al. (2011) were able to consider returns from optimal portfolios constructed according to variance minimization arguments in order to verify their methods' superior performance. However, portfolio optimization

typically requires an assumption on the (non-zero) mean return level of the assets under consideration.

We feel that typical assumptions (for instance Carvalho and West (2007) use the previous day's returns while Rodriguez et al. (2011) use the mean return level for the previous 50 days) that have been used in the absence of detailed professional return forecasts are unsuitable for the data collected over the time period we consider due to the extreme swings in asset value. Furthermore, our goal is to characterize the entire distribution of the next day's returns, instead of forming an investment rule, since such distributional forecasts could be used to subsequently inform investment decisions.

Let $s_t = \sum_{i=1}^{20} Y_{ti}$ be the sum of returns of the 20 assets at time point t and let $F^{(t)}$ be a given forecast distribution of s_t. We evaluate the competing methods using the continuous ranked probability score (CRPS) (see Gneiting and Raftery, 2007, for additional details of the properties of the CRPS) which is calculated as

$$CRPS(F^{(t)}, s_t) = \mathbb{E}_{F^{(t)}}|U - s_t|^2 - \frac{1}{2}\mathbb{E}_{F^{(t)}}|U - U'|^2 \tag{12}$$

where U and U' are independent random variates sampled according to $F^{(t)}$.

It has frequently been shown in the literature on distributional forecasting of weather, that using only a small amount of previous data can be beneficial (Raftery et al., 2005; Gneiting et al., 2005; Thorarinsdottir and Gneiting, 2010). The variance discounting method naturally embeds these considerations (this feature is shown quite clearly in our results section). For the stochastic volatility model we consider manually dropping data that come a given number of days before the forecast time $t + 1$.

Similar to what is done in Raftery et al. (2005) and Thorarinsdottir and Gneiting (2010) in the weather context, we isolate the first 700 days as pure training data, with the sole intention of determining the appropriate length window. We then consider the last 500 days in this set for which we make predictions. For each day in this period, we fit our stochastic volatility model using windows between 30 and 200 days in length, in increments of 10 days. For each window-length, we form a predictive distribution for s_{t+1} and compare this to the observed level using the CRPS. We then calculate the mean CRPS over these 500 days for each window length considered.

Figure 3 shows the mean CRPS for each window length under consideration over this period. Such U shaped results are common when training lead time, and we see that using somewhere between 80 to 140 days of data offers the best predictive performance. For both the stochastic volatility and variance discounting models, we therefore present two results. One set of results uses all data available before timepoint $t + 1$ for forming the predictive distribution of s_{t+1}, while the other uses just returns between time $t - 120$ and t.

After conducting our training exercise, we use the last 1800 days to validate the performance of our proposed methodologies. We compare the stochastic volatility model to variance discounting when $\vartheta = .97$ and $\vartheta = .99$. Table 2 shows the mean CRPS of the six methodologies. We see that the stochastic

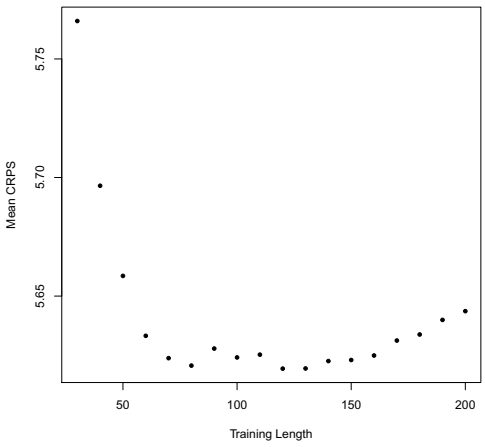


FIG 3. *Mean CRPS for different length of training period used to form the forecast of* s_{t+1}.

TABLE 2
Mean CRPS of predictive distributions taken over the validation period when the models are fit using all previous data and the last 120 *days*

	All Previous	Last 120
Stochastic Volatility	6.27	5.61
Variance Discounting $\vartheta = .99$	6.47	6.45
Variance Discounting $\vartheta = .97$	6.62	6.61

volatility model outperforms the variance discounting approaches in both situations. However, there is a considerably larger improvement if the forecast distribution for s_{t+1} is formed using the stochastic volatility model over only the previous 120 days. The variance discounting approaches, by contrast, are relatively agnostic to the use of a reduced sample to form predictions, since these models naturally down-weight previous observations. Our general intuition is that the stochastic volatility model is able to incorporate additional uncertainty over the variance discounting approach, and the CRPS rewards this.

