Archival Version including Appendicies: Experiments in Stochastic Computation for High-Dimensional Graphical Models

Beatrix Jones[*], Carlos Carvalho[*], Adrian Dobra[*], Chris Hans[*], Chris Carter[†] and Mike West[*]

January 14, 2004

[*]Institute of Statistics and Decision Sciences, Duke University, Durham NC
[†]CSIRO, Sydney, Australia

Abstract

We discuss the implementation, development and performance of methods of stochastic computation in Gaussian graphical models, with a particular interest on the scalability of MCMC and other stochastic search methods with dimension. Our perspective is that of high-dimensional model search – we are interested in exploring the complex, high-dimensional spaces of undirected graphical models that arise due to uncertainty about model form. We review the structure and context of undirected graphical models, Gaussian models and model uncertainty (the so-called covariance selection problem). We discuss prior specifications, including new priors over models and hyper-Markov priors on covariance patterns within models, and then explore a number of examples using various methods of stochastic computation. This discussion represents both a review of the theoretical structure of these graphical models and of a number of key aspects and details of existing computational ideas, as well as the point of departure for experimentation with MCMC methods. We then discuss alternative stochastic search ideas, and in examples compare and contrast MCMC methods with a novel stochastic model search approach. We summarize our experiences in trying to use these methods in problems in low (12-20) to moderate (150) dimensions. The examples combine simple synthetic examples with data analysis from gene expression studies. We conclude with comments about scalability of the approaches studied and the need and potential for new computational methods in far higher dimensions, the need for new theoretical insights, and alternative constructive approaches to Gaussian graphical modeling and computation.

Keywords: Decomposable Models, Non-decomposable models, Markov chain Monte Carlo, Shotgun Stochastic Search, Parallel implementation

1 Introduction

The last decade or so has witnessed an essential revolution in the statistical sciences, based on developments in stochastic simulation methods for scientific computation. The impact on applied Bayesian statistics has, of course, been particularly notable, with the development of MCMC methods enabling application of increasingly rich and more relevant mathematical models. In

tandem with model complexity is the radically increasing capacity to generate data sets involving many, many variables. From high-frequency finance and enormous marketing databases to gene expression studies in functional genomics, we are now faced with applied problems typified by very high-dimensional variables and/or parameter spaces. The canonical use of methods of stochastic computation in searching over spaces of candidate models raises challenges of both statistical and computational efficiency as well as basic feasibility. As dimension escalates, the increasing need to address these challenges promotes a research focus on the scalability of existing computational methods, and an increasing interest in novel computational strategies.

We are interested in precisely these questions – statistical and computational efficacy, and scalability with dimension – of stochastic computational methods used to explore spaces of Gaussian graphical models. In a graphical model of a multivariate distribution, nodes represent variables and edges represent pairwise dependencies, with the edge set defining the global conditional independence structure of the distribution. The methodological issues faced, as dimension grows, include questions of the nature and consistency of prior specification (priors over graphical structure, and priors for parameters on any single, specified graph) and then the very challenging problem of searching over the space of graphs to identify subsets of interest under the theoretically implied posterior distributions. This represents a complex variable/model selection problem for which a number of computational methods have been suggested. But, in problems of many variables, little is known about the performance and scalability of stochastic computational approaches.

Understanding the conditional independence structure is a critical, indeed central element of exploratory and confirmatory data analysis, and especially so in trying to "make sense" of problems in which the number of variables far exceeds the number of observations. This is the case in examples from genomics involving gene expression data analysis – a key example and motivating context for us.

We focus on undirected graphical models – a central context that has received a good deal of interest in the computational statistics literature in recent years. An undirected Gaussian graphical model is an average (with respect to the posterior of the covariance matrix) over multivariate normal distributions with a common conditional independence structure. The graph G consists of a vertex for every variable, and a set of edges E. Two variables a and b are conditionally independent given the remaining variables if, and

only if, $\{a, b\} \notin E$. The analysis challenge is inherently one of model uncertainty and model selection: we are interested in exploring "graph space" and identifying graphs (conditional independence structures) that are most appropriate for a given data set as measured by the posterior distribution over graphs; inference on variable dependencies, and prediction, is then based on parametric inferences within a set of selected graphs.

A number of recent papers have addressed the questions of improving computational methods for estimating undirected Gaussian graphical models. A key focus in this literature has been on decomposable graphical models; for example, Giudici and Green (1999) give new, easily and efficiently computable conditions for determining if adding an edge to a graph maintains the decomposable property (see Appendix B for details). Wong and Carter (2002) detail how the decomposition changes between two such graphs differing by a single edge. Understanding the "local" nature of these changes leads to an efficient formula for computing marginal likelihood ratios for comparing two neighboring graphs. This general line of development has recently been extended beyond decomposable graphs, with Monte Carlo methods for computing marginal likelihoods for non decomposable models, Roverato (2002), Atay-Kayis and Massam (2005), and Dellaportas, Giudici and Roberts (2003) that are of immediate utility. Alternative, deterministic methods using a combination of approximations and prior specifications that substantially simplify the computation of these marginal likelihoods are developed by Wong, Carter and Kohn (2003).

Despite these advances, and the pressing need to develop methodology for increasingly large high-dimensional problems, the recent literature maintains a general focus on small problems (Wong Carter and Kohn, 2003, is a notable exception) treating undirected Gaussian graphical models and high dimensional data as an unlikely pairing. This motivates our focus on model search and selection approaches in higher-dimensional contexts. After reviewing some of the structure and recent advances for undirected graphical models (Sections 2, 3 5, 6), we summarize some of our experiences in trying to utilize some of these methods in problems with a moderate (12-20) to large (150) number of variables (Section 9). We also introduce new methodology motivated by these experiences, including priors over graph space that encourage sparsity (Section 4) and a parallelizable stochastic search method for rapid traversal of spaces of graph (Section 8). The examples combine simple synthetic examples with data analysis from gene expression studies. We conclude the paper, covering review of prior specification and compu-

tational approaches in Gaussian graphical models, with discussion of novel, alternative constructive approaches that are able to move to far higher dimensions, comments about the potential for theoretical advances to lead to improvements in stochastic computation in these models, and also for hybrid approaches that combine "aggressive" moves in complex model spaces with the tried-and-tested "local move" approaches that underlie current MCMC methods. We also comment on the need for increased development of distributed computational tools.

2 Graphical Models and Graph Structure

Graphical models provide representations of the conditional independence structure of a multivariate distribution as well as access to efficient algorithms for computation of conditional and marginal densities (Whittaker 1990, Lauritzen 1996, Andersson, Madigan, Perlman, and Richardson 1998, Cowell, Dawid, Lauritzen, and Spiegelhalter 1999) . The computational efficiencies arise through decompositions of the sample space into subsets of variables (graph vertices) based on their graphical relationships. The joint distribution of the variables is Markov over its graph, so likelihoods, and prior and posterior densities, can be computed separately on the subsets of vertices and then reassembled into a likelihood or density incorporating all variables (Hammersley and Clifford 1971, Dawid and Lauritzen 1993). Subsets of variables that are *complete* (have all possible edges between them filled in, or equivalently have no conditional independencies between them) play a special role.

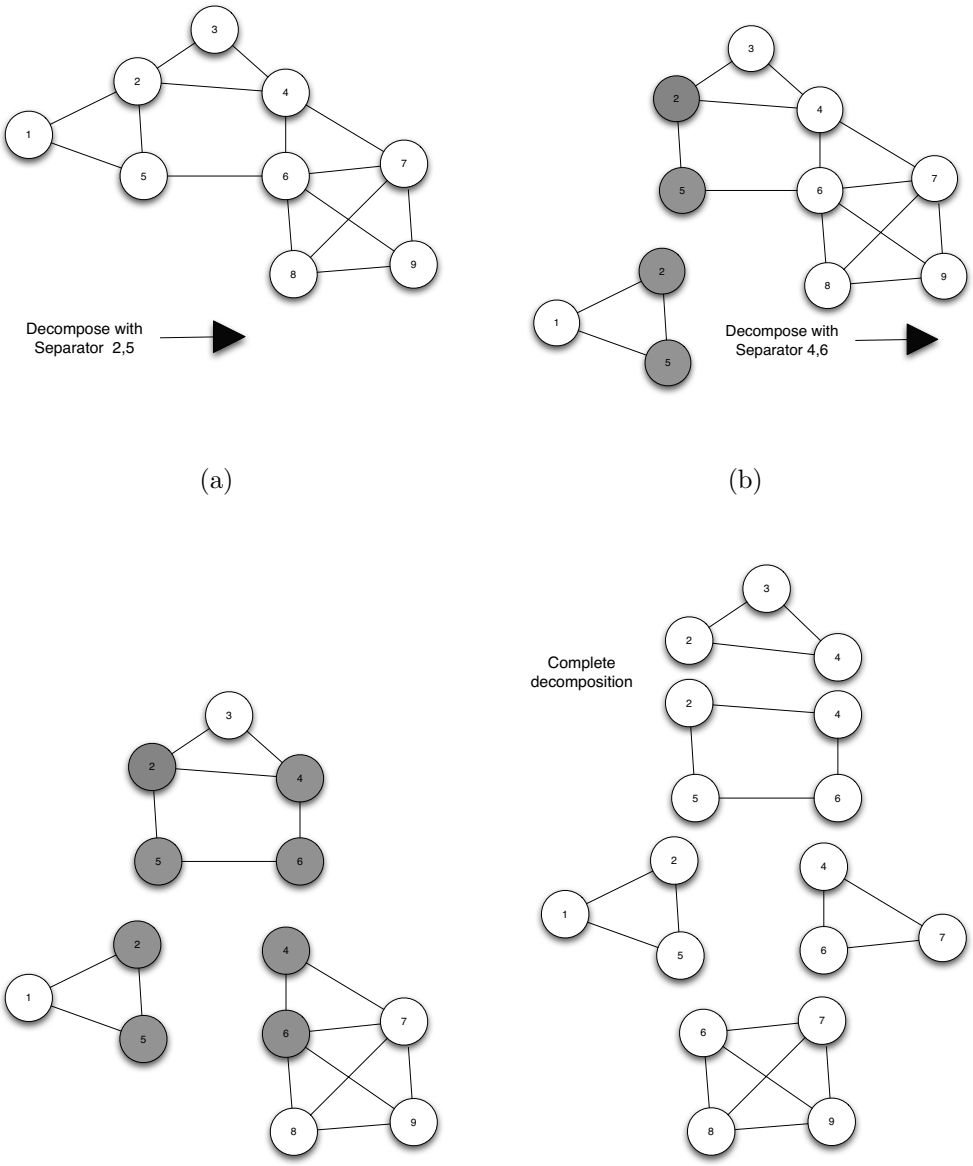
The basic terminology and ideas for graphical models (Cowell, Dawid, Lauritzen, and Spiegelhalter 1999), and the notation used here, begin with a graph $G = \{V, E\}$ where G is defined over a the set of vertices (the variables) V by the edge-set E; two variables a and b are neighbors in G if, and only if, the edge $(a, b) \in E$. A complete graph on p vertices has all $\binom{p}{2}$ edges; otherwise, the graph is incomplete. The incomplete graph G *decomposes* into disjoint subgraphs A, B and C (with $A \cup B \cup C = G$) if C is complete and *separates* A and B (any path from a vertex in A to a vertex in B goes through C). The subgraph C is a *separator*. The decomposition is *proper* if neither A nor B is empty. If the separator C is always chosen to be minimal (so that it does not contain a proper subgraph that separates A and B) then iterative, proper decomposition of the subgraphs $A \cup C$ and $B \cup C$

ultimately results in the *prime components* of a graph: a sequentially defined collection of subgraphs that cannot be further decomposed. See Figure 1 for an example.

