A MCMC Approach for Learning the Structure of Gaussian Acyclic Directed Mixed Graphs

Ricardo Silva

Abstract Graphical models are widely used to encode conditional independence constraints and causal assumptions, the directed acyclic graph (DAG) being one of the most common families of models. However, DAGs are not closed under marginalization: that is, if a distribution is Markov with respect to a DAG, several of its marginals might not be representable with another DAG unless one discards some of the structural independencies. Acyclic directed mixed graphs (ADMGs) generalize DAGs so that closure under marginalization is possible. In a previous work, we showed how to perform Bayesian inference to infer the posterior distribution of the parameters of a given Gaussian ADMG model, where the graph is fixed. In this paper, we extend this procedure to allow for priors over graph structures.

Key words: Graphical models, Bayesian inference, sparsity.

1 Acyclic Directed Mixed Graph Models

Directed acyclic graphs (DAGs) provide a practical language to encode conditional independence constraints (see, e.g., [8]). However, such a family is not *closed under marginalization*. As an illustration of this concept, consider the following DAG:

$$Y_1 \rightarrow Y_2 \leftarrow X \rightarrow Y_3 \leftarrow Y_4$$

This model entails several conditional independencies. For instance, it encodes constraints such as $Y_2 \perp\!\!\!\perp Y_4$, as well as $Y_2 \not\perp\!\!\!\perp Y_4 \mid Y_3$ and $Y_2 \perp\!\!\!\perp Y_4 \mid \{Y_3, X\}$. Directed graphical models are non-monotonic independence models, in the sense that conditioning on extra variables can destroy and re-create independencies, as the sequence $\{\emptyset, \{Y_3\}, \{Y_3, X\}\}$ has demonstrated.

Ricardo Silva
University College London, Gower Street, London WC1E 6BT, e-mail: ricardo@stats.ucl.ac.uk

If X is a latent variable which we are not interested in estimating, there might be no need to model explicitly its relationship to the observed variables $\{Y_1, Y_2, Y_3, Y_4\}$ – a task which would require extra and perhaps undesirable assumptions.

However, marginalizing X results in a model that cannot be represented as a DAG structure without removing some of the known independence constraints. Since any constraint that conditions on X has to be dropped in the marginal for $\{Y_1, Y_2, Y_3, Y_4\}$ (for instance, $Y_2 \perp\!\!\!\perp Y_4 \mid \{Y_3, X\}$), we are forced to include extra edges in the DAG representation of the remaining variables. One possibility is $\{Y_1 \rightarrow Y_2 \leftarrow Y_3 \leftarrow Y_4, Y_4 \rightarrow Y_2\}$, where the extra edge $Y_4 \rightarrow Y_2$ is necessary to avoid constraints that we know should not hold, such as $Y_2 \perp\!\!\!\perp Y_4 \mid Y_3$. However, with that we lose the power to express known constraints such as $Y_2 \perp\!\!\!\perp Y_4$.

Acyclic directed mixed graphs (ADMGs) were introduced in order to provide independence models that result from marginalizing a DAG. ADMGs are *mixed* in the sense they contain more than one type of edge. In this case, *bi-directed* edges are also present. They are *acyclic* in the sense that there is no directed cycle composed of directed edges only. In principle, it is possible for two vertices to be linked by both a directed and a bi-directed edge. Moreover, let $sp(i)$ denote the "spouses" of Y_i in the graph (i.e., those Y_j such that $Y_i \leftrightarrow Y_j$ exists) and define $nsp(i)$ to be the non-spouses (Y_i is neither a spouse nor a non-spouse of itself).

