Sampling decomposable graphs using a Markov chain on junction trees

Peter J. Green* University of Bristol.
Alun Thomas† University of Utah.

April 20, 2011

Abstract

This paper makes two contributions to the computational geometry of decomposable graphs, aimed primarily at facilitating statistical inference about such graphs where they arise as assumed conditional independence structures in stochastic models. The first of these provides sufficient conditions under which it is possible to completely connect two disconnected cliques of vertices, or perform the reverse procedure, yet maintain decomposability of the graph. The second is a new Markov chain Monte Carlo sampler for arbitrary positive distributions on decomposable graphs, taking a junction tree representing the graph as its state variable. The resulting methodology is illustrated with two numerical experiments.

Some key words: conditional independence graph, graphical model, Markov chain Monte Carlo, Markov random field, model determination.

1 Introduction

Giudici and Green (1999) introduced a reversible jump Markov chain Monte Carlo (MCMC) sampler for posterior sampling of decomposable graphical models – described and implemented for the Gaussian case – which exploited a junction tree representation of decomposable graphs. This allows rapid checking of decomposibility for modified graphs and implementation of modifications, through local computation. The state variable in any such Markov chain must include a representation of the graph, along with associated parameter values. In that previous work, the decomposable graph itself is represented explicitly in the state variable. In this paper, we derive a more efficient sampler that augments the state variable by using a particular junction tree representation of the decomposable graph and dispenses with a direct representation of the graph itself.

We also make an important generalisation applicable to both samplers, allowing certain multiple-edge updates to the graph, maintaining decomposibility, and not only the single-edge moves seen in earlier work. Using this broader class of moves may improve performance in some situations. Our characterisation of a class of multiple-edge perturbations to a graph that maintain decomposability is likely to find broader application in computational graph theory, not only in the MCMC sampling of such graphs considered here.

In statistical science, the use of graphical models in inference is now very well-established, and it is not necessary to give a literature review here. Methodologies in which the graph itself is

*School of Mathematics, University of Bristol, Bristol BS8 1TW, UK.
Email: P.J.Green@bristol.ac.uk.
†Division of Genetic Epidemiology, Department of Internal Medicine, University of Utah.
Email: Alun.Thomas@utah.edu

one of the unknowns in the model are becoming common-place, as inference about the conditional independence properties of models fitted to data is a key part of understanding the stucture of data. More specifically, the single edge MCMC sampler described below has already been implemented in the `FitGMLD` program described by Abel and Thomas (2011). This program uses the sampler to fit a graphical model to the inter locus correlations between alleles at proximal genetic markers, a phenomenon usually referred to as linkage disequilibrium. By enforcing some model restrictions that allow a walking window approach, this implementation has been used on data representing over 100,000 variables assayed on hundreds of individuals.

1.1 Preliminaries on graphical models

We begin by reviewing some definitions and standard properties of decomposable graphs and junction trees.

Consider a graph $G = (V, E)$ with vertices V and (undirected) edges E. A subset of vertices $U \subseteq V$ defines an *induced subgraph* of G which contains all the vertices U and any edges in E that connect vertices in U. A subgraph induced by $U \subseteq V$ is *complete* if all pairs of vertices in U are connected in G. A *clique* is a complete subgraph that is maximal, that is, it is not a subgraph of any other complete subgraph.

A graph G is *decomposable* if and only if the set of cliques of G can be ordered as $(C_1, C_2, \ldots, C_c)$ so that for each $i = 1, 2, \ldots, c-1$

$$\text{if } \; S_i \; = \; C_i \cap \bigcup_{j=i+1}^{c} C_j \; \text{ then } \; S_i \subset C_k \; \text{ for some } \; k > i. \tag{1}$$

This is called the *running intersection property*. Note that decomposable graphs are also known as *triangulated* or *chordal* graphs and that the running intersection property is equivalent to the requirement that every cycle of length 4 or more in G is chorded.

The sets $S_1, \ldots S_{c-1}$ are called the *separators* of the graph. The set of cliques $\{C_1, \ldots C_c\}$ and the collection of separators $\{S_1, \ldots S_{c-1}\}$ are uniquely determined from the structure of G, however, there may be many orderings that have the running intersection property. The cliques of G are distinct sets, but the separators are generally not all distinct.

The *junction graph* of a decomposable graph has nodes $\{C_1, \ldots, C_c\}$ and every pair of nodes is connected. Each link is associated with the intersection of the two cliques that it connects.

Note that for clarity we will reserve the terms *vertices* and *edges* for the elements of G, and call those of the junction graph and its subgraphs *nodes* and *links*.

Let J be any spanning tree of the junction graph. J has the *junction property* if for any two cliques C and D of G, every node on the unique path between C and D in J contains $C \cap D$. In this case J is said to be a *junction tree*.

Some authors first partition a graph into its disjoint components before making a junction tree for each component, combining the result into a *junction forest*. The above definition, however, will allow us to state results more simply without having to make special provision for nodes in separate components. In effect, we have taken a conventional junction forest and connected it into a tree by adding links between the components. Each of these new links will be associated with the empty set and have zero weight. Clearly, this tree has the junction property. Results for junction forests can easily be recovered from the results we present below for junction trees.