Any connected graph can be represented as a tree of its prime components – a *junction tree*. In the junction tree, each prime component, denoted by P_i, is a node; if two nodes share a set of vertices, every prime component on the path between them in the junction tree also contains that set of vertices. A set of vertices shared between two prime components forms a complete subgraph (from the definition of decomposition). The sets of vertices shared by adjacent nodes in the junction tree are called the separators of the junction tree, denoted by S_i. An example is shown in Figure 2. For graphs with more than one connected component, a junction tree exists for each connected component; the collection of junction trees is called a junction forest. Efficient algorithms for producing the junction tree representation of any specified graph are discussed in Appendix A.1. A full development of the basic graph theory useful in computation of probabilities and densities on decomposable can be found in Cowell, Dawid, Lauritzen, and Spiegelhalter (1999); Dobra and Fienberg (2000) reviews this material and extends it to non-decomposable graphs.

As a junction forest contains no cycles (loops) among its nodes, we can define a *perfect ordering* of the prime components and separators. That is, an ordering of prime components (P_i) and separators (S_i) as $P_1; S_2, P_2; S_3, P_3; \ldots$ where S_i is the intersection of P_i and all lower numbered components. We call the prime component sequence G^i and the separator sequence S^i More than one perfect ordering may exist for any given graph.

If all the prime components of a graph are complete, the graph is said to be *decomposable*. Maximal complete subgraphs are called *cliques*, so the prime components of a decomposable graph are all cliques. When we are referring exclusively to prime components that are cliques we will use C to denote the component rather than P. In Gaussian graphical models, the distributional properties of sets of variables represented by complete graphs are well understood: they have unrestricted multivariate normal distributions. Thus decomposable graphs have distributional properties that make them particularly tractable, as we shall see below.



(a) (b)

(c) (d)

Figure 1: An example of how iterative decomposition of a graph produces its prime components.

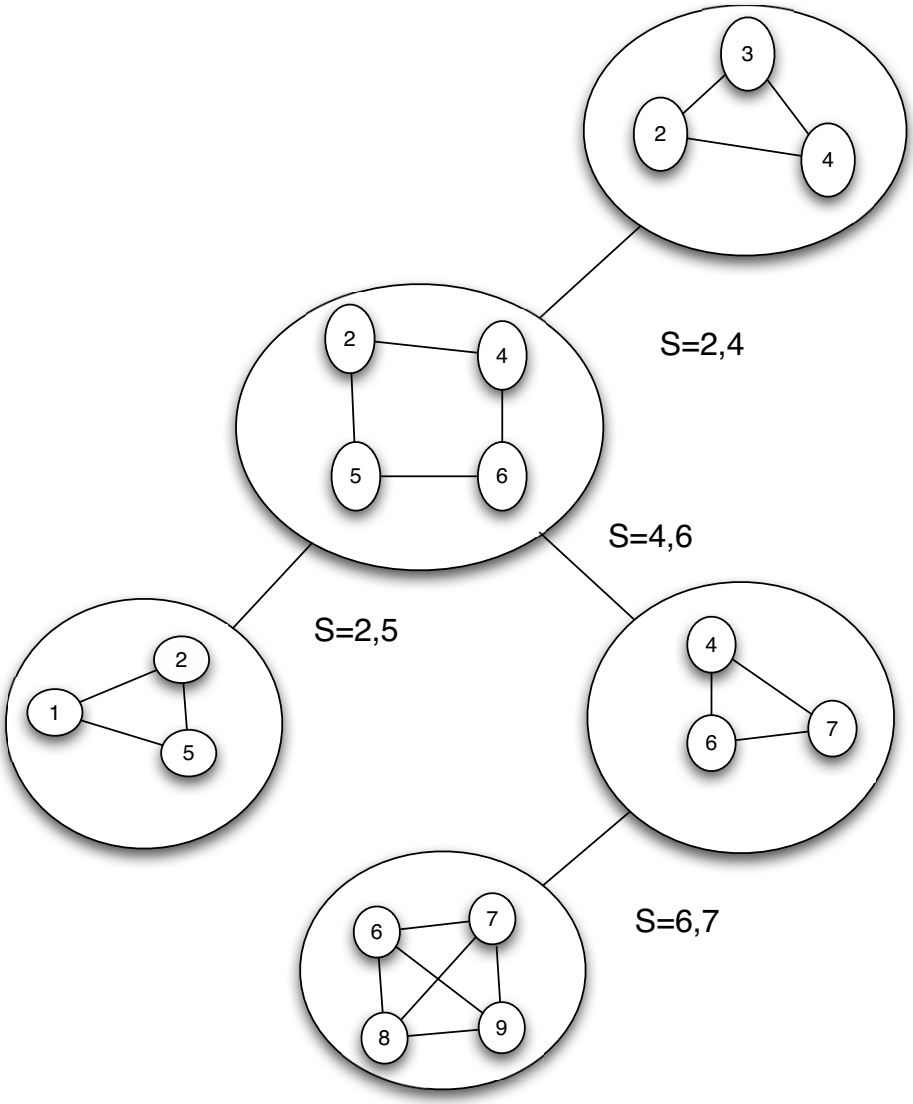


Figure 2: Representing a graph as a junction tree, a tree of its prime components

3 Gaussian Graphical Models

3.1 Density Factorization and Likelihood

The factorization of joint distributions that satisfy the conditional independencies implied by the edge structure of a given graph is key to the development of graphical model analyses. In general, a $p-$vector random variable y has a multivariate distribution $p(y)$ that, on the specified graph G, factorizes into terms corresponding to the prime components and separators of any junction tree representation of G, i.e.,

$$p(y) = \frac{\prod_{P \in G^i} p(y_P)}{\prod_{S \in S^i} p(y_S)} \tag{1}$$

where y_P and y_S represent the variable subsets on the components and separators. In the special case of a multivariate Gaussian distribution, the formulation in terms of a structured covariance matrix – or, more directly, for the inverse covariance matrix, or *precision* matrix – clearly isolates the key structure. With a non-singular covariance matrix Σ, hence precision matrix $\Omega = \Sigma^{-1}$, each term in the decomposition is a multivariate Gaussian with covariance matrices Σ_{PP}, Σ_{SS} on prime components and separators. Hence, for a random sample of size n, $Y = \{y_1, \ldots, y_n\}$, the joint density function on the graph G has the representation

$$p(Y|\Sigma_G) = \frac{\prod_{P \in G^i} p(Y_P|\Sigma_{PP})}{\prod_{S \in S^i} p(Y_S|\Sigma_{SS})}. \tag{2}$$

This provides the full likelihood function for Σ in problems of inference on covariance structure for a given graph G. Dempster (1972) referred to the problem of identifying relevant graphical structures via patterns of zeros in the precision matrix as *covariance selection*, hence the use of that terminology by subsequent authors including Wong, Carter, and Kohn (2003). Key to this approach are the characterizing constraints on Ω implied by G via its edge set E; that is, for any pair of variables (nodes) i, j, $\Omega_{ij} \neq 0$ if, any only if, the edge $(i, j) \in E$. Hence, as we move across potential candidate graphs, the implied structural zeros in the precision matrix induce constraints on covariance patterns and hence parameters in Σ. (At the expense of notational complication, we should properly index Σ by G; we avoid that for clarity of notation, as it is understood throughout). Formal inference is inherently structured by composition; from a Bayesian perspective, we are interested

in posterior distributions $p(G, \Sigma|Y) = p(\Sigma|G, Y)p(G|Y)$ for specified priors $p(G, \Sigma) = p(\Sigma|G)p(G)$.

3.2 Priors and Posteriors for Covariance Matrices

Giudici (1996) discusses the major approaches to prior specification for Σ, comparing the "local priors" described in Dawid and Lauritzen (1993), and the "global priors" based on the conditional approach in Dickey (1971). These priors, based on inverse Wishart or constrained inverse Wishart forms, have the desirable property that $p(\Sigma|G)$ is consistent over graphs in the sense of maintaining a common prior distribution for the (i, j) element of Ω whenever the graph does not constrain the (i, j) element to be zero. Giudici (and other authors) previously stated the local priors were only suitable for decomposable models, but Roverato (2002) has extended this class of priors to general, non-decomposable models. Giudici (1996) suggests that the local priors encourage sparser graphs; for that reason, we will use the local priors. The computational issues are similar whichever class is chosen.

The prior $p(\Sigma|G)$ is *hyper-inverse Wishart*, $HIW(G, \delta, \Phi)$. Here Φ is a positive definite matrix and $\delta > 0$ are defining parameters, to be discussed further below. The prior density factors in a form related to the likelihood (2); in decomposable models,

$$p(\Sigma|G) = \frac{\prod_{C \in G^i} p(\Sigma_{CC}|G)}{\prod_{S \in S^i} p(\Sigma_{SS}|G)} \tag{3}$$

For each complete prime component C of G (and each separator), the corresponding sub-matrix of the covariance, Σ_{CC}, has an inverse Wishart(δ, Φ_{CC}) prior:

$$p(\Sigma_{CC}|G) = \frac{|\frac{\Phi_{CC}}{2}|^{\left(\frac{\delta+|C|-1}{2}\right)}}{\Gamma_{|C|}\left(\frac{\delta+|C|-1}{2}\right)} |\Sigma_{CC}|^{-\frac{\delta+2|C|}{2}} \exp\{-\frac{1}{2} tr(\Phi_{CC}\Sigma_{CC}^{-1})\} \tag{4}$$

where $\Gamma_k(a)$ is the multivariate gamma function:

$$\Gamma_k(a) = \pi^{\frac{k(k-1)}{4}} \prod_{i=0}^{k-1} \Gamma(a - \frac{i}{2}).$$

Decomposable graphs consist entirely of complete prime components, so equations (3) and (4) are sufficient to express the density of Σ when we restrict consideration to decomposable graphs.

