The Hidden Life of Latent Variables: Bayesian Learning with Mixed Graph Models

Ricardo Silva
Statistical Laboratory
University of Cambridge, CB3 0WB, UK

Zoubin Ghahramani†
Department of Engineering
University of Cambridge, CB2 1PZ, UK

Abstract
Directed acyclic graphs (DAGs) have been widely used as a representation of conditional independence in machine learning and statistics. Moreover, hidden or latent variables are often an important component of graphical models. However, DAG models suffer from an important limitation: the family of DAGs is not closed under marginalization of hidden variables. This means that in general we cannot use a DAG to represent the independencies over a subset of variables in a larger DAG. Directed mixed graphs (DMGs) are a representation that includes DAGs as a special case, and overcomes this limitation. This paper introduces algorithms for performing Bayesian inference in Gaussian and probit DMG models. An important requirement for inference is the characterization of the distribution over parameters of the models. We introduce a new distribution for covariance matrices of Gaussian DMGs. We discuss and illustrate how several Bayesian machine learning tasks can benefit from the principle presented here: the power to model dependencies that are generated from hidden variables, but without necessarily modelling such variables explicitly.

Keywords: Graphical models, Bayesian inference, Markov chain Monte Carlo, latent variable models

1. Contribution
The introduction of graphical models (Pearl, 1988; Lauritzen, 1996; Jordan, 1998) changed the way multivariate statistical inference is performed. Graphical models provide a suitable language to factorize many complex real-world processes through conditional independence constraints.

Different families of independence models exist. The directed acyclic graph (DAG) family is a particularly powerful representation. Besides providing a language for encoding causal statements (Spirtes et al., 2000; Pearl, 2000), it is in a more general sense a family that allows for non-monotonic independence constraints: that is, models where some independencies can be destroyed by conditioning on new information (also known as the “explaining away” effect – Pearl, 1988), a feature to be expected in many real problems.

∗. Part of this work was done while RS was at the Gatsby Computational Neuroscience Unit, UCL.
†. Also affiliated with the Machine Learning Department, Carnegie Mellon University.

©2000 Ricardo Silva and Zoubin Ghahramani.
Figure 1: Consider the DAG in (a). Suppose we want to represent the marginal dependencies and independencies that result after marginalizing out $Y_6$. The simplest resulting DAG (i.e., the ones with fewer edges) is depicted in (b). However, notice that this graph does not encode some of the independencies of the original model. For instance, $Y_3$ and $Y_4$ are no longer marginally independent in the modified DAGs. A different family of graphical models, encoded with more than one type of edge (directed and bi-directed), is the focus of this paper. The graph in (c) depicts the solution using this “mixed” representation.

However, DAG independence models have an undesirable feature: they are not closed under marginalization, as we will illustrate. Consider the regression problem where we want to learn the effect of a cocktail of two drugs for blood pressure, while controlling for a chemotherapy treatment of liver cancer. We refer to $Y_1$, $Y_2$ as the dosage for the blood pressure drugs, $Y_3$ as a measure of chemotherapy dosage, $Y_4$ as blood pressure, and $Y_5$ as an indicator of liver status. Moreover, let $Y_6$ be an hidden physiological factor that affects both blood pressure and liver status. It is assumed that the DAG corresponding to this setup is given by Figure 1(a).

In this problem, predictions concerning $Y_6$ are irrelevant: what we care is the marginal for $\{Y_1, \ldots, Y_5\}$. Ideally, we want to take such irrelevant hidden variables out of the loop. Yet the set of dependencies within the marginal for $\{Y_1, \ldots, Y_5\}$ cannot be efficiently represented as a DAG model. If we remove the edge $Y_3 \rightarrow Y_4$ from Figure 1(b), one can verify this will imply a model where $Y_3$ and $Y_4$ are independent given $Y_5$, which is not true in our original model. To avoid introducing unwanted independence constraints, a DAG such as the one in Figure 1(b) will be necessary. Notice that in general this will call for extra dependencies that did not exist originally (such as $Y_4$ and $Y_3$ now being marginally dependent). Not only learning from data will be more difficult due to the extra dependencies, but specifying prior knowledge on the parameters becomes less intuitive and therefore more error prone.

In general, it will be the case that variables of interest have hidden common causes. This puts the researcher using DAGs in a difficult position: if she models only the marginal comprising the variables of interest, the DAG representation might not be suitable anymore. If she includes all hidden variables for the sake of having the desirable set of independencies, extra assumptions about hidden variables will have to be taken into account. In this sense, the DAG representation is flawed. There is a need for a richer family of graphical models, for which mixed graphs are an answer.
Directed mixed graphs (DMGs) are graphs with directed and bi-directed edges. In particular, acyclic directed mixed graphs (ADMGs) have no directed cycle, i.e., no sequence of directed edges $X \rightarrow \cdots \rightarrow X$ that starts and ends on the same node. Such a representation encodes a set of conditional independencies among random variables, which can be read off a graph by using a criterion known as m-separation, a natural extension of the d-separation criterion used for directed acyclic graphs (Richardson, 2003).

In an ADMG, two adjacent nodes might be connected by up to two edges, where in this case one has to be bi-directed and the other directed. A cyclic model can in principle allow for two directed edges of opposite directions. Figure 2 provides a few examples of DMGs. The appeal of this graphical family lies on the representation of the marginal independence structure among a set of observed variables, assuming they are part of a larger DAG structure that includes hidden variables. This is illustrated in Figure 3\(^1\). More details on DMGs are given in Sections 2 and 8. In our blood pressure/liver status multiple regression problem, the suitable directed mixed graph is depicted in Figure 2(c).

The contribution of this paper is how to perform Bayesian inference on two different families of mixed graph models: Gaussian and probit. Markov chain Monte Carlo (MCMC) and variational approximations will be discussed. Bayesian inference for DMG models has not been accomplished satisfactorily before, as discussed in Section 2, despite the fact that such models are widely used in several sciences.

The rest of the paper is organized as follows. Section 3 describes a special case of Gaussian mixed graph models, where only bi-directed edges are allowed. Priors and a Monte Carlo algorithm are described. This case will be a building block for subsequent sections, such as Section 4, where Gaussian DMG models are treated. Section 5 covers a type of discrete model for binary and ordinal data that factorizes according to an acyclic DMG. In Section 6 we discuss more sophisticated algorithms that are useful for scaling up Bayesian learning to higher-dimensional problems. Section 7 presents several empirical studies. Since the use of mixed graph models in machine learning applications is still in its early stages, we describe in Section 8 a variety of possible and largely unexplored uses of such graphs in machine learning applications.

---

1. Notice that it is not necessarily the case that the probability model itself is closed under marginalization. This will happen to some models, including the Gaussian model treated in this paper. But the basic claim of closure concerns the graph, i.e., the representation of independence constraints.
Figure 3: After marginalizing variables $H_1$ and $H_2$ from the DAG on the left, one possible DMG representation of the same dependencies is shown by the graph in the middle. Notice that there multiples DMGs within a same Markov equivalence class, i.e., encoding the same set of conditional independencies (Richardson and Spirtes, 2002). The two last graphs above are on the same class.

2. Basics of DMGs, Gaussian models and related work

In this section, we describe the Gaussian DMG model and how it complements latent variable models. At the end of the section, we also discuss a few alternative approaches for the Bayesian inference problem introduced in this paper.

2.1 Notation and terminology

In what follows, we will use standard notions from the graphical modelling literature, such as vertex (node), edge, parent, child, ancestor, descendant, DAG, undirected graph, Markov condition and d-separation. Refer to Pearl (1988) and Lauritzen (1996) for the standard definitions if needed. Less standard definitions will be given explicitly when appropriate. A useful notion is that of m-separation (Richardson, 2003). This can be reduced to d-separation (Pearl, 1988) by the following trick: for each bi-directed edge $Y_i \leftrightarrow Y_j$, introduce a new hidden variable $X_{ij}$ and the edges $X_{ij} \rightarrow Y_i$ and $X_{ij} \rightarrow Y_j$. Remove then all bi-directed edges and apply d-separation to the resulting directed graph.

Also, as usual, we will refer to vertices (nodes) in a graph and the corresponding random variables in a distribution interchangeably. Data points are represented by vectors with an upper index, such as $Y^{(1)}, Y^{(2)}, \ldots, Y^{(n)}$. The variable corresponding to node $Y_i$ in data point $Y^{(j)}$ is represented by $Y_i^{(j)}$.

2.2 On latent variable models and DMGs

The origins of mixed graph models can be traced back to Sewall Wright (Wright, 1921), who used special cases of mixed graph representations in genetic studies. Generalizing Wright’s approach, many scientific fields such as psychology, social sciences and econometrics use linear mixed graph models under the name of structural equation models (Bollen, 1989). Only recently the graphical and parametrical aspects of mixed graph models have been given a thorough theoretical treatment (Richardson and Spirtes, 2002; Richardson, 2003; Kang and Tian, 2005; Drton and Richardson, 2008). In practice, many structural equation
Bayesian Learning with Mixed Graph Models

models today are multivariate Gaussian distributions. We will work under this assumption unless stated otherwise.

For a DMG $G$ with a set of vertices $Y$, a standard parameterization of structural equation model Markov to $G$ is given as follows.

For each variable $Y_i$ with a (possibly empty) parent set $\{Y_{i1}, \ldots, Y_{ik}\}$, one provides a "structural equation"

$$Y_i = \mu_i + b_{i1}Y_{i1} + b_{i2}Y_{i2} + \cdots + b_{ik}Y_{ik} + \epsilon_i$$

(1)

where $\epsilon_i$ is a Gaussian random variable with zero mean and variance $v_{ii}$. Notice that this parameterization allows for cyclic models.

Unlike in standard regression models, “error term” $\epsilon_i$ is not necessarily constructed to be independent of parent $Y_p \in \{Y_{i1}, \ldots, Y_{ik}\}$. Dependencies arise when error terms of different equations are not independent. Independence is asserted by the graphical structure: $\epsilon_i$ and $\epsilon_j$ (the error term for some $Y_j$) are marginally independent if $Y_i$ and $Y_j$ are not connected by a bi-directed edge.

By this parameterization, each directed edge $Y_i \leftarrow Y_j$ in the graph corresponds to a parameter $b_{ij}$. Each bi-directed edge $Y_i \leftrightarrow Y_j$ in the graph is associated with a covariance parameter $v_{ij}$, the covariance of $\epsilon_i$ and $\epsilon_j$. Each node $Y_j$ in the graph is associated with variance parameter $v_{jj}$, the variance of $\epsilon_j$. Algebraically, let $B$ be a $m \times m$ matrix, $m$ being the number of observed variables. This matrix is such that $B_{ij} = b_{ij}$ if $Y_i \leftarrow Y_j$ exists in the graph, and 0 otherwise. Let $V$ be a $m \times m$ matrix, where $V_{ij} = v_{ij}$ if $i = j$ or if $Y_i \leftrightarrow Y_j$ is in the graph, and 0 otherwise. Let $Y$ be the column vector of observed variables, $\mu$ the column vector of intercept parameters, and $\epsilon$ the corresponding vector of error terms. The set of structural equations can be given in matrix form as

$$Y = BY + \mu + \epsilon \Rightarrow Y = (I - B)^{-1}(\epsilon + \mu)$$

(2)

where $A^{-T}$ is the transpose of the inverse of matrix $A$ and $\Sigma(\Theta)$ is the implied covariance matrix of the model, $\Theta = \{B, V, \mu\}$.

2.2.1 Completeness of parameterization and ancestral graphs

An important class of ADMGs is the directed ancestral graph. Richardson and Spirtes (2002) provide the definition and a thorough account of the Markov properties of ancestral graphs. One of the reasons for the name “ancestral graph” is due to one of its main properties: if there is a directed path $Y_i \rightarrow \cdots \rightarrow Y_j$, i.e., if $Y_i$ is an ancestor of $Y_j$, then there is no bi-directed edge $Y_i \leftrightarrow Y_j$. Thus directed ancestral graphs are ADMGs with this constraint.

In particular, they show that any Gaussian distribution that is Markov with respect to a given ADMG can be represented by some Gaussian ancestral graph model that is parameterized as above. The given parameterization is complete: i.e., the parameterization imposes no further constraints on the distribution besides the graphical independence constraints.

2. Notice this rules out the possibility of having both edges $Y_i \rightarrow Y_j$ and $Y_i \leftrightarrow Y_j$ in the same ancestral graph.
Since the methods described in this paper apply to general DMG models, they also apply to directed ancestral graphs.

Further constraints beyond independence constraints are certainly not undesirable. For instance, general ADMGs that are not ancestral graphs may impose other constraints (Richardson and Spirtes, 2002), and such graphs can still be sensible models of, e.g., the causal processes for the problem at hand. However, it is useful to be able to build upon parametric independence models that are known to have a complete parameterization. In particular, it is interesting to have a model for the distribution of the error terms that has a complete parameterization, as shown below.

2.2.2 Example

To better appreciate what a complete parameterization buys us, let us consider the case where $B = 0$ and $\mu = 0$, i.e., a zero-mean structural equation model where the DMG $G$ contains only bi-directed edges. The model is then parameterized by a covariance matrix $V$. This structured covariance matrix has entry $v_{ij}$ equal to zero if vertices $Y_i$ and $Y_j$ are not connected. In general, bi-directed graph models are models of marginal independence. This lies in contrast to undirected Gaussian models, where the absence of an edge $Y_i \rightarrow Y_j$ implies a zero entry in the inverse covariance matrix (Lauritzen, 1996).

Estimation in Gaussian bi-directed models is not computationally straightforward. Only recently a maximum likelihood estimation algorithm for this family has been proposed (Drton and Richardson, 2003) (many other implementations used in practice are not guaranteed to respect the positive-definiteness constraint of the problem). Yet any Gaussian bi-directed model corresponds to the observed marginal of some latent variable model. We should therefore ask ourselves why not just use a latent variable model.

There are several reasons. One of theoretical interest is that some parametric maximal ancestral graphs models are curved exponential models (Richardson and Spirtes, 2002; Drton and Richardson, 2008). Therefore, they inherit all the nice properties of such a class, such as the possibility of using BIC as an asymptotically consistent model selection measure. This is not true of latent variable models in general (Rusakov and Geiger, 2005).

An important practical reason is that the set of latent variable models that factorize according to a bi-directed graph is undetermined from the independence constraints alone. Different latent variable models will imply different constraints and different models, and picking up the appropriate one is not an easy and often not an elegant alternative. Figure 4 illustrates a trivial but often surprising example: by introducing arbitrary “ancillary” latent variables corresponding to each bi-directed edge, we introduce a strong bias in the model. The learned covariance matrix is very different from the true one. The reason for the major failure of the model in Figure 4(d) should be clear: for each observed variable, the variances from the hidden parents are always added independently to its variance, restricting the class of covariance matrices that can be represented.

This “ancillary latent” representation is of course not the only way of generating a DAG latent variable model from an DMG, but the problem in general would require either adding lots of redundant parameters or allowing for other (intentional and non-intentional) constraints. This defeats the point of modelling without concerns about the structure of the corresponding latent variables.
Figure 4: The figure illustrates problems that can happen to any model whose graph contains a substructure of three mutually adjacent nodes, as in (a). We will examine the simplest case, which is a saturated (unconstrained) family of models. Consider, for instance, the model in (b), where in the figure each number is the corresponding entry in the covariance matrix, e.g., the covariance of $Y_1$ and $Y_2$ is 0.8, the variance of $Y_3$ is 2.0, and so on. This marginal could have been generated by a latent variable model such as in (c), for instance. One common way of parameterizing DMG models (used in, e.g., Dunson et al., 2005), however, is to create a new hidden variable $X_{ij}$ for each bi-directed edge $Y_i \leftrightarrow Y_j$, and add edges $X_{ij} \rightarrow Y_i$, $X_{ij} \rightarrow Y_j$. This can have disastrous consequences, however. For instance, consider variable $Y_3$ with its latent parents $X_{13}$ and $X_{23}$, which are also parents of $Y_1$ and $Y_2$, respectively. If the covariance of, e.g., $Y_1$ and $Y_3$ is strong, the parameters associate with the path $Y_1 \leftarrow X_{12} \rightarrow Y_3$ should be large. But since $X_{13}$ and $X_{23}$ are independent, variances associated with paths $X_{13} \rightarrow Y_3$ and $X_{23} \rightarrow Y_3$ will be added individually, inflating the variance of $Y_3$. This will create a compromise where parameters in paths such $Y_1 \leftarrow X_{12} \rightarrow Y_3$ have to be large so that they account for the covariance, but not too large so that the variance is not overestimated. But such compromises can be far from optimal. Figure (d) illustrates this model, with parameters estimated by maximum likelihood given a sample size of 10,000 from the model in (b). The numbers shown in this Figure are the covariance matrix estimates for the marginal of $Y_1$, $Y_2$ and $Y_3$ (e.g., the estimated covariance of $Y_1$ and $Y_2$ is 0.116, while the variance of $Y_1$ is 0.851). Using the (trivial) saturated DMG model in (e) results in proper estimates from the same data.
Moreover, after deciding for a proper choice of latent structure, the corresponding model will usually be unidentifiable. There are two options in this case. One option is to introduce extra constraints, by default or manually (such as setting some coefficient parameters to constants), getting an “asymmetric” parameterization which might be identifiable but cumbersome to sample from or set priors for. Another option is to completely ignore identifiability issues, at the expense of having many more random variables to integrate over than in the original DMG parameterization\(^3\).

This discussion by no means implies that one should not use latent variable models when appropriate. Latent variables are essential in several cases. Structural equation models often include latent variables, as in the application described in Section 7. Silva (2005) provides other examples in machine learning. However, even latent variables can have hidden common causes: the problem of where to stop adding hidden variables and starting modelling only the remaining marginals will, in general, call for a mixed graph representation. When it comes to solving a problem, it is up to the modeler (or learning algorithm) to decide if some set of latent variables should be included, or if they should be implicit, living their hidden life through the marginals.

2.3 Other approaches for Bayesian inference

Efficient algorithms for maximum likelihood estimation of Gaussian ancestral graph models are given by Drton and Richardson (2004). However, the current practice in Bayesian inference for Gaussian DMG models relies on either introducing extra latent variables or using rejection sampling.

One example of common practice is given by Dunson et al. (2005). A latent variable is introduced as a common parent of every pair of nodes connected by a bi-directed edge. The result is a DAG model, and inference is done on it using Gibbs sampling. This parameterization has all the disadvantages mentioned in the previous section.

Scheines et al. (1999) use the standard parameterization, with a single parameter corresponding to each bi-directed edge. However, the global constraint of positive-definiteness in the covariance matrix is enforced only by rejection sampling, which in models with moderate covariance values might be inefficient. Moreover, the prior is setup in an indirect way. “Local” Gaussian priors are independently defined for each error covariance \(\nu_{ij}\). The real prior, however, is the result of multiplying all “local priors” and the indicator function that discards non-positive definite matrices, which is then renormalized. This might cause some unwanted, but hard to predict, consequences when priors are elicited from experts.