In our example, the corresponding ADMG could be

$$Y_1 \rightarrow Y_2 \leftrightarrow Y_3 \leftarrow Y_4$$

Independences can be read off an ADMG using a criterion analogous to d-separation. More than one Markov equivalent ADMG can exist as the result of marginalizing a DAG (or marginalizing another ADMG). Moreover, other types of (non-independence) constraints can also result from an ADMG formulation if one allows two edges between two vertices. A detailed account of such independence models and a Gaussian parameterization are described at length by [9, 10]. Generalizations are discussed by [11]. An algorithm for maximum likelihood estimation in Gaussian ADMGs was introduced by [5]. A Bayesian method for estimating parameters of Gaussian ADMGs was introduced by [12]. In this paper, we extend [12] by allowing the ADMG structure to be estimated from data, besides the parameters. Section 2 reviews the Bayesian formulation of the problem while Section 3 describes a sampler for inferring structure. A simple demonstration is given in Section 4.

2 A Review of the Gaussian Parametrization and Priors

Given a ADMG $\mathscr{G}$ and a p-dimensional distribution $\mathscr{P}$, each random variable in the distribution corresponds to a vertex in the graph. Let Y_i be a vertex with parents $Y_{i[1]}, Y_{i[2]}, \ldots, Y_{i[M_i]}$, $1 \leq i \leq p$, $1 \leq i[j] \leq p$, $1 \leq j \leq M_i$. We define a set of parameters $\{\lambda_{ij}\}$ according to the regression equation $Y_i = \sum_{j=1}^{M_i} \lambda_{ij} Y_{i[j]} + \varepsilon_i$, where each error term ε_i is distributed as a zero mean Gaussian. Therefore, given the covariance ma-

trix $\mathbf{V}$ of the error terms, we have a fully specified zero-mean Gaussian distribution. The parameterization of $\mathbf{V}$ is given by a sparse positive definite matrix: if there is no bi-directed edge $Y_i \leftrightarrow Y_j$, then we define $(\mathbf{V})_{ij} \equiv v_{ij} \equiv 0$. The remaining entries are free parameters within the space of (sparse) positive definite matrices. Priors for such models were described by [12]. Priors for each λ_{ij} are defined as independent zero-mean Gaussians, which in our experiments were given a prior variance of 3. The prior for $\mathbf{V}$ is given by

$$\pi_{\mathscr{G}}(\mathbf{V}) \propto |\mathbf{V}|^{-(\delta+2p)/2} \exp\left\{-\frac{1}{2} tr(\mathbf{V}^{-1}\mathbf{U})\right\} \tag{1}$$

for $\mathbf{V} \in M^+(\mathscr{G})$, the cone of positive definite matrices where $v_{ij} \equiv 0$ if there is no edge $Y_i \leftrightarrow Y_j$ in $\mathscr{G}$. This is called a *$\mathscr{G}$-inverse Wishart prior*. In general, there is no closed-form expression for the normalizing constant of this density function. Draws from the posterior distribution for parameters can be generated by a Gibbs sampler scheme as introduced by [12].

3 A Sampler for Bi-directed Structures

In this Section we consider the case where a given p-dimensional observed vector $\mathbf{Y}$ is generated according to a Gaussian ADMG model without directed edges. In this special case, the corresponding graph is called a *bi-directed graph*. That is, $\mathbf{Y}$ follows a zero-mean multivariate Gaussian with sparse covariance matrix $\mathbf{V}$. Conditional on a bi-directed graph $\mathscr{G}$, $\mathbf{V}$ is given the $\mathscr{G}$-inverse Wishart prior (1). For each pair of variables (Y_i, Y_j), $i < j$, we define a Bernoulli random variable z_{ij}, where $z_{ij} = 1$ if and only if there is an edge $Y_i \leftrightarrow Y_j$ in $\mathscr{G}$. Vector $\mathbf{z}$ denotes the vector obtained by stacking all z_{ij} variables. The prior for $\mathscr{G}$ is defined as the product of priors for each z_{ij}, $i < j$ (z_{ij} is also defined for $i > j$ and equal to z_{ji}), where $p(z_{ij} = 1) \equiv \eta_i \eta_j$, with each $\eta_i \sim Uniform(0,1)$ a priori.