In addition to the results regarding posterior predictive performance, we also provide some indication of the posterior inference provided by our stochastic volatility model, shown in Figure 4. The left panel of the Figure 4 shows the estimate of the X_t when using the entire dataset, thereby giving an indication the daily estimated latent volatility. Since these factors enter into the precision, lower levels indicate higher market volatility. We see the obvious increase in market volatility associated with the market crash of 2009. However, we also see a smaller, but non-negligible increase in volatility near the beginning of our dataset. We have determined that this period corresponds exactly to the American invasion of Iraq in 2003.

The right panel of Figure 4 shows the posterior distribution of the autocorrelation parameter ϕ for each day, when the model was fit using only the previous

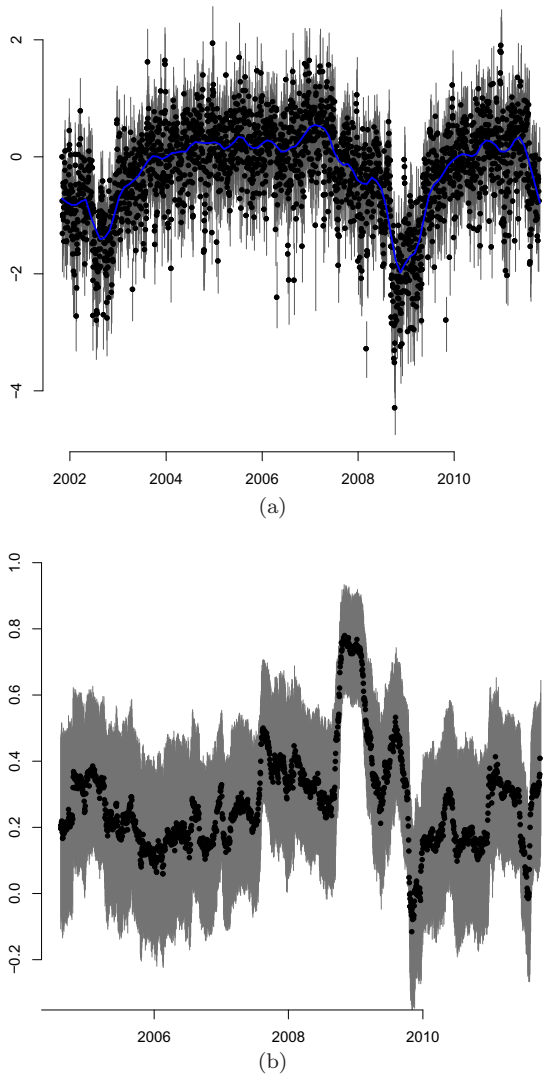


FIG 4. *The left panel shows the distribution of the latent volatility parameters, showing median value (black dot) and (.025,.975) posterior interval (gray lines) as well as a simple lowess line through the medians. The right panel shows the estimate of* ϕ *on each day using the previous* 120 *days of data.*

120 days of data. This figure also shows interesting dynamics. In particular, while ϕ is typically positive and between .2 and .6, during the recent financial crisis there appears to have been considerably greater autocorrelation, with ϕ hovering near .8 for a period. In our mind, this feature is what led to the use of a smaller window having considerably better predictive performance for the stochastic volatility model. By being able to quickly adjust the overall level of

the parameters, predictive distributions were able to focus more quickly on the changing dynamics throughout the period under consideration.

5. Conclusions

We have synthesized a number of recent results related to the G-Wishart distribution. This has allowed for an algorithm that does not rely on RJ methods, obviates the need for expensive and numerically unstable MC approximation of prior normalizing constants and does so with minimal computational effort. The improvement in computation time is sufficient that at each stage of the algorithm, all edges may be evaluated for inclusion or exclusion in the graphical model. This algorithm allows the GGM to be embedded in more sophisticated hierarchical Bayesian models and opens the possibility of replacing standard Wishart distributions with G-Wishart variates, leveraging the improvement in predictive performance offered by sparse precision matrices.