In contrast, the Bayesian approach delineated in the next sections uses the standard parameterization, does not appeal to rejection sampling, makes use of a family of priors which we believe is the natural choice for the problem, and leads to convenient ways of computing marginal likelihoods for model selection. We will also see that empirically they also lead to much better behaved Markov chain Monte Carlo samplers.

\(^3\) To make an analogy with undirected networks, it is also true in general that a given Markov random field can be parameterized by a DAG model with latent variables, where latent variables are conditioned on, instead of marginalized (Richardson and Spirtes, 2002). Yet it would not be sensible to drop the whole family of Markov random field models in favor of an often cumbersome DAG parameterization (however, see Chu et al. (2006) for a nice example of a sensible DAG-parameterized Markov random field).
3. Gaussian models of marginal independence

This section concerns priors and sampling algorithms for zero-mean Gaussian models factorized according to a bi-directed graph. Focusing on bi-directed graphs only simplifies the presentation, while providing a convenient starting point to solve the full DMG case in the sequel.

Concerning the notation: the distribution we introduce in this section is a distribution over covariance matrices. In the interest of generality, we will refer to the matrix random variable as $\Sigma$. In the context of the previous section, $\Sigma \equiv \Sigma(\Theta) = V$, since we are assuming $B = 0, \mu = 0$.

3.1 Priors

Gaussian bi-directed graph models are sometimes called covariance graph models. Covariance graphs are models of marginal independence: each edge corresponds to a single parameter in the covariance matrix (the corresponding covariance); the absence of an edge $Y_i \leftrightarrow Y_j$ is a statement that $\sigma_{Y_iY_j} = 0$, $\sigma_{XY}$ being the covariance of random variables $X$ and $Y$. If $\Sigma$ is a random covariance matrix generated by a covariance model, the distribution of $\Sigma$ is actually a distribution over the set of non-fixed entries corresponding to variances and covariances of adjacent nodes$^4$.

In a model with a fully connected bi-directed graph, this reduces to a space of unrestricted covariance matrices. A common distribution for covariance matrices is the inverse Wishart $IW(\delta, U)$. In this paper, we adopt the following inverse Wishart parameterization:

$$p(\Sigma) \propto |\Sigma|^{-(\delta+2m)/2} \exp \left\{ -\frac{1}{2} tr(\Sigma^{-1}U) \right\}, \Sigma \text{ positive definite,}$$

$p(\cdot)$ being the density function, $tr(\cdot)$ the trace function, and $m$ the number of variables (nodes) in our model$^5$. We will overload the symbol $p(\cdot)$ wherever it is clear from the context which density function we are referring to. It is assumed that $\delta > 0$ and $U$ is positive definite.

Following Atay-Kayis and Massam (2005), let $M^+(\mathcal{G})$ be the cone of positive definite matrices such that, for a given bi-directed graph $\mathcal{G}$ and $\Sigma \in M^+(\mathcal{G})$, $\sigma_{ij} = 0$ if nodes $Y_i$ and $Y_j$ are not adjacent in $\mathcal{G}$. It is convenient to choose a distribution that is conjugate to the Gaussian likelihood function, since one can use the same algorithms for doing inference both in the prior and posterior. In a zero-mean Gaussian model, the likelihood of dataset $\mathcal{D} = \{Y^{(1)}, Y^{(2)}, \ldots, Y^{(n)}\}$ is given by the sufficient statistic $S = \sum_{d=1}^{n} (Y^{(d)})(Y^{(d)})^T$ as follows:

$$\mathcal{L}(\mathcal{D}|\Sigma) = (2\pi)^{-nm/2}|\Sigma|^{-n/2} \exp \left\{ -\frac{1}{2} tr(\Sigma^{-1}S) \right\}$$

$^4$ As such, the density over $\Sigma$ is defined with respect to the Lebesgue measure of the non-zero, independent elements of this matrix.

$^5$ We adopt this non-standard parameterization of the inverse Wishart because it provides a more convenient reparameterization used in the sequel. Notice is the parameterization used by Brown et al. (1993) and Atay-Kayis and Massam (2005), which developed other distributions for covariance matrices.
One cannot stray away from the kernel of the inverse Wishart distribution to preserve conjugacy. We define the following distribution:

$$p(\Sigma) = \frac{1}{I_G(\delta, U)} |\Sigma|^{-\left(\frac{\delta+2m}{2}\right)} \exp \left\{ -\frac{1}{2} \text{tr}(\Sigma^{-1} U) \right\}, \Sigma \in M^+(G)$$  \hspace{1cm} (5)

which is basically a re-scaled inverse Wishart prior with a different support and, consequently, different normalizing constant $I_G(\delta, U)$. An analogous concept exists for undirected graphs, where $\Sigma^{-1} \in M^+(G)$ is given a Wishart-like prior: the “$G$-Wishart” distribution (Atay-Kayis and Massam, 2005). We call the distribution with density function defined as in Equation (5) the $G$-Inverse Wishart distribution ($G$-IW). It will be the basis of our framework. There are no analytical formulas for the normalizing constant.

### 3.2 The normalizing constant

We now derive a Monte Carlo procedure to compute $I_G(\delta, U)$. This will provide a basis to sample from the $G$-IW distribution in the sequel. Moreover, the normalizing constant is essential for model selection of covariance graphs. By combining the likelihood equation (4) with the prior (5), we obtain the joint

$$p(D, \Sigma|G) = (2\pi)^{-\frac{nm}{2}} I_G(\delta, U)^{-1} \times |\Sigma|^{-\frac{\delta+2m+n}{2}} \exp \left\{ -\frac{1}{2} \text{tr}(\Sigma^{-1}(S + U)) \right\}$$  \hspace{1cm} (6)

where we make the dependency on the graphical structure $G$ explicit. By the definition of $I_G$, integrating $\Sigma$ out of the above equation implies the following marginal likelihood:

$$p(D|G) = \frac{1}{(2\pi)^{\frac{nm}{2}}} \frac{I_G(\delta + n, S + U)}{I_G(\delta, U)}$$  \hspace{1cm} (7)

from which a posterior $P(G|D)$ can be easily derived as a function of quantities of the type $I_G(\cdot, \cdot)$.

The normalizing constant $I_G(\delta, U)$ is given by the following integral:

$$I_G(\delta, U) = \int_{M^+(G)} |\Sigma|^{-\frac{\delta+2m}{2}} \exp \left\{ -\frac{1}{2} \text{tr}(\Sigma^{-1} U) \right\} d\Sigma$$  \hspace{1cm} (8)

The space $M^+(G)$ can be described as the space of positive definite matrices conditioned on the event that each matrix has zero entries corresponding to non-adjacent nodes in graph $G$. We will reduce the integral (8) to an integral over random variables we know how to sample from. The given approach follows the framework of Atay-Kayis and Massam (2005) using the techniques of Drton and Richardson (2003).

Atay-Kayis and Massam (2005) show how to compute the marginal likelihood of non-decomposable undirected models by reparameterizing the precision matrix through the Cholesky decomposition. The zero entries in the inverse covariance matrix of this model correspond to constraints in this parameterization, where part of the parameters can be sampled independently and the remaining parameters calculated from the independent ones.

---

6. Notice this integral is always finite for any choice of $\delta > 0$ and positive definite $U$, since it is smaller than the normalizing constant of the inverse Wishart.
We will follow a similar framework but with a different decomposition. This provides an easy way to calculate the desired integral.

3.2.1 Bartlett’s decomposition

Brown et al. (1993) attribute the following result to Bartlett: a positive definite matrix Σ, written as the partitioned matrix

\[ \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} \]  

(9)

can be decomposed as \( \Sigma = T \Delta T^T \) where

\[ \Delta = \begin{pmatrix} \Sigma_{11} & 0 \\ 0 & \Gamma \end{pmatrix} \text{ and } T = \begin{pmatrix} I & 0 \\ B & I \end{pmatrix} \]  

(10)

such that

\[ B = \Sigma_{21} \Sigma_{11}^{-1} \text{ and } \Gamma = \Sigma_{22,1} = \Sigma_{22} - \Sigma_{21} \Sigma_{11}^{-1} \Sigma_{12} \]  

(11)

That is, \( \Sigma \) can be parameterized by \((\Sigma_{11}, B, \Gamma)\) provided a partition \( \{Y_1, Y_2 \} \) of its random variables and the mapping \( \Sigma \rightarrow \{\Sigma_{11}, B, \Gamma\} \) is bijective. In this case, \( \Sigma_{11} \) is the covariance matrix of \( Y_1 \), \( B \) is equivalent to the coefficients obtained by least-squares regression of \( Y_2 \) on \( Y_1 \), and \( \Gamma \) is the covariance matrix of the residuals of this regression.

Expressing \( \Sigma \) as a function of \( \{\Sigma_{11}, B, \Gamma\} \) gives

\[ \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{11} B^T \\ BS_{11} & \Gamma + BS_{11} (\Sigma_{11})^T \end{pmatrix} \]  

(12)

This decomposition can be applied recursively. Let \( \{i\} \) represent the set of indices \( \{1, 2, \ldots, i\} \). Let \( \Sigma_{i,\{i-1\}} \) be the row vector containing the covariance between \( Y_i \) and all elements of \( \{Y_1, Y_2, \ldots, Y_{i-1}\} \). Let \( \Sigma_{\{i-1\},\{i-1\}} \) be the marginal covariance matrix of \( \{Y_1, Y_2, \ldots, Y_{i-1}\} \). Let \( \sigma_{ii} \) be the variance of \( Y_i \). Define the mapping

\[ \Sigma \rightarrow \{\gamma_1, B_2, \gamma_2, B_3, \gamma_3, \ldots, B_m, \gamma_m\}, \]  

(13)

such that \( B_i \) is a row vector with \( i - 1 \) entries, \( \gamma_i \) is a scalar, \( \sigma_{11} = \gamma_1 \), and

\[ \Sigma_{i,\{i-1\}} = B_i \Sigma_{\{i-1\},\{i-1\}}, \quad i > 1 \]

\[ \sigma_{ii} = \gamma_i + B_i \Sigma_{\{i-1\},i}, \quad i > 1 \]  

(14)

We call this set \( \{\gamma_1, B_2, \gamma_2, B_3, \gamma_3, \ldots, B_m, \gamma_m\} \) the Bartlett parameters of \( \Sigma \).

For a random inverse Wishart matrix, Bartlett’s decomposition allows the definition of its density function by the joint density of \( \{\gamma_1, B_2, \gamma_2, B_3, \gamma_3, \ldots, B_m, \gamma_m\} \). Define \( U_{\{i-1\},\{i-1\}} \), \( U_{\{i-1\},i} \) and \( U_{ii,\{i-1\},\{i-1\}} \) in a way analogous to the \( \Sigma \) definitions. The next lemma follows directly from Lemma 1 of Brown et al. (1993):

Lemma 1 Suppose \( \Sigma \) is distributed as \( IW(\delta, U) \). Then, after the transformation \( \Sigma \rightarrow \Phi = \{\gamma_1, B_2, \gamma_2, B_3, \gamma_3, \ldots, B_m, \gamma_m\} \):

1. \( \gamma_i \) is independent of \( \Phi \backslash \{\gamma_i, B_i\} \)
Lemma 2 The determinant of the Jacobian for the change of variable $\Sigma \to \Phi_G$ is

$$|J(\Phi_G)| = \prod_{i=2}^{m} |R_i| = \frac{1}{\prod_{i=2}^{m} |\Sigma_{sp,<(i),sp,<(i)}|} \prod_{i=1}^{m-1} \gamma_i^{m-i}$$

(17)

where $R_i \equiv \Sigma_{sp,<(i),sp,<(i)} - \Sigma_{sp,<(i),sp,<(i)}^{-1} \Sigma_{sp,<(i),sp,<(i)} \Sigma_{sp,<(i),sp,<(i)}^{-1} \Sigma_{sp,<(i),sp,<(i)}$, i.e., the covariance matrix of the residual of the linear regression of $sp,<(i)$ on $sp,<(i)$. If $sp,<(i) = \emptyset$, $R_i$ is defined as $\Sigma_{sp,<(i),sp,<(i)}$ and $|\Sigma_{sp,<(i),sp,<(i)}|$ is defined as 1.
This implies that the gradient of the block is the Jacobian \( \partial \Sigma_G / \partial \Phi_G \) as the gradient of the \( i \)-th row of the Jacobian matrix \( \partial \Sigma_G / \partial \Phi_G \) as the gradient of the \( i \)-th element of \( \Sigma_G \) with respect to \( \Phi_G \), following the specified order.

Notice that \( \partial \sigma_{ij} / \partial \beta_{st} = 0 \) and \( \partial \sigma_{ij} / \partial \gamma_s = 0 \) for \( s > i \) (by construction, \( i \geq j \) and \( s \geq t \)). This implies that \( J(\Phi_G) \) is a (lower) block triangular matrix of \( 2m - 1 \) blocks: for \( k \) odd, the \( k \)-th block is the singleton \( \partial \sigma_{ii} / \partial \gamma_i = 1 \), where \( i = (k + 1)/2 \). For \( k \) even, the \( k \)-th block is the Jacobian \( \partial \Sigma_{i,sp(i)} / \partial \Phi_{i,sp(i)} \), where \( i = 1 + k/2 \) and \( \Sigma_{i,sp(i)} \) is the vector of covariances of \( Y_i \) and its preceding spouses. This submatrix Jacobian turns out to be \( R_i \).

Since the determinant of the block triangular Jacobian \( J(\Phi_G) \) is given by the determinant of the blocks, this implies

\[
|J(\Phi_G)| = \prod_{i=2}^{2m} |R_i| \tag{18}
\]

By the matrix identity

\[
\begin{vmatrix}
A & B \\
C & D \\
\end{vmatrix} = |A||D - CA^{-1}B|, 
\]

\[
|\Sigma_{\{i-1\},\{i\}|} = |\Sigma_{nsp_{<i}(i),nsp_{<i}(i)}|^{\Sigma_{sp_{<i}(i),sp_{<i}(i)} - \Sigma_{sp_{<i}(i),nsp_{<i}(i)} - \Sigma_{nsp_{<i}(i),sp_{<i}(i)} - \Sigma_{nsp_{<i}(i),nsp_{<i}(i)}}| = |\Sigma_{nsp_{<i}(i),nsp_{<i}(i)}|^{R_i}. \]

Using the fact that \( |\Sigma_{\{i-1\},\{i\}|} = \prod_{t=1}^{2m} \gamma_t \), the second equality holds.

Notice that it corresponds to the Jacobian of the unconstrained covariance matrix, \( |J(\Phi)| \), in the case where there are no non-spouses:

\[
|J(\Phi)| = \prod_{i=1}^{m-1} \gamma_i^{m-i} \tag{20}
\]

Now that we have the original Jacobian (20), the constrained Jacobian (17), the distribution over Bartlett’s parameters given by Lemma 1, and the identities of Drton and Richardson (2003) given in Equation (16), we have all we need to provide a Monte Carlo algorithm to compute the normalizing constant of a \( G-IW \) with parameters \( (\delta, U) \).

Let \( \Sigma(\Phi_G) \) be the implied covariance matrix given by our set of parameters \( \Phi_G \). We start from the integral in (16), and rewrite it as a function of \( \Phi_G \). This can be expressed by substituting \( \Sigma \) for \( \Sigma(\Phi_G) \) and multiplying the integrand by the determinant of the Jacobian. Notice that the parameters in \( \Sigma(\Phi_G) \) are variation independent: that is, their joint range is given by the product of their individual ranges (positive reals for the \( \gamma \) variables and the real line for the \( \beta \) coefficients). This range will replace the original \( M^+(\mathcal{G}) \) space, which we omit below for simplicity of notation:

\[
I_G(\delta, U) = \int |J(\Phi_G)||\Sigma(\Phi_G)|^{-\frac{d+2m}{2}} \exp \left\{ -\frac{1}{2} tr(\Sigma(\Phi_G)^{-1}U) \right\} d\Phi_G \tag{21}
\]
We now multiply and divide the above expression by the normalizing constant of an inverse Wishart \((\delta, \mathbf{U})\), which we denote by \(I_{IW}(\delta, \mathbf{U})\):

\[
I_G(\delta, \mathbf{U}) = I_{IW}(\delta, \mathbf{U}) \int |J(\Phi_G)| \times I_{IW}^{-1}(\delta, \mathbf{U}) |\Sigma(\Phi_G)|^{-\frac{\delta+2m}{2}} \exp \left\{ -\frac{1}{2} \text{tr}(\Sigma(\Phi_G)^{-1} \mathbf{U}) \right\} \, d\Phi_G
\]

(22)

The expression

\[
I_{IW}^{-1}(\delta, \mathbf{U}) |\Sigma|^{-\frac{\delta+2m}{2}} \exp \left\{ -\frac{1}{2} \text{tr}(\Sigma^{-1} \mathbf{U}) \right\}
\]

(23)

corresponds to the density function of an inverse Wishart \(\Sigma\) and is embedded in Equation (22) written in our parameterization. Lemma 1 allows us to rewrite the inverse Wishart density function as the density of Bartlett parameters, but this is assuming no independence constraints. We can do a lengthy but analogous derivation of a similar result in our parameterization \(\Phi_G\), but instead we reuse Lemma 1 through the following shortcut:

1. write the density of the inverse Wishart as the product of gamma-normal densities given in Lemma 1;

2. this expression contains the original Jacobian determinant \(|J(\Phi)|\). We have to remove it, since we are plugging in our own Jacobian determinant. Hence, we divide the reparameterized density by the expression in Equation (20).

This ratio \(|J(\Phi_G)|/|J(\Phi)|\) can be rewritten as

\[
\frac{|J(\Phi_G)|}{|J(\Phi)|} = \prod_{i=1}^{m} |\tilde{R}_i|^{-\frac{1}{m-1}} = \prod_{i=2}^{m} \frac{1}{|\Sigma_{nsp, sp}(i), nsp, sp(i)|}
\]

(24)

where \(|\Sigma_{nsp, sp}(i), nsp, sp(i)| \equiv 1\) if \(nsp, sp(i) = \emptyset\);

3. substitute each vector \(\mathcal{B}_{i, nsp, sp(i)}\), which is not a free parameter, by the corresponding expression \(-\mathcal{B}_{i, sp, sp(i)} \Sigma_{sp, sp(i), nsp, sp(i)}^{-1} \Sigma_{nsp, sp(i), nsp, sp(i)}^{-1}\).