For a general ADMG with directed and bi-directed edges, directed edge coefficients can be sampled conditioned on $\mathbf{V}$ using a variety of off-the-shelf methods (e.g., using spike-and-slab priors corresponding to parameters associated with directed edges). Conditioned on edge coefficients $\{\lambda_{ij}\}$, the joint residual vector given by entries $Y_i - \sum_{j=1}^{M_i} \lambda_{ij} Y_{i[j]}$ follows a Gaussian bi-directed graph model, where sampling requires novel methods. Therefore, for simplicity of exposition, we describe only the sampler for the bi-directed structure. We present an (approximate) Gibbs sampler to generate posterior samples for $\mathbf{z}$ given a dataset $\mathscr{D} = \{\mathbf{Y}^{(1)}, \mathbf{Y}^{(2)}, \ldots, \mathbf{Y}^{(N)}\}$.

Let $\mathbf{V}_{\setminus i,\setminus i}$ the submatrix of $\mathbf{V}$ obtained by dropping its i-th column and row. Let $\mathbf{z}_{\setminus ij}$ be the set of edge indicator variables $\mathbf{z}$ without indicator z_{ij} (or z_{ji}). The sampler iterates over each vertex Y_i and performs the following:

1. for each $j \in \{1,2,\ldots,p\}\setminus\{i\}$, we sample z_{ij} given $\mathbf{V}_{\setminus i,\setminus i}$ and $\mathbf{z}_{\setminus ij}$;
2. we sample the i-th row/column entries of $\mathbf{V}$, $\{v_{i1}, v_{i2}, \ldots, v_{ip}\}$ given $\mathbf{V}_{\setminus i,\setminus i}$ and $\mathbf{z}$

The second step above is parameter sampling for sparse covariance matrices, described in detail by [12]. In the remainder of this Section, we focus on the step of structure sampling. The conditional distribution for z_{ij} is given by

$$p(z_{ij} \mid \mathbf{V}_{\setminus i,\setminus i}, \mathbf{z}_{\setminus ij}, \mathscr{D}) \propto p(\mathscr{D} \mid \mathbf{V}_{\setminus i,\setminus i}, \mathbf{z}) \times p(\mathbf{V}_{\setminus i,\setminus i} \mid \mathbf{z}) \times p(z_{ij} \mid \mathbf{z}_{\setminus ij}) \tag{2}$$

One difficulty is introduced by the factor $p(\mathbf{V}_{\setminus i,\setminus i} \mid \mathbf{z})$, which is the marginal of a $\mathscr{G}$-inverse Wishart and where in general $p(\mathbf{V}_{\setminus i,\setminus i} \mid \mathbf{z}_{\setminus ij}, z_{ij} = 1) \neq p(\mathbf{V}_{\setminus i,\setminus i} \mid \mathbf{z}_{\setminus ij}, z_{ij} = 0)$. Computing this factor is expensive. However, in 1,000 preliminary runs with $p = 10$, $\delta = 1$ and $\mathbf{U}$ as the identity matrix, we found that errors introduced by the approximation

$$p(\mathbf{V}_{\setminus i,\setminus i} \mid \mathbf{z}_{\setminus ij}, z_{ij} = 1) \approx p(\mathbf{V}_{\setminus i,\setminus i} \mid \mathbf{z}_{\setminus ij}, z_{ij} = 0) \tag{3}$$

are minimal. No convergence problems for the Markov chains could be detected either. We adopt this approximation due to the large computational savings it brings, and as such the factor $p(\mathbf{V}_{\setminus i,\setminus i} \mid \mathbf{z})$ will be dropped without further consideration.