The applied example shows the usefulness of this combination. We are able to sparsely model the interactions in financial assets while simultaneously addressing the issues of stochastic volatility prevalent in markets undergoing turbulence, successfully characterizing the distribution of asset returns during periods of rapidly fluctuating volatility.

The stochastic volatility model we develop remains parsimonious and several adjustments could be made. The first such development would be to replace the univariate term X_t with a multivariate factor that allows the variance of each asset to follow its own path, while potentially tying the evolution of these factors together with a separate GGM. Furthermore, employing some form of the iHMM framework of Rodriguez et al. (2011) could allow for the matrix $\boldsymbol{K}$ to change throughout the period as well. Such developments will be considered in future work.

Furthermore, while the evaluation of GGMs outside of the RJ framework has proven to be a useful development, much work remains computationally. In particular, the growing prevalence of options of parallel computing, through multi-chip processors and graphics processing units has yet to be harnessed. Unfortunately, it is not yet clear how parallel computation can best be used in the confines of MCMC and further research is necessary to determine how these resources can best be leveraged in the hierarchical modeling context.

Acknowledgments

Both authors gratefully acknowledge partial support from the German Science Foundation (DFG), research grant GRK 1653. Alex Lenkoski's work is also funded by Statistics for Innovation $(\text{sfi})^2$, one of the 14 Norwegian Centers for Research-based Innovation. We thank Thordis Thorarinsdottir, Tilmann Gneiting, the editor, associate editor and anonymous referee for their helpful suggestions.

Appendix A: Determination of CBF for using $\boldsymbol{\Phi}^{-f}$

Consider

$$\frac{pr(\mathcal{D}, \boldsymbol{\Phi}^{-f}|G')}{pr(\mathcal{D}, \boldsymbol{\Phi}^{-f}|G)}$$

we note that

$$pr(\mathcal{D}, \boldsymbol{\Phi}^{-f}|G') = \int_{\Phi_{p-1,p}} \int_{\Phi_{pp}} pr(\mathcal{D}, \boldsymbol{\Phi}|G') d\boldsymbol{\Phi}_f$$

and

$$pr(\mathcal{D}, \boldsymbol{\Phi}^{-f}|G) = \int_{\Phi_{pp}} pr(\mathcal{D}, \boldsymbol{\Phi}|G) d\Phi_{pp}$$

up to common terms we thus have that

$$pr(\mathcal{D}, \boldsymbol{\Phi}^{-f}|G') \propto \frac{\Phi_{p-1,p-1}}{I_{G'}(\delta, \boldsymbol{D})} \int_{\Phi_{p-1,p}} \exp\left(-\frac{1}{2} D^*_{p,p}(\Phi_{p-1,p} + \mu)^2\right) d\Phi_{p-1,p}$$

recognizing the integral as the kernel of a normal distribution, this yields

$$pr(\mathcal{D}, \boldsymbol{\Phi}^{-f}|G') \propto \frac{\Phi_{p-1,p-1}}{I_{G'}(\delta, \boldsymbol{D})} \left(\frac{2\pi}{D^*_{pp}}\right)^{1/2}.$$

Further, again up to common terms

$$pr(\mathcal{D}, \boldsymbol{\Phi}^{-f}|G) \propto \frac{1}{I_G(\delta, \boldsymbol{D})} \exp\left(-\frac{1}{2} D^*_{p,p}(\Phi_0 + \mu)^2\right)$$

and thus

$$\frac{pr(\mathcal{D}, \boldsymbol{\Phi}^{-f}|G')}{pr(\mathcal{D}, \boldsymbol{\Phi}^{-f}|G)} = N(\boldsymbol{\Phi}^{-f}, D^*) \frac{I_G(\delta, \boldsymbol{D})}{I_{G'}(\delta, \boldsymbol{D})}$$

Appendix B: Posterior determination of the stochastic volatility model

We outline the posterior determination of

$$(\alpha, \phi, \boldsymbol{X}, \boldsymbol{K}, G)|\boldsymbol{Y}^{(1:T)}$$

which is broken into three steps

Step 1: Update $(\boldsymbol{K}, \boldsymbol{G})$

Conditional on $\boldsymbol{X}$ have that

$$\boldsymbol{K} \sim \mathcal{W}_G(\delta + T, \boldsymbol{D} + \sum \exp(X_t) \boldsymbol{Y}_t \boldsymbol{Y}'_Y)$$

and therefore we update $(\boldsymbol{K}, G)$ as a block using the CL algorithm discussed in Section 2.4