This substitution takes place into the original factors given by Bartlett’s decomposition, as introduced in Lemma 1:

\[
p(\mathcal{B}_i, \gamma_i) = (2\pi)^{-\frac{(I-1)/2}{2}} \gamma_i^{-\frac{(I-1)/2}{2}} |\mathbf{U}_{(i-1), (i-1)}|^{-1/2} \times \exp \left\{ -\frac{1}{2\gamma_i} (\mathcal{B}_i^T - \mathbf{M}_i)^T \mathbf{U}_{(i-1), (i-1)} (\mathcal{B}_i^T - \mathbf{M}_i) \right\} \times \frac{(u_{ii,(i-1), (i-1)}/2)^{(\delta+i-1)/2} \gamma_i^{-(\delta+i+1)/2}}{\Gamma((\delta+i-1)/2)} \exp \left\{ -\frac{1}{2\gamma_i} u_{ii,(i-1), (i-1)} \right\}
\]

(25)

where \(\mathbf{M}_i \equiv \mathbf{U}_{(i-1), (i-1)}^{-1} \mathbf{U}_{(i-1), i}\). Plugging this in (22) results in

\[
I_G(\delta, \mathbf{U}) = I_{IW}(\delta, \mathbf{U}) \int \prod_{i=2}^{m} \frac{1}{|\Sigma_{nsp, sp(i), nsp, sp(i)}|} \times p(\gamma_i) \prod_{i=2}^{m} p(\mathcal{B}_i, \gamma_i) \, d\Phi_G
\]

(26)
However, after substitution, each factor \( p(B_i, \gamma_i) \) is not in general a density function for \( \{B_{i,sp}(i), \gamma_i\} \) and will include also parameters \( \{B_{j,sp}(j), \gamma_j\}, j < i \). Because of the non-linear relationships that link Bartlett parameters within marginal independence models, we cannot expect to reduce this expression to a tractable distribution we can easily sample from. Instead, we rewrite each original density factor \( p(B_i, \gamma_i) \) such that it includes all information about \( B_{i,sp}(i) \) and \( \gamma_i \) within a canonical density function. That is, factorize \( p(B_i, \gamma_i) \) as

\[
p(B_i, \gamma_i | \Phi_{i-1}) = p_b(B_{i,sp}(i) | \gamma_i, \Phi_{i-1}) p_g(\gamma_i | \Phi_{i-1}) \times f_i(\Phi_{i-1})
\]

where we absorb any occurrence of \( B_{i,sp}(i) \) within the sampling distribution and factorize the remaining dependence on previous parameters \( \Phi_{i-1} \equiv \{\gamma_1, \gamma_2, B_{2,sp}(2), \ldots, \gamma_{i-1}, B_{i-1,sp}(i-1)\} \) into a separate function\(^7\). We derive the functions \( p_b(\cdot), p_g(\cdot) \) and \( f_i(\cdot) \) in Appendix A. The result is as follows.

The density \( p_b(B_{i,sp}(i) | \gamma_i, \Phi_{i-1}) \) is the density of a Gaussian \( N(K_i, m_i, \gamma_iK_i) \) such that

\[
m_i = (U_{ss} - A_i U_{ns}) M_{sp}(i) + (U_{sn} - A_i U_{nn}) M_{nsp}(i)
\]

\[
K_i^{-1} = U_{ss} - A_i U_{ns} - U_{sn} A_i^T + A_i U_{nn} A_i^T
\]

\[
A_i = \Sigma_{sp}(i) \Sigma_{nsp}(i)^{-1} \Sigma_{nsp}(i, nsp)(i)
\]

where

\[
\begin{bmatrix} U_{ss} & U_{sn} \\ U_{ns} & U_{nn} \end{bmatrix} = \begin{bmatrix} U_{sp}(i, sp)(i) & U_{sp}(i, nsp)(i) \\ U_{nsp}(i, sp)(i) & U_{nsp}(i, nsp)(i) \end{bmatrix}
\]

The density \( p_g(\gamma_i | \Phi_{i-1}) \) is the density of an inverse gamma \( IG(g_1, g_2) \) such that

\[
g_1 = \frac{\delta + i - 1 + \#nsp(i)}{2}
\]

\[
g_2 = \frac{u_{ii, \{i-1\}, \{i-1\}} + \mathcal{U}_i}{2}
\]

\[
\mathcal{U}_i = M_{i}^T U_{\{i-1\}, \{i-1\}} M_i - m_i^T K_i m_i
\]

where \( u_{ii, \{i-1\}, \{i-1\}} \) was originally defined in Section 3.2.1.

Finally,

\[
f_i(\Phi_{i-1}) \equiv (2\pi)^{-\frac{(i-1) - \#sp(i)}{2}} |K_i|^{1/2} |U_{\{i-1\}, \{i-1\}}|^{-1/2} \times \frac{\Gamma((\delta + i - 1 + \#nsp(i))/2)}{\Gamma((\delta + i - 1)/2)} \frac{\Gamma((\delta + i - 1)/2)}{((u_{ii, \{i-1\}, \{i-1\}} + \mathcal{U}_i)/2)^{(\delta + i - 1 + \#nsp(i))/2}}
\]

Density function \( p_b(B_{i,sp}(i) | \cdot, \cdot) \) and determinant \( |K_i|^{1/2} \) are defined to be 1 if \( sp(i) = \emptyset \). \( \mathcal{U}_i \) is defined to be zero if \( nsp(i) = \emptyset \).

\(^7\) A simpler decomposition was employed by Silva and Ghahramani (2006) (notice however that paper used an incorrect expression for the Jacobian). The following derivation, however, can be adapted with almost no modification to define a Gibbs sampling algorithm, as we show in the sequel.
The original normalizing constant integral is the expected value of a function of \( \Phi_G \) over a factorized gamma-normal distribution. The density function of this distribution is given below:

\[
p_{I}(\delta, U)(\Phi_G) = \left( \prod_{i=1}^{m} p_{g}(\gamma_{i}|\Phi_{i-1}) \right) \left( \prod_{i=2}^{m} p_{b}(B_{i,sp_{\prec}(i)}|\gamma_{i}, \Phi_{i-1}) \right)
\]

Let \( \langle f(X) \rangle_{p(X)} \) be the expected value of \( f(X) \) where \( X \) is a random vector with density \( p(X) \). We summarize the main result of this section through the following theorem:

**Theorem 3** The normalizing constant of a \( G \)-Inverse Wishart with parameters \((\delta, U)\) is given by

\[
I_{G}(\delta, U) = I_{IW}(\delta, U) \times \left\langle \prod_{i=1}^{m} \frac{f_{i}(\Phi_{i-1})}{\Sigma_{nsp_{\prec}(i),nsp_{\prec}(i)}} \right\rangle_{p_{I}(\delta, U)(\Phi_G)}
\]

Notice this can be further simplified to

\[
I_{G}(\delta, U) = \left\langle \prod_{i=1}^{m} \frac{f'_{i}(\Phi_{i-1})}{\Sigma_{nsp_{\prec}(i),nsp_{\prec}(i)}} \right\rangle_{p_{I}(\delta, U)(\Phi_G)}
\]

where

\[
f'_{i}(\Phi_{i-1}) \equiv (2\pi)^{-\frac{\#sp_{\prec}(i)}{2}} |K_{i}(\Phi_{i-1})|^{-1/2} \frac{\Gamma((\delta + i - 1 + \#nsp_{\prec}(i))/2)}{(\sum_{ti,\{i-1\},i-1} + U_{i})/(\delta + i - 1 + \#nsp_{\prec}(i))/2}
\]

which, as expected, reduces \( I_{G}(\delta, U) \) to \( I_{IW}(\delta, U) \) when the graph is complete.

A Monte Carlo estimate of \( I_{G}(\delta, U) \) is then given from (34) by obtaining samples \( \{\Phi_{G}^{(1)}, \Phi_{G}^{(2)}, \ldots, \Phi_{G}^{(M)}\} \) according to \( p_{I(\delta, U)}(\cdot) \) and computing:

\[
I_{G}(\delta, U) \approx \frac{1}{M} \sum_{s=1}^{M} \prod_{i=2}^{m} \frac{f'_{i}(\Phi_{i-1}^{(s)})}{\Sigma_{nsp_{\prec}(i),nsp_{\prec}(i)}(\Phi_{i-1}^{(s)})}
\]

where here we emphasize that \( \Sigma_{nsp_{\prec}(i),nsp_{\prec}(i)} \) is a function of \( \Phi_{G} \).

### 3.3 Sampling from a \( G \)-inverse Wishart

If \( Y \) follows a normal \( N(0, \Sigma) \) where \( \Sigma \) is given a \( G-IW(\delta, U) \) prior, then from a sample \( D = \{Y^{(1)}, \ldots, Y^{(n)}\} \) with sufficient statistic \( S = \sum_{d=1}^{n} (Y^{(d)})^{T}(Y^{(d)}) \), the posterior distribution for \( \Sigma \) given \( S \) will be a \( G-IW(\delta+n, U+S) \). In order to obtain samples from the posterior or to compute its functionals, one can adapt the algorithm for computing normalizing constants. We describe an importance sampler and a Gibbs sampling algorithm.
Algorithm \textsc{SampleGIW-1}

\textbf{Input:} matrix $U$, scalar $\delta$, bi-directed graph $G$, an ordering $\prec$

1. Let $\Sigma$ be a $m \times m$ matrix, with $m$ being the number of rows in $U$
2. Index $(\Sigma, U)$ and define functions $sp\prec(i)$, $nsp\prec(i)$ according to $G$ and ordering $\prec$
3. Sample $\sigma_{11}$ from $IG(\delta/2, u_{11}/2)$
4. For $i = 2, 3, \ldots, m$
5. Sample $\gamma_i \sim IG((\delta + i - 1 + \#nsp\prec(i))/2, (u_{ii,d_{i-1},d_{i-1}} + U_i)/2)$
6. Sample $B_{i,sp\prec(i)} \sim N(K_i m_i, \gamma_i K_i)$
7. Set $B_{i,nsp\prec(i)} = -B_{i,sp\prec(i)} \Sigma_{sp\prec(i),nsp\prec(i)}^{-1}$
8. Set $\Sigma_{i,i-1} = \Sigma_{i-1,i} = \Sigma_{i,\{i-1\}} \Sigma_{\{i-1\},i} B_i$
9. Set $\sigma_{ii} = \gamma_i + \Sigma_{i,\{i-1\}} B_i$
10. Set $w = \prod_{i=1}^M f_i'(\Phi_{i-1})/|\Sigma_{nsp\prec(i),nsp\prec(i)}|$
11. Return $(w, \Sigma)$.

\textbf{Figure 5:} A procedure for generating a sample $\Sigma$ and importance weight $w$ for a $G$-Inverse Wishart distribution. Variables $\{M_i, m_i, K_i, U_i\}$ and function $f'_i(\Phi_{i-1})$ are defined in Section 3.2.2.

### 3.3.1 The Importance Sampler

One of way computing functionals of the posterior, i.e., functions of the type

$$g(\delta, U; G) \equiv \int_{M^+(G)} g(\Sigma)p(\Sigma; \delta, U, G) d\Sigma$$

is through the numerical average

$$g(\delta, U; G) \approx \frac{\sum_{s=1}^M w_s g(\Sigma(s))}{\sum_{s=1}^M w_s}$$

where weights $\{w_1, w_2, \ldots, w_M\}$ and samples $\{\Sigma^{(1)}, \Sigma^{(2)}, \ldots, \Sigma^{(M)}\}$ generated by an importance sampler. The procedure for computing normalizing constants can be readily adapted for this task using $p_1(\delta, U; \cdot)$ as the importance distribution and the corresponding weights from the remainder factors. The sampling algorithm is shown in Figure 5.
Algorithm \textsc{SampleGIW-2}
\begin{enumerate}
\item Let $\Sigma$ be a copy of $\Sigma^{start}$
\item For $i = 1, 2, 3, \ldots, m$
\item Sample $\gamma_i \sim IG((\delta + (m-1) + \#nsp(i))/2, (u_{ii.}\{i\},\{i\} + U_i)/2)$
\item Sample $B_{i,sp(i)}$ from a $N(K_{ii}, \gamma_i K_{ii})$
\item Set $B_{i,nsp(i)} = -B_{i,sp(i)} \Sigma_{sp(i),nsp(i)} \Sigma_{nsp(i),nsp(i)}^{-1}$
\item Set $\Sigma^T_{i,\{i\}} = \Sigma_{\{i\},i}^T = \Sigma_{\{i\},\{i\}} B_i$
\item Set $\sigma_{ii} = \gamma_i + \Sigma_{i,\{i\}} B_i$
\item Return $\Sigma$
\end{enumerate}

Figure 6: A procedure for generating a sample $\Sigma$ within a Gibbs sampling procedure.

3.3.2 The Gibbs sampler

In the Gibbs sampling procedure, we sample the whole $i$-th row of $\Sigma$, for each $1 \leq i \leq m$, by conditioning on the remaining independent entries of the covariance matrix as obtained on the previous Markov chain iteration.

Let $\{\setminus i\}$ denote the set $\{1, 2, \ldots, i-1, i+1, \ldots, m\}$. The Gibbs algorithm is analogous to the previous algorithms. Instead of $sp \prec (i)$, one now has $sp_{\setminus (i)} = sp(i) \cap \{\setminus i\}$. The set $nsp_{\setminus (i)}$ is defined analogously. Matrix $U_{\setminus (i),\{i\}}$ is defined by deleting the $i$-th row and $i$-th column of $U$. Row matrix $U_{i,\{i\}}$ and scalar $u_{ii.}\{i\}$ are defined accordingly. The same indexing scheme defines $U_{sp_{\setminus (i)},sp_{\setminus (i)}}$, $U_{nsp_{\setminus (i)},nsp_{\setminus (i)}}$ and $U_{sp_{\setminus (i)},nsp_{\setminus (i)}}$. This index notation also applies to the corresponding $\Sigma$ submatrices. The algorithm is described in Figure 6. The procedure can be interpreted as having a dynamic ordering $\prec_i$ that at every step moves $Y_i$ to the end of the global ordering $\prec$.

The importance sampler suffers from the usual shortcomings in high-dimensional problems, by having a highly skewed weight distribution (where “high-dimensional” can be as low as a covariance matrix for a dozen variables). This can result in unstable estimates of functionals of the posterior. The Gibbs sampler is more computationally demanding considering the cost per step, but we expect it to require many fewer steps in such problems.

4. Gaussian directed mixed graph models

To extend our method to directed mixed graphs with parameters $\Theta = \{V, B, \mu\}$ for a fixed DMG $\mathcal{G}$, we apply a Gibbs sampling procedure.

Our prior takes the form $p(\Theta) = p(B)p(\mu)p(V)$. We assign priors for the parameters of directed edges (non-zero entries of matrix $B$) in a standard way: each parameter $b_{ij}$ is given a Gaussian $N(c_{ij}^B, s_{ij}^B)$ prior, where all parameters are marginally independent in the
prior, i.e., \( p(B) = \prod_{ij} p(b_{ij}) \). The prior for intercept parameters \( \mu \) is analogous, with \( \mu_i \) being a Gaussian \( N(c_i^\mu, s_i^\mu) \).

Recall from Equation (2) that the implied covariance of the model is given by matrix
\[
\Sigma(\Theta) = (I - B)^{-1} V (I - B)^{-T}.
\]
The likelihood function of dataset \( D = \{Y^{(1)}, Y^{(2)}, \ldots, Y^{(n)}\} \) is given by
\[
L(D|\Theta) = |\Sigma(\Theta)|^{-n/2} \prod_{d=1}^n \exp \left( -\frac{1}{2} (Y^{(d)} - \mu)^\top \Sigma(\Theta)^{-1} (Y^{(d)} - \mu) \right)
\]
\[
= \left\{ |(I - B)^{-1}V||(I - B)^{-T}| \right\}^{-n/2} \exp \left( -\frac{1}{2} tr(V^{-1}(I - B)S(I - B)^T) \right),
\]
where now \( S \equiv \sum_{d=1}^n (Y^{(d)} - \mu)(Y^{(d)} - \mu)^\top \).

Given a prior \( G-IW(\delta, U) \) for \( V \), it immediately follows that the posterior distribution of \( V \) given the data and other parameters is
\[
V \mid \{B, \mu, D\} \sim G-IW(\delta + n, U + (I - B)S(I - B)^T)
\]
(40)

Therefore it can be sampled using the results from the previous section. Notice this holds even if the directed mixed graph \( G \) is cyclic.

Sampling \( \mu_i \) given \( \{D, \Theta \mid \{\mu_i\}\} \) can also be done easily for both cyclic and acyclic models: the posterior is given by a normal \( N(c_i^\mu/s_i^\mu, 1/s_i^\mu) \) where
\[
s_i^{\mu'} = 1/s_i^\mu + n(V^{-1})_{ii},
\]
\[
c_i^{\mu'} = c_i^\mu/n - \sum_{t=1, t\neq i}^m (V^{-1})_{it}\mu_t + \sum_{d=1}^n \sum_{t=1}^m (V^{-1})_{it} \left( Y_t^{(d)} - \sum_{p_t} b_{tp_t} Y_{p_t}^{(d)} \right),
\]
(41)

with \( p_t \) being an index running over the parents of \( Y_t \) in \( G \).

However, sampling the non-zero entries of \( B \) results in two different cases depending whether \( G \) is cyclic or not. We deal with them separately.

### 4.1 Sampling from the posterior: acyclic case

The acyclic case is simplified by the fact that \( I - B \) can be rearranged in a way it becomes lower triangular, with each diagonal element being 1. This implies the identity
\[
|(I - B)^{-1}V||(I - B)^{-T}| = |V|,
\]
with the resulting log-likelihood being a quadratic function of the non-zero elements of \( B \). Since the prior for coefficient \( b_{ij} \) is Gaussian, its posterior given the data and all other parameters will be the Gaussian \( N(c_{ij}^b/s_{ij}^b, 1/s_{ij}^b) \) where
\[
s_{ij}^{b'} = 1/s_{ij}^b + \sum_{d=1}^n (Y_i^{(d)})^2,
\]
\[
c_{ij}^{b'} = c_{ij}^b + \sum_{d=1}^n Y_i^{(d)} \sum_{t=1}^m (V^{-1})_{it} \left( Y_t^{(d)} - \sum_{p_t \neq (i,j)} b_{tp_t} Y_{p_t}^{(d)} - \mu_t \right),
\]
(42)

As before, \( p_t \) runs over the indices of the parents of \( Y_t \) in \( G \). Notice that in the innermost summation we exclude \( b_{ij} Y_{ij}^{(d)} \). This Gaussian can be used to sample \( b_{ij} \).
It is important to notice that, in practice, a much better mixing behavior can be obtained by sampling the coefficients (and intercepts) jointly. The joint distribution is Gaussian and can be obtained in a way similar to the above derivation. The derivation of the componentwise conditionals is nevertheless useful in the algorithm for cyclic networks.