The first factor in (2) can be rewritten by completing and integrating away the remaining non-zero entries of $\mathbf{V}$, which we denote here by $\mathbf{V}_{i\cdot}$:

$$p(\mathscr{D} \mid \mathbf{V}_{\setminus i,\setminus i}, \mathbf{z}) = \int \prod_{d=1}^{N} p(\mathbf{Y}^{(d)} \mid \mathbf{V}) p(\mathbf{V}_{i\cdot} \mid \mathbf{V}_{\setminus i,\setminus i}, \mathbf{z}) \prod_{j \in E(\mathbf{z},i)} dv_{ij} \tag{4}$$

where $E(\mathbf{z}, i)$ is the set of indices for the spouses of Y_i in $\mathscr{G}$ (as given by $\mathbf{z}$), including Y_i itself. By definition, $v_{ij} = 0$ if Y_j is not a spouse of Y_i in $\mathscr{G}$.

In order to solve this integral, we appeal to some of the main results of [12]. Let $\mathscr{B}_i$ be a $1 \times (p-1)$ vector and γ_i a positive scalar such that

$$\mathbf{V}_{i,\setminus i} = \mathscr{B}_i \mathbf{V}_{\setminus i,\setminus i}, \quad v_{ii} = \gamma_i + \mathscr{B}_i \mathbf{V}_{\setminus i,\setminus i} \mathscr{B}_i^T \tag{5}$$

where $\mathbf{V}_{i,\setminus i}$ is the i-th row of $\mathbf{V}$ after removing entry v_{ii}. We define $\mathscr{B}_{sp(i)}$ and $\mathscr{B}_{nsp(i)}$ to be the subvectors of $\mathscr{B}_i$ that match the corresponding rows of $\mathbf{V}_{\setminus i,\setminus i}$. The "non-spouse" entries are not free parameters when considering the structural zeroes of $\mathbf{V}$.

By our definition of $\mathscr{B}_i$, we have that $\mathscr{B}_i \mathbf{V}_{\setminus i,nsp(i)}$ gives the covariance between Y_i and its non-spouses (where $\mathbf{V}_{\setminus i,nsp(i)}$ is the corresponding submatrix of $\mathbf{V}$). By assumption these covariances are zero, that is $\mathscr{B}_i \mathbf{V}_{\setminus i,nsp(i)} = 0$. It follows that

$$\mathscr{B}_{sp(i)} \mathbf{V}_{sp(i),nsp(i)} + \mathscr{B}_{nsp(i)} \mathbf{V}_{nsp(i),nsp(i)} = 0 \Rightarrow \mathscr{B}_{nsp(i)} = -\mathscr{B}_{sp(i)} \mathbf{V}_{sp(i),nsp(i)} \mathbf{V}^{-1}_{nsp(i),nsp(i)} \tag{6}$$

As in the unconstrained case, the mapping between the non-zero entries of $\mathbf{V}_{i\cdot}$ and $\{\mathscr{B}_{sp(i)}, \gamma_i\}$ is one-to-one. Following [12], conditional density $p(\mathbf{V}_{i\cdot} \mid \mathbf{V}_{\setminus i,\setminus i} \mathbf{z})$ can be rewritten as:

$$p(\mathbf{V}_{i\cdot} \mid \mathbf{V}_{\setminus i,\setminus i}, \mathbf{z}) = p_{\mathscr{B}}(\mathscr{B}_{sp(i)} \mid \gamma_i, \mathbf{V}_{\setminus i,\setminus i}; \mu_{\mathscr{B}}, \gamma_i \mathbf{K}_{\mathscr{B}}) p_{\gamma_i}(\gamma_i \mid \mathbf{V}_{\setminus i,\setminus i}; \alpha_i, \beta_i) \tag{7}$$