The tractability of decomposable graphs is explained by the fact that while the graphical structure determines which entries of the covariance matrix appear in the density, the entries that do appear are in some sense unconstrained. Grone (1984) showed that when considering an incomplete covariance matrix (where only the entries corresponding to edges or on the diagonal are filled in), if the matrix can be completed to be a positive definite matrix consistent with the graph, this completion is unique. In this sense the entries of the graph on the diagonal and corresponding to edges are "free" and the other entries are functions of the these free entries. Grone also shows that if the submatrices corresponding to the cliques in a decomposable graph are positive definite, then a positive definite completion consistent with the graph always exists. This is reflected in the density for decomposable graphs: none of the "non-free" elements appear in either (3) or (4), so they do not affect the density at all. The free elements are constrained only to define full rank multivariate normal distributions on the cliques of the graph.

To deal with non-decomposable graphs we need an expression analogous to (4) for a non-complete prime component P. Roverato (2002) provides a generalization of the inverse Wishart as a prior density for Σ_{PP} (thus we will call the prior over Σ a hyper-inverse Wishart distribution just as in the decomposable case). The prior is derived as the Diaconis-Ylvisaker conjugate (Diaconis and Ylvisaker 1979) of the likelihood for Ω. In this density, some of the non-free elements of Σ_{PP} will appear; however, the true dimension of the density corresponds to the number of free elements. The free elements are determined by the edge set E, so we give the density argument as Σ^E_{PP}. The expression for the prior density is then:

$$p(\Sigma^E_{PP}|G) \propto |\Sigma_{PP}|^{-\frac{\delta-2}{2}} J(\Sigma^E_{PP}) \exp\{-\frac{1}{2}\Sigma^{-1}_{PP}\Phi_{PP}\} \tag{5}$$

where Σ_{PP} is the positive definite completion of Σ^E and $J(\Sigma^E_{PP})$ is the Jacobian of the transformation from Ω^E_{PP} (which has zeroes for off-diagonal entries not corresponding to edges in E) to Σ^E_{PP}. This density is obtained from a Wishart prior on Ω_{PP}, conditioned on Ω_{PP} consistent with G, by a change of variables. While also based on conditioning, this prior differs from the global prior of Giudici (1996) in that the conditioning is only used within the prime components. When using the hyper-inverse Wishart prior with an unrestricted graph space (non-decomposable models allowed) we constrain δ to be strictly greater than 2.0; it has not been shown that (5) has a finite integral for smaller δ.

The hyper-inverse Wishart prior is conjugate in either the decomposable or unrestricted case; the posterior is hyper-inverse Wishart$(G, \delta^* = \delta + n, \Phi^* = \Phi + S_y)$, where S_y is the sum of products matrix, $\sum_{i=1}^n y_i y_i'$. In examples below, we use $\Phi = \tau I$ for specified constants τ (other choices for Φ, such as an intra-class correlation structure, are considered in Giudici and Green 1999). This choice is consistent with problems in which variables represent measures of similarly defined quantities on a common scale. The form of the posterior makes it clear that it is important to choose τ to be on the appropriate scale, or it may dominate the effect of the data. In fact, increasing τ promotes increased prior probability on more complicated graphs (more edges); see Figure 3. The marginal prior mode for each variance term (σ_{ii}) is $\tau/(\delta + 1)$; we use this quantity to set an appropriate value for τ. For example, if the data has been standardized so all the variances are 1.0, τ might be set to $\delta + 1$.

3.3 Marginal Likelihood Functions over Graphs

The marginal likelihood function evaluated at any graph G is

$$p(Y|G) = \int_{\Sigma|G} p(Y|\Sigma)p(\Sigma|G)d\Sigma.$$

By noting that the prior normalizing constant and a factor of $(2\pi)^{-np/2}$ from the likelihood can be pulled outside the integral, this expression becomes a simple function of the prior and posterior normalizing constants, $h(G, \delta, \Phi)$ and $h(G, \delta^*, \Phi^*)$:

$$p(Y|G) = (2\pi)^{-np/2} \frac{h(G, \delta, \Phi)}{h(G, \delta^*, \Phi^*)} \tag{6}$$

For a decomposable graph, the hyper-inverse Wishart normalizing constants are a function of the normalizing constants for the inverse Wishart clique and separator densities given in (4):

$$h(G, \delta, \Phi) = \frac{\prod_{C \in G^i} |\frac{\Phi_{CC}}{2}|^{\left(\frac{\delta+|C|-1}{2}\right)} \Gamma_{|C|}\left(\frac{\delta+|C|-1}{2}\right)^{-1}}{\prod_{S \in S^i} |\frac{\Phi_{SS}}{2}|^{\left(\frac{\delta+|S|-1}{2}\right)} \Gamma_{|S|}\left(\frac{\delta+|S|-1}{2}\right)^{-1}}. \tag{7}$$

For non-decomposable graphs, the normalizing constant factors over the prime components of the graph as implied by (3), but the normalizing constants for non-complete prime components do not have closed form. Monte Carlo methods for estimating these normalizing constants are discussed in Section 5.

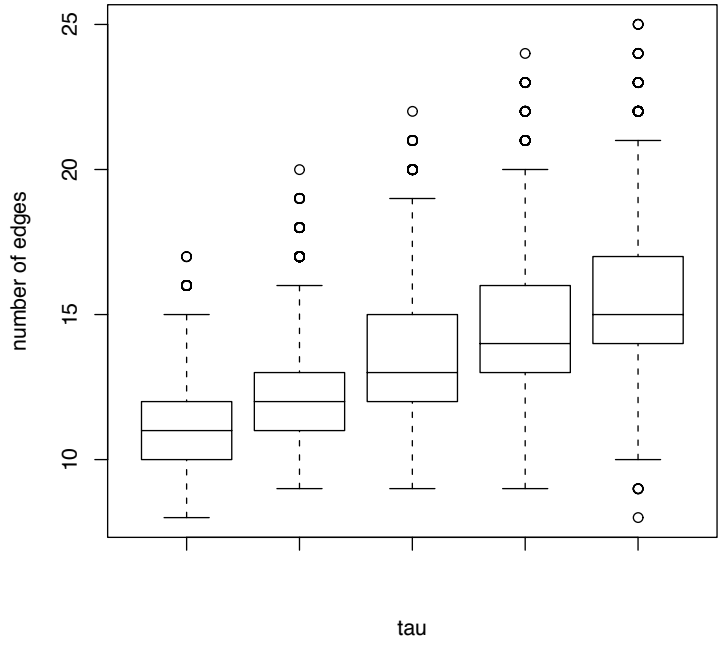


Figure 3: Boxplot of posterior samples of the number of edges, for different values of τ. The model and data are taken from the 12 node data in Section 9, with the posterior restricted to decomposable models.

4 Priors over Graphs

A uniform prior over all graphs, or all decomposable graphs, assigns most of its mass on graphs with a "medium" number of edges. The number of possible edges in a graph with p nodes is $T = p(p-1)/2$, so for large p a medium number of edges is quite large. The mass function peaks around $p(p-1)/4$ for general graphs; an estimate of the distribution when we restrict to decomposable graphs is seen in Figure 4. In both cases the average number of edges explodes very quickly as the number of nodes increases.

We would like to encourage parsimonious representations of the conditional independence structure, and discourage the inclusion of spurious edges; that is, to encourage *sparse* graphs, especially as dimension increases. To do this we use a Bernoulli prior on each edge inclusion probability with parameter $\beta = 2/(p-1)$. Thus a particular graph with $|E|$ edges has prior probability $\beta^{|E|}(1-\beta)^{T-|E|}$. This distribution has its peak at p edges for an unrestricted p node graph; the mode is somewhat lower when we restrict to decomposable graphs, as seen in Figure 5. (Both Figure 4 and Figure 5 are produced by sampling from the prior over graphs; the sampler used is the Metropolis Hastings algorithm described in Section 7, but with the marginal likelihood for each graph set to 1.)

Our approach to prior specification penalizes the number of edges, with the view that if choosing between two edges we want the edge resulting in the greatest increase of the graph's marginal likelihood, regardless of the rest of the graph's structure. One could, of course, penalize other measures of complexity such as the maximum or average prime component size. Wong, Carter and Kohn (2003) developed an approach that equalizes the prior probability of graphs with different numbers of edges; for decomposable graphs, this requires estimating the fraction of the total number of decomposable graphs with each number of edges.

5 Likelihood Computations for Non-Decomposable Models

For non-decomposable models, the normalizing constants corresponding to factors in equation (3) that represent non-complete prime components do not have closed form. They can be expressed as integrals over the space of Σ_{PP} compatible with the edge set E of the prime component P. To simplify

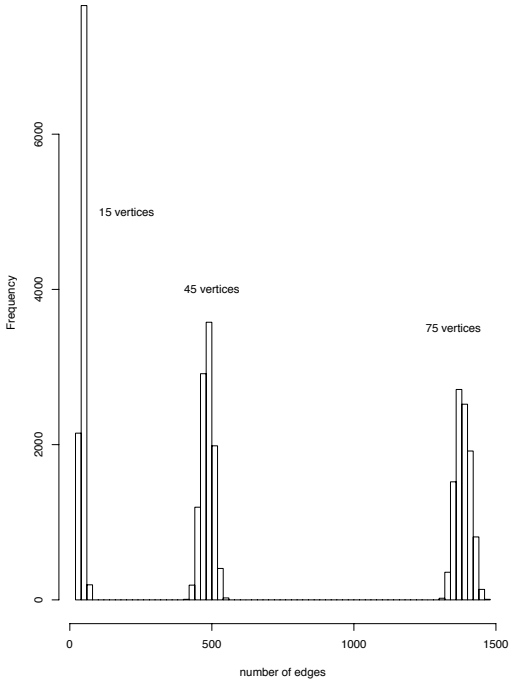


Figure 4: For each different number of nodes listed, the histogram represents prior mass on different numbers of edges, under a prior that is uniform over decomposable graphs. The histogram is based on sampling from the prior with a Metropolis-Hastings algorithm.