4.2 Sampling from the posterior: cyclic case

Cyclic directed graph models have an interpretation in terms of causal systems in equilibrium. The simultaneous presence of directed paths \( Y_i \rightarrow \cdots \rightarrow Y_j \) and \( Y_j \rightarrow \cdots \rightarrow Y_i \) can be used to parameterize instantaneous causal effects in a feedback loop (Spirtes, 1995). This semantics appear also in the structural equation modeling literature (Bollen, 1989). In terms of cyclic graphs as families of conditional independence constraints, methods for reading off constraints in linear systems also exist (Spirtes et al., 2000).

The computational difficulty in the cyclic case is that the determinant \( |I - B| \) is no longer a constant, but a multilinear function of coefficients \( \{b_{ij}\} \). Because \( b_{ij} \) will appear outside the exponential term, its posterior will no longer be Gaussian.

From the definition of the implied covariance matrix \( \Sigma(\Theta) \), it follows that \( |\Sigma(\Theta)|^{-n/2} = (|I - B||V|^{-1}|I - B|)^{n/2} \). As a function of coefficient \( b_{ij} \),

\[
|I - B| = (-1)^{i+j+1}C_{ij}b_{ij} + \sum_{k=1, k \neq j}^{k=m} (-1)^{i+k+1}C_{ik}b_{ik},
\]

where \( C_{ij} \) is the determinant of respective co-factor of \( I - B \), \( b_{ik} \equiv 0 \) if there is no edge \( Y_i \leftarrow Y_j \), and \( b_{ii} \equiv -1 \). The resulting density function of \( b_{ij} \) given \( D \) and \( \Theta \backslash \{b_{ij}\} \) is

\[
p(b_{ij}|D \backslash \{b_{ij}\}, D) \propto |b_{ij} - \kappa_{ij}|^n \exp \left\{ -\frac{(b_{ij} - \ell_{ij}'/s_{ij}')^2}{2s_{ij}'} \right\},
\]

where

\[
\kappa_{ij} \equiv C_{ij}^{-1} \sum_{k=1, k \neq j}^{k=m} (-1)^{k-j+1}C_{ik}b_{ik}
\]

and \( \{\ell_{ij}', s_{ij}'\} \) are defined as in Equation (42). Notice that if \( \ell_{ij}'/s_{ij}' = \kappa_{ij} \), this posterior could be transformed into two renormalized and symmetric gamma density functions stiched at \( \kappa_{ij} \), but in general this will not be true. Standard algorithms such as Metropolis-Hastings can be applied to sample from this posterior within a Gibbs procedure.

4.3 Sampling latent variables

In Gaussian DMG models with latent variables, we sample hidden values in order to complete the data and use the above posteriors to sample parameters conditioned on the completed dataset. Sampling latent variables is part of the Gibbs procedure: we condition on the data and the parameter set \( \Theta \).

Sampling a latent variable \( X_h^{(d)} \) given the observed variables is a matter of using the properties of the multivariate Gaussian distribution. Since one is conditioning on \( \Theta \), the joint distribution of latents \( X^{(d)} \) and observables \( Y^{(d)} \) is a Gaussian with mean \( \mu(\Theta) \equiv (I - B)^{-1}\mu \)
Bayesian Learning with Mixed Graph Models

The mean and covariance of $X^{(d)}$ given $\{Y^{(d)}, \Theta\}$ can be obtained as in multiple linear regression. The independence properties of the distribution, which can be read off the graph (Richardson, 2003), might be exploited to simplify this procedure. We highly recommend sampling the latent variables jointly, since the autocorrelation of the chain will be much higher otherwise. In our implementation, we always sample latent variables jointly.

4.4 Marginal likelihood: a variational Monte Carlo approach

While model selection of bi-directed graphs can be performed using a simple Monte Carlo procedure as seen in the previous Section, the same is not true in the full Gaussian DMG case. Approaches such as nested sampling (Skilling, 2006) can in principle be adapted to deal with the full case. For problems where there are many possible candidates to be evaluated, such a computationally demanding sampling procedure might be undesirable (at least for an initial ranking of graphical structures). As an alternative, we describe an approximation procedure for the marginal likelihood $p(D|G)$ by combining variational bounds (Jordan et al., 1998) with the $G$-Inverse Wishart samplers, and therefore avoiding a Markov chain over the joint model of coefficients and error covariances. This is described for acyclic DMGs only.

We adopt the following approximation in our variational approach, accounting also for latents $X$:

$$p(V, B, \mu, X|D) \approx q(V)q(B, \mu)$$

with $q(B, \mu)$ being a multivariate Gaussian over the non-zero elements of $B$ and $\mu$, and $q(X^{(d)})$ being also a Gaussian. The distribution $q(V)$ is a $G$-Inverse Wishart.

From Jensen’s inequality, we obtain the following lower-bound (Beal, 2003, p. 47):

$$\ln p(D|G) = \ln \int p(Y, X|V, B, \mu) p(V, B, \mu) dX dB dV d\mu$$

$$\geq \left< \ln p(Y, X|V, B, \mu) \right>_q(V)q(B, \mu)q(X)$$

$$+ \left< \ln p(V) \right>_q(V)$$

$$+ \left< \ln p(B, \mu) \right>_q(B, \mu) - \left< \ln q(X) \right>_q(X)$$

(47)

where this lower bound can be optimized with respect to functions $q(V), q(B), q(X)$. This can be done by iterative coordinate ascent, maximizing the bound with respect to a single $g(\cdot)$ function at a time.

The update of $q(V)$ is given by

$$q^{new}(V) = p_{G-IW}(\delta + d, U + \left< (I - B)S(I - B)^T \right>_q(X)q(B, \mu))$$

(48)

where $p_{G-IW}(\cdot)$ is the density function for a $G$-Inverse Wishart, and $S$ is the empirical second moment matrix summed over the completed dataset $(X, Y)$ (hence the expectation over $q(X)$) centered at $\mu$.

The updates for $q(B, \mu)$ and $q(X)$ are tedious but straightforward derivations, and described in Appendix B. The relevant fact about these updates is that they are functions of $\left< V^{-1} \right>_{q(V)}$. Fortunately, there is a relatively small cost to obtain these inverses from the
Monte Carlo sampler of Figure 5: from Bartlett’s decomposition, one can create a lower triangular matrix $B$ (by placing on the $i$th line the row vector $B_i^T$, followed by zeroes) and a diagonal matrix $\Gamma$ from the respective vector of $\gamma_i$’s. The matrix $V^{-1}$ can be computed from $(I - B)^T \Gamma^{-1} (I - B)$, and the relevant expectation computed according to the importance sampling procedure. However, for problems of moderate dimensionality\(^8\), the importance sampler is not recommended. The Gibbs sampler can be used instead. The disadvantage of the Gibbs sampler is that it is not as straightforward to obtain the inverse matrices $V^{-1}$ nor the scaling-up procedures of Section 6. The advantage of using the Gibbs procedure for computing averages is that the chains tend to mix very fast, being more reliable than the importance sampler with fewer draws in high dimensions.

At the last iteration of the variational maximization, the samples from $q(V)$ can then be used to compute the required averages in (47), obtaining a bound on the marginal log-likelihood of the model. Notice that the expectation $\langle \ln p(V)/q(V) \rangle_{q(V)}$ contains the entropy of $q(V)$, which will require the computation of $\mathcal{G}$-inverse Wishart normalizing constants.

For large problems, the cost of this approximation might still be prohibitive. One might consider decoupling parameters of the error covariance matrix $V$ to obtain a simpler approximation. It is interesting to notice that a natural deterministic approximation derived by choosing a specific ordering $\prec$, a reparameterization of $V$ in terms of $\Phi_G$, and the approximation family

$$q(\Phi_G) = \prod_{i=1}^m q(\gamma_i) q(B_{i,sp_{\prec}(i)}|\gamma_i)$$

will still not result in closed-form update steps for $q(\Phi_G)$, since $p(\Phi_G)$ is a complicated non-linear function of its parameters.

An option is to partially parameterize $V$ in terms of ancillary latents and another submatrix distributed as a $\mathcal{G}$-inverse Wishart, but details on how to best do this partition are left as future work (this approximation will be worse but less computationally expensive if ancillary latents are independent of the coefficient parameters in the variational $q(\cdot)$ distribution).

We emphasize that the results present in this section are alternatives that did not exist before in previous approaches for learning mixed graph structures through variational methods (e.g., Silva and Scheines, 2006).

### 5. Discrete models: the probit case

Constructing a discrete mixed graph parameterization is not as easy as in the Gaussian case. Advances in this area are described by Drton and Richardson (2008), where a complete parameterization of binary bi-directed graph models is given. Since the main goal of this paper is to present Bayesian approaches for learning using mixed graph models, inference with unconstrained mixed graph discrete models is not going to be any computationally

---

\(^8\) We observed a high ratio of the highest importance weight divided by the median weight in problems with dimensionality as low as 15. However, notice that in practice the error covariance matrix $V$ has a block diagonal structure, and only the size of the largest block is relevant. This is explained in more detail in Section 6.
Bayesian Learning with Mixed Graph Models

easier than the case for Markov random fields, which has been labeled as doubly-intractable (Murray et al., 2006).

Instead, in this paper we will focus on a class of discrete models that has been widely used in practice: the probit model (Bartholomew and Knott, 1999). This model is basically a projection of a Gaussian distribution into a discrete space. It also allows us to build on the machinery developed in the previous sections. We will describe the parameterization of the model for acyclic DMGs, and then proceed to describe algorithms for sampling from the posterior distribution.

5.1 Parameterizing models of observable independencies

A probit model for the conditional probability of discrete variable \( Y_i \) given a set of variables \( \{Y_{i1}, ..., Y_{ik}\} \) can be described by the two following relationships:

\[
Y_i^* = \mu_i + b_{i1}Y_{i1} + b_{i2}Y_{i2} + \cdots + b_{ik}Y_{ik} + \epsilon_i
\]

\[
P(Y_i = v_i^j \mid Y_i^*) = 1(\tau_{i-1}^j \leq Y_i^* < \tau_i^j)
\] (50)

where \( P(\cdot) \) is the probability function of a given random variable, as given by the context, and \( 1(\cdot) \) is the indicator function. \( Y_i \) assumes values in \( \{v_i^1, v_i^2, ..., v_{\kappa(i)}^i\} \). Thresholds \( \{\tau_0^i < \tau_1^i < \cdots < \tau_{\kappa(i)}^i = \infty\} \) are used to define the mapping from continuous \( Y_i^* \) to discrete \( Y_i \). This model has a sensible interpretation for ordinal and binary values as the discretization of some underlying latent variable \( (UV) \) \( Y_i^* \). Such a UV is a conditionally Gaussian random variable, which follows by assuming normality of the error term \( \epsilon_i \). This formulation, however, is not appropriate for general discrete variables, which are out of the scope of this paper. Albert and Chib (1993) describe alternative Bayesian treatments of discrete distributions not discussed here.

Given this binary/ordinal regression formulation, the natural step is how to define a graphical model accordingly. As a matter of fact, the common practice does not strictly follow the probit regression model. Consider the following example: for a given graph \( \mathcal{G} \), a respective graphical representation of a probit model can be built by first replicating \( \mathcal{G} \) as a graph \( \mathcal{G}^* \) with continuous underlying latent variables (UVs) in place of each respective original vertex. To each vertex \( Y_i^* \) in \( \mathcal{G}^* \), we then add a single child \( Y_i \). We call this the Type-I UV model. Although there are arguments for this approach (see, for instance, the arguments by Webb and Forster (2006) concerning stability to ordinal encoding), this is a violation of the original modelling assumption as embodied by \( \mathcal{G} \): if the given graph is a statement of conditional independence constraints, it is expected that such independencies will be present in the actual model. The Type-I formulation does not fulfill this basic premise: by construction there are no conditional independence constraints among the set of variables \( \mathbf{Y} \) (the marginal independencies are preserved, though). This is illustrated by Figure 7(b), where the conditional independence of \( Y_1 \) and \( Y_3 \) given \( Y_2 \) disappears.

An alternative is illustrated in Figure 7(c). Starting from the original graph \( \mathcal{G} \) (as in Figure 7(a)), the probit graph model \( \mathcal{G}^* \) shown in the Figure is built from \( \mathcal{G} \) by the following algorithm:

1. add to empty graph \( \mathcal{G}^* \) the vertices \( \mathbf{Y} \) of \( \mathcal{G} \), and for each \( Y_i \in \mathbf{Y} \), add a respective UV \( Y_i^* \) and the edge \( Y_i^* \rightarrow Y_i \);
Figure 7: The model in (a) has at least two main representations as a probit network. In (b), the original structure is given to the underlying variables, with observed variables being children of their respective latents. In (c), the underlying variable inherits the parents of the original variable and the underlying latents of the spouses.

2. for each $Y_i \rightarrow Y_j$ in $G$, add edge $Y_i \rightarrow Y_j^*$ to $G^*$;

3. for each $Y_i \leftrightarrow Y_j$ in $G$, add edge $Y_i^* \leftrightarrow Y_j^*$ to $G^*$;

We call this the Type-II UV model, which has the following property:

**Theorem 4** Suppose $G$ is acyclic with vertex set $Y$. $Y_i$ and $Y_j$ are m-separated given $Z \subseteq Y \setminus \{Y_i, Y_j\}$ in $G$ if and only if $Y_i$ and $Y_j$ are m-separated given $Z$ in $G^*$.

**Proof:** We first show that there is a mapping from each path in $G$ to a path in $G^*$, and vice-versa (such mappings are not inverse functions of each other, since the number of paths in $G^*$ is larger than in $G$). By construction, all bi-directed edges in $G^*$ have two UVs as endpoints, with an one-to-one mapping between each $Y_s^* \leftrightarrow Y_t^*$ in $G^*$ and each $Y_s \leftrightarrow Y_t$ in $G$. All directed edges in $G^*$ are of two types: $Y_s \rightarrow Y_t^*$, with $s \neq t$, or $Y_s^* \rightarrow Y_t$. Therefore, one can show that any path $P^*$ in $G^*$ corresponds to an unique path $P$ in $G$ obtained by relabeling each $Y^*$ as $Y$, and by collapsing any $Y \rightarrow Y$ edges that might result from this relabeling into a single vertex $Y$. A mapping in the opposite direction is analogous as given by the construction rule of Type-II models.

A collider in a path is any vertex within a head-to-head collision in the path, i.e., any vertex $Y_t$ where the preceding and the next vertex in the path are connected to $Y_t$ with an edge (directed or bi-directed) into $Y_t$. $Y_i$ and $Y_j$ are m-separated by $Z$ in an acyclic DMG if and only if there is no active path connecting $Y_i$ and $Y_j$. Like in d-separation, a path is active if all of its colliders have some descendant in $Z$, and none of its non-colliders is in $Z$ (Richardson, 2003). The mapping between paths $P$ and $P^*$ is such that, $Y_t$ is a collider in $P$ if and only if $Y_t$ is in $P^*$ and is a collider, or $Y_t^*$ is in $P^*$ and is a collider. Since by construction any $Y_t^*$ will have the same $Y$-descendants in $G^*$ as $Y_t$ has in $G$, and $Z \subset Y$, the result follows. □

The parameterization of the Type-II UV model follows from the definition of probit regression: the relationship of $Y_i$ given its parents in $\{Y_{i1}, ..., Y_{ik}\}$ in $G$ is given as in Equation (50), while the vector of error terms $\{\epsilon_1, \epsilon_2, \ldots, \epsilon_m\}$ follows the multivariate normal $N(0, V)$. 24
The entry corresponding to the covariance of $\epsilon_i$ and $\epsilon_j$ is assumed to be zero if there is no bi-directed edge $Y_i \leftrightarrow Y_j$ in $\mathcal{G}$.

In this paper, we will focus on algorithms for Type-II models only. The approach here described can be easily adapted to cover Type-I models. We say that Type-II models are models of observable independencies, since independencies hold even after marginalizing all UVs.

5.2 Priors

As in the Gaussian case, there are three parameters $\{\mu, B, V\}$. We will use exactly the same family of priors.

The other parameters in the probit family are the thresholds $T_i = \{\tau^1_i = -\infty < \tau^2_i < \cdots < \tau^{\kappa(i)}_i = \infty\}$ for each random variable $Y_i$ with $\kappa(i)$ different values. We treat each $T_i$ independently. For simplicity, we adopt as the joint prior for $\{\tau^1_i, \tau^2_i, \ldots, \tau^{\kappa(i)}_i\}$ proportional to a joint Gaussian subject to ordering constraints

$$p(\tau^1_i, \tau^2_i, \ldots, \tau^{\kappa(i)}_i) \propto \prod_{i=1}^{\kappa(i)-1} (s_{\tau^i})^{-1/2} \exp \left( -\frac{(\tau^i - c_{\tau^i})^2}{2s_{\tau^i}} \right), \text{ s.t. } \tau^1_i < \tau^2_i < \cdots < \tau^{\kappa(i)-1}_i \quad (51)$$

For any given binary variable $Y_i$, we fix the threshold vector to $\{\tau^0_0 = \infty, \tau^1_1 = 0, \tau^2_2 = \infty\}$.

5.3 Algorithm

As before, we provide a Gibbs sampling scheme to sample parameters $\Theta = \{\mu, B, V, T\}$ from the posterior distribution given dataset $\mathcal{D} = \{Y^{(1)}, Y^{(2)}, \ldots, Y^{(n)}\}$. For the purposes of the Gibbs procedure, we augment the dataset with the underlying variables $\mathcal{D}^* = \{Y^{*(1)}, Y^{*(2)}, \ldots, Y^{*(n)}\}$ at each sampling step.

From the set of structural equations

$$Y^{*(d)} = \mu + BY^{(d)} + \epsilon \quad (52)$$

it follows that the conditional distribution of $Y^{*(d)}$ given the $\mathcal{D} \cup \mathcal{D}^* \cup \Theta$ is a truncated Gaussian with mean $\mu + BY^{(d)}$ and covariance matrix $V$. The truncation levels are given by the thresholds and observed data $Y^{(d)}$; for each $Y^{(d)}_i = s_{\tau^i}$, the range for $Y^{*(d)}_i$ becomes $[\tau^i_{i-1}, \tau^i_i]$. Sampling from a truncated Gaussian is a standard procedure. We used the algorithm of Kotecha and Djuric (1999) in our implementation.

To sample $V$ from its conditional, we will rely on the following result.

Proposition 5 Let $\mathcal{G}$ be an acyclic DMG, and $(\mu, B, V, T)$ be the respective set of parameters that defines the probit model. For a fixed $(\mu, B, T)$, there is a bijective function $f_{B\mu T}(\cdot)$ mapping $Y^*$ to $\epsilon$. This is not true in general if $\mathcal{G}$ is cyclic.

Proof: If the graph is acyclic, this follows directly by recursively solving the model equations, starting from those corresponding to $Y^*_j$ vertices with no parents.