where $p_{\mathscr{B}}(\cdot;\mu_{\mathscr{B}},\gamma_i\mathbf{K}_{\mathscr{B}})$ is a Gaussian density function with mean $\mu_{\mathscr{B}}$ and covariance matrix $\gamma_i\mathbf{K}_{\mathscr{B}}$, and function $p_{\gamma_i}(\cdot;\alpha_i,\beta_i)$ is an inverse gamma density function with parameters α_i and β_i. Parameters $\{\mu_{\mathscr{B}},\mathbf{K}_{\mathscr{B}},\alpha_i,\beta_i\}$ are described in the Appendix. Moreover, as a function of $\{z_{ij},\mathscr{B}_{sp(i)},\gamma_i,\mathbf{V}_{\setminus i,\setminus i}\}$, we can rewrite $p(\mathbf{Y}^{(d)} \mid \mathbf{V})$ as:

$$p(\mathbf{Y}^{(d)} \mid \mathbf{V}) \propto \gamma_i^{-1/2} \exp\left\{-\frac{1}{2\gamma_i}\left(Y_i^{(d)} - \mathscr{B}_{sp(i)}\mathbf{H}_{sp(i)}^{(d)}\right)^2\right\} \tag{8}$$

where $\mathbf{H}_{sp(i)}^{(d)}$ are the residuals of the regression of the spouses of Y_i on its non-spouses for datapoint d, as given by $\mathbf{V}_{\setminus i,\setminus i}$. That is

$$\mathbf{H}_{sp(i)}^{(d)} \equiv \mathbf{Y}_{sp(i)}^{(d)} - \mathbf{V}_{sp(i),nsp(i)}\mathbf{V}_{nsp(i),nsp(i)}^{-1}\mathbf{Y}_{nsp(i)}^{(d)} \tag{9}$$

Combining (7) and (8) allows us to rewrite (4) as

$$p(\mathscr{D} \mid \mathbf{V}_{\setminus i,\setminus i},\mathbf{z}) \propto |\mathbf{K}_{\mathscr{B}}|^{-\frac{1}{2}}|\mathbf{T}|^{-\frac{1}{2}}\frac{\beta_i^{\alpha_i}}{\Gamma(\alpha_i)}\frac{\Gamma(\alpha_i')}{\beta_i'^{\alpha_i'}} \tag{10}$$

where $\{\mathbf{T},\alpha_i',\beta_i'\}$ are defined in the Appendix. Every term above depends on the value of z_{ij}. Finally, $p(z_{ij}=1 \mid \mathbf{V}_{\setminus i,\setminus i},\mathbf{z}_{\setminus ij},\mathscr{D}) \propto p(\mathscr{D} \mid \mathbf{V}_{\setminus i,\setminus i},\mathbf{z}_{\setminus ij},z_{ij}=1)\eta_i\eta_j$ and $p(z_{ij}=0 \mid \mathbf{V}_{\setminus i,\setminus i},\mathbf{z}_{\setminus ij},\mathscr{D}) \propto p(\mathscr{D} \mid \mathbf{V}_{\setminus i,\setminus i},\mathbf{z}_{\setminus ij},z_{ij}=0)(1-\eta_i\eta_j)$.

After resampling z_{ij} for all $1 \leq j \leq p, j \neq i$, we resample the corresponding non-zero covariances, as described at the beginning of this Section, and iterate, alternating with steps to sample latent variables, regression coefficients and hyperparameters η_i as necessary.

4 Illustration: Learning Measurement Error Structure

One application of the methodology is learning the structure of measurement error for a latent variable model. Consider, for illustration purposes, the DAG in Figure 1. Assume the goal is to learn from observed measurements $Y_1,Y_2,\ldots,Y_8$ what values the corresponding latent variables X_1 and X_2 should take (more precisely, to calculate functionals of the conditional distribution of $\{X_1,X_2\}$ given $\mathbf{Y}$). Other sources of variability explain the marginal distribution of $\mathbf{Y}$, but they are not of interest. In this example, X_3 and X_4 are the spurious sources. Not including them in the model introduces bias. Sometimes background knowledge is useful to provide which observed variable measures which target latent variable (e.g., Y_1 should be a child of X_1 but not of X_2). The literature in structural equation models and factor analysis [3, 2] provides some examples where observed variables are designed so that latent concepts of interest are measured (up to some measurement error). Background knowledge about other hidden common causes of the observed variables is less clear, though.