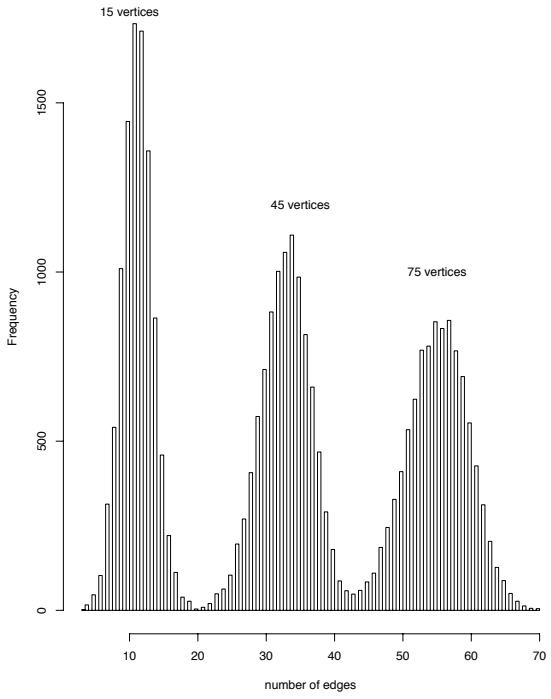


Figure 5: For each different number of nodes listed, the histogram represents prior mass on different numbers of edges, where consideration is restricted to decomposable graphs. The prior mass of a graph is proportional to $\beta^{|E|}(1-\beta)^{T-|E|}$, where $\beta = 2/(p-1)$. The histogram is based on sampling from the prior with a Metropolis-Hastings algorithm.

notation throughout this section, we will assume that P constitutes the whole graph, so that $\Sigma = \Sigma_{PP}$. We then have

$$h(P, \delta, \Phi) = \int_{\Sigma^E|P} |\Sigma|^{-\frac{\delta-2}{2}} \exp(-\frac{1}{2}\Sigma^{-1}\Phi) J(\Sigma^E), \tag{8}$$

or more simply, in terms of an integral over $(\Sigma)^{-1} = \Omega$:

$$h(P, \delta, \Phi) = \int_{\Omega^E|P} |\Omega|^{\frac{\delta-2}{2}} \exp(-\frac{1}{2}\Omega\Phi) d\Omega^E, \tag{9}$$

To estimate these integrals, we use the method presented in Atay-Kayis and Massam (2003), here after AM05. They exploit two changes of variables: from Ω^E to ϕ^E, the free elements of the upper triangular matrix produced by the Cholesky decomposition of Ω; and from ϕ^E to ψ^E, where $\psi = \phi T^{-1}$, and $T'T$ is the Cholesky decomposition of Φ. The point of this change is that the free elements of ψ are independent normals and square roots of χ^2 random variables, and thus are easily generated; the non-free elements can be straightforwardly computed from the free elements. Equation (9), written in terms of ψ, becomes:

$$h(P, \delta, \Phi) = \left(\prod_{i=1}^{|P|} 2^{\frac{\delta+\nu_i}{2}} (2\pi)^{\frac{\nu_i}{2}} \Gamma(\frac{\delta+\nu_i)}{2}) T_{ii}^{\frac{\delta+b_i-1}{2}} \right) \mathrm{E}_{\psi^E}(f_T(\psi^E)), \tag{10}$$

where ν_i is the number of neighbors of node i subsequent to it in the ordering of vertices, b_i is the total number of neighbors of node i plus 1, and

$$f_T(\psi^E) = \exp(-\frac{1}{2} \sum_{(i,j)\notin E, i<j} \psi_{ij}^2). \tag{11}$$

Because the distribution of ψ^E can be sampled from, it is straightforward to estimate the expectation of (11) by Monte Carlo. Note that when P is a clique, (11) evaluates to 1 and (10) simplifies to the inverse Wishart normalizing constant that appears in (4).

The method of AM05 builds on some of the ideas in Roverato (2002), where an importance sampling method to compute (9) is developed. Roverato uses an approximating decomposable model, with edge set E^* containing E. For this model, the change of variables from Ω^{E^*} to ϕ^{E^*} is performed; the distribution of the elements in ϕ^{E^*} is easily sampled from because of the

decomposability. If P is not decomposable, ϕ^E has more constrained elements than ϕ^{E^*}; however, values for the constrained entries compatible with E can be straightforwardly computed from the elements of ϕ^{E^*} corresponding to the free elements of ϕ^E. The resulting matrix is not precisely from the distribution of ϕ^E, and thus is re-weighted in order to compute the relevant expectation.

Dellaportas, Giudici and Roberts (2003) explore computing these normalizing constants in the context of global priors. However, at the component level the global priors differ from the local priors only in the degrees of freedom for the conditional Wishart distributions; thus their method could be used with local priors as well. Their method, like Roverato's, uses a change of variables, writes the normalizing constant as an expectation over those variables, and uses importance sampling to estimate that expectation. The sampling is based on multivariate normal random variables Z_i. For a complete graph with edge set E^*, Ω^{E^*} has a Wishart density. If the parameters are δ, Φ, a value for Ω^{E^*} can be generated as $\sum_{i=1}^{\delta} Z_i Z_i'$ where the Z_i have covariance Φ. Note that this restricts them to sampling from Wisharts with integral degrees of freedom. As for ψ and ϕ in the previous approaches, the constraints on Ω^E translate into non-free elements in the collection of Z_i's that can be computed as functions of the free elements. After modifying the non-free elements, the resulting variables (with an accompanying importance weight) can be used to estimate the desired expectation.

An additional contrast between the methods is in the maximum prime component size for a given number of data points. In Dellaportas's method and the method of AM05, the prime component size cannot exceed $n - 1$ where n is the sample size. Otherwise, the relevant submatrix of Φ^* is not necessarily invertible. In Roverato's method, it is the number of variables in the prime components of the triangulated graph that cannot exceed $n - 1$.

Throughout our examples we use the method of AM05 to compute marginal likelihoods for non-decomposable graphs. We prefer this method to the importance sampling approaches because it avoids the worries about the efficiency of the importance sampler, i.e. how far the distribution of the generated ϕ's or Z's may be from the desired distribution, as it is based on direct sampling via composition.

6 Local Updates in MCMC Methods for Decomposable Models

In addition to having analytical expressions for their normalizing constants, decomposable graphs have attractive properties in connection with "local updates" in model search based on MCMC or other related methods. In particular, for any two decomposable graphs G, G' that differ by one edge only, computing the marginal likelihood ratio $p(Y|G)/p(Y|G')$ is easy, requiring far less computation than computation from scratch of two likelihoods. This property is exploited in Giudici and Green (1999), and more fully explained in Wong and Carter (2002). Decomposable graphs differing by one edge have very similar cliques and separators. Suppose the graphs differ by an edge $\{a, b\}$. As both graphs are decomposable, we know that in the graph including $\{a, b\}$ this edge lies in a single clique (see Appendix B for more details). Call this clique C_q. At most one of a and b lies in the separator S_q. Suppose the larger graph has k cliques. Theorem 3 of Wong and Carter (2002) states that if $a \notin S_q$ the smaller graph replaces C_q with 2 cliques, $C_{q_1} = C_q/a$ and $C_{q_2} = C_q/b$, and a perfect ordering of the cliques in the smaller graph is $C_1, \ldots, C_{q-1}, C_{q_1}, C_{q_2}, C_{q+1}, \ldots, C_k$, with corresponding separators $S_2, \ldots S_{q-1}, S_q, S_{q_2} = C_q/\{a, b\}, S_{q+1} \ldots, S_k$. This fact, combined with the factorization of the graph marginal likelihood implied by (7) means taking the likelihood ratio between the two graphs results in cancellation of all terms except those involving C_q, C_{q_1}, C_{q_2} and S_{q_2}. In fact, Wong and Carter (2002) also show that the determinants needed for (4) can be computed using just two Cholesky decompositions, of Φ_{C_q,C_q} and $\Phi^*_{C_q,C_q}$ (See Appendix C for details).

In contrast, when we do not restrict ourselves to decomposable graphs there is no such guarantee of significant cancellations in the likelihood ratio between graphs that differ by one edge. While the likelihoods still factor over prime components, a single edge change may radically alter the junction tree of components. Imagine starting with a graph where all the nodes are connected in a chain, and then adding the edge that completes the full cycle. The single edge change moves us from a situation with $p-1$ prime components to a single prime component; there is no cancellation in the likelihood ratio.

7 Markov Chain Monte Carlo Algorithms

MCMC is a much used tool for exploring the space of graphical structures, (e. g. Madigan and York 1995, Dellaportas and Forster 1999, Giudici and Castelo 2003). In the context of Gaussian graphical models, Wong and Carter (2002) use their results to construct a fixed scan Gibbs sampler for decomposable graphs, where each edge was updated according to its full conditional distribution. Their results are also easily exploited in a Metropolis-Hastings sampler. We constructed three samplers to traverse the space of decomposable graphs: fixed scan Gibbs, Metropolis-Hastings where the edge to be updated was picked at random, and Metropolis-Hastings where the choice to add or delete an edge was made, and then an edge was selected at random from those appropriate for that type of move. There was no noticeable difference in performance between these closely related MCMC algorithms; the results presented are from the add/delete Metropolis-Hastings sampler.

We also implemented the add/delete Metropolis-Hastings sampler for an unrestricted search of graph space. When evaluating a proposal involving a non-decomposable graph, the algorithm described in Section 5 is used to evaluate the likelihood. This adds considerable computational burden; see Tables 1 and 2. In addition, because the local computation properties described in Section 6 no longer hold, we recompute the junction tree and entire likelihood for each proposed graph.

For problems with even a moderate number of variables (either in the decomposable or unrestricted space), the space to be explored is so large that a graph's frequency in the sample of graphs produced cannot be viewed as reflecting its posterior probability. Indeed, many graphs are not revisited after the chain leaves them. Posterior graph probability estimates must be based on normalizing the posterior mass function using the visited graphs, and these quantities will reflect the true posterior mass only to the extent that the majority of the mass has been visited. However, the frequencies of other quantities, such as the marginal probabilities of edge inclusion, can be viewed as posterior probabilities.

8 Shotgun Stochastic Search Algorithms

If Markov chain Monte Carlo is viewed merely as a tool for visiting high probability regions of graph space, there are certainly competing algorithms.

The following algorithm is attractive because step two (which contains most of the computational burden) can be easily parallelized.

1. Start with a graph G.
2. Select at random X_1 neighbors (graphs differing by 1 edge), compute their unnormalized posterior mass, and retain the top $X_2 \leq X_1$.
3. From among the X_2 top neighbors, propose the ith graph G_i as a new starting graph with probability proportional to p_i^α, where p_i is the unnormalized posterior probability of graph i and α is an annealing parameter.
4. Return to step 2 and iterate. Maintain a list of the overall best X_3 graphs visited.

In experimenting with this approach, we have typically used $X_1 = X_2 = T$, so all the neighbors are examined at each stage. We refer to this as a *Shotgun Stochastic Search* (SSS) method; at each step, we generate a large number of candidate models, "shooting out" candidates in all directions, and then following one (or, in a variant of the above, more than one) plausible candidate. Algorithms of this type can accommodate either unrestricted search of graph space, or restriction to decomposable graphs. When restricting to decomposable graphs, step 2 contains a check for decomposability; non-decomposable graphs are considered to have zero posterior probability.

Normalizing the (unnormalized) posterior probabilities within the list of the top X_3 graphs reflects their posterior probability to the extent that they contain most of the posterior mass. Unlike the Markov chain Monte Carlo algorithms, edge frequencies, weighted by the estimated posterior graph probabilities, also can be viewed as posterior probabilities only to the extent that the whole posterior mass is captured in the top X_3 graphs.

9 Simulated Examples

We first consider two simulated examples where the true underlying graph is known. The first graph, pictured in Figure 6, has 15 nodes and is decomposable. The second graph, pictured in Figure 7 consists of 12 nodes in a single non-complete prime component. Each data set consists of 250 observations.