For cyclic graphs, the following model provides a counter-example. Let the graph be $Y^*_1 \rightarrow Y_1 \rightarrow Y^*_2 \rightarrow Y_2 \rightarrow Y^*_1$. Let the model be $Y^*_1 = Y_2 + \epsilon_1, Y^*_2 = Y_1 + \epsilon_2$, i.e., $b_{12} = b_{21} = 1$.
and \( \mu = 0 \). Let the variables be binary, with a threshold at zero \((Y_i = 1 \text{ if and only if } Y_i^* \geq 0)\). Then the two instantiations \((Y_1^* = -0.8, Y_2^* = 0), (Y_1^* = 0.2, Y_2^* = 1)\) imply the same pair \((\epsilon_1 = -0.8, \epsilon_2 = 0)\). \(\Box\)

The negative result for discrete models with cycles is the reason why such models are out of the scope of the present paper.

Due to this bijection (and the determinism linking \(Y\) to \(Y^*\)), the density \(p(V | \Theta \setminus V, D, D^*) = p(V | \Theta \setminus V, D^*) = p(V | \Theta \setminus V, Y^*)\) is equivalent to

\[
p(V | \Theta \setminus V, Y^* = y^*) &= p(V | \mu, B, T, Y^* = y^*, \epsilon = f_{B\mu T}(y^*)) \\
&= p(V | \mu, B, T, \epsilon = f_{B\mu T}(y^*)) \\
&\propto p(\epsilon = f_{B\mu T}(y^*), T, V) p(V | \mu, B, T) \tag{53}
\]

For the given dataset \(D \cup D^*\), define \(S^*\) as the sum of \((Y_{t}^{*}(d) - \mu - B Y_t^{(d)}) (Y_{t}^{*(d)} - \mu - B Y_t^{(d)})^T\) over all \(d \in \{1, \ldots, n\}\). Since \(p(V | \epsilon) \propto p(\epsilon | V) p(V)\), where \(p(\epsilon | V)\) is normal with zero mean and covariance matrix \(V\), the posterior for \(V\) given all other parameters and variables is

\[
V | \{\Theta \setminus V, D, D^*\} \sim \mathcal{G}-IW(\delta + n, U + S^*) \tag{54}
\]

Sampling \(B\) and \(\mu\) is analogous to the Gaussian case, except that we have to consider that the left side of the structural equations now refer to \(Y^*\). We give the explicit conditional for \(\mu_i\), with the conditional for \(b_{ij}\) being similarly adapted from Section 4. The posterior for \(\mu_i\) is given by a normal \(N((s_i^\mu)^{-1}m_i^\mu, s_i^\mu)\) where

\[
\begin{align*}
 s_i^\mu &= \frac{1}{s_i^\mu} + n (V^{-1})_{ii} \\
 c_i^\mu &= c_i^\mu - n \sum_{t=1,t\neq i}^m (V^{-1})_{it} \mu_t + \sum_{d=1}^n \sum_{t=1}^m (V^{-1})_{it} \left( Y_t^{*}(d) - \sum_{p_t} b_{tp_t} Y_{tp_t}^{(d)} \right), \tag{55}
\end{align*}
\]

Finally, in order to sample a threshold \(\tau_i^f\), one has to consider two things:

- \(\tau_i^f\) cannot be large to the point where some \(Y_i^{*}(d)\) associated with \(Y_i^{(d)} = v_{i+1}^f\) falls out of the interval \([\tau_i^f, \tau_{i+1}^f]\)
- \(\tau_i^f\) cannot be small to the point where some \(Y_i^{*}(d)\) associated with \(Y_i^{(d)} = v_i^f\) enters the interval \([\tau_i^f, \tau_{i+1}^f]\)

Hence, the infimum of \(\tau_i^f\) given all other parameters, underlying latent variables (UVs) and data is

\[
\tau_{\text{inf}} = \max_{d \in \{1, \ldots, n\}} Y^{*}(d) | Y^{*(d)} \in [\tau_{i-1}^f, \tau_i^f] \tag{56}
\]

and the supremum is
\[ \tau_{sup} \equiv \min_{d \in \{1, \ldots, n\}} Y^{*\star}(d) \mid Y^{*\star}(d) \in [\tau_i^l, \tau_i^u] \] (57)

and the conditional distribution of \( \tau_i^l \) is therefore a truncated Gaussian with mean \( c_i^{\tau_l} \), variance \( s_i^{\tau_l} \), and support \( (\tau_{inf}, \tau_{sup}) \).

5.4 A note on identifiability

The scale of the underlying latent variables in the probit model is arbitrary. As such, it has been often suggested that such latents should have constant (e.g., unity) variance (Pitt et al., 2006). There are two usual arguments for fixing the variance: improving the interpretability of the model, and improving the mixing of the Markov chain.

The interpretability argument is not particularly appealing within the Bayesian setting with proper priors, such as the one proposed in this paper: the posterior distribution of the parameters is well-defined by the prior uncertainty and the data.

The goal of improving the mixing of the chain might be important: if some parameters can assume arbitrary values and still allow for the same model over the observables, then fixing such parameters may help sampling by eliminating largely flat regions from the posterior (which will happen for large datasets and broad priors). In practice, however, scaling UVs might not be advantageous. In some cases it might increase the computational cost of each sampling step, while sampling from the non-scaled model might work just fine. Many MCMC algorithms work well on highly unidentifiable models such as multilayer perceptrons (Neal, 1996). In our experiments, we do not use any scaling.

5.5 Remarks

It is clear that the given approach can be generalized to other generalized linear models by changing the link function that maps underlying latent variables (UVs) to observables. For instance, a model containing discrete and continuous variables can be constructed by using the identity link function instead of probit for the continuous variables. Notice that the continuous variables will not necessarily be marginally Gaussian if some of its parents are discrete. Other link functions will have different parameters besides thresholds, such as in multivalued ("polychotomous") discrete distributions. A Bayesian account of Gaussian copula models is given by Pitt et al. (2006), to which a DMG-based family could in principle be defined. For continuous, marginally non-Gaussian, variables joined by a Gaussian copula, it is possible that all link functions are invertible. In this case, it is easier in principle to define cyclic models through Type-I UV models (e.g., Figure 7(b)) while preserving the observable independencies.

6. Scaling up: factorizations and perfect sequences

Each Monte Carlo sampling step for the given mixed graph models is theoretically tractable, but not necessarily practical when the dimensionality \( m \) of the data is high. By using clever factorizations of the graph and ordering of the variables, it is possible to sometimes scale to high-dimensional problems. The main goal of this section is to describe approaches to

---

9. For simplicity of exposition, this assumes that \( Y_i \) will assume each possible value at least once in \( D \).
minimize the run-time of the marginal likelihood computation for bi-directed graphs, which is also important for computing variational bounds on full DMGs. We start, however, with a discussion on the proper factorization of the coefficient matrix $B$ in the Gibbs sampler for acyclic models.

### 6.1 Factorizations

Our prior for coefficients $\{b_{ij}\}$ is fully factorized. In directed acyclic graphs, this is particularly advantageous: coefficients corresponding to edges into different nodes are independent in the posterior\(^{10}\). One can then jointly sample a whole set of $\{b_{ij}\}$ coefficients with same $i$ index, with no concern for the other coefficients. Figure 8(a) illustrates this factorization. This means that, in Equation (42), the summation over $t$ does not go over all variables, but only for $t = i$. This also follows from the fact that $(V)_{ii}^{-1} = 0$ unless $i = t$, since $V$ is diagonal.

In ADMGs, however, this is not true anymore. For any pair of vertices linked by a path of bi-directed edges, e.g., $Y_i \leftrightarrow Y_{i+1} \leftrightarrow \cdots \leftrightarrow Y_t$, one will have in general that $(V)_{it}^{-1} \neq 0$. This can be shown by using the graphical properties of the model when conditioning on some arbitrary datapoint $Y$:

**Proposition 6** Suppose $G$ is acyclic DMG with vertex set $Y$. Let $G'$ be the DMG obtained by augmenting $G$ with a vertex for each parameter $b_{ij}$ and a respective edge $b_{ij} \rightarrow Y_i$. Then if there is a bi-directed path $Y_i \leftrightarrow \cdots \leftrightarrow Y_t$ in $G$, $\{b_{ij}, b_{tv}\}$ are not m-separated given $Y$ in $G'$.

**Proof:** The joint model for $\{Y, B\}$ with factorized priors on $B$ is clearly Markov with respect to $G'$. The sequence of bi-directed edges between $Y_i$ and $Y_t$ implies a path between $b_{ij}$ and $b_{tv}$ where every vertex but the endpoints is a collider in this path. Since every collider

---

10. Sampling in Gaussian DAG models is still necessary if the model includes latent variables (Dunson et al., 2005).
is in \( \mathbf{Y} \), this path is active. □

This Proposition is illustrated by Figure 8(b). The practical implication is as follows: m-connection means that there is no further graphical property that would entail \( (\mathbf{V})^{-1} = 0 \) (i.e., only particular parameter cancellations on the determinant, unlikely to happen in practice, would happen to generate such zeroes).

Consider the maximal sets of vertices in an ADMG such that each pair of elements in this set is connected by a path of bi-directed edges. Following Richardson (2003), we call this a district\(^{11} \). It follows that it is not possible in general to factorize the posterior of \( \mathbf{B} \) beyond the set of districts of \( \mathcal{G} \). Figure 8(c) illustrates a factorization. Fortunately, for many DMG models with both directed and bi-directed edges found in practical applications (e.g., Bollen, 1989), the maximum district size tends to be considerably smaller than the dimensionality of the problem.

### 6.2 Perfect sequences

Despite the previous claim, there are applications where the size of the maximum district is naturally large. This is particularly true in structured classification and regression models where the dependencies within vector \( \mathbf{Y} \) given covariates \( \mathbf{Z} \) are represented as a large bi-directed graph. For instance, this appears in the relational model of Silva et al. (2007), where a simplified DMG model is used due to computational constraints. It is thus important to speed up marginal likelihood evaluations for models with large districts.

Without loss of generality, assume our graph \( \mathcal{G} \) is a bi-directed graph with a single district, since the problem can be trivially separated into the disjoint bi-directed components. We will consider the case where the bi-directed graph is sparse: otherwise there is little to be gained by exploring the graphical structure. In that case, we will assume that the largest number of spouses of any node in \( \mathcal{G} \) is bounded by a constant \( \kappa \) that is independent of the total number of nodes, \( m \). The goal is to derive algorithms that are of lower complexity in \( m \) than the original algorithms.

The bottleneck of our procedure is the computation of the \( \Sigma^{-1}_{\text{sp}(i),\text{sp}(i)} \) matrices, required in the mapping between independent and dependent Bartlett parameters (Equation 16), as well as computing the determinants \( |\Sigma_{\text{sp}(i),\text{sp}(i)}| \). Since in sparse districts \( \text{sp}(i) \) grows linearly with \( m \), the cost of a naive algorithm for a single sampling step is \( O(m^3) \) per node. Iterating over all nodes implies a cost of \( O(m^4) \) for a Monte Carlo sweep. Therefore, our goal is to find a procedure by which such mappings can be computed in less than \( O(m^3) \) time. The general framework is reusing previous inverses and determinants instead of performing full matrix inversion and determinant calculation for each \( Y_i \). The difficulty on applying low-rank updates when we traverse the covariance matrix according to \( \prec \) is that the sets of non-spouses \( \text{sp}(i) \) and \( \text{sp}(i+1) \) might differ arbitrarily. We want sensible orderings where such sets vary slowly and allow for efficient low-rank updates, if any.

The foundation of many scaling-up procedures for graphical models is the graph decomposition by clique separators (Tarjan, 1985), usually defined for undirected graphs: the

---

\(^{11}\) Kang and Tian (2005) call such structures *c-components* and reserve the word “district” to refer to the function mapping a vertex to its respective c-component, as originally introduced by Richardson (2003). We choose to overload the word and call “district” both the structure and the mapping.
definition for bi-directed graphs is identical. Such a decomposition identifies overlapping prime subgraphs \( \{G_{P(1)}, G_{P(2)}, \ldots, G_{P(k)} \} \) of the original graph \( G \). A prime graph is a graph that cannot be partitioned into a triple \((Y', S, Y'')\) of non-empty sets such that \( S \) is a complete separator (i.e., \( S \) is a clique and removing \( S \) disconnects the graph). Notice that a clique is also a prime subgraph.

The prime components of a graph can be ordered in what is usually called a perfect sequence \( \{Y_{P(1)}, \ldots, Y_{P(k)} \} \) of subsets of \( Y \) (Roverato, 2002). Define \( H_j \equiv Y_{P(1)} \cup \cdots \cup Y_{P(j)} \) as the history of the perfect sequence up to the \( j \)-th subgraph. Let \( R_j \equiv Y_{P(j)} \setminus H_{j-1} \) be the residual of this history (with \( R_1 \equiv Y_{P(1)} \)), and \( S_j \equiv H_{j-1} \cap Y_{P(j)} \) the separator. The triple \((H_{j-1} \setminus S_j, S_j, R_j)\) forms a decomposition of the subgraph of \( G \) induced by the vertex set \( H_j \).

Surprisingly, although bi-directed and undirected graph models have very different Markov properties (in undirected models, conditioning removes dependencies; in bi-directed models, it adds dependencies), perfect prime graph sequences prove to be also useful, but in an entirely different way. The next subsection describes the use of prime graph decompositions in a particularly interesting class of bi-directed graphs: the decomposable case. The general case is treated in the sequel.

### 6.2.1 Decomposable models

In a recursively decomposable graph, all prime subgraphs are cliques. The resulting decomposition can be interpreted as a hypergraph where nodes are the maximal cliques of the original graph, and edges correspond to the separators. In the statistics literature, a decomposable model is defined as a model that is Markov to a recursively decomposable undirected graph (Lauritzen, 1996). Its widespread presence on applications of Markov random fields is due to nice computational properties, with tree-structured distributions being a particular case.

Our definition of bi-directed decomposable models is analogous: a model that is Markov to a recursively decomposable bi-directed graph.

Given the residual sequence \( \{R_1, R_2, \ldots, R_k\} \) obtained through a perfect sequence of maximal cliques of \( G \), we define a perfect ordering \( < \) by numbering nodes in \( R_t \) before nodes in \( R_1, \ldots, R_{t-1}, 1 \leq t \leq k \) and ordering nodes according to this numbering\(^{12}\). Any ordering that satisfies this restriction is a perfect ordering. Such an ordering has the following property.

**Theorem 7** Let \( G \) be a recursively decomposable bi-directed graph such that the index of its vertices \( Y = \{Y_1, Y_2, \ldots, Y_m\} \) follows a perfect ordering \( < \). Then for each \( 1 < i \leq m \), the set \( \{Y_1, Y_2, \ldots, Y_{i-1}\} \) can be partitioned as \( \bigcup_{t=1}^{K(i)} \mathcal{V}_t \) such that

1. for each \( Y_t \in \mathcal{V}_t \) and \( Y' \in \mathcal{V}_{t'}, t \neq t', \) there is no path in \( G \) connecting these two vertices that includes a vertex in \( \{Y_1, Y_2, \ldots, Y_{i-1}\} \setminus \mathcal{V}_t \cup \mathcal{V}_{t'} \);

2. for each \( \{Y_p, Y_q\} \subseteq \mathcal{V}_t \), if \( Y_p \) is a spouse of \( Y_i \), and \( Y_q \) is a non-spouse of \( Y_i \), then \( p > q \);

\(^{12}\) Lauritzen (1996) describes other uses of perfect numberings in undirected graphs.
Bayesian Learning with Mixed Graph Models

\[ V_1 = \{Y_1, Y_2, Y_3\} \]
\[ V_2 = \{Y_4, Y_5\} \]
\[ V_3 = \{Y_6\} \]

Figure 9: On the left, we have a bi-directed graph of 7 vertices arranged and ordered such that nodes are numbered by a depth-first numbering starting from “root” \( Y_7 \), with \( \{Y_1, Y_2, Y_4, Y_6\} \) being leaves. Vertices \( \{Y_1, Y_2, \ldots, Y_6\} \) can be partitioned as the union \( \bigcup_{t=1}^{3} V_t \), as illustrated on the right.

**Proof:** The result is easier to visualize in trees. One can take as a perfect ordering some depth-first ordering for a given choice of root. Then for each vertex \( Y_i \), the set \( \{Y_1, Y_2, \ldots, Y_{i-1}\} \) is partitioned according to the different branches “rooted” at \( Y_i \). The starting point of each branch is a spouse of \( Y_i \), and all other vertices are non-spouses of \( Y_i \). The ordering result then follows directly from the definition of depth-first traversal. This is illustrated in Figure 9.

For the general decomposable case, we will consider a proof by contradiction. If the premise is not true, then there is some bi-directed path \( Y_q \leftrightarrow \cdots \leftrightarrow Y_p \), where: all elements in this path are indexed before \( Y_i \); \( Y_q \) is a non-spouse of \( Y_i \); \( Y_p \) is a spouse of \( Y_i \); and \( q > p \). Assume without loss of generality that this path is of length 1, i.e., \( Y_q \leftrightarrow Y_p \) is in \( G \) (otherwise longer paths satisfying the given condition will not exist).