Step 2: Update α, ϕ

Let $\boldsymbol{V}$ be the $(T-1) \times 2$ matrix with 1 in the first column of each row and X_r in the second column of row r. Finally let $\boldsymbol{X}_{-1}$ be the length $T-1$ vector formed by dropping X_1 from $\boldsymbol{X}$. Then

$$(\alpha \ \phi)' \sim \mathcal{N}(\beta, \Omega^{-1})$$

where

$$\Omega = V'V + \mathbb{I}_2$$

and

$$\beta = \Omega^{-1}(\boldsymbol{V}\boldsymbol{X}_{-1})$$

Then the pair α, ϕ are updated by sampling from this distribution given the current state of $\boldsymbol{X}$. If $|\phi| > 1$, the proposal is rejected. In practice, this never occurred in the course of our study.

Step 3: Updating X

Updating the latent factors $\boldsymbol{X}$ constitutes the most challenging step in the algorithm. We largely follow the method outlined in Rue and Held (2005), pages 167-168.

Remember that for identification purposes we assume $X_1 = 0$. Then note that

$$X_j | X_{j-1}, X_{j+1}, \phi, \alpha \sim N(\alpha + \phi X_{j-1} + \phi X_{j+1}, 1/(1+2\phi^2)).$$

when $j \in \{1, \ldots, T\}$ while

$$X_T | X_{T-1}, \phi, \alpha \sim N(\alpha + \phi X_{j-1}, 1/(1+\phi^2))$$

and thus in general we write

$$X_j | X_{-j}, \phi, \alpha \sim N(\mu_j, \kappa_j^{-1})$$

and letting $r_j = \langle K, \boldsymbol{Y}_j \boldsymbol{Y}'_j \rangle$ we have that

$$pr(r_j | X_j) \propto \exp\left(-\frac{1}{2}\exp(X_j) r_j + \frac{p}{2} X_j\right) = \exp(f(X_j, r_j)).$$

Note that

$$f'(X_j) = \frac{df(X_j, r_j)}{dX_j} = -\frac{1}{2}\exp(X_j) r_j + \frac{p}{2}$$

and

$$f''(X_j) = \frac{d^2 f(X_j, r_j)}{dX_j^2} = -\frac{1}{2}\exp(X_j) r_j.$$

Following Rue and Held (2005), set

$$\begin{aligned} b(X_j) &= \mu_j + f'(X_j) - f''(X_j)X_j \\ &= \mu_j + \frac{p}{2} - \frac{r_j}{2}\exp(X_j) + \frac{r_j}{2}X_j\exp(X_j) \\ c(X_j) &= 1 + \frac{r_j}{2}\exp(X_j) \end{aligned}$$

and sample

$$X_j' \sim \mathcal{N}\left(\frac{b(X_j)}{c(X_j)}, \frac{1}{c(X_j)}\right).$$

The proposal is then accepted with probability $\min\{\alpha, 1\}$ where

$$\alpha = \frac{pr(r_j|X_j')pr(X_j'|\mu_j,\kappa_j)pr(X_j|b(X_j'),c(X_j'))}{pr(r_j|X_j)pr(X_j|\mu_j,\kappa_j)pr(X_j'|b(X_j),c(X_j))}.$$

Each X_j is updated in this fashion, in a random order to reduce dependence. We note that these updates require no pre-specified tuning parameter for proposal and further lead to between 80% and 90% acceptance probabilities of updates throughout our study.

Appendix C: Details for mixing over G in the variance discounting model

First, note that

$$\boldsymbol{K}_t|\boldsymbol{Y}_{(1:t)} \sim \mathcal{W}_G(\delta_t, \boldsymbol{D}_t).$$

Thus, for a given timepoint we run the CL algorithm for $50,000$ iterations, after a burn-in of $10,000$ iterations. For each graph G that appears in the output of the MCMC, a sample $\boldsymbol{Y}^{(t+1)}|\{\boldsymbol{Y}^{(1:t)}, G\}$ is obtained by running the block Gibbs sampler for 1000 iterations over the distribution $\mathcal{W}_G(\vartheta\delta_t, \vartheta\boldsymbol{D}_t)$. For each of the last 100 matrices $\boldsymbol{K}$ that are returned from this run, a sample

$$\boldsymbol{Y}^{(t+1)} \sim \mathcal{N}_p(0, \boldsymbol{K}^{-1})$$

is drawn and retained. By collecting these samples, the predictive distribution s_{t+1} is formed.