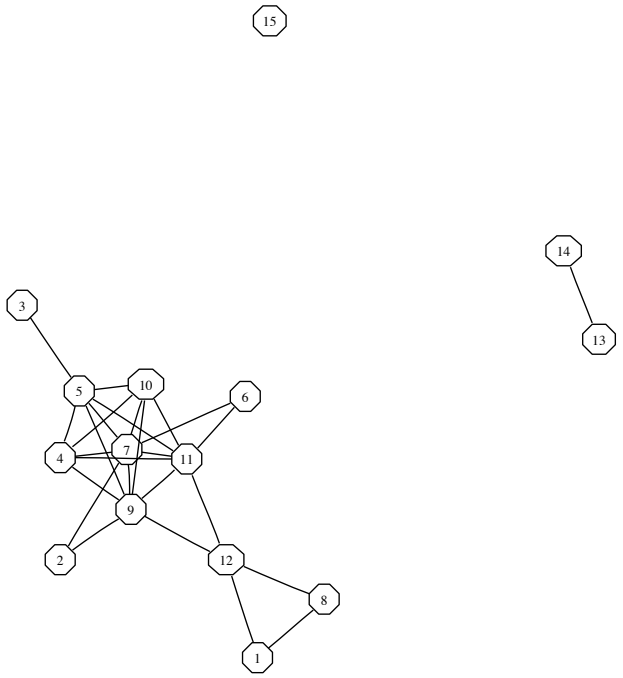


Figure 6: The true underlying decomposable graph on $p = 15$ nodes - the first simulated example

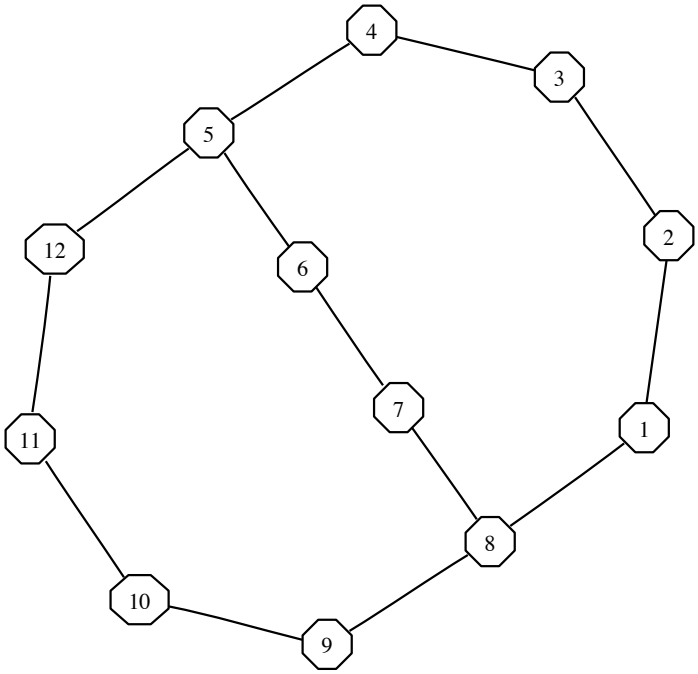


Figure 7: The true underlying non-decomposable graph on $p = 12$ nodes - the second simulated example

The first simulated data set was inspired by patterns of daily currency exchange fluctuations against the US dollar. Consequently, the data ranges approximately between $\pm 2\%$. We assume this range is about two standard deviations, so $\sigma_{ii}^2 \approx 0.0001$. We choose $\delta = 3$ so $\tau = .0004$. For the second data set, Σ is actually a random draw from the inverse Wishart(I,3) constrained to obey the graph; thus we use $\tau = 1$ and $\delta = 3$. In both cases the prior over graphs is the sparsity encouraging prior suggested in Section 4. For the shotgun stochastic search, the annealing parameter was set at 1.0 for simplicity. Performance of the algorithm in larger examples is very sensitive to the annealing parameter; see Section 10 for details.

9.1 Difficulties in Estimating Normalizing Constants in Non-Decomposable Models

To search the unrestricted model space, we must specify the number of random draws that will be used to estimate the normalizing constants for non-decomposable prime components. Initial runs with an insufficient number of draws (1000, regardless of prime component size) revealed an important and, we believe, both generic and limiting problem: high variance in the Monte Carlo draws of values of the marginal likelihood (standard deviation on the order of 2 units of log likelihood) created artificial local modes that it was difficult to escape from, so greatly inhibiting the movement of the MCMC chain.

To explore the behavior of the normalizing constant estimates, we examined non-complete prime components with different numbers of nodes. Two examples for each size were selected from those which occurred during the model search for the 15 variable data set. Because our search strategies depend on likelihood ratios, it is the variance of the log normalizing constants that are relevant for our purposes. Figures 8A and 8B show the variances of the estimated log of the prior and posterior normalizing constants (where the estimate is based on 100 random draws). The plotted variances are of course estimates themselves, each based on 1000 separate normalizing constant estimations.

The estimate of the log prior normalizing constant for a prime component P has a systematically smaller variance than the corresponding estimate for the posterior. This is not surprising since ψ, the sampled matrix from which the estimate are computed, has on its diagonal χ^2 random variables,

whose degrees of freedom range from δ to $\delta + |P|$ in the prior, but $\delta + n$ to $\delta + n + |P|$ for the posterior. In addition, the diagonal structure of Φ reduces the variance for the prior normalizing constant. The effect of the differing degrees of freedom along the diagonal of ψ is also reflected in how the variance of the prior normalizing constant can increase with the size of the prime component. However, in the posterior this effect can be dwarfed by the inherent difficulty in estimating the normalizing constant for very low probability components (see Figure 8B).

We also note that the ordering of the variables used when setting up ψ effects the variance of the log normalizing constant. Figure 8C shows variance differences in the estimated log of the prior normalizing constant that are the result of different orderings. Each prime component considered is a cycle; in the "optimal" configuration, each variable, except the first and the last, has exactly one neighbor preceding it in the rows of ψ and one following it. The "worst" configuration has the first $|P|/2$ variables each with two neighbors occurring further down in the matrix.

The cause of this phenomenon can be seen by factoring equation (10) into a constant C, and the part estimated by Monte Carlo, M:

$$C = \left(\prod_{i=1}^{|P|} 2^{\frac{\delta+\nu_i}{2}} (2\pi)^{\frac{\nu_i}{2}} \Gamma(\frac{\delta+\nu_i}{2}) T_{ii}^{\frac{\delta+b_i-1}{2}} \right) \tag{12}$$

$$M = \mathrm{E}_{\psi^E}(f_T(\psi^E)) \tag{13}$$

Recall that ν_i is the number of neighbors of node i subsequent to it in the ordering of vertices, so the relative sizes of C and M clearly depend on the order in which the variables are listed in the matrix (although their product is constant). In our experiments the variances of the estimates of M have roughly the same order of magnitude, regardless of the ordering; however, it is the variance of $\log(M)$ that is reflected in our plot. The first order approximation to the variance of $\log(M)$ is $1/M^2 Var(M)$. Thus, when an ordering is chosen that increases C and decreases M, the variance of $\log(M)$ increases. The "optimal" ordering for the prior normalizing constant of cycles discussed above minimizes C, the "worst" ordering maximizes it.

Similar multiplicative differences due to different orderings were observed in estimates of the log posterior normalizing constant; however, because of the appearance of the data (through the T_{ii}) in the expression for C, the ordering of the vertices that minimizes C depends on the data as well as the graph structure.

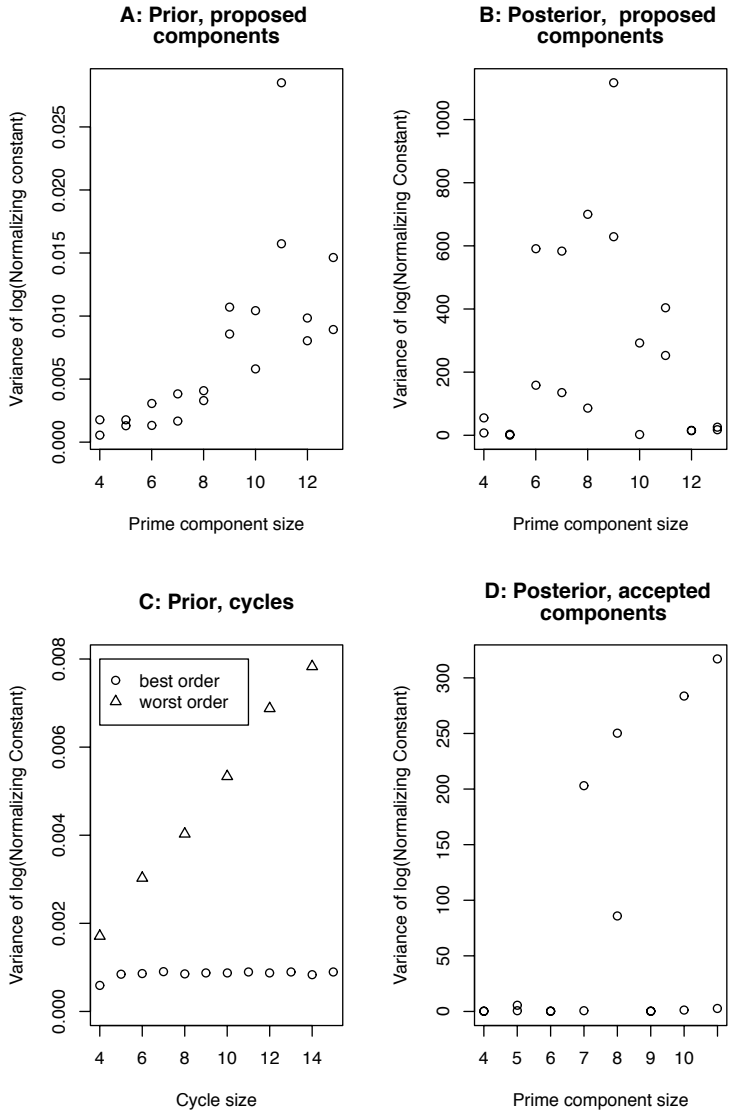
Attempts to reduce variances of all log marginal likelihoods considered and increase the number of Monte Carlo samples until the variance fell below a fixed level resulted in unacceptable computation times. However, the plot in Figure 8B is an arbitrary sample of graphs considered, and contains some very low likelihood graphs. These low likelihood graphs have both M and C small; the small M results in high variance for $\log(M)$. Figure 8D, a plot of variances of log posterior normalizing constants for prime components in *accepted* graphs, shows a trend more consistent with that in the plots of the variance of the log prior normalizing constants. The variance of the "worst case" for each component size seems to be a function of the size of the component considered, $|P|$. Based on this, a scheme using $|P|$ to determine the Monte Carlo iterations used was developed. We used $1.5|P|^3$ for the posterior normalizing constant and $0.5|P|^3$ for the prior normalizing constant. This scheme does not monitor the variance of our estimated log posteriors, but solved the problem with chain mobility discussed at the beginning of this section. At the end of our run, all graphs with a log posterior within 2.0 of the top log posterior were reexamined with enough Monte Carlo runs to ensure graph listed as "best" did indeed have the highest log posterior.