By assumption, since \( \{Y_i, Y_q, Y_p\}, i > q > p \), is not a clique, there is some history-separator-residual decomposition \( \{H_{t-1} \backslash S_t, S_t, R_t\} \), such that \( Y_i \in H_{t-1}, \{Y_q, Y_p\} \subseteq R_t \), and \( Y_q \in Y_{P(t)} \). \( Y_i \) cannot be in the separator, since it is not adjacent to \( Y_q \) (remember \( S_t \cup \left( R_t \cap Y_{P(t)} \right) = Y_{P(t)} \) is a clique). But this is contradicts the fact that \( Y_i \) and \( Y_p \) are adjacent, since \( S_t \) should be a separator for this pair. □

Let \( \Sigma \) be the covariance matrix of a bi-directed decomposable model with graph \( G \), where \( \Sigma \) follows a \( G \)-inverse Wishart distribution. Let \( \prec \) be a perfect ordering for \( G \). What follows is useful to both the algorithm for computing normalizing constants and the importance sampler algorithm of Figure 5. By the construction of Bartlett’s decomposition, mapping between independent and dependent Bartlett parameters is given by

\[
\Sigma_{sp \prec (i), nsp \prec (i)} = \Sigma_{nsp \prec (i), sp \prec (i)}^{-1}
\]

the computational bottleneck being the inversion. Notice this corresponds to the multiple regression coefficients of \( sp \prec (i) \) on \( nsp \prec (i) \). But according to Theorem 7, using a perfect
6.2.2 Non-decomposable models

In a non-decomposable model, some prime graphs \( Y_{P(t)} \) will no longer be cliques. As such, some preceding non-spouses of \( Y_i \in R_t \) will also be in \( Y_{P(t)} \), and nodes in separator \( S_t \) ordering implies that within each \( V_s \), all preceding non-spouses of \( Y_i \) are ordered before the preceding spouses. Elements \( \{Y_p, Y_q\} \) in different \( V_s \) are marginally independent given \( \{Y_1, \ldots, Y_{i-1}\} \backslash \{Y_p, Y_q\} \). This implies that the regression coefficient of spouse \( Y_p \) on non-spouse \( Y_q \) will be zero if \( Y_p \) and \( Y_q \) are on different components \( V_s \), and will be identical to the previously computed \( B_{p,q} \) if they are in the same component. Splitting the set \( \{Y_1, Y_2, \ldots, Y_{i-1}\} \) into preceding spouses \( Y_{sp,<}(i) \) and non-spouses \( Y_{nsp,<}(i) \), we have

\[
Y_{sp,<}(i) = B_{sp,<}(i,sp,<)(i) Y_{sp,<}(i) + B_{sp,<}(i,nsp,<)(i) Y_{nsp,<}(i) + \epsilon_{sp,<}(i)
\]

where each \( \epsilon_j \) is an independent Gaussian with variance \( \gamma_j \), and each element \( (sp,<)(i, q) \) in \( B_{sp,<}(i,nsp,<)(i) \) corresponds to the known regression coefficient of \( Y_{sp,<}(i) \) on the non-spouse \( Y_q \). Matrix \( B_{sp,<}(i,sp,<)(i) \) is defined analogously. Due to Theorem 7, all \( Y_{nsp,<}(i) \) precede \( Y_{sp,<}(i) \), and hence the regression coefficients of \( Y_{sp,<}(i) \) on \( Y_{nsp,<}(i) \) are given by

\[
\Sigma_{sp,<}(i,nsp,<)(i) = \Sigma_{sp,<}(i,nsp,<)(i) = (I - B_{sp,<}(i,sp,<)(i))^{-1} B_{sp,<}(i,nsp,<)(i)
\]

No inversion of \( \Sigma_{nsp,<}(i,nsp,<)(i) \) is ever necessary. Moreover, the determinant \( |\Sigma_{nsp,<}(i,nsp,<)(i)| \) is given by \( \prod_{q \text{ s.t. } Y_q \in nsp,<(i)} \gamma_q \).

Hence, calculating \( B_{i,nsp,<}(i) \) for all \( 1 \leq i \leq m \) according to a perfect ordering has as a bottleneck the inversion (of a triangular matrix) and multiplication in Equation (60), with a cost of \( O(m \kappa^2 + \kappa^2) \), \( \kappa \) being the maximum number of spouses for any given node. The cost of the remaining operations for the \( i \)-th stage in the importance sampler is \( O(\kappa^3) \). As a function of \( m \), the cost of the step falls from \( O(m^3) \) to \( O(m^2) \). Figure 10 illustrates the derivation of the new ordering in a tree-structured model.

Figure 10: The tree-structured (i.e., cycle-free) bi-directed graph in (a) has as maximal cliques the adjacent pairs. Such cliques can be ordered in a perfect sequence as shown in (b), where rectangles indicate the separators. Notice that \( R_1 = \{Y_A, Y_C\}, R_2 = \{Y_B\}, R_3 = \{Y_D\} \). One possible perfect ordering is \( \{Y_D, Y_B, Y_C, Y_A\} \).
are no longer guaranteed to be linked to all elements in $Y_{P(t-1)} \cup Y_{P(t)}$. As we shall see, the function of the perfect sequence is now to provide a sensible choice of which inverse submatrices $\{\Sigma_{W,W}^{-1}\}$, $W \subseteq Y$, to cache and reuse when computing $\Sigma_{\text{nsp},(i)\text{nsp},(i)}^{-1}$. The same will be done to compute determinants $|\Sigma_{\text{nsp},(i)\text{nsp},(i)}|$. 

Recall that the inverse of a partitioned matrix can be given by the following identity:

$$
\begin{pmatrix}
A & B \\
C & D
\end{pmatrix}^{-1} = \begin{pmatrix}
A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1} & A^{-1}B(D - CA^{-1}B)^{-1} \\
-(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1}
\end{pmatrix}
$$

(61)

In what follows, we once again assume that $\prec$ is a perfect ordering. Unlike in the decomposable case, the product $\Sigma_{\text{sp},(i)\text{sp},(i)}\Sigma_{\text{nsp},(i)\text{nsp},(i)}^{-1}$ does not simplify in general. Instead we will focus only on fast methods to compute $\Sigma_{\text{nsp},(i)\text{sp},(i)}^{-1}$.

Define $R_i \equiv R_{t,i} \cup R_{t,i+1} \cup \cdots \cup R_{t,k}$. We will consider first the case for $Y_i \in R_t \setminus S_{t+1}$. Let $\text{nsp},(i) \prec (t)$ be the preceding non-spouses of $Y_i$ in $R_{t+1}$, and let $\text{nsp},(i) \prec (i)$ be the preceding non-spouses of $Y_i$ in $R_t$. We index $\text{nsp},(i) \prec (i)$ by $i$ instead of $t$, since all elements in $R_{t+1}$ are non-spouses of elements in $R_t \setminus S_{t+1}$ by construction. In order to compute $\Sigma_{\text{nsp},(i)\text{nsp},(i)}^{-1}$, we consider its partitioned version

$$
\Sigma_{\text{nsp},(i)\text{nsp},(i)}^{-1} = \begin{pmatrix}
\Sigma_{\text{nsp},(i)\text{nsp},(i)}^{-1} & \Sigma_{\text{nsp},(i)\text{nsp},(i)}^{-1} \\
\Sigma_{\text{nsp},(i)\text{nsp},(i)}^{-1} & \Sigma_{\text{nsp},(i)\text{nsp},(i)}^{-1}
\end{pmatrix}^{-1}
$$

(62)

Let $\kappa_{\text{nsp}}$ be the maximum number of non-spouses among all $Y_i$ within any prime subgraph induced by $Y_{P(t)}$. By using the relation (61) where we assume the inverse of $A \equiv \Sigma_{\text{nsp},(i)\text{nsp},(i)}^{-1}$ was cached in the previous iteration, the cost of computing (62) is $O(m^2\kappa_{\text{nsp}}) + O(\kappa_{\text{nsp}}^3) = O(m^2\kappa_{\text{nsp}}) + O(m^3)$, while the cost of inverting it is $O(\kappa_{\text{nsp}}^3)$. This is dominated by $O(m^3)$. Treating $\kappa_{\text{nsp}}$ as a constant, this reduces the complexity of sampling the $i$-th row of $\Sigma$ from $O(m^3)$ to $O(m^2)$. A similar procedure applies to the computation of the determinant $|\Sigma_{\text{nsp},(i)\text{nsp},(i)}|$, using in this case the relationship (19).

The case for $Y_i \in R_{t,i} \cap S_{t+1}$ is similar. In this case, the common set of non-spouses will be $R_{t+2}$ (notice that is not possible for a vertex to belong both to $R_{t,i} \cap S_{t+1}$ and $R_{t+1} \cap S_{t+2}$, since $R_{t,i} \cap R_{t+1} = \emptyset$). The set $\text{nsp},(i)$ is defined with respect to $R_{t+1}$ instead of $R_t$.

6.3 Remarks

In Gaussian undirected models, the problem of covariance matrix sampling can also be reduced to sampling within each prime graph at the cost of $O(|P|^4)$, $|P|$ being the size of the largest prime component (Atay-Kayis and Massam, 2005). Since both $\kappa$ and $\kappa_{\text{nsp}}$ are $O(|P|)$, our procedure costs $O(m^2|P|^2 + |P|^4)$ per prime graph. Considering a number of $m/|P|$ prime graphs and $|P| < m$, the total cost is $O(m^3|P|)$, down from $O(m^4)$ (or $O(m^2|P|)$ for the decomposable case. The algorithm is quadratic for trees, for instance, with $|P| = 2$). For undirected models, the corresponding cost by sampling step using the perfect ordering decomposition is $O(m|P|^3)$. The higher-order dependency on $m$ in bi-directed models is to be expected, since the Markov blanket of any node $Y_i$ in a connected bi-directed graph is $V \setminus \{Y_i\}$. It is clear that inference with a given bi-directed graph model
will never scale at the same rate of a undirected model with the same adjacencies, but this does not justify adopting an undirected representation if it is ill-suited to the problem at hand. One has also to consider that in problems with directed and bi-directed edges, the actual maximum district size might be much smaller than the number of variables. For large problems, however, further approximation schemes will be necessary. Drton and Richardson (2005) describe some reduction techniques for transforming bi-directed edges into directed edges such that the resulting Gaussian model remains the same. As future work, such methods could be adapted to the $\mathcal{G}$-inverse Wishart sampling procedures and combined with the ordering techniques developed here into a single framework. It will also be interesting to develop similar schemes for the Gibbs sampler.

7. Experiments

We now evaluate the advantages of the Gaussian and probit models in Bayesian inference on real problems.

7.1 Industrialization and democratization study

Bollen (1989) describes a structural equation model of political and democratization factors within nations. “Democratization” and “industrialization” levels are abstract notions, but nevertheless of clearly observable impact. They are tied to empirical observations through different sets of indicators. For instance, an indicator of industrialization level is the gross national product. Hence, democratization and industrialization levels are here defined as scalar latent variables never observed directly, while the observed data is composed of indicators. In this model, there is a total of three indicators of industrialization, and four indicators of democratization. Democratization is measured in a longitudinal study, where data was collected in two years (1960 and 1965). The indicators of democratization are pooled expert opinions summarized in an ordinal number scaled from 1 to 10. Following Bollen, we will treat the model as multivariate Gaussian, which provides an excellent fit (a $p$-value greater than 0.3 using a chi-square test) for a sample of 75 countries.

The corresponding mixed graph is depicted in Figure 11, along with a description of all indicators. The graph is taken from Bollen (1989). Other hidden common causes affect the democratization indicators over time, but the nature of such hidden variables is irrelevant to the problem at hand: that is, the bi-directed edges are motivated by unmeasured causes of variability in the observed indicators that exist over time. For instance, the records of freedom of press in 1960 ($Y_4$) and 1965 ($Y_8$) co-vary due to other unmeasured factors not accounted by democratization factors. Two examples of applications with both implicit and explicit latent variables are illustrated as follows.

Example of application: embedding and clustering One important application of structural equation models is to compute what is sometimes known as factor scores (Dunson et al., 2005): an embedding of the observed data points into the latent space. This is useful for ranking objects according to different abstract criteria while making use of background knowledge on the relationship between the latent space and the observed data. The low-dimensional projections are also useful for visualization purposes. Figure 12 illustrates an application to the industrialization/democratization domain. Using our MCMC
1. Gross national product (GNP) 1960
2. Energy consumption per capita 1960
3. Percentage of labor force in industry 1960
5. Freedom of opposition 1960
6. Fairness of elections 1960
7. Elective nature of legislative body 1960
10. Fairness of elections 1965
11. Elective nature of legislative body 1965

Figure 11: A directed mixed graph representing dependencies between 11 observed political and economical indicators and three latent concepts (shaded nodes). Further discussions about this model are provided by Dunson et al. (2005) and Bollen (1989).
algorithm applied to the DMG model (details are given in the next section), we compute the expected value of Industrialization Level 1960 for each of the 75 countries. Those countries are then sorted by the increasing order of such levels, and the respective boxplots of Democratization Level 1960 and Democratization Level 1965 are generated. Dunson et al. (2005) use this information to, for instance, find clusters of countries in the latent space. An example of a cluster is the one formed by the bottom 16 countries in the industrialization level ranking: there is a visible break in the pattern of democratization levels at that point, as depicted in Figure 12.

Figure 12: An embedding of 75 countries in a two-dimensional latent space: democratization level in 1960 and 1965. Boxplots of the Bayesian posterior distribution of the projection in the two dimensions are depicted in the vertical axis. Countries are arranged in the horizontal axis by the increasing order of their posterior expected industrialization level. Figure adapted from Dunson et al. (2005).

Example of application: causal inference Another fundamental application of such models is to derive predictions of policies that are entailed by the given assumptions and data (Pearl, 2000; Spirtes et al., 2000; Bollen, 1989). Under the assumption that the edge directions in the graph in Figure 11 follow the causal direction in the world, and that there are no other unmeasured common causes between any pair of variables apart from those given by the bi-directed edges, conclusions about causal relations can be made. For instance, the effect of variations on the industrialization level of a country into its democratic level happens to be given by the coefficient parameters. The coefficient related to Industrialization Level 1960 → Democratization Level 1960 corresponds to a direct causal effect, while the coefficient of Industrialization Level 1960 → Democratization Level 1965 corresponds to a residual causal effect after conditioning on Democratization Level 1960.
Figure 13: Posterior distribution of parameters associated with the respective edges in the industrialization/democratization domain. Smoothed posterior obtained using the output of our Gibbs sampler and the density function of R 2.6.0.

See Pearl (2000) and Spirtes et al. (2000) for more information on how to read causal effects from a causal graphical model. Figure 13 illustrates the posterior distribution of said parameters given the data and the model. In particular, these plots provide evidence that the associations encoded by the respective parameters are significantly away from zero, and that much of the effect of industrialization on democratization can be explained by the previous democratization levels (by comparing the distribution in Figure 13(b) to 13(a)).

7.1.1 Evaluating the MCMC algorithm for different models

We will compare the behavior of the MCMC algorithm for three different models. The first is our standard DMG model. We fix to unity the coefficients corresponding to the edges Industrialization 1960 → Y₁, Democratization 1960 → Y₄ and Democratization 1965 → Y₈, since the scale and sign of the latent variables in arbitrary. The intercept terms of the equations for Y₁, Y₄ and Y₈ are set to zero, since the mean of the latents is also arbitrary. The resulting model is identifiable.

We apply the Gibbs sampling procedure to three different models. The Gaussian DMG model as described above, and two modified DAG models. The first DAG model is the one described by Dunson et al. (2005), where each bi-directed edge is substituted by an “ancillary” latent (as mentioned in Section 2.3). For instance, the pathway corresponding to Y₄ ↔ Y₈ is substituted by the chain Y₄ ← D₄₈ → Y₈, where D₄₈ is unobserved. Dunson et al. further assume that all covariances due to such ancillary latents are positive. As such, the coefficients from D_ij into {Y_i, Y_j} are set to unity, with the variance of D_ij corresponding to the residual covariance of {Y_i, Y_j} given their parents. Means of ancillary latents are fixed at zero.

However, even for covariance matrices with positive covariances, this parameterization is not complete. This result is evident from the fact that the variances of Y_i and Y_j will both be greater than their covariance, which is not true of covariance matrices in general. For this particular problem, however, this extra restriction provides no measureable difference
Figure 14: The first three plots show the initial 5,000 iterations of a run of the Gibbs sampling algorithm for the DMG model for three different parameters associated with edges in the graph. The last plot depicts the posterior distribution the error covariance associated with the edge $Y_7 \leftrightarrow Y_{11}$ (smoothed with the kernel density estimator in the R statistical software).

Figure 15: Comparison of the effective sample size of the MCMC algorithm applied to the three models, DMG, DAG with positive covariances (posDAG) and general DAG, as explained in the main text. The horizontal axis is the boxplot for each independent entry of the observed covariance matrix, 66 in total. The boxplots are obtained from 80 independent chains initialized randomly, where each chain runs for 50,000 iterations.

in terms of fitness. It does serve as a reminder, however, that “intuitive” parameterizations might hide undesirable constraints.

The second DAG model is an extension of the DAG model suggested by Dunson et al., the only difference being that the coefficients corresponding to edges $D_{ij} \rightarrow Y_i, i < j,$
Bayesian Learning with Mixed Graph Models

Figure 16: Comparison of the effective sample size of the MCMC algorithm applied to the three models. Here we plot the average effective sample sizes over 80 trials of 50,000 samples for each of the 66 entries of the covariance matrix. Points over the line indicate parameters where the DMG approach performed better.

are free to vary (instead of being fixed to 1). For this particular graphical structure, this parameterization is complete (there are no bi-directed cliques of size 3 or larger), but in general there are DMG models that cannot be parameterized this way, as discussed at length in Section 2.2.2. Notice also that because of chains such as Democratization 1960 → Y₄ ↔ Y₈ ← Democratization 1965, the set of independence constraints in this graph can only be represented by a DAG if we include the ancillary latents Dᵢⱼ. That is, just substituting bi-directed edges Yᵢ ↔ Yⱼ by some directed edge, Yᵢ → Yⱼ or Yⱼ → Yᵢ, will result in a different set of independence constraints.

We study the behavior of the MCMC algorithm for these three models\textsuperscript{13}. It turns out that the mixing properties of the chain are considerably affected by the choice of model. Recall that, in the Gibbs sampling algorithm for the DMG model, a whole row of the error

\textsuperscript{13} A few technical notes: we used the priors suggested in Dunson et al. (2005), except that we changed the confidence in the prior of the covariance of the error terms V to be smaller (in order to minimize the influence of the priors in the models, since in this particular problem the DMG and DAG models are nearly likelihood equivalent but not posterior distribution equivalent – the priors belong to different families). We used 1 degree of freedom in our G-Inverse Wishart, with the matrix parameter being the expected value of Dunson et al.’s prior. For the DAG models, we also used the G-inverse Wishart prior for the error terms, but where all error terms are independent. For the DAG model with a free coefficient per ancillary latent, we assigned a standard Gaussian prior to such coefficients. The chains were initialized randomly by sampling standard Gaussians for the coefficients and latent variables. Error covariance matrices were initialized to diagonal matrices with diagonal entries sampled uniformly in [1, 2]. Coefficient parameters were sampled jointly given the error covariance matrix and latent variables. Latent variables were also sampled jointly, given the parameters.
covariance matrix is sampled jointly conditioning on the other parameters. For the DAG models all entries of the error covariance matrix are independent and can be sampled jointly, but this requires conditioning on the ancillary latents, which do not exist in the DMG model and have to be sampled only in the DAG case.

For the majority of the covariance entries, the MCMC procedure mixed quite well, as illustrated in Figure 14. Notice how about 12% of the sampled DMG error covariances for $Y_7 \leftrightarrow Y_{11}$ were under zero, what could raise suspicion over the assumption of positive covariances. Autocorrelation is essentially zero for most parameters at a lag of 50. The degree of autocorrelation, however, varied significantly between the DMG model and each DAG model. The chains for the DMG model mixed considerably better. To summarize such behavior, we calculated the effective sample size of the samples obtained from several chains. The parameters of interest in this comparison are the independent entries in the $11 \times 11$ dimensional observed covariance matrix. This is a total of 66 parameters. The effective sample size statistics were obtained by 80 independent chains of 50,000 samples each, for the three models. For each chain and each parameter, we compute the desired statistic using the EFFECTIVESIZE function implemented in the R package coda, freely available in the Internet.