In this Section, we provide a simple illustration on how to combine background knowledge about measurement with an adaptive methods that generates extra condi-

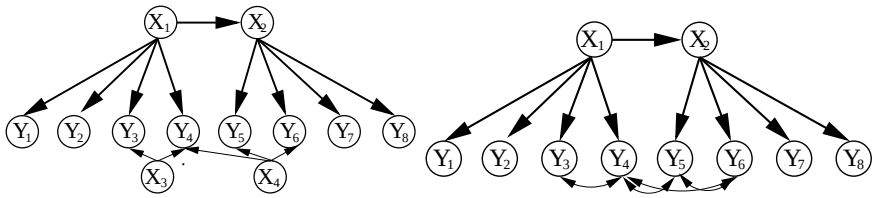

Fig. 1 In the left, a model where latent variables X_3 and X_4 provide an extra source of dependence for some of the observed variables that is not accounted by the target latent variables X_1 and X_2. In the right, a graphical representation of the marginal dependencies after marginalizing away some (X_3 and X_4) but not all of the latent variables.

tional dependencies among observed variables. Consider a more complex synthetic model given by a latent variable ADMG with four latent variables and 12 observed variables. Each observed variable has a single latent parent: the first three have X_1 as a common parent, the next three have X_2, and so on. The covariance matrix of the latent variables was sampled from an inverse Wishart distribution. Bi-directed edges among indicators were generated randomly with probability 0.2. To ensure identifiability, we pick 2 out the each 3 children of each latent variable and enforce that no bi-directed edges should exist within this set of 8 indicators. More flexible combinations can be enforced in the future using the results of [6].

The goal is: given knowledge about which directed edges exist and do not exist, learn the bi-directed structure. The algorithm in the previous Section is used to sample error covariance matrices among observed variables (non-zero error covariances between a latent variable and an observed variable are prohibited for simplicity, and the covariance matrix among latent variables has no independence constraints). This is done as part of a Gibbs sampling procedure where the values of the latent variables are also sampled so that the procedure in Section 3 can be used without modification as if all variables were observed.

Figure 2 summarizes the analysis of the error covariance matrix and its corresponding bi-directed structure using a sample size of 2000 (and a Markov chain with 5000 iterations). A bi-directed structure estimate is generated using the posterior samples. In this case, instead of using the most common structure as the estimator, we use a thresholding mechanism. Edges $Y_i \leftrightarrow Y_j$ such that the posterior expected value of the corresponding z_{ij} is greater than 0.5 are kept, while the others are estimated to be non-existent. A thresholding estimator for the structure is a practical alternative to choosing the most probable graph: a difficult task for Markov chain Monte Carlo in discrete structures. An analysis of thresholding mechanisms is provided in other contexts by [1] and [4]. However, since the estimated graph might not have occurred at any point during sampling, further parameter sampling conditioned on this graph will be necessary in order to obtain as estimator for the covariance matrix with structural zeroes matching the missing edges.

We also found that the choice of prior $p(z_{ij} = 1) \equiv \eta_i \eta_j$ to be particularly important. An alternative prior $p(z_{ij} = 1) = 0.5$ resulted in graphs with considerably more edges than the true one. A more extended discussion on how to enforce sparsity by

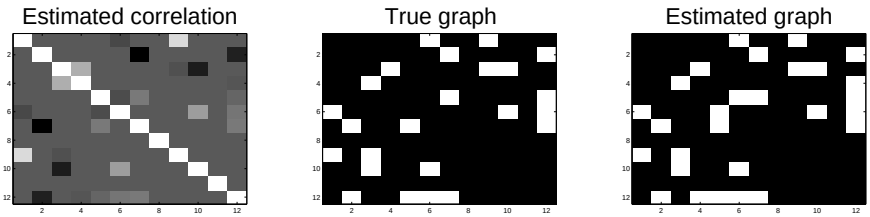

Fig. 2 In the left, the estimated error correlation matrix as given by the expected value of the marginal (hence, not sparse) posterior distribution of the rescaled error covariance **V**. Black dots mean correlation of -1, white dots mean correlation of 1. In the right, the estimator of the structure (edge appears if its posterior probability is greater than 0.5). The procedure added two spurious edges, but the corresponding estimated correlations are still close to zero.

priors over graphical structures is presented by [7]. An important line of future work will consist on designing and evaluating priors for mixed graph structures.