9.2 Results

For each example, the add-delete Metropolis was run for $10{,}000 \times T$ steps, both restricting to decomposable graphs and in unrestricted graph space. The search was started at the empty graph. The shotgun stochastic search algorithm was run so that it had the same number of graph likelihood evaluations as the Metropolis-Hastings. (For the decomposable case, a count of the "likelihood evaluations" includes examinations of graphs that are not decomposable and thus have zero likelihood.) Each iteration of the shotgun search algorithm includes evaluation of the T graphs that differ by one edge from the current graph, so 10,000 stochastic search iterations were run.

The algorithms clearly use a similar amount of computing resources; they are performing essentially the same calculations. However, the stochastic search algorithm is parallelizable. The run times for both types of algorithm, over the decomposable and unrestricted spaces, are given in Tables 1 and 2 to demonstrate the advantage of being able to exploit multiple processors. The Metropolis-Hastings was run on a Dell PC with a 1.8 MHz Xeon processor in a Linux environment, and the shotgun stochastic search on a Beowulf cluster with 26 dual processor, 1.4Mhz nodes.

Figure 8: Relationship between the variance of the estimated normalizing constants, based on 100 samples, and the size of the prime component. Four cases are considered: A, the prior normalizing constant for components proposed during the unrestricted model search for the 15 node data set, B, the posterior normalizing constants for these components, C, prior normalizing constants for cycles, using different variable orderings, and D, posterior normalizing constants for components considered during the unrestricted model search and subsequently accepted by the Metropolis-Hastings algorithm.



Method	Runtime (secs)	Max log posterior	Graphs to first top graph visit	Time to first top graph visit
MH-d	36	−2591.18	912	1
SSS-d	183	−2591.18	792	2
MH-u	15,220	−2590.94	415	2
SSS-u	2773	−2590.94	13,266	5

Table 1: Comparison between Algorithms of runtime, and quality of best graph found, for the 12 node example. MH-d(u) refers to the Metropolis-Hastings algorithm on decomposable (unrestricted) models, while SSS-d(u) refers to the shotgun stochastic search method on decomposable (unrestricted) models.

Method	Runtime (secs)	Max log posterior	Graphs to first top graph visit	Time to first top graph visit
MH-d	93	15633.76	349,484	36
SSS-d	234	15633.76	33,495	9
MH-u	513,077	15633.83	666,425	309,222
SSS-u	5930	15636.38	82,845	112

Table 2: Comparison between Algorithms of runtime, and quality of best graph found, for the 15 node example. MH-d(u) refers to the Metropolis-Hastings algorithm on decomposable (unrestricted) models, while SSS-d(u) refers to the shotgun stochastic search method on decomposable (unrestricted) models.

The "top" decomposable graphs – those identified with highest posterior probability – are pictured in Figures 9 and 10; the top graphs from the unrestricted search appear in Figures 11 and 9.2. Likelihood comparison with true graphs show that each of these graphs have greater likelihood support (as well as greater posterior support) than the true graph. For maximum *a posteriori* graphs, it is also seen that in general the edges included have higher estimated posterior probability than those not included. For all but the 15 node decomposable graph, the included edges all have higher probability than the excluded ones. (The 15 node decomposable graph includes one exception to this: the lowest probability included edge has probability 0.58, while the highest probability excluded edge has probability 0.60.) Thus aggregating high probability edges into a graph does not result in dramatically different graphs than taking the best graph found.

The most probable graph found in the 12 node case and the decomposable cases was insensitive to the starting point: the same graph was found starting at the complete graph. The unrestricted search for the 15 node case starting at the complete graph did not attain the likelihood for the top graph shown in the table.

10 150 Node Example: Gene Expression Data

A more challenging problem is the analysis of expression data from $p = 150$ genes associated with the estrogen receptor pathway, taken from $n = 49$ individuals; the data come from the study of West et al (2001). The data was standardized and the prior specified with $\delta = 3$, $\tau = 4$. In this context our sparsity-encouraging prior can be interpreted as a belief that on average, each gene has major interactions with a relatively small number of other genes. In this large example, we add to the decomposable model prior the restriction that the clique size not exceed the $n - 1$, in order to maintain the identifiability of the model. For the unrestricted model, we constrain each prime component to have fewer than $n - 1$ vertices.

The results from three algorithms are shown in Table 3. Times are now given in hours. Because the unrestricted search Metropolis-Hastings showed such poor performance, it was not used. In addition the best results for the shotgun search algorithm were obtained when an annealing parameter of 50 was used. This amounts essentially to deterministic hill climbing. In this large example we see that even in the decomposable case the shotgun

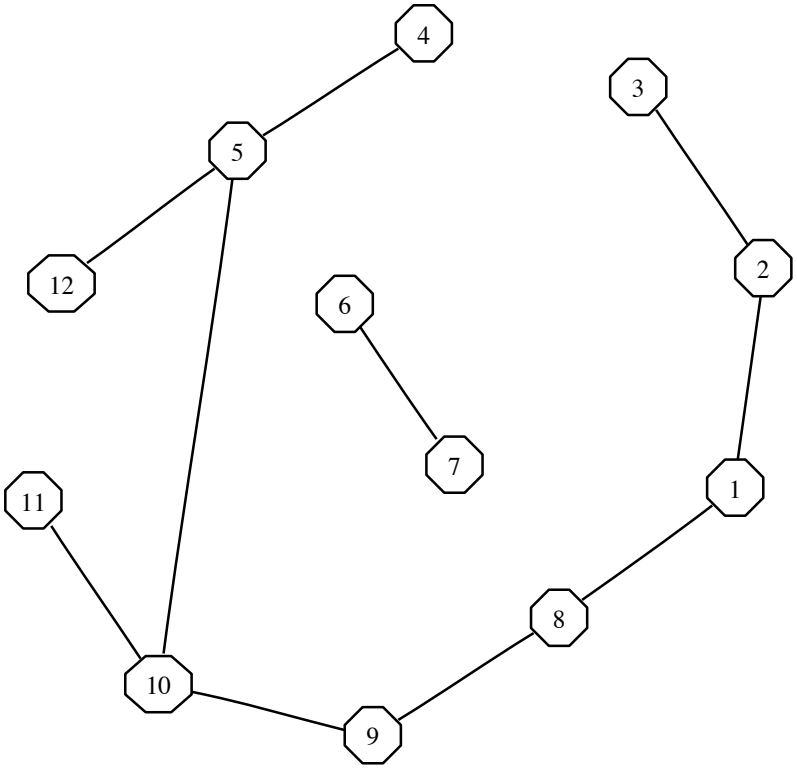


Figure 9: Highest log posterior graph for the 12 node example when the search is restricted to decomposable models.

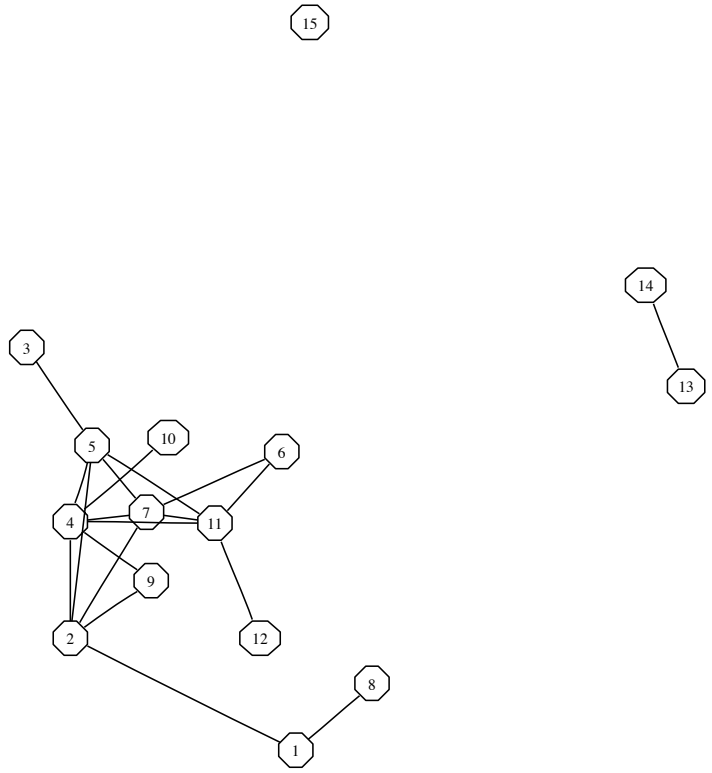


Figure 10: Highest log posterior graph for the 15 node example when the search is restricted to decomposable models.

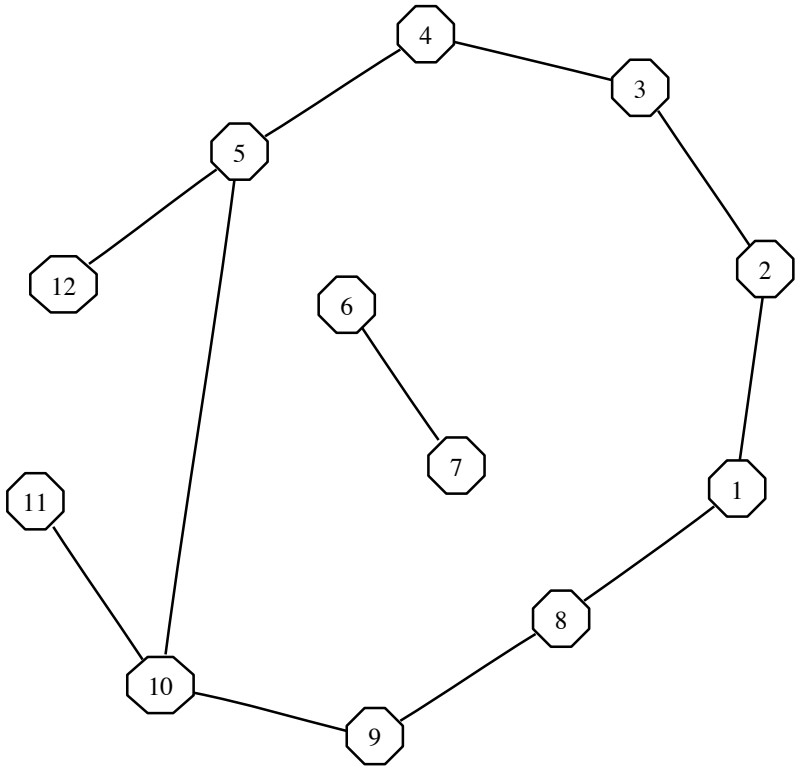


Figure 11: Highest log posterior graph for the 12 node example when the search is unrestricted.