A junction tree for G will exist if and only if G is decomposable, and algorithms such as the *maximal cardinality search* of Tarjan and Yannakakis (1984) allows a junction tree representation to be found in time of order $|V| + |E|$ (where $|\cdot|$ denotes the cardinality of a set). The collection of clique intersections associated with the $c-1$ links of any junction tree of G is equal to the collection

of separators of G. The junction property ensures that the subgraph of a junction tree induced by the set of cliques that contain any set $U \subseteq V$ is a single connected tree.

A complete treatment of graphical models is given by Lauritzen (1996), to whom we refer readers for terminology not defined above; see also Thomas and Green (2009a).

1.2 Elaborating the model to include the junction tree

Each decomposable graph G can be equivalently represented by one or more junction trees. Thomas and Green (2009b) derived an expression for $\mu(G)$, the number of equivalent junction trees. Given a probability distribution $\pi(G)$ on decomposable graphs (which might, for example, be the posterior distribution of the conditional independence graph of a multivariate distribution, given data), we can define a distribution on junction trees simply by

$$\widetilde{\pi}(J) = \frac{\pi(G(J))}{\mu(G(J))}$$

where $G(J)$ is the decomposable graph represented by J – that is, conditional on G distributed as $\pi(G)$, J is distributed uniformly at random from among the $\mu(G)$ equivalent junction trees. We assume throughout that $\pi(G) > 0$ for all decomposable G, so that $\widetilde{\pi}(J) > 0$ for all junction trees J.

We will construct an ergodic Markov chain whose states are junction trees, with invariant distribution $\widetilde{\pi}$.

1.3 Structure of the paper

In section 2, we discuss perturbations to a decomposable graph through adding and removing edges that maintain decomposability; this include some new results on multiple-edge updates. We go on in Section 3 to define the junction tree sampler, a Markov chain Monte Carlo method for sampling from a prescribed distribution over junction trees. Finally in Section 4, we present numerical examples: one that serves to verify the correctness of the sampler and a second one demonstrating successful posterior sampling for a real graphical gaussian model on 50 variables.

2 Allowable perturbations to decomposable graphs

2.1 Single-edge perturbations

We first follow previous work in concentrating on MCMC moves that perturb the graph in a very simple way – they connect or disconnect two vertices x and y by adding or removing an edge between them. In general, such a move may destroy the decomposability of the graph, and it is therefore necessary either to test that the perturbed graph is decomposable, or in some way to limit the choice of (x, y) to guarantee in advance that it is decomposable.

Frydenberg and Lauritzen (1989) and Giudici and Green (1999) gave efficient methods for checking that the perturbed graph G' is decomposable, given that G is, when the perturbation scheme involves either connecting or disconnecting an arbitrary pair of vertices. Using our definition of a junction tree, we can restate their results as follows.

(C) Connecting x and y by adding an edge (x, y) to G will result in a decomposable graph if and only if x and y are contained in cliques that are adjacent in some junction tree of G.

(D) Disconnecting x and y by removing an edge (x, y) from G will result in a decomposable graph if and only if x and y are contained in exactly one clique.

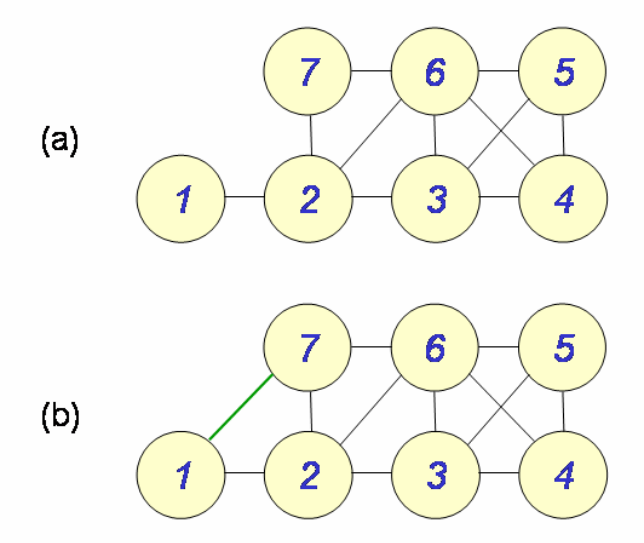

Figure 1: Two small decomposable graphs, differing by the presence of a single edge.

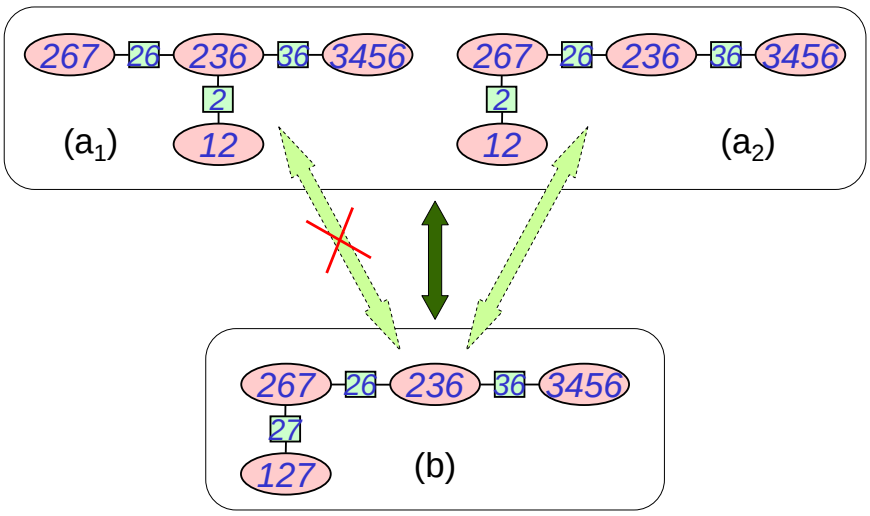

Figure 2: Junction trees corresponding to the decomposable graphs in Figure 1: ellipses represent cliques, and boxes the separators. Trees (a_1) and (a_2) correspond to graph (a), and tree (b) to graph (b).