References

1. Barbieri, M.M., Berger, J.: Optimal predictive model selection. The Annals of Statistics **32**, 870–897 (2004)
2. Bartholomew, D., Knott, M.: Latent Variable Models and Factor Analysis. Arnold Publishers (1999)
3. Bollen, K.: Structural Equations with Latent Variables. John Wiley & Sons (1989)
4. Carvalho, C., Polson, N.: The horseshoe estimator for sparse signals. Biometrika **97**, 465–480 (2010)
5. Drton, M., Richardson, T.: Iterative conditional fitting for Gaussian ancestral graph models. Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (2004)
6. Grzebyk, M., Wild, P., Chouaniere, D.: On identification of multi-factor models with correlated residuals. Biometrika **91**, 141–151 (2004)
7. Jones, B., Carvalho, C., Dobra, A., Hans, C., Carter, C., West, M.: Experiments in stochastic computation for high-dimensional graphical models. Statistical Science **20**, 388–400 (2005)
8. Lauritzen, S.: Graphical Models. Oxford University Press (1996)
9. Richardson, T.: Markov properties for acyclic directed mixed graphs. Scandinavian Journal of Statistics **30**, 145–157 (2003)
10. Richardson, T., Spirtes, P.: Ancestral graph Markov models. Annals of Statistics **30**, 962–1030 (2002)
11. Sadeghi, K., Lauritzen, S.: Markov properties for loopless mixed graphs. arXiv:1109.5909v1 (2012)
12. Silva, R., Ghahramani, Z.: The hidden life of latent variables: Bayesian learning with mixed graph models. Journal of Machine Learning Research **10**, 1187–1238 (2009)

Appendix

We describe the parameters referred to in the sampler of Section 3. The full derivation is based on previous results described by [12]. Let **HH** be the statistic

$\sum_{n=1}^{d} \mathbf{H}_{sp(i)}^{(d)} \mathbf{H}_{sp(i)}^{(d)\ T}$. Likewise, let $\mathbf{YH} \equiv \sum_{n=1}^{d} Y_i^{(d)} \mathbf{H}_{sp(i)}^{(d)}$ and $\mathbf{YY} \equiv \sum_{n=1}^{d} Y_i^{(d)^2}$. Recall that the hyperparameters for the $\mathscr{G}$-inverse Wishart are δ and $\mathbf{U}$, as given by Equation (1) and as such we are computing a "conditional normalizing constant" for the posterior of $\mathbf{V}$ integrating over only *one* of the row/columns of $\mathbf{V}$.

First, let

$$\begin{aligned} \mathbf{A}_i &\equiv \mathbf{V}_{sp(i),nsp(i)} \mathbf{V}_{nsp(i),nsp(i)}^{-1} \\ \mathbf{M}_i &\equiv (\mathbf{U}_{\setminus i,\setminus i})^{-1} \mathbf{U}_{\setminus i,i} \\ \mathbf{m}_i &\equiv (\mathbf{U}_{ss} - \mathbf{A}_i \mathbf{U}_{ns}) \mathbf{M}_{sp(i)} + (\mathbf{U}_{sn} - \mathbf{A}_i \mathbf{U}_{nn}) \mathbf{M}_{nsp(i)} \end{aligned} \tag{11}$$