Results are summarized by boxplots in Figure 15. Parameters are ordered in the x-axis following the upper triangular covariance matrix, scanning it in the order \{$\sigma_{Y_1 Y_1}, \sigma_{Y_1 Y_2}, \ldots, \sigma_{Y_1 Y_{11}}, \sigma_{Y_2 Y_2}, \ldots, \sigma_{Y_{11} Y_{11}}$\}. White boxplots correspond to the distribution of effective sample size statistics with the DMG model across the 80 independent chains. Grey boxplots correspond to the two DAG variants. There is no significant difference between the behaviour of the Gibbs sampling procedure for the two DAG models. The procedure with the DMG model is clearly better behaved. As a summary statistic, the average effective sample size over 80 trials was steadily larger in the DMG outcome than in the positive DAG outcome (61 out of 66 parameters) and unconstrained DAG (59 out of 66). The comparison of averages is illustrated by Figure 16.

By caching the sufficient statistics of the data and factorizing the sampling procedure according to the districts of the graph, the running time for generating 50,000 samples out of the DMG model was of 34 seconds in a dual core Pentium IV 2.0 GHz. The time for the DAG models were substantially slower, at 58 and 60 seconds for the positive covariance and unconstrained covariance variants. This can be explained by the fact that sampling latent variables is very expensive, especially considering that in the given DAG models all ancillary latents become dependent when conditioning on the data. To summarize, the DMG approach allowed for a complete parameterization with significantly better mixing properties, while still resulting in a faster MCMC procedure.

### 7.2 Structure learning applications

When trying to find a point estimate of graphical structures (i.e., returning a single graph that explains the data well), simple approaches such as testing for marginal independencies are reasonable learning algorithms under the Gaussian assumption. The Bayesian approach, however, allows one to compute odds and distributions over graphs and graph statistics, e.g., the joint probability of small substructures (Friedman and Koller, 2003). Moreover, it is not clear how the independence test procedure controls for the predictive ability of the
Bayesian Learning with Mixed Graph Models

model, which is not a straightforward function of the edges that are selected due to the quantitative aspects of the dependencies.

We evaluate our Bayesian model selection contribution, focusing on the Monte Carlo sampler for bi-directed models. Concerning priors for graphs, Jones et al. (2005) points out that an uniform prior over symmetric graphs puts most of its mass in graphs with a number of edges that increases quadratically with the number of vertices. If the true model is sparse, this might not be a good choice of prior. Instead, we use the prior of Jones et al.:

\[ P(G|\beta) = \beta^{|E|}(1 - \beta)^{0.5m(m-1)-|E|} \]  

(63)

where \( \beta \) is a hyperparameter, \(|E|\) is the number of edges in \( G \), and \( m \) is the number of nodes. As suggested by Jones et al., we choose \( \beta = 0.5/(m - 1) \), which puts more mass on graphs with \( O(m) \) edges than the uniform prior.

We start with a brief synthetic study to compare the approach against a simple but effective approach based on the BIC approximation\(^{14}\). An experiment with gene expression data closes this subsection.

7.2.1 Synthetic studies

As a sanity check for the procedure, we generate synthetic 10-dimensional Gaussian data from models that are Markov with respect to a bi-directed graph. One hundred datasets of 50 datapoints each are generated, each coming from a different model\(^{15}\). We initially find a structure by marginal independence tests using the Fisher’s Z statistic at a 0.05 level. From this starting point, we perform two searches: one using the BIC score, and the other using the marginal likelihood with a \( G-IW \) prior\(^{16}\). Given the best model for each procedure, we evaluate the predictive log-likelihood on a test set of 2,000 points which are independently sampled for each of the 100 models.

14. The BIC approach is an asymptotically consistent score for selecting the maximum a posteriori Gaussian bi-directed graph model (Richardson and Spirtes, 2002).

15. The details of the simulated data are as follows: we start with DAG with no edges, with observed nodes \( \{Y_1, Y_2, \ldots, Y_{10}\} \) and hidden nodes \( \{X_1, X_2, X_3, X_4\} \). Each individual edge \( X_i \rightarrow Y_j \) is added with probability 0.35, and no other edges are allowed. We reject graphs with fewer than 10 edges. All coefficient parameters are sampled from a standard Gaussian, and variances from an uniform distribution in \([0, 1]\). The model over \( Y \) corresponds to a bi-directed graph, where the edge \( Y_i \leftrightarrow Y_j \) exists if and only if \( Y_i \) and \( Y_j \) have a common latent parent \( X_k \) in the DAG. We then store 50 samples for the \( Y \) variables in a dataset. The procedure is repeated 100 times with different parameters and graphical structures each time. The average number of edges in the resulting simulation was of 18.4 edges per graph.

16. In both cases, we center the data at the empirical mean of the training set and assume the data to have been generated from a zero-mean Gaussian. The \( G \)-Inverse Wishart is an empirical prior: a diagonal matrix with the training variance of each variable used as the diagonal. The number of degrees of freedom is set to 1. The search is a standard greedy procedure: we evaluate the marginal log-likelihood or BIC score for each graph that differs from the current candidate by one edge (i.e., graphs with one more or one fewer edge) and pick the one with the highest score. We stop when no improvement is possible.

41
The average difference in log-likelihood prediction\textsuperscript{17} between the structure learned with the fully Bayesian prior and the BIC-induced model is depicted in Figure 17(a). This is computed by conditioning on the learned structures (fully Bayesian vs. BIC maximum a posteriori graphs) and marginalizing over the posterior of the parameters. The parameter priors are those used for the structure learning step. This might be unfair for the BIC procedure, since it does not seek to maximize the finite sample posterior: hence we also show in Figure 17(b) the results obtained when the predictions given the BIC model are obtained by using the maximum likelihood estimators of the parameters. The average difference in the first case is 400.07, and only slightly less for the second case (389.63).

In terms of incorrect edge additions and deletions, the procedures behave about the same: an average of one third of the edges is missed, and 7\% of edges are incorrectly added (individual percentages are with respect to total number of possible mistakes in each graph). Unlike BIC, however, our procedure allows for different trade-offs by using different priors. It should also be pointed out that counting edge errors is just one possible measure. A more global quantitative score such as predictive log-likelihood takes into account, indirectly, the magnitude of the errors – although it is not a direct measure of model fitness.

\textsuperscript{17}The average difference in log-likelihood prediction is calculated by conditioning on the learned structures and marginalizing over the posterior of the parameters. This allows for a fair comparison between the fully Bayesian and BIC models. However, it might be unfair for the BIC procedure as it does not seek to maximize the finite sample posterior. Hence, we also show the results obtained when the predictions given the BIC model are obtained by using the maximum likelihood estimators of the parameters.
Bayesian Learning with Mixed Graph Models

<table>
<thead>
<tr>
<th>Folder</th>
<th>BIC (MLE)</th>
<th>BIC (Bayesian)</th>
<th>Fully Bayesian</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1142.117103</td>
<td>-1139.140452</td>
<td>-1138.383068</td>
</tr>
<tr>
<td>2</td>
<td>-1191.379395</td>
<td>-1186.751332</td>
<td>-1185.532145</td>
</tr>
<tr>
<td>3</td>
<td>-1172.032072</td>
<td>-1171.37862</td>
<td>-1171.047614</td>
</tr>
<tr>
<td>4</td>
<td>-1127.233931</td>
<td>-1125.155156</td>
<td>-1125.476156</td>
</tr>
<tr>
<td>5</td>
<td>-1164.221219</td>
<td>-1161.62244</td>
<td>-1160.147087</td>
</tr>
<tr>
<td>6</td>
<td>-1144.453722</td>
<td>-1143.423074</td>
<td>-1142.651106</td>
</tr>
<tr>
<td>7</td>
<td>-1168.776494</td>
<td>-1169.259566</td>
<td>-1168.826299</td>
</tr>
<tr>
<td>8</td>
<td>-1187.434765</td>
<td>-1185.747506</td>
<td>-1180.706025</td>
</tr>
<tr>
<td>9</td>
<td>-1162.638587</td>
<td>-1160.396862</td>
<td>-1158.91037</td>
</tr>
</tbody>
</table>

Table 1: Results for the 10-fold cross-validation evaluation with the gene expression data (the last folder has more test points). The first column shows the predictions made in the respective test set using the graph structure learned with BIC and greedy search, with parameters given by the maximum likelihood estimator. The second column reports the results using the same graph given by the BIC greedy search, but where now we average over the posterior distribution of the parameters. The third column reports results with the graph given by the estimated mode of the posterior distribution, found by greedy search. Priors are described in the body of the text.

Most of the mass of the difference distribution is positive (85 out of 100 for the first case, 89 out of 100 in the second case), which passes a sign test at a 0.05 level.

7.2.2 Gene expression analysis

To illustrate the use of Bayesian model selection approaches, we analyse gene expression data previously studied by Drton and Perlman (2008), also as Gaussian bi-directed models. As before, our goal will be to compare the predictive power of models learned by greedy search with BIC and greedy search with the full Bayesian posterior.

The data consists of measurements of 13 gene expression profiles from a metabolic network. A total of 118 points is available. Using all data, the BIC-induced graph has 39 edges, while the finite sample posterior graph had 44. The same procedure used in the synthetic studies, on initializing graphs and choosing priors and centering the data, was applied in this case. Predictive results from a 10-fold cross-validation procedure is shown in Table 1. There is a steady advantage for the Bayesian approach, although by not much. Notice that using Bayesian averaging over parameters given the BIC graph improves prediction significantly when compared to using the maximum likelihood point estimate, despite the simplistic choice of prior in this study.
Figure 18: Two learning problems with discrete data. In (a), the graph shows dependencies concerning alcoholism ($A_i$) and depression ($D_i$) symptoms for paired twins $\{1, 2\}$. In (b), a model for dependencies among features of a study on parole appeals, including the success of the parole, if the type of offense was a person offense or not, and if the offender had a dependency on drugs and was over 25 years old. All variables in these studies are binary and further details and references are provided by Drton and Richardson (2008).

7.2.3 Remarks

The procedure based on the sampler is doable for reasonably sized problem on the order of a few dozen variables in single core machines. For problems with directed and bi-directed edges, the scale of the problem can actually be higher depending on the size of the largest district. However, further improvements are necessary for larger problems. One aspect that was not explored here was re-using previous computations when calculating the probability of a new candidate, in a way similar to the local updates in DAG models (Chickering, 2002). How to combine local updates with the ordering-based improved sampler of Section 6 is left as future research. Several practical variations can also be implemented, such as vetoing the inclusion of edges associated with high p-values in the respective independence tests. Such tabu lists can significantly decrease the search space.

7.3 Discrete data applications

We now show results on learning a discrete distribution that factorizes according to a mixed graph. Drton and Richardson (2008) describe two applications on real-world binary data modeled according to bi-directed graphs. The empirical contingency tables for the two studies can be found in the given paper. They used a complete parameterization for bi-directed binary models and a maximum likelihood estimation procedure. In this section, we use these two studies to illustrate the behavior of our Bayesian procedure using the probit model. Our model imposes probit constraints that are not required by Drton and Richardson, but it allows us to obtain Bayesian credible intervals and predictions.

The graphs used in the two studies are depicted in Figure 18. The first problem is a study on the dependency between alcoholism and depression, as shown in Figure 18(a). A
data point is collected for a given pair of mono-zygotic twins. For each sibling $S_i$, it is recorded whether $S_i$ is/is not alcoholic ($A_i$), and whether $S_i$ suffers/does not suffer from depression ($D_i$). The hypothesis encoded by the graph is that alcoholism and depression do not share a common genetic cause, despite $A$ and $D$ having some hidden (but different) genetic causes. If $A$ and $D$ did have genetic common causes, one would expect that the edges $A_1 \leftrightarrow D_2$ and $A_2 \leftrightarrow D_1$ would be also required. The compounded hypothesis of marginal independencies for $A_i$ and $D_j$, $i \neq j$, can be tested jointly by testing a bi-directed model. Notice that no reference to particular genetic hidden causes of alcoholism and depression is necessary, which again illustrates the power of modelling by marginalizing out latent variables.

The second study, as shown in Figure 18(b), concerns the dependencies among several variables in an application for parole. The model implies, for instance, that the success of a parole application ($Success$ node, in the Figure) is independent of the age of the offender being under 25 ($Age$ node). However, if it is known that the offender had a prior sentence, these two variables become dependent (through the path $Success \leftrightarrow Prior sentence \leftrightarrow Age$). As reported by Drton and Richardson (2008), their binary bi-directed model passes a significance test. Drton and Richardson also attempted to learn an undirected (Markov) network structure with this data, but the outcome was a fully connected graph. This is expected, since Markov networks cannot represent marginal independencies unless the graph is disconnected, which would introduce all sorts of other independencies and possibly not fit the data well. If many marginal independencies exist in the data generating process, Markov networks might be a bad choice of representation. For problems with symmetries such as the twin study, DAGs are not a natural choice either.

### 7.3.1 Results

For the twin data problem, we used a simple prior for the covariance matrix of the underlying latent variables: a $\mathcal{G}$-inverse Wishart with 1 degree of freedom and a complete covariance with a value of 2 for each element in the diagonal and 1 in the diagonals. Thresholds are fixed at zero, since we have binary data. We present the expected posterior values of the
<table>
<thead>
<tr>
<th>Entry</th>
<th>Estimates</th>
<th>Entry</th>
<th>Estimates</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1A_2D_1D_2$</td>
<td>$E[\Theta</td>
<td>D]$</td>
<td>MLE</td>
</tr>
<tr>
<td>$A_1A_2D_1D_2$</td>
<td>$E[\Theta</td>
<td>D]$</td>
<td>MLE</td>
</tr>
<tr>
<td>0000</td>
<td>0.461</td>
<td>0.461</td>
<td>0.482</td>
</tr>
<tr>
<td>0001</td>
<td>0.136</td>
<td>0.138</td>
<td>0.134</td>
</tr>
<tr>
<td>0010</td>
<td>0.157</td>
<td>0.159</td>
<td>0.154</td>
</tr>
<tr>
<td>0011</td>
<td>0.097</td>
<td>0.096</td>
<td>0.085</td>
</tr>
<tr>
<td>0100</td>
<td>0.032</td>
<td>0.032</td>
<td>0.025</td>
</tr>
<tr>
<td>0101</td>
<td>0.022</td>
<td>0.021</td>
<td>0.015</td>
</tr>
<tr>
<td>0110</td>
<td>0.007</td>
<td>0.008</td>
<td>0.012</td>
</tr>
<tr>
<td>0111</td>
<td>0.012</td>
<td>0.012</td>
<td>0.017</td>
</tr>
</tbody>
</table>

Figure 20: The posterior expected value of the 16 entries in the twin study table ($E[\Theta|D]$). Results generated with a chain of 5,000 points. We also show the maximum likelihood estimates of Drton and Richardson (2008) (MLE) and the maximum likelihood values obtained using an unconstrained model (uMLE). As discussed by Drton and Richardson, the data strongly supports the marginal independence constraints encoded in the mixed graph, as can be verified by the fact that the estimates implied by the model provide a good match to the empirical uMLE estimates that follow from the unconstrained contingency table.

contingency table entries in Figure 20. The outcome is essentially identical to the maximum likelihood estimates of Drton and Richardson (2008) despite the probit parameterization. Moreover, with our procedure we are able to generate Bayesian confidence intervals, as illustrated in Figure 19. The results are very stable for a chain of 1,000 points.

For the parole data, we used a $G$-inverse Wishart prior for the covariance matrix of underlying variables $Y^*$ with 1 degree of freedom and the identity matrix as hyperparameters. We compare the effective sample size of the Gibbs sampler for our DMG model and the DAG model obtained by using the ancillary latent parameterization of Section 7.1 for the underlying latent variable covariance matrix $^{18}$. Boxplots for the 16 contingency table entries of the twin network and the 32 entries of the parole study are shown in Figure 21. The setup is the same as in the democratization and industrialization experiment, where we run 80 independent chains and plot the distribution of the effective sample sizes to measure the mixing time. We ran a shorter chain of 2,000 points, since computing the contingency table entries is expensive.

There is a substantial difference in effective sample size for the parole study. Notice that we are comparing MCMC samples for the entries in the contingency table, which in the DAG case requires integrating out not only the underlying latent variables implicit in the probit parameterization, but also the ancillary latents that account for the bi-directed edges. This hierarchy of latent variables, which does not exist in the DMG case, adds a

---

18. The priors used are as follows: the ancillary representation was given a prior with mean 1 and variance 1 for the coefficients $X_{ij} \rightarrow Y^*_j$, for $j > i$, and set constant to 1, if $i < j$. The means of the ancillary latents were fixed at 0. Variance parameters were given (0.5, 0.5) inverse gamma priors, which approximately matches the priors in the DMG model.
considerable amount of autocorrelation to the chain. The standard DMG parameterization can be seen as a way of obtaining a collapsed Gibbs sampler, where the parameterization by construction reflects latent variables that were analytically marginalized.

![MCMC comparison: DMG vs. DAG (twin data)](image1)

![MCMC comparison: DMG vs. DAG (parole data)](image2)

Figure 21: Comparison of effective sample sizes for the twin data (a) and parole data (b). 80 independent chains of 2,000 points were obtained using the Gibbs sampling algorithm, and the respective box-plots shown above. The Markov chain with the DMG approach easily dominates the DAG one. For the parole data, the average effective sample size for the DAG was as low as 60 points.

8. Conclusion

Directed mixed graph models are a generalization of directed graph models. Whenever a machine learning application requires directed graphs, one should first consider whether directed mixed graphs are a better choice of representation instead. DMGs represent conditional independencies of DAGs where hidden variables have been marginalized out. Given that in most applications it is unlikely that all relevant variables are known, DMGs are a natural representation to use. In this paper, we introduced priors and inference algorithms for Bayesian learning with two popular families of mixed graph models: Gaussian and probit. We discussed some implementations and approximations to scale up algorithms. We showed examples of applications with real data, and demonstrated that Bayesian inference in Gaussian and probit DMG models using MCMC can have substantially faster mixing than in comparable DAGs.

It is part of the machine learning folklore that factor graphs can subsume directed networks. In a very important sense, this is known not to be true: as a language, a graphical model is only interesting to the extent the constraints entailed by the representation allow
Figure 22: In (a), a simple bi-directed chain with four random variables. In (b), the respective factor graph that is obtained from a Bartlett parameterization using the ordering \( \preceq \{ Y_1, Y_2, Y_3, Y_4 \} \). In this case, the factors are \( p(Y_1) \times p(Y_2|Y_1) \times p(Y_3|Y_1, Y_2) \times p(Y_4|Y_1, Y_2, Y_3) \). A different choice of ordering (e.g., the perfect ordering) could provide simpler factors on average, but the presence of a factor linked to all variables is unavoidable. In this case, a generic message-passing algorithm that does not take into account the linearity of the parameterization would be of little value.

for a proper parameterization of the intended family of distributions. Factor graphs cannot represent the same family of constraints found in directed networks. Undirected and factor graphs only allow for monotonic independence models, where explaining away is ruled out. This excludes a vast number of realistic, non-monotonic, models. While factor graphs are perhaps the data structures of choice for general message-passing algorithms (e.g., Yedidia et al., 2005), they are far from being universal modelling languages for independencies.