References

Amestoy, P. R., Davis, T. A., and Duff, I. S. (2004). Algorithm 837: AMD, an approximate minimum degree ordering algorithm. *ACM Transactions on Mathematical Software*, 30:381–388. MR2124398

Atay-Kayis, A. and Massam, H. (2005). A Monte Carlo method for computing the marginal likelihood in nondecomposable Gaussian graphical models. *Biometrika*, 92:317–335. MR2201362

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

Dawid, A. P. and Lauritzen, S. L. (1993). Hyper Markov laws in the statistical analysis of decomposable graphical models. *Ann. Statist.*, 21:1272–1317. MR1241267

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

Dickey, J. M. and Gunel, E. (1978). Bayes factors from mixed probabilities. *J. R. Statist. Soc. B*, 40:43–46. MR0512141

Dobra, A. and Lenkoski, A. (2011). Copula Gaussian graphical models and their application to modeling functional disability data. *Annals of Applied Statistics*, 5:969–993. MR2840183

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*, 106:1418–1433.

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction and estimation. *Journal of the American Statistical Association*, 102:359–378. MR2345548

Gneiting, T., Raftery, A. E., Westveld, A. H., and Goldman, T. (2005). Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation. *Monthly Weather Review*, 133:1098–1118.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(711-732). MR1380810

Jacquier, E., Polson, N. G., and Rossi, P. E. (1994). Bayesian analysis of stochastic volatility models. *Journal of Business and Economic Statistics*, 12:371–389.

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

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

Letac, G. and Massam, H. (2007). Wishart distributions for decomposable graphs. *Ann. Statist.*, 35:1278–323. MR2341706

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

Mitsakakis, N., Massam, H., and Escobar, M. D. (2011). A Metropolis-Hastings based method for sampling from the G-Wishart distribution in Gaussian graphical models. *Electronic Journal of Statistics*, 5:18–30. MR2763796

Murray, I., Ghahramani, Z., and MacKay, D. (2006). MCMC for doubly-intractable distributions. *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence*.

Piccioni, M. (2000). Independence structure of natural conjugate densities to exponential families and the Gibbs sampler. *Scand. J. Statist.*, 27:111–27. MR1774047

Raftery, A. E., Gneiting, T., Balabdaoui, F., and Polakowski, M. (2005). Using Bayesian model averaging to calibrate forecast ensembles. *Monthly Weather Review*, 133:1155–1174.

Rajaratnam, B., Massam, H., and Carvalho, C. M. (2008). Flexible covariance estimation in graphical Gaussian models. *Ann. Statist.*, 36:2818–2849. MR2485014

Rodriguez, A., Dobra, A., and Lenkoski, A. (2011). Sparse covariance estimation in heterogeneous samples. *Electronic Journal of Statistics*, 5:981–1014. MR2836767

Roverato, A. (2002). Hyper inverse Wishart distribution for non-decomposable graphs and its application to Bayesian inference for Gaussian graphical models. *Scand. J. Statist.*, 29:391–411. MR1925566

Rue, H. (2001). Fast sampling of Gaussian Markov random fields. *Journal of the Royal Statistical Society, Series B*, 63:325–338. MR1841418

Rue, H. and Held, L. (2005). *Gaussian Markov Random Fields*. Chapman & Hall. MR2130347

Thorarinsdottir, T. L. and Gneiting, T. (2010). Probabilistic forecasts of wind speed: Ensemble model output statistics using heteroskedastic censored regression. *Journal of the Royal Statistical Society Ser. A*, 173:371–388. MR2751882

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

Wang, H. and Li, S. Z. (2012). Efficient Gaussian graphical model determination under G-Wishart prior distributions. *Electronic Journal of Statistics*, 6:168–198. MR2879676

Wang, H., Reeson, C., and Carvalho, C. M. (2011). Dynamic financial index models: Modeling conditional dependencies via graphs. *Bayesian Analysis*, 6:639–664. MR2869960

West, M. and Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models*. Springer. MR1482232