$$\begin{aligned} \mathbf{K}_{\mathscr{B}}^{-1} &\equiv \mathbf{U}_{ss} - \mathbf{A}_i \mathbf{U}_{ns} - \mathbf{U}_{sn} \mathbf{A}_i^T + \mathbf{A}_i \mathbf{U}_{nn} \mathbf{A}_i^T \\ \mu_{\mathscr{B}} &\equiv \mathbf{K}_{\mathscr{B}} \mathbf{m}_i \end{aligned}$$

where

$$\begin{bmatrix} \mathbf{U}_{ss} & \mathbf{U}_{sn} \\ \mathbf{U}_{ns} & \mathbf{U}_{nn} \end{bmatrix} \equiv \begin{bmatrix} \mathbf{U}_{sp(i),sp(i)} & \mathbf{U}_{sp(i),nsp(i)} \\ \mathbf{U}_{nsp(i),sp(i)} & \mathbf{U}_{nsp(i),nsp(i)} \end{bmatrix} \tag{12}$$

Moreover, let

$$\begin{aligned} \mathscr{U}_i &\equiv \mathbf{M}_i^T \mathbf{U}_{\setminus i,\setminus i} \mathbf{M}_i - \mathbf{m}_i^T \mathbf{K}_i \mathbf{m}_i \\ u_{ii.\setminus i} &\equiv \mathbf{U}_{ii} - \mathbf{U}_{i,\setminus i} (\mathbf{U}_{\setminus i,\setminus i})^{-1} \mathbf{U}_{\setminus i,i} \end{aligned}$$

$$\begin{aligned} \alpha_i &\equiv (\delta + p - 1 + \#nsp(i))/2 \\ \beta_i &\equiv \left(u_{ii.\setminus i} + \mathscr{U}_i\right)/2 \\ \mathbf{T} &\equiv \mathbf{K}_{\mathscr{B}}^{-1} + \mathbf{HH} \\ \mathbf{q} &\equiv \mathbf{YH} + \mathbf{K}_{\mathscr{B}}^{-1} \mu_{\mathscr{B}} \end{aligned} \tag{13}$$

where $\#nsp(i)$ is the number of non-spouses of Y_i (i.e., $(p-1) - \sum_{j=1}^{p} z_{ij}$).

Finally,

$$\begin{aligned} \alpha_i' &\equiv \frac{N}{2} + \alpha_i, \\ \beta_i' &\equiv \frac{\mathbf{YY} + \mu_{\mathscr{B}}^T \mathbf{K}_{\mathscr{B}}^{-1} \mu_{\mathscr{B}} - \mathbf{q}^T \mathbf{T}^{-1} \mathbf{q}}{2} + \beta_i \end{aligned} \tag{14}$$

Notice that each calculation of $\mathbf{A}_i$ (and related products) takes $\mathscr{O}(p^3)$ steps (assuming the number of non-spouses is $\mathscr{O}(p)$ and the number of spouses is $\mathscr{O}(1)$, which will be the case in sparse graphs). For each vertex Y_i, an iteration could take $\mathscr{O}(p^4)$ steps, and a full sweep would take prohibitive $\mathscr{O}(p^5)$ steps. In order to scale this procedure up, some tricks can be employed. For instance, when iterating over each candidate spouse for a fixed Y_i, the number of spouses increases or decreases by 1: this means fast matrix update schemes can be implemented to obtain a new $\mathbf{A}_i$ from its current value. However, even in this case the cost would still be $\mathscr{O}(p^4)$. More speed-ups follow from solving for $\mathbf{V}_{sp(i),nsp(i)} \mathbf{V}_{nsp(i),nsp(i)}^{-1}$ using sparse matrix representations, which should cost less than $\mathscr{O}(p^3)$ (but for small to moderate p, sparse matrix inversion might be slower than dense matrix inversion). Moreover, one might not try to evaluate all pairs $Y_i \leftrightarrow Y_j$ if some pre-screening is done by looking only at pairs where the magnitude of corresponding correlation sampled in the last step lies within some interval.