What is true is that for any distribution that is Markov with respect to a DAG or DMG there is at least one corresponding factor graph model, but this is a vacuous claim of little interest: any distribution can be represented by a single factor involving all variables. Some will require a factor with all variables, even under the presence of a large number of independence constraints. For instance, a factor graph corresponding to any given bi-directed chain will necessarily include a factor node adjacent to all variable nodes, as illustrated in Figure 22. When parameterizing a distribution with many marginal independencies (e.g., a bi-directed tree), the respective factor graph would be no more than a unhelpful drawing. A better strategy for solving real-world problems is to define a family of models according to the (directed/undirected/factor) graphs of choice, and let the inference algorithm decide which re-expression of this factorization suits the problem. This has been traditional in graphical modelling literature (Lauritzen, 1996). The strategy adopted in this paper followed this spirit.

Many challenges remain. For instance, much more flexible models for DMG discrete models are being developed (Drton and Richardson, 2008), but for large graphs they pose a formidable computational cost. An important question is which other intermediate parameterizations could be used, and which approximation algorithms to develop. The probit family discussed here was a choice among many. The parameterization by Drton and Richardson (2008) could be a starting point for trading-off flexibility and computational effort. And while it is true that Gaussian copula models (Pitt et al., 2006) can be adapted
to generalize the approach introduced here, it remains to be seen if other copula parameterizations easily lead to DMG models.

The goal of this paper was not only to introduce “one more” graphical language into Bayesian machine learning, but to also force the designers of new models and algorithms to think about the reasons why a graphical model is being used in the first place. Should independence constraints be monotonic or non-motonic (i.e., should we allow for “explaining away”)? If a directed representation is used, which observables should be considered? What is the role of the hidden variables? If one accepts the possibility of hidden common causes, what is the space of graphs that should be considered?

To further motivate the development of Bayesian tools for learning with mixed graph models, we describe how some machine learning tasks can benefit from using such models.

Measurement models
In robotics, social sciences, psychology and several other areas, one important task is to model the distribution of latent variables that are measured indirectly from a set of multiple sensors, or indicators (Silva, 2005). Such variables are usually noisy measures of the target latent variables. Since the goal is to give a point estimate or distribution of such hidden variables, it is essential to represent them explicitly. Yet there might be several other hidden variables that are not directly relevant to the study, as illustrated in Section 7.1. If there is some sparse structure of associations implied by other latent variables, such sparseness is better captured by a mixed graph model.

Structure learning for density estimation and causal inference
A standard task in machine learning is learning the structure of graphical models (Heckerman, 1998). Learning the structure of DAGs is a well-studied problem, either due to the need for non-monotonic independence constraints, or for the estimation of causal relations (Spirtes et al., 2000; Chickering, 2002; Sachs et al., 2005).

However, once one accepts the DAG representation to be a suitable representation of real-world structure, one has also to accept that possible hidden common causes of the observed variables might exist. In general the corresponding marginal dependencies cannot be represented as DAGs, since DAGs are not closed under marginalization. This aspect is even more crucial if one is interested in causal relations, since connections will appear in DAG representations where there are no direct causal effects but only unmeasured confounding.

An important class of acyclic DMGs is the directed mixed ancestral graph, DMAG (Richardson and Spirtes, 2002) (all directed MAGs are ADMGs, but the opposite is not true). When the search is restricted to Markov equivalence classes, it suffices to score a Markov equivalence class by scoring a MAG model that encodes the corresponding constraints (the independence constraints of any DMG can be represented by some MAG). Early studies in learning Gaussian MAG models are described by Spirtes et al. (1996). The usefulness of MAG representations is illustrated in Figure 23. While Bayesian learning is admittedly a computationally demanding task, different levels of approximations can be used at different stages: starting from the fast constraint-satisfaction FCI algorithm of Spirtes et al. (2000), through BIC and variational approximations to the marginal likelihood, end-

19. Notice that MAGs do not need to be directed: they allow for the possibility of undirected edges with semantics analogous to undirected networks.
Figure 23: Consider learning the DMG structure in (a) given data sampled from variables \{Y_1, Y_2, \ldots, Y_5\}. A typical algorithm that searches over DAGs would return a structure such as in (b). In contrast, Spirtes et al. (2000) introduced an algorithm that searches over Markov equivalence classes of DMGs. The expected output is shown in (c), where the circle notation indicates that the endpoints of the edges at \(Y_1\) and \(Y_2\) are undetermined within the equivalence class. The graph in (d) is an example of a MAG within the equivalence class of the original graph (a).

Structured and relational prediction

The task of structured prediction can be described as predicting vector \(Y\) given vector \(X\) when the distribution (or density) \(P(Y|X)\) is constrained. Conditional independence constraints are among the most common. Some topics in structured prediction with independence constraints are discussed by Pérez-Cruz et al. (2007) and Taskar (2004).

One of the earliest and simplest models of structured prediction was the seemingly unrelated regression (SUR) model of Zellner (1962). Consider the problem of predicting the stock of a company at the end of year from the profits given at the beginning of the year. Figure 24(a) illustrates the case where the prediction is performed for two companies simultaneously. This model encodes the assumption that Sony’s profit (\(SP\)) is not a predictor of Microsoft’s stock (\(MS\)) given Microsoft’s profit (\(MP\)). The reciprocal is also assumed to be true. In this sense, the regression problems are seemingly unrelated. However, \(MS\) and \(SS\) (Sony’s stock) are dependent given \(MP\) and \(SP\), which can be explained by several hidden market factors that are common causes of \(MS\) and \(SS\). Notice that the indepen-

---

20. Since there were no reasonable Bayesian methods for learning DMG models at that time, Silva and Scheines (2006) implicitly used extra latent variables in place of the bi-directed edges.
Bayesian Learning with Mixed Graph Models

Figure 24: The graph in (a) encodes the following constraints: Microsoft’s stock ($M_S$) is independent of Sony’s profit ($SP$) given Microsoft’s profit ($MP$) (the structure between $MP$ and $SP$ is unspecified, since this is a regression problem and, as such, these variables are always fixed). When Sony’s stock ($SS$) is observed, however, $M_S$ and $SP$ become dependent. This emerges when several unspecified common causes affect both stock prices, as depicted in (b). The inverse relationship is encoded in the undirected network in (c): in this case, $M_S$ and $SP$ are dependent and, when observing $SS$, they become independent. A variation of this regression problem including extra components and dependencies is shown in (d). Philip’s stock ($PS$) is considered independent of $M_S$ for them not being direct competitors (while hidden factors in the videogame market affect $\{MS, SS\}$, and electronics market factors affect $\{SS, PS\}$).
Figure 25: Sparse models with no independencies. The chain in (a) is a graph where no conditional independencies exist among the observables. Its Gaussian parameterization, discussed in Section 2 is requires few parameters, and it is much simpler than a latent variable model such as the one in (b). Characterizing the class of non-independence constraints entailed by parametric families with DMG representation is a topic that has been mostly unexplored.

\[ MP \rightarrow MS \leftrightarrow SS \leftarrow SP \]

has no equivalent in undirected graphs and DAGs (Proposition 2 of Drton and Richardson, 2003), or chain graphs (Richardson, 1998). For comparison, an undirected graph with the same adjacencies is given in Figure 24(c). Generalizations of SUR when there are other output variables (Figure 24(d)) and other connections from input to output can be easily managed with the proposed algorithms, since they fall under the class of mixed graph models\(^{21}\).

**Other sparse models**

Mixed graph models can also be seen as networks of “long-distance” dependencies with sparse parameterizations. This is made explicit in Figure 25(a): as a Gaussian model, this has very few parameters, yet one can verify using the m-separation criterion (Richardson, 2003) that there are no (marginal or conditional) independence constraints. The elegant parameterization of Gaussian models in this case can be contrasted with the simplest latent variable model equivalent in Figure 25(b): there are now two parameters corresponding to each bi-directed edge (the coefficients of the edges, where we fix the distribution of the latent to a standard normal), unless extra constraints are introduced. There is also, for each bi-directed edge, a new latent random variable that has to be integrated over. This is not without its advantages, since during sampling we can condition on a few latent vertices to break long chains. For long chains where the cost of inference with DMG models might be too high, a hybrid representation can be considered: replacing a few of the bi-directed links by latent variables.

It is possible to explore the DMG properties in many creative ways, such as in constructing new classes of time series models and networks of regression models where the “signal” corresponds to parameters associated with directed edges, while “noise” has a joint distribution that factorizes according to a bi-directed graph. Moreover, since sparseness comes from constraints other than independence constraints, this raises very interesting theoreti-

\[^{21}\text{Silva et al. (2007) treat the parameters from bi-directed edges as hyperparameters in a generalized Gaussian process classifier for relational data, although the parameterization has to be much more constrained due to the nature of the problem.}\]
Bayesian Learning with Mixed Graph Models

cal and practical questions. Which approximate inference algorithms to develop and which constraints (besides independence constraints) can be read off the graph given a parametric family are examples of such questions.

Acknowledgments

Thanks to Kenneth Bollen for providing us with the industrialization and democratization dataset, and to Robert Gramacy for helpful discussions. An earlier version of this paper (Silva and Ghahramani, 2006) appeared in the proceedings of the Uncertainty in Artificial Intelligence conference. This work was funded by the Gatsby Charitable Foundation and a EPSRC grant #EP/D065704/1.

References


R. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, 1996.


Appendix A: deriving the sampling distribution for the Monte Carlo computation of normalizing constants

We give here the details on how to derive the sampling distribution used for computing normalizing constants $I_Q(\delta,\mathbf{U})$, as described in Section 3.2.2.

Let $\mathbf{A}_i \equiv \Sigma_{sp,(i),nsp,(i)}^{-1} \Sigma_{sp,(i),nsp,(i)}^{-1}$. Recall from Equation (16) that $\mathbf{B}_{i,nsp,(i)} = -\mathbf{B}_{i,sp,(i)} \mathbf{A}_i$. The original density $p(\mathbf{B}_i|\gamma_i)$, as given by Lemma 1, is a multivariate Gaussian with the following kernel:

$$
\exp\left(-\frac{1}{2\gamma_i} \begin{bmatrix} \mathbf{B}_{sp,(i)}^T - \mathbf{M}_{sp,(i)} \\ \mathbf{B}_{nsp,(i)}^T - \mathbf{M}_{nsp,(i)} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{ss} & \mathbf{U}_{sn} \\ \mathbf{U}_{ns} & \mathbf{U}_{nn} \end{bmatrix} \begin{bmatrix} \mathbf{B}_{sp,(i)}^T - \mathbf{M}_{sp,(i)} \\ \mathbf{B}_{nsp,(i)}^T - \mathbf{M}_{nsp,(i)} \end{bmatrix} \right) 
$$

(64)

where $\mathbf{U}_{\{i-1\},\{i-1\}}$ in Lemma 1 was rearranged above as the partitioned matrix in (29). The pair $\{\mathbf{M}_{sp,(i)},\mathbf{M}_{nsp,(i)}\}$ corresponds to the respective partition of the mean vector $\mathbf{M}_i$. Plugging in the expression for $\mathbf{B}_{i,nsp,(i)}$ in (64), we obtain the modified kernel

$$
\exp\left(-\frac{1}{2\gamma_i} \begin{bmatrix} \mathbf{B}_{sp,(i)}^T - \mathbf{M}_{sp,(i)} \\ -\mathbf{A}_i^T \mathbf{B}_{nsp,(i)} - \mathbf{M}_{nsp,(i)} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{ss} & \mathbf{U}_{sn} \\ \mathbf{U}_{ns} & \mathbf{U}_{nn} \end{bmatrix} \begin{bmatrix} \mathbf{B}_{sp,(i)}^T - \mathbf{M}_{sp,(i)} \\ -\mathbf{A}_i^T \mathbf{B}_{nsp,(i)} - \mathbf{M}_{nsp,(i)} \end{bmatrix} \right) 
$$

(65)

which can be rewritten as

$$
p_b(\mathbf{B}_{i,sp,(i)}; \mathbf{K}_i, \mathbf{m}_i; \gamma_i) \propto (2\pi)^{\# sp,(i)/2} \gamma_i^{\# sp,(i)/2} |\mathbf{K}_i(\Phi_{i-1})|^{1/2} \exp\left\{-\frac{1}{2\gamma_i} \mathbf{U}_i \right\} 
$$

(66)

where $\# sp,(i)$ is the size of set $sp,(i)$, $p_b(\cdot; \mu, \Sigma)$ is the density function of a multivariate Gaussian distribution with mean $\mu$ and covariance $\Sigma$, $\mathbf{K}_i(\Phi_{i-1}) \equiv \mathbf{K}_i$ to emphasize the contribution of previous parameters, and
\[ \mathbf{m}_i = (\mathbf{U}_{ss} - \mathbf{A}_i \mathbf{U}_{ns}) \mathbf{M}_{sp,z(i)} + (\mathbf{U}_{sn} - \mathbf{A}_i \mathbf{U}_{nn}) \mathbf{M}_{nsp,z(i)} \]

\[ \mathbf{K}_i^{-1} = \mathbf{U}_{ss} - \mathbf{A}_i \mathbf{U}_{ns} - \mathbf{U}_{sn} \mathbf{A}_i^T + \mathbf{A}_i \mathbf{U}_{nn} \mathbf{A}_i^T \]

(67)

\[ \mathcal{U}_i = \mathbf{M}_i^T \mathbf{U}_{(i-1),(i-1)} \mathbf{M}_i - \mathbf{m}_i^T \mathbf{K}_i \mathbf{m}_i \]

If \( sp_z(i) = \emptyset \), it follows that \( \mathcal{B}_i = \mathcal{B}_{i,sp_z(i)} = 0 \). The kernel (65) reduces to \( \exp(-0.5 \mathcal{U}_i / \gamma_i) \), and \( \mathcal{U}_i \equiv \mathbf{M}_i^T \mathbf{U}_{(i-1),(i-1)} \mathbf{M}_i \). If \( nsp_z(i) = \emptyset \), then the expression for the kernel does not change \( (\mathcal{U}_i \equiv 0) \), and Equation (66) corresponds to the original kernel in Equation (25).

Inserting the re-expressed kernel into the original function (25), we obtain

\[ p_b(\mathcal{B}_{i,sp_z(i)}, \mathbf{K}, \mathbf{m}_i, \gamma_i \mathbf{K}_i) p_g \left( \frac{\delta + i - 1 + \#nsp_z(i)}{2}, \frac{u_{ii, (i-1),(i-1)} + \mathcal{U}_i}{2} \right) f_i(\Phi_{i-1}) \]  

(68)

where \( p_g(\cdot; \alpha, \beta) \) is a gamma density function and

\[ f_i(\Phi_{i-1}) \equiv \frac{(2\pi)^{-\frac{(i-1)-\#sp_z(i)}{2}} |\mathbf{K}_i(\Phi_{i-1})|^{1/2} |\mathbf{U}_{(i-1),(i-1)}|^{-1/2}}{\Gamma((\delta + i - 1 + \#nsp_z(i))/2)} \frac{\Gamma((\delta + i - 1)/2)}{(u_{ii, (i-1),(i-1)} + \mathcal{U}_i)/2} \] 

(69)

**Appendix B: variational updates for Gaussian mixed graph models**

The variational updates for the coefficient and intercept parameters are essentially identical to their joint conditional distribution given \( \mathbf{V} \) and \( \mathbf{X} \), where occurrences of \( \mathbf{V} \) and \( \mathbf{X} \) are substituted by expectations \( \mathcal{V}_{ij} \) and \( \mathcal{X}_{ij} \), respectively. Let \( \mathcal{V}_{ij} \) be the \( ij \)-th entry of \( \langle \mathbf{V}^{-1} \rangle_q(\mathbf{V}) \). The covariance matrix of \( (\mathbf{B}, \mu) \) is the covariance matrix of the vector \( \text{vec}(\mathbf{B}, \mu) \), where such vector is constructed using all coefficients and intercepts. We denote this covariance matrix by \( \Sigma_{B,\mu} \). For simplicity of notation, we will treat \( \mu_i \) as the coefficient \( b_{i(m+1)} \), \( m \) being the number of variables. We will also adopt the notation \( Y_{m+1}^{(d)} \equiv 1 \) in the following derivations. As an abuse of notation, let \( \mathbf{Y} \) also refer to latent variables. In this case, if \( Y_i \) and \( Y_j \) refer to latent variables \( X_{h_i} \) and \( X_{h_j} \), then define \( Y_i \equiv \langle X_{h_i} \rangle_q(\mathbf{X}) \), and

\[ Y_i Y_j \equiv \langle X_{h_i} X_{h_j} \rangle_q(\mathbf{X}) \cdot \]

Let \( b_{ij} \) and \( b_{uv} \) be the \( r \)-th and \( s \)-th entries of \( \text{vec}(\mathbf{B}, \mu) \), respectively. The \( rs \)-th entry of the inverse matrix \( \Sigma_{B,\mu}^{-1} \) is given by

\[ (\Sigma_{B,\mu}^{-1})_{rs} = \mathcal{V}_{is} \sum_{d=1}^{n} Y_j^{(d)} Y_v^{(d)} + 1(i = t)1(j = v) \frac{c_{ij}}{s_{ij}} \]

(70)

where \( b_{xps} \equiv 0 \) if no edge \( Y_x \leftrightarrow Y_p \) exists in the graph, \( 1(\cdot) \) is the indicator function, and \( c_{ij}^{(d)}, s_{ij}^{(d)} \) are the given prior parameters defined in Section 4. Similarly to the factorization criterion explained in Section 6, the matrix \( q(\mathbf{V}) \) will in general be block-diagonal, and this summation can be highly simplified.

Define now a vector \( \mathbf{e}^b \) analogous to the Gibbs sampling case, where
\[ c_r^b = \sum_{t=1}^{m} V_{it} \sum_{d=1}^{n} Y_j^{(d)} Y_t^{(d)} + \frac{c_{ij}^b}{s_{ij}^b} \]  

(71)

The variational distribution \( q(B, \mu) \) is then a \( N(\Sigma_{B,\mu} c, \Sigma_{B,\mu}) \). The variational distribution for the latent variables will exactly the same as the Gibbs distribution, except that references to \( B, \mu, V^{-1} \) are substituted by \( \langle B \rangle_{q(B,\mu)}, \langle \mu \rangle_{q(B,\mu)} \) and \( \langle V^{-1} \rangle_{q(V)} \).