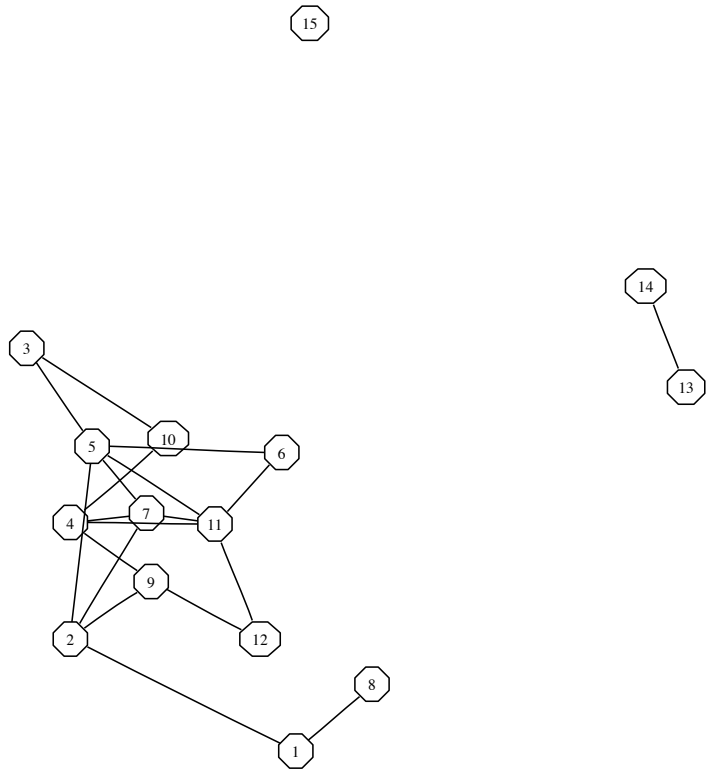


Figure 12: Highest log posterior graph for the 15 node example when the search is unrestricted.

Method	Runtime (hrs)	Max log posterior	Graph to first top graph visit	Time to first top graph visit
MH-d	18.02	−9417.97	100,466,818	6.51
SSS-d	0.03	−9260.84	1,698,600	0.03
SSS-u	6.29*	−9227.68	44,700	3.39

Table 3: Comparison between Algorithms of runtime, and quality of best graph found, for the gene expression example. *Starting from the best decomposable graph found. MH-d refers to the Metropolis-Hastings algorithm on decomposable models, while SSS-d(u) refers to the shotgun stochastic search method on decomposable (unrestricted) models.

stochastic search algorithm finds much better graphs.

A large annealing parameter was also used for the shotgun stochastic search in the unrestricted case. However, in this case the large annealing parameter does not eliminate the stochasticity of the search, as the marginal likelihoods are estimated with substantial error. Increasing the number of iterations enough to get a sharp evaluation of the likelihood resulted in an unacceptable computation time. Settling for a standard deviation of the log likelihood of 1.0 resulted in evaluating the neighbors of 1 graph (a single step in our stochastic search procedure) taking up to 40 computer days (1 day on a 40 node cluster). Using this procedure, starting from the empty graph and running until the estimated log posterior stopped improving, the best graph found had log posterior -9364.67, worse than the best decomposable graph. This graph may in fact represent a local mode not present in the decomposable framework, or be the result of sub-optimal moves resulting from the imprecise likelihood evaluation. The table shows the best graph found starting at the best decomposable graph (the final estimate of the log posterior for this graph was run with enough iterations to put the standard deviation below .1 units of log likelihood.) A total of 10 cycles of evaluating all neighbors were done. As these graphs were "close" to decomposable graphs, the time required to evaluate them was also reduced versus graphs with similar numbers of edges produced by the search starting at the empty graph.

11 Discussion and Other Approaches

Fitting of decomposable Gaussian graphical models using local move methods is feasible for large numbers of variables, certainly up to 100 or so. Exploration of model space to find high posterior probability graphs can be successfully carried out using direct search such as with our shotgun stochastic search method; traditional MCMC is competitive for relatively small graphs. However, unrestricted (non-decomposable and/or decomposable) model search is very much more problematic; it is easily accomplished for up to around 15 variables, but becomes very challenging quickly thereafter. Large prime components induce a major computational burden via the Monte Carlo estimation of the needed normalization constants; this estimation can be very unstable as dimension increases. Other methods are needed to deal with this computational problem. Local search of unrestricted graphs around "good" decomposable graphs or other candidate graphs is possible for 150 variables and represents a promising strategy. In both these cases, the method of choice is not a Markov chain Monte Carlo algorithm that attempts to sample from the posterior, but rather the shotgun stochastic search that is designed to generate many candidate graphs around a "current" graph, and then very rapidly traverse graph model space around sequences of "promising" models. The specific stochastic search algorithm we have introduced and exemplified here is easily parallelizable and, indeed, designed for distributed implementation. More experimentation with the annealing schedules is needed to find optimal strategies for different situations. For the 150 node decomposable model search presented as an example here, deterministic hill climbing produced the best results in terms of rapid identification of high probability graphs.

In the case of unrestricted search, new theoretical insights and methods are needed to improve the capacity to estimate the normalizing constants associated with non-complete prime components in a junction tree representation. One potential direction for research that would have immediate payoff involves a characterization of the changes in prime component structure when one edge moves are made from a current graph. Flores, Gomes and Olesen (2003) address this problem in the context of directed graphs; their results could be applied to provide characterization of prime component changes analogous to the results for clique changes in decomposable graphs used in Giudici and Green (1999). Correlating the marginal likelihood estimates of graphs that are to be compared by using the same random number draws to

estimate the normalizing constants involved may also improve computational efficiency.

A rather different view – a *constructive* approach – is to approach the development of graphical models through construction of specific classes of directed models. Dobra, Hans, Jones, Nevins, Yao, and West (2004) have recently introduced such an approach to high-dimensional Gaussian graphical model construction, building the full joint distribution up via composition using a triangular set of regressions representing the relationships between variables. This is related both to the dependency network framework of Heckerman, Chickering, Meek, Rounthwaite, and Kadie (2000) and approaches that model structure in the Cholesky decomposition of variance matrices; it is innovative in the creation of a constructive approach that scales with dimension and also utilizes priors consistent across graphs. While the probability models in that paper do not directly correspond to those considered here, our model can be easily evaluated for the conditional independence structure implied by the set of regressions. The regression based approaches are an appealing complement to those discussed here because the sets of "promising" models generated can be widely separated in term of one edge moves. Their method also can handle large sets of variables by using a prescreening procedure that limits which variables will considered possible predictors of others.

The method of Dobra et al (2004) also includes a complexity penalizing parameter; however, it does not directly correspond to our parameter β. We found that setting their complexity penalty to 0.5/(N-1) resulted in graphs with similar numbers of edges to ours. For the 150 node data set, we generated a candidate conditional independence structures and evaluated its posterior probability. The process took 24 hours. While longer than the total time used to produce the best unrestricted graph in Table 3, it should be noted this search procedure never restricts itself to decomposable graphs, and consequently may visit much different regions of graph space than our procedure for large variable sets, which starts from the best decomposable graph.

Follow-on research to understand the theoretical differences, in terms of prior specifications and the resulting impact on model search algorithms, between such constructive approaches and the MCMC/stochastic search framework described above is of some interest. Constructive approaches based on regressions inherently require an ordering of variables, and this can evidently have a major impact on computational efficiency but, more critically, on the

regions of graphical model space visited. However, our experiments with MCMC methods and greedy stochastic search related to MCMC methods lead us to conclude that a constructive approach of some form is needed to scale beyond moderate dimensions. The example in Dobra et at (2004) concerns gene expression data on over 12,000 genes, indicating that the approach is at least implementable with very large sets of variables; also, that example apparently identifies graphs that are, in the biological context, interpretable and consonant with known biology. Questions of adequacy of search over graph model space are challenging, however, as they are for our MCMC and search method. One general concept that seems very promising is to develop methods that are able to routinely generate "larger" jumps in graphical model space – such as the referenced constructive method – and then to integrate that with MCMC or search-based local-move methods applied around regions of model space so identified.

Finally, it is apparent that radical progress in this area, as in other areas of model and variable selection/search in the face of increasing dimension, is unlikely, in the near term, if computations are restricted to serial, single processors. Our experiments have heavily utilized a Beowulf cluster, and the development of search and constructive methods beyond moderate dimensions is, currently, simply not an option without embracing distributed computation. With increasing access to larger clusters for distributed computing, the computational statistics research community stands at an opportune time to substantially advance our ability to explore complex, high-dimensional model spaces based on more aggressively embracing technology and integrating it into day-to-day research.

Acknowledgments

This work was developed under the auspices of the inaugural SAMSI program on *Stochastic Computation* during 2003. The authors acknowledge the support of the National Science Foundation through the SAMSI grant DMS-0112069, on grants DMS-0102227 and 0112340 to Duke University, and by grants from the Keck Foundation and NIH. Graphical displays (Figures 6-12) are based on the AT&T GraphViz software.

References

Andersson, S. A., D. Madigan, M. D. Perlman, and T. Richardson (1998). Graphical Markov models in multivariate analysis. In S. Ghosh (Ed.), *Multivariate Analysis, Design of Experiments and Survey Sampling*. New York: Marcel Dekker Inc.

Atay-Kayis, A. and H. Massam (2005). The marginal likelihood for decomposable and non-decomposable graphical Gaussian models. *Biometrika*, to appear.

Cowell, R. G., A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter (1999). *Probabilistic Networks and Expert Systems*. New York: Springer-Verlag.

Dawid, A. P. and S. L. Lauritzen (1993). Hyper Markov laws in the statistical analysis of decomposable graphical models. *The Annals of Statistics 3*, 1272–1317.

Dellaportas, P. and J. J. Forster (1999). Markov chain Monte Carlo model determination for hierarchical and graphical log-linear models. *Biometrika 86*, 615–633.

Dellaportas, P., P. Giudici, and G. Roberts (2003). Bayesian inference for nondecomposable graphical Gaussian models. *Sankhya, Series A 65*, 43–55.

Dempster, A. P. (1972). Covariance selection. *Biometrika 32*, 95–108.

Deshpande, A., M. N. Garofalakis, and M. I. Jordan (2001). Efficient stepwise selection in decomposable models. In J. Breese and D. Koller (Eds.), *Uncertainty in Artificial Intelligence (UAI), Proceedings of the Seventeenth Conference*.

Diaconis, P. and D. Ylvisaker (1979). Conjugate priors for exponential families. *Annals of Statistics 7*, 269–281.

Dickey, J. M. (1971). The weighted likelihood ratio, linear hypotheses on normal location parameters. *Annals of Mathematical Statistics 42*, 204–223.

Dobra, A. and S. E. Fienberg (2000). Bounds for cell entries in contingency tables given marginal totals and decomposable graphs. *Proceedings of the National Academy of Science 97*, 11885–11892.