Figure 1 illustrates two small decomposable graphs, differing by the presence of a single edge. The conditions (C) and (D) stated above are clearly satisfied for this example. Junction trees corresponding to these graphs are shown in Figure 2.

These observations are key to deriving both the sampler in Giudici and Green (1999) and that in the present work. The key difference in the two approaches lies in the phrase "adjacent in some junction tree of G" which we replace with "adjacent in *this* junction tree of G". In Giudici and Green (1999), where the graph is (part of) the state variable, we manipulate the junction tree, searching for one for which the cliques containing x and y are adjacent, and then use that junction tree to effect the perturbation. In the present work, where the junction tree is (part of) the state variable, there is no such manipulation, and the proposal mechanism is modified so that x and y are only selected if the cliques containing x and y are already adjacent. Figure 2 illustrates this point. In the sampler of Giudici and Green (1999), moves between graphs (a) and (b) are possible, even if graph (a) is currently represented by junction tree (a_1); the first stage of the move is manipulation from tree (a_1) to (a_2). However, in the sampler introduced here, moves between trees (a_2) and (b) are possible, in either direction, but not between (a_1) and (b).

Thus the computational cost savings in our new approach come from the more restrictive choice of proposed pairs (x, y) specifying edges to be added, and avoidance of the manipulation from one junction tree to another, and the price paid is that the space of possible (junction tree) states of the chain is in some sense less connected. We shall see in Section 4.2 that this price is worth paying, especially in larger graphs.

It might be useful at this point to consider for illustration the specific but extreme case when G is the trivial graph with n vertices and no edges. The cliques all contain a single vertex, and any tree J connecting these vertices is a valid junction tree, using our generalized formulation. J will have $n-1$ edges, and by Cayley's theorem (Cayley 1889) we know that it is one of n^{n-2} possible junction tree representations of G.

Under the scheme of Giudici and Green (1999), one of the $n(n-1)/2$ possible pairs of vertices would be selected at random and on inspection and manipulation of J, connecting this pair would be found to make a valid decomposable graph. With very high probability, $(1-2/n)$, this will require changing J into an alternative junction tree J' in which the cliques comprising the selected vertices of G are connected.

Under our new scheme, one of the $n-1$ links of J would be selected at random. The vertices making up the cliques that the link joins would be found to contain a pair of vertices whose connection forms a decomposable graph. The computational saving is that no manipulation of J is required to establish this. The cost is that only $n-1$ of the possible $n(n-1)/2$ pairs of vertices can be thus sampled. This may be aleviated to some extent by occasionally using the randomization step described by Thomas and Green (2009b) which allows a junction tree to be replaced by an equivalent one chosen uniformly at random from the n^{n-2} junction tree representations of G.

In both algorithms, the effect on the junction tree of connecting or disconnecting x and y is shown schematically in Figure 3. In each case the upper panel shows part of the junction tree with x and y unconnected; the lower panel the same part of the tree with them connected. The figures can be "read" in both directions. The symbol S denotes the separator between the cliques containing x and y referred to in the condition (C) for connecting by adding an edge, and $XYS = \{x, y\} \cup S$ is the clique containing both x and y referred to in the condition (D) for disconnecting by removing an edge. The 4 cases correspond to the 2×2 possibilities that the cliques containing x and y are exactly $XS = \{x\} \cup S$ and $YS = \{y\} \cup S$ respectively, or supersets thereof.

These single-edge perturbations to G are a special case of the multiple-edge perturbations defined and justified in the next section, so we omit the proofs that the modifications maintain decomposability.

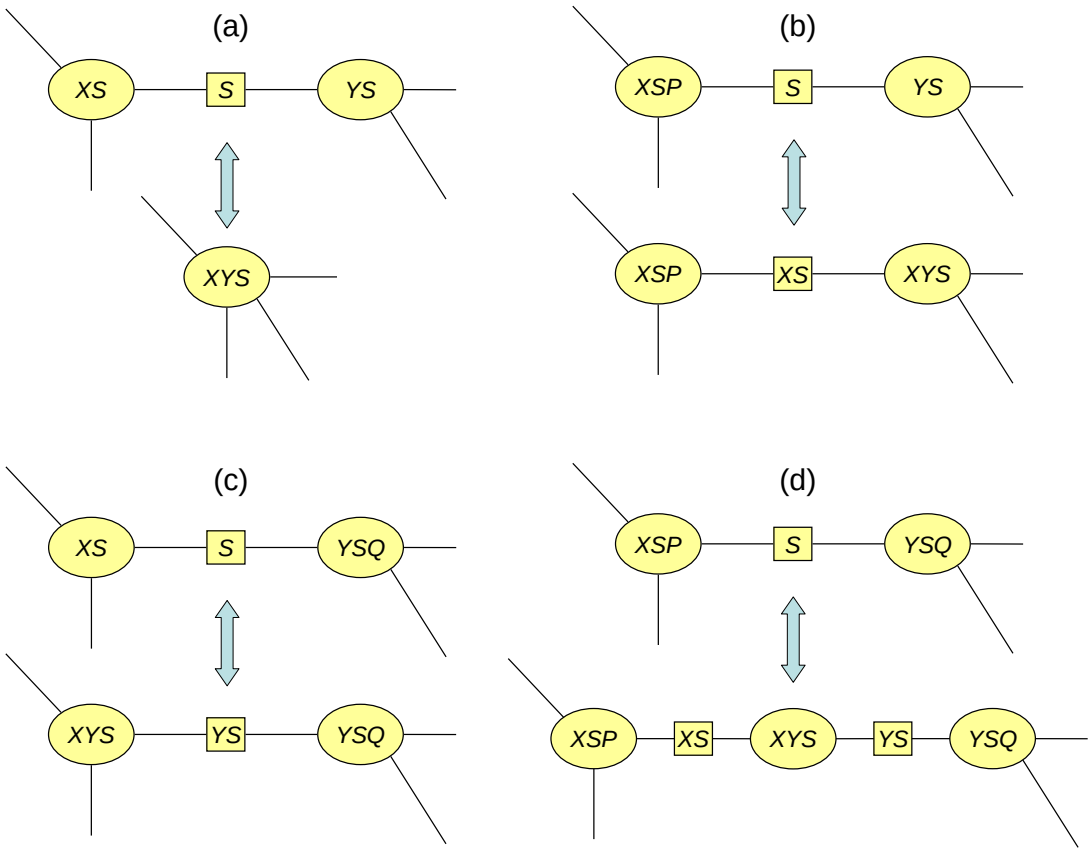

Figure 3: The 4 possible cases: the clique containing $X = \{x\}$ and S before connecting $X = \{x\}$ and $Y = \{y\}$ is in cases (a) and (c) exactly $XS = X \cup S$, while in cases (b) and (d) it is a proper superset; similarly the clique containing $Y = \{y\}$ and S before the connection is in cases (a) and (b) exactly YS and in (c) and (d) a proper superset. These four cases have to be considered both in the proof that decomposability is maintained (Section 2.2 and Appendix 1) and in the algorithm for making valid connections and disconnections (Section 3).

2.2 Multiple-edge perturbations

In this section, we present perturbations to decomposable graphs that make multiple connections and disconnections simultaneously, yet are guaranteed to maintain decomposability. Unlike the single-edge moves of the previous section, however, these provide only sufficient, not necessary, conditions for the validity of the perturbations to G.

Two disjoint non-empty connected sets of vertices X and Y are said to be *completely connected* if every vertex in X is connected to every vertex in Y. They are *completely disconnected* if no vertices in X are connected to any vertices in Y.

Proposition 1. *Suppose $G = (V, E)$ is a decomposable graph, and that X and Y are two disjoint non-empty subsets of V that are each complete in G, and which are completely disconnected (i.e. there are no edges (x, y) between any element $x \in X$ and $y \in Y$). Suppose X and Y are subsets of cliques that are adjacent in some junction tree representing G.*

Let G' be the graph formed from G by completely connecting X and Y (i.e. inserting an edge between every pair of vertices (x, y) with $x \in X$ and $y \in Y$).

Then G' is decomposable.

Proposition 2. *Suppose $G = (V, E)$ is a decomposable graph, and that X and Y are two disjoint non-empty subsets of V that are completely connected (i.e., $X \cup Y$ is complete in G), such that X and Y are subsets of exactly one clique, $X \cup Y \cup S$, say, where $S \cap (X \cup Y) = \emptyset$. Suppose that one of the following holds:*

(a) there is no other clique containing $X \cup S$ or $Y \cup S$,

(b) there is one more clique containing $X \cup S$ but then no other cliques intersecting X, and there are no more cliques containing $Y \cup S$,

(c) there is one more clique containing $Y \cup S$ but then no other cliques intersecting Y, and there are no more cliques containing $X \cup S$, or

(d) there are two more cliques containing $X \cup S$ and $Y \cup S$ respectively, but then no other cliques intersecting X or Y, and there is a junction tree J representing G such that there are no other cliques adjacent to $X \cup Y \cup S$ in J.

Let G' be the graph formed from G by disconnecting X and Y (i.e. removing all edges between pairs of vertices (x, y) with $x \in X$ and $y \in Y$).

Then G' is decomposable.

Remarks. These propositions are presented separately, and it may not be immediately clear that there is a unity to them (in particular it may seem that the conditions in Proposition 2 are much more stringent that those in Proposition 1). In fact, however, they are perfectly matched, since as implemented they precisely delineate the circumstances in which particular moves applied to a junction tree form a reversible pair.

Thus in practical use, the junction tree J representing G is already determined before the connection or disconnection of X and Y is considered. Indeed, given J the only X and Y that will ever be considered are those for which this particular junction tree satisfies the conditions mentioned in Proposition 1 and Proposition 2, part (d).

Finally, we will see that the moves that these propositions confirm are valid (i.e. maintain decomposability) can always be implemented by modest local perturbations to the current junction tree. These local perturbations are illustrated in Figure 3.

The proofs of these propositions are deferred to Appendix 1, following specification in Section 3 of the algorithms that will implement these perturbations to G, which provides further notation and describes the local perturbations of the junction tree in detail.

Other variant multiple-edge perturbations are possible, but not considered here.

3 The junction tree sampler

An early version of our junction tree sampler employed only single-edge connect and disconnect moves, but was later generalised to allow multiple-edge connects and disconnects, following the analysis of multiple-edge perturbations in Section 2.2. These multiple-edge moves involve choices of appropriate random sets of vertices X and Y in the algorithms detailed below; for the single-edge versions these choices are restricted to be singletons $\{x\}$ and $\{y\}$ respectively, and there are no other changes; therefore, we do not describe the single-edge moves separately.

3.1 Multiple-edge connect move

We first choose a separator S uniformly at random from the collection of separators $\mathcal{S}(J)$ in the current junction tree J, respecting multiplicities of course. If $\mathcal{S}(J)$ is empty, which is the case only if the graph consists of a single clique, no further connection is possible, and we reject immediately.

Suppose S separates cliques C_X and C_Y: we choose non-empty sets of vertices X and Y from $C_X \setminus S$ and $C_Y \setminus S$, (whose joint probability distribution is to be decided later). By criterion (C), completely connecting X and Y yields a new decomposable graph, one junction tree representation of which, J', can be easily formed as follows:

(a) If $C_X = X \cup S$ and $C_Y = Y \cup S$, then C_X, C_Y and S are removed from the junction tree, and replaced by a new clique $X \cup Y \cup S$, connected to all those cliques previously connected to C_X or C_Y, through the same separators as before.

(b) If $C_X \supset X \cup S$ and $C_Y = Y \cup S$, then the vertices in X are added into S and C_Y, and the junction tree otherwise left unchanged.

(c) If $C_X = X \cup S$ and $C_Y \supset Y \cup S$, then the vertices in Y are added into S and C_X, and the junction tree otherwise left unchanged.

(d) If $C_X \supset X \cup S$ and $C_Y \supset Y \cup S$, then the separator S is replaced by a separator / clique / separator triple: $X \cup S$, $X \cup Y \cup S$, $Y \cup S$, with C_X connected to the first, and C_Y to the last, and the junction tree otherwise left unchanged.

These four possibilities are represented graphically in Figure 3, reading downwards.

3.2 Multiple-edge disconnect move

For the reverse move, we first draw a clique C at random from the collection of cliques $\mathcal{C}(J)$ of the current junction tree J. If C contains a single vertex, the proposal is rejected. Using a probabilistic mechanism to be decided later, we then partition C at random into three sets X, Y and S, where X and Y at least are non-empty.

The neighbours of C in the junction tree J are then scanned; if any neighbour intersects both X and Y, then disconnecting X and Y is not possible, and the proposal is rejected.

Otherwise, we partition the neighbours into 3 sets: $\mathcal{N}$, those intersecting neither X nor Y, $\mathcal{N}_X$, those intersecting only X, and $\mathcal{N}_Y$, those intersecting only Y. Among the cliques in $\mathcal{N}_X$, we select an arbitrary one of any encountered that contains all of $X \cup S$ and identify this as C_X; if none are encountered, the set is left undefined. Similarly, we look in $\mathcal{N}_Y$ to try to identify C_Y.

(a) If neither of C_X and C_Y are defined, then X and Y are disconnectible: we create new cliques $C \setminus Y = X \cup S$ and $C \setminus Y = Y \cup S$, with a separator S between them. The first of these is connected to those cliques in $\mathcal{N}_X$ and the second to those in $\mathcal{N}_Y$. Those in $\mathcal{N}$ are connected at random to one of the new cliques. Finally the clique C is deleted.

(b) If C_X is defined, but not C_Y, then disconnection is possible if and only if $\mathcal{N}_X$ contains exactly one clique, C_X itself: in this case, X is removed from the clique C and from the adjacent separator $C \setminus Y = X \cup S$ connecting it to C_X; the junction tree is otherwise unchanged.

(c) If C_Y is defined, but not C_X, then disconnection is possible if and only if $\mathcal{N}_Y$ contains exactly one clique, C_Y itself: in this case, Y is removed from the clique C and from the adjacent separator $C \setminus X = Y \cup S$ connecting it to C_Y; the junction tree is otherwise unchanged.

(d) If both of C_X and C_Y are defined, then X and Y can only be disconnectible if $\mathcal{N}$ is empty, and both $\mathcal{N}_X$ and $\mathcal{N}_Y$ contain exactly one clique. In this case, the clique C and its adjacent separators $C \setminus Y = X \cup S$ and $C \setminus X = Y \cup S$ are removed from the junction tree, and replaced by a separator S linking the cliques C_X and C_Y.

These four possibilities are represented graphically in Figures 3, reading upwards.

3.3 Choices of X and Y, and associated proposal probabilities

Whether using single-edge or multiple-edge updates, in each of the connect and disconnect moves we have at one point to choose sets of vertices X and Y at random, subject to the stated constraints. Providing that the probabilities with which these choices are made are correctly encoded into the Metropolis–Hastings acceptance calculation through the proposal probabilities $q(J, J')$, the junction tree sampler satisfies detailed balance whatever probability distribution for X and Y is used. Varying this choice allows scope for improving performance, although we have not conducted any systematic experiments on this issue.

In the single-edge case, $X = \{x\}$ and $Y = \{y\}$ are both singletons, and we have few options. For the connect move, we choose x and y uniformly at random from $C_X \setminus S$ and $C_Y \setminus S$ respectively. The probability $q(J, J')$ that starting from J leads to the proposed modified junction tree J' specified in Section 3.1, following the uniform random choice of S, is easily seen to be $1/[|\mathcal{S}(J)| \times (m_X - s) \times (m_Y - s)]$, where $m_X = |C_X|$, $m_Y = |C_Y|$ and $s = |S|$.

For the disconnect move, we choose x and y uniformly at random without replacement from C; then the process in Section 3.2 yields the proposal probability $[1/|\mathcal{C}(J)|] \times [2/m(m-1)] \times 2^{-|\mathcal{N}|}$ in case (a), and otherwise $[1/|\mathcal{C}(J)|] \times [2/m(m-1)]$, where $m = |C|$. The factor 2 in the numerator accounts for the fact that the effect of the move on the junction tree is not affected by the order in which X and Y are drawn.

Turning to the multiple-edge case, out of wider ranges of options we choose the simplest. For the connect move, to select X from $C_X \setminus S$, we first pick N_X uniformly at random between 1 and $|C_X \setminus S|$ and then choose X to be a subset of $C_X \setminus S$ of that size chosen uniformly at random from all such. We choose N_Y and Y similarly, and independently. The proposal probability is

$$\frac{1}{|\mathcal{S}(J)|} \times \frac{1}{m_X - s} \frac{N_X!(m_X - s - N_X)!}{(m_X - s)!} \times \frac{1}{m_Y - s} \frac{N_Y!(m_Y - s - N_Y)!}{(m_Y - s)!}$$

For the disconnect move, we choose M uniformly at random between 2 and $m = |C|$, then N uniformly at random between 1 and $M - 1$. We then partition C into sets X, Y and S of sizes N, $M - N$ and $m - M$, respectively, uniformly at random from all such partitions. This sampling can

be conducted efficiently in a single pass through C. The proposal probability is

$$\frac{1}{|\mathcal{C}(J)|} \times \frac{2}{(m-1)(M-1)} \times \frac{N!(M-N)!(m-M)!}{m!}.$$

Again this has to be multiplied by an additional factor $2^{-|\mathcal{N}|}$ in case (a).

In the multiple-edge moves, note that the proposal probabilities are random, even conditional on J. Although unusual, this is valid, as shown by Besag *et al.* (1995), in their Appendix 1.

3.4 Acceptance probabilities for detailed balance

The well-known standard 'Metropolis–Hastings' acceptance probability for this proposal (Hastings 1970) is

$$\alpha(J, J') = \min\left\{1, \frac{\widetilde{\pi}(J')q(J', J)}{\widetilde{\pi}(J)q(J, J')}\right\}$$

which ensures detailed balance with respect to the target distribution $\widetilde{\pi}(J)$.

A fact that is well known but not commonly exploited is that the acceptance probability expression cited above is not the only choice yielding detailed balance. For example, consider the alternative choice of acceptance probability

$$\widetilde{\alpha}(J, J') = \min\left\{1, \frac{\widetilde{\pi}(J')}{\widetilde{\pi}(J)}\right\} \times \min\left\{1, \frac{q(J', J)}{q(J, J')}\right\}.$$

Then the equilibrium joint probability of the chain being in state J followed by $J' \neq J$ is

$$\widetilde{\pi}(J)q(J, J')\widetilde{\alpha}(J, J') = \min\left\{\widetilde{\pi}(J), \widetilde{\pi}(J')\right\} \times \min\left\{q(J, J'), q(J', J)\right\},$$

an expression evidently symmetric in J and J'. Thus this chain is also reversible, with the same invariant distribution $\widetilde{\pi}(J)$. We are not aware of this expression being given before, in spite of its simplicity and broad applicability.

According to the important result of Peskun (1973), since $\widetilde{\alpha}(J, J') \leq \alpha(J, J')$ for all $J \neq J'$, this new chain is inferior to the Metropolis–Hastings one, in respect of the asymptotic variance of any ergodic average. However, in computational terms, it may still be advantageous. An accept/reject decision taken with probability $\widetilde{\alpha}(J, J')$ will involve computing the ratios $\widetilde{\pi}(J')/\widetilde{\pi}(J)$ and $q(J', J)/q(J, J')$ separately, and comparing with two independent uniform random numbers. The proposal is rejected if either test fails. Thus in situations where either of these probability ratios is costly to compute, there is scope for saving time by delaying computing the more expensive of the two ratios, only doing so when the pre-test using the first ratio is passed.

3.5 The chain is irreducible, hence ergodic

Recall that $\widetilde{\pi}(J) > 0$ for all J. Since the moves of our chain are reversible, it is sufficient to show that there is a path of junction trees, formed by successively adding edges one by one, from any J up to the trivial junction tree (with all vertices in a single clique) corresponding to the completely connected graph. But we can always add an edge to any junction tree other than this trivial one, simply by selecting a pair of disconnected vertices in adjacent cliques, and connecting them.

The state space of the chain is finite, so it follows from this irreducibility that the chain is ergodic, and so ergodic averages converge to expectations under the invariant distibution $\widetilde{\pi}$.

4 Numerical experiments and performance of the new sampler

We present two numerical illustrations of the new sampler in operation. First we show that for decomposable graphs on $n = 7$ vertices we can correctly sample either uniformly over junction trees or uniformly over decomposable graphs. For the second illustration we introduce a novel graphical Gaussian intra-class model from which we simulate data and then use our approach to sample from the posterior distribution of models given the simulated data.

The programs to carry out these computations were written in Java and are included in the *Java Programs for Statistical Genetics and Computational Statistics (JPSGCS)* package that can be obtained in from **http:/balance.med.utah.edu/wiki/index.php/JPSGCS** .

4.1 Decomposable graphs of size 7

Using a brute force approach we iterated through all 2,097,152 undirected graphs on 7 labelled vertices and identified the 617,675 decomposable ones. A list of the cliques of each decomposable graph was found and used as an index into a table of counters. The storage required for the indexed table on the 30,888,596 decomposable graphs on 8 vertices seemed excessive for the purpose of this illustration.

The number of possible junction tree representations for each graph was found using the algorithm given by Thomas and Green (2009b) and recorded. The decomposable graphs were sorted from those with most representations (16,807 for the trivial graph) to least (187,447 have a single junction tree).

We began with G set as the trivial graph and chose J uniformly at random from the 16,807 possible representations. For each simulated junction tree, the list of cliques comprising its nodes were used to find the appropriate counter in the indexed table, which was updated.

In the first case we sampled uniformly over junction trees, that is with $\widetilde{\pi}(J) \propto 1$, and, hence, $\pi(G(J)) \propto \mu(G(J))$. In the second case we set $\widetilde{\pi}(J) \propto \frac{1}{\mu(G(J))}$ which should give a uniform sample of decomposable graphs. Note that $\mu(G(J))$ is directly computable from J and does not require the construction of $G(J)$. In each case we sampled 1,000,000 graphs. The times taken for the runs were 70 and 76 seconds respectively, but note that the first 60 seconds in each case was used to make the indexed table, a step not typically required in a real application.

Figure 4 compares the expected and empirical distribution functions for both of these runs, and shows an excellent correspondence. Similar performance was observed for both standard Metropolis–Hastings and the variant described above.

4.2 A graphical Gaussian intra-class model

Given a decomposable graph G on v vertices labelled $1, 2, \ldots, v$, and real scalar parameters $\sigma^2 > 0$ and ρ, we define a non-negative definite matrix $V = V_G(\sigma^2, \rho)$ by

$$V_{ij} = \begin{cases} \sigma^2 & \text{if } i = j \\ \rho\sigma^2 & \text{if } (i, j) \text{ is an edge in } G, \end{cases}$$

and $(V^{-1})_{ij} = 0$ if (i, j) is not an edge in G.

By Grone *et al.* (1984), since G is decomposable and V restricted to each clique is positive definite, V exists and is unique, in fact the unique completion of the specified entries that is positive definite; it is the variance matrix of a v–variate Gaussian distribution for which G is the conditional independence graph. We call this the graphical Gaussian intra-class model (GGIM).

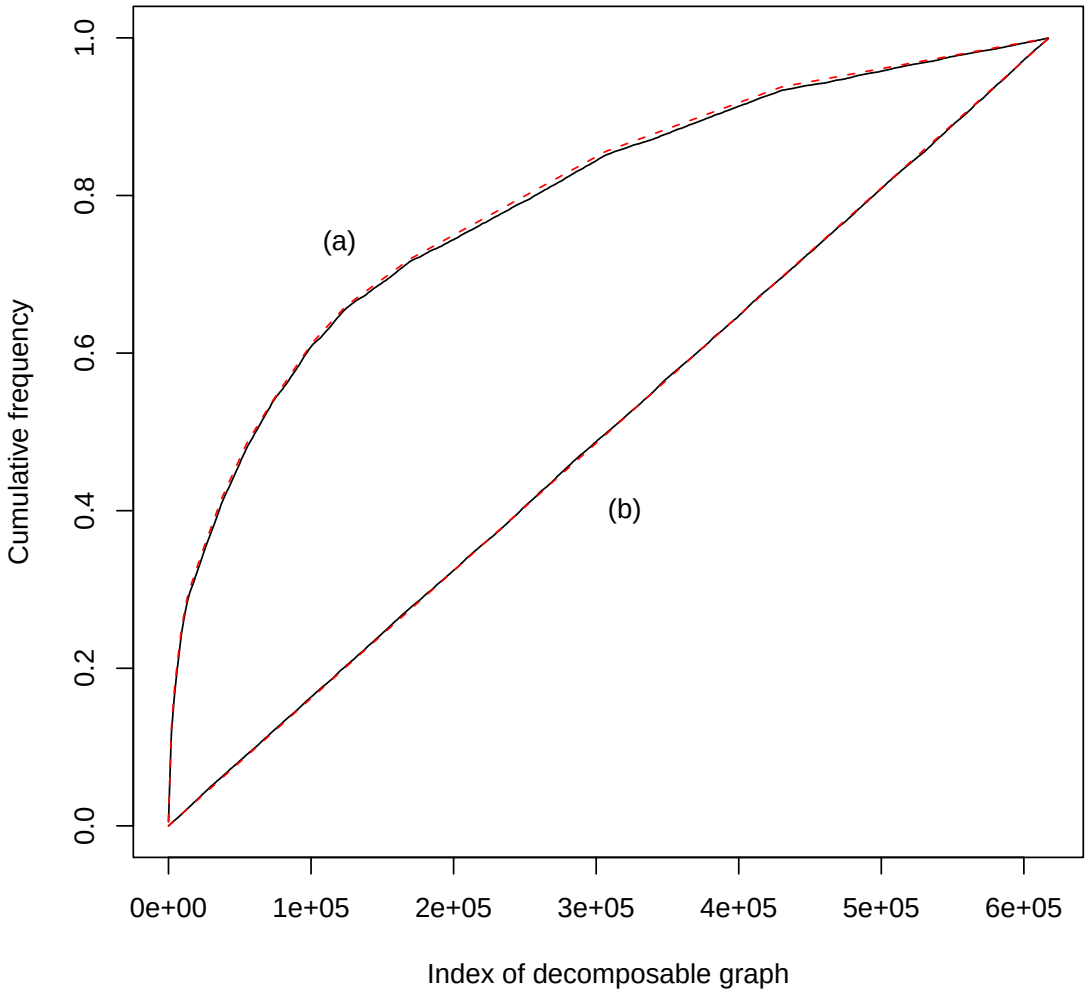

Figure 4: Cumulative distribution functions for decomposable graphs of size 7 sampled (a) with probability proportional to the number of junction tree representations and (b) uniformly. The solid lines give the observed frequencies and the expected distributions are shown by the dashed lines. The graphs are indexed from left to right in decreasing order by number of junction tree representations.

Suppose that $y \sim N(0, V_G(\sigma^2, \rho))$. Then if $\mathcal{C}$ and $\mathcal{S}$ denote the sets of cliques and separators of G, we have the well-known clique-separator factorisation

$$p(y|G, \sigma^2, \rho) = \frac{\prod_{C\in\mathcal{C}} p(y_C|G, \sigma^2, \rho)}{\prod_{S\in\mathcal{S}} p(y_S|G, \sigma^2, \rho)}. \tag{2}$$

Since each C is a complete subgraph of G, $\mathrm{var}(y_C)$ is explicitly specified in the assumptions above, it is the intra-class model $\sigma^2[(1-\rho)I_C+\rho J_C]$ where I_C and J_C are respectively the identity matrix and the matrix of all ones, with rows and columns both indexed by C. But the inverse and determinant of this variance matrix may be written down explicitly, and so we have

$$p(y_C|G, \sigma^2, \rho) = (2\pi)^{-v_C/2}\sigma^{-v_C}\left[(1-\rho)^{v_C-1}(1-\rho+v_C\rho)\right]^{-1/2}\times \\ \exp\left(\frac{-1}{2\sigma^2(1-\rho)}(y_C^T y_C - \frac{\rho}{1-\rho+v_C\rho}y_C^T J_C y_C)\right)$$

where v_C is the number of vertices in C. Replacing C by S throughout, the same holds for each $p(y_S|G, \sigma^2, \rho)$. Noting that $\sum_{C\in\mathcal{C}} v_C - \sum_{S\in\mathcal{S}} v_S = v$, we thus have the joint distribution explicitly, from (2):

$$p(y|G, \sigma^2, \rho) = (2\pi)^{-v/2}\sigma^{-v}(1-\rho)^{-v/2}\prod_{C\in\mathcal{C}}(1+v_C\rho/(1-\rho))^{-1/2}\prod_{S\in\mathcal{S}}(1+v_S\rho/(1-\rho))^{+1/2}\times \\ \exp\left(\frac{-1}{2\sigma^2(1-\rho)}\left\{\sum_{C\in\mathcal{C}}(y_C^T y_C - \frac{\rho}{1-\rho+v_C\rho}y_C^T J_C y_C) - \sum_{S\in\mathcal{S}}(y_S^T y_S - \frac{\rho}{1-\rho+v_S\rho}y_S^T J_S y_S)\right\}\right),$$

which can be simplified to

$$p(y|G, \sigma^2, \rho) = (2\pi)^{-v/2}\sigma^{-v}(1-\rho)^{-v/2}\prod_{C\in\mathcal{C}} f(C)^{-1/2}\prod_{S\in\mathcal{S}} f(S)^{+1/2}\times \\ \exp\left(\frac{-1}{2\sigma^2(1-\rho)}\left\{y^T y - \rho\sum_{C\in\mathcal{C}} H(C) + \rho\sum_{S\in\mathcal{S}} H(S)\right\}\right), \tag{3}$$

where $f(D) = (1 + v_D\rho/(1-\rho))$ and $H(D) = (\sum_{i\in D} y_i)^2/(1-\rho+v_D\rho)$ for any $D \subseteq \{1, 2, \ldots, v\}$.

The necessary and sufficient condition on ρ for this distribution to be well-defined for all decomposable graphs G on v vertices is that $-1/(v-1) < \rho < 1$.

Computational issues

Certain likelihood ratios, ratios of this joint density for two different G, can simplify greatly. For example for disjoint sets A, B and S, writing, e.g. AS for $A \cup S$,

$$\frac{p(y_{ABS}|G, \sigma^2, \rho)p(y_S|G, \sigma^2, \rho)}{p(y_{AS}|G, \sigma^2, \rho)p(y_{BS}|G, \sigma^2, \rho)} = \\ [\{f(ABS)f(S)\}/\{f(AS)f(BS)\}]^{-1/2}\times\exp\left(\frac{\rho}{2\sigma^2(1-\rho)}\left\{H(ABS) + H(S) - H(AS) - H(BS)\right\}\right).$$

This is the cross-ratio relevant to a single observation.

Given replicate observations $y^{(r)} \sim N(0, V_G(\sigma^2, \rho))$, independently for $r = 1, 2, \ldots, n$, we need the ratio

$$\prod_{r=1}^{n}\left(\frac{p(y_{ABS}^{(r)}|G, \sigma^2, \rho)p(y_S^{(r)}|G, \sigma^2, \rho)}{p(y_{AS}^{(r)}|G, \sigma^2, \rho)p(y_{BS}^{