Dobra, A., C. Hans, B. Jones, J. Nevins, G. Yao, and M. West (2004). Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis 90*, 196–212.

Flores, M. J., J. Gamez, and K. G. Olesen (2003). Incremental compilation of Bayesian networks. In *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)*, San Francisco, CA, pp. 233–240. Morgan Kaufmann Publishers.

Frydenberg, M. and S. L. Lauritzen (1989). Decomposition of maximum likelihood in mixed models. *Biometrika 76*, 539–555.

Giudici, P. and R. Castelo (2003). Improving Markov chain Monte Carlo model search for data mining. *Machine Learning 50*, 127–158.

Giudici, P. and P. J. Green (1999). Decomposable graphical Gaussian model determination. *Biometrika 86*, 785–801.

GraphViz. Open source graph drawing software. AT&T Research Labs., http://www.research.att.com/sw/tools/graphviz/.

Grone, R., C. R. Johnson, E. M. Sà, and H. Wolkowice (1984). Positive definite completions of partial hermitian matricies. *Linear algebra and its applications 58*, 109–124.

Guidici, P. (1996). Learning in graphical Gaussian models. In J. M. Bernado, J. O. Berger, A. P. Dawid, and A. M. Smith (Eds.), *Bayesian Statistics 5*, pp. 621–628. Oxford Univeristy Press.

Hammersley, J. M. and P. E. Clifford (1971). Markov fields on finite graphs and lattices. Unpublished manuscript.

Heckerman, D., D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie (2000). Dependency networks for inference, collaborative filtering, and data visualization. *Journal Of Machine Learning Research 1*, 49–75.

Lauritzen, S. L. (1996). *Graphical Models*. Clarendon Press, Oxford.

Madigan, D. and J. York (1995). Bayesian graphical models for discrete data. *International Statistical Review 63*, 215–232.

Roverato, A. (2002). Hyper-inverse Wishart distribution for non-decomposable graphs and its application to Bayesian inference for Gaussian graphical models. *Scandinavian Journal of Statistics 29*, 391–411.

West, M., C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, M. J.R., and J. R. Nevins (2001). Predicting the clinical status of human breast cancer using gene expression profiles. *Proceedings of the National Academy of Sciences 98*, 11462–11467.

Whittaker, J. (1990). *Graphical Modles in Applied Multivariate Statistics*. Chichester, United Kingdom: John Wiley and Sons.

Wong, F., C. Carter, and R. Kohn (2003). Efficient estimation of covariance selection models. *Biometrika 90*, 809–830.

Wong, K. and C. Carter (2002). An efficient sampler for decomposable covariance selection models. preprint.

A Building the Junction Forest

A.1 Maximum Cardinality Search and Decomposable Graphs

In this section we will consider how to obtain a junction tree representation of a connected decomposable graph. To obtain a junction forest of a disconnected graph, the algorithm can be used on each connected component. Obtaining this representation for non-decomposable graphs builds on this algorithm and is considered in Section A.2. The junction forest is created by first establishing a perfect ordering of the nodes of the graph, using the following maximum cardinality search algorithm:

1. Pick a vertex v, and label it 1. While some unlabeled vertices remain, iterate the following procedure:

2. Suppose k unlabeled vertices remain. From among the vertices with the most labeled neighbors, pick a vertex and label it $p - k + 1$.

One can use this algorithm to check for decomposability of a graph by checking at each iteration of step (2) that all the labeled neighbors of the vertex to be added form a complete subgraph.

For decomposable graphs, the ordering of vertices established defines an ordering of cliques, where the cliques are ordered by the highest numbered node contained in each. This sequence has the running intersection property: for all $j > 1$, let S_j be the set of nodes shared with lower numbered cliques.

There is an $i < j$ such that $S_j \subset C_i$, and the S_j's are all complete. Thus S_k is a separator between $C_1, \ldots, C_{k-1}$ and C_k. This property shows us that the cliques can be arranged in a junction tree, where cliques are nodes, and if two nodes share a set of vertices, every prime component on the path between them in the junction tree also contains that set of vertices. Clique C_j may contain the separators of and therefore be connected to many higher numbered cliques in the junction tree, but it is connected to at most one lowered number clique in the tree. This prevents loops in the connections among cliques, telling us the structure is a tree. The highest numbered clique is a leaf, connected to only one other clique. While there may be many perfect orderings (for examples, leaves of the tree may be listed in any order among themselves) the junction tree is a unique representation.

A.2 Non-decomposable Graphs

Non decomposable graphs also have a junction forest representation, but in terms of the prime components $P_1 \ldots P_k$ rather than cliques. To get at this representation, we first triangulate the graph (add edges so that it is decomposable). A perfect ordering is then built as in Section A.1. The set of edges added during triangulation are called the *fill-in* edges. Now we will remove the fill-in edges and consolidate the prime components that were decomposed after the addition of these edges, while maintaining the running intersection property in our ordering of prime components. Any of the fill in edges not in $S_2, \ldots, S_k$ can simply be removed. To deal with the other edges, we start with the highest numbered separator S_j containing fill-in edges. We consolidate C_j and the lower numbered component adjacent in the junction tree that contains S_j, C_i. The sequence of cliques then reads $C_1, \ldots C_{j-1}, C_{j+1}, \ldots C_k$. This maintains the running intersection property– any separators contained in C_j are now contained in the lower numbered clique C_i. We repeat this process in sequence for each separator containing fill-in edges.

B One edge changes that maintain decomposability

It has long been known that an edge deletion maintains decomposability if that edge is contained in exactly one clique (see, for example, Frydenberg

and Lauritzen 1989). Giudici and Green (1999) give an efficient condition for checking whether an edge addition maintains decomposability. Decomposability is maintained if the vertices to be joined (a and b) are in different connected components or if there exist $R, T \subset V$ such that $a \cup R$ and $b \cup T$ are cliques, and $S = R \cap T$ is a separator on the path between $a \cup R$ and $b \cup T$ in the junction forest representation of the graph G. In our program, the junction forest representation of the graph is maintained, listing the cliques and separators of each component. When considering adding an edge between a and b in the same component, each possible combination of values of R and T are considered (these are defined by the clique memberships of a and b). For each of these combinations, it is determined whether $R \cap T$ is a separator. As demonstrated in Giudici and Green (1999), checking these conditions results in substantial time savings over checking the decomposability of the new graph with maximum cardinality search each time. Other conditions for checking whether edge addition maintains decomposability are given in Deshpande, Garofalakis, and Jordan (2001), however we found them more difficult to implement in practice.

C Computing likelihood ratios for decomposable graphs differing by one edge

This algorithm computes the likelihood ratio between two decomposable graphs, where the first differs from the second by the deletion of the edge a, b. As established in Section 6, this ratio involves the subsets of Φ and Φ^* corresponding to C_q, $C_{q_1} = C_q/a$, $C_{q_2} = C_q/b$, and $S_{q_2} = C_q/\{a, b\}$. Wong and Carter (2002) give a technique whereby the ratio can be computed using just two Cholesky decompositions, of $\Phi_{C_q C_q}$ and $\Phi^*_{C_q C_q}$. They partition $\Phi_{C_q C_q}$ as

$$\Phi_{C_q C_q} = \begin{pmatrix} \Phi_{S_{q_2} S_{q_2}} & \Phi_{S_{q_2} D} \\ \Phi_{D S_{q_2}} & \Phi_{DD} \end{pmatrix} \tag{14}$$

where $D = \{a, b\}$. The likelihood ratio can be written in terms of

$$\begin{aligned} \Phi_{DD|S_q} &= \Phi_{DD} - \Phi_{DS_{q2}} (\Phi_{S_{q2} S_{q2}})^{-1} \Phi_{DS_{q2}} \\ \Phi_{aa|S_q} &= \Phi_{aa} - \Phi_{aS_{q2}} (\Phi_{S_{q2} S_{q2}})^{-1} \Phi_{iS_{q2}} \\ \Phi_{bb|S_q} &= \Phi_{bb} - \Phi_{bS_{q2}} (\Phi_{S_{q2} S_{q2}})^{-1} \Phi_{bS_{q2}} \end{aligned}$$

and the corresponding quantities for Φ^* because

$$\begin{aligned}
\left|\Phi_{C_qC_q}\right| &= \left|\Phi_{DD|S_{q2}}\right|\left|\Phi_{S_{q2}}\right| \\
\left|\Phi_{C_{q1}C_{q1}}\right| &= \left|\Phi_{aa|S_{q2}}\right|\left|\Phi_{S_{q2}}\right| \\
\left|\Phi_{C_{q2}C_{q2}}\right| &= \left|\Phi_{bb|S_{q2}}\right|\left|\Phi_{S_{q2}}\right|
\end{aligned}$$

(and similarly for Φ^*). The Cholesky decomposition LL' of $\Phi_{C_qC_q}$ as partitioned in (14) is

$$L = \begin{pmatrix} L_{S_{q2}S_{q2}} & 0 \\ L_{DS_{q2}} & L_{DD} \end{pmatrix}$$

where

$$L_{DD} = \begin{pmatrix} l_{aa} & 0 \\ l_{ba} & l_{bb} \end{pmatrix}.$$

Then

$$\begin{aligned}
\Phi_{DD|S_q} &= L_{DD}L'_{DD} \\
\Phi_{aa|S_q} &= (l_{aa})^2 \\
\Phi_{bb|S_q} &= (l_{ba})^2 + (l_{bb})^2
\end{aligned}$$

giving all the necessary quantities to compute the likelihood ratio.

D Computer Code

C++ code related to the work reported here is available at the web site *www.isds.duke.edu* under the *Software* link. The algorithms for computing the prime component decomposition are based on those in Dobra and Fienberg (2000). Each of the four main approaches developed and explored here – Metropolis Hastings for decomposable graphs, Metropolis Hastings for unrestricted graphs, stochastic search for decomposable graphs and stochastic search for unrestricted graphs – is represented via a corresponding C++ program that was used to perform the analyses presented. The two stochastic search algorithms are designed to be implemented on a Beowulf cluster of computers using MPI. The programs are designed for use with variables that are centered at zero and on a common scale. In current form they accommodate priors over graphs of the form $\beta^{|E|}(1-\beta)^{T-|E|}$, where the user sets β, and inverse-Wishart priors over the graph parameters where the matrix

parameter is of the form τI. Users can freely edit the code to modify aspects of prior specification or search.

The two Metropolis-Hastings programs produce lists of all the graphs visited and their unnormalized posterior probabilities. The two stochastic search algorithms generate lists of the top X graphs in posterior probability, (and the associated unnormalized posterior probabilities) where X is specified by the user. As the stochastic search programs run they also list the incremental changes to the graph and its posterior for purposes of monitoring the extent of movement around graph space. For the algorithms that are not restricted to searches over decomposable graphs, the posterior probabilities of non-decomposable graphs are evaluated via Monte Carlo.

Additional details can be found at the web site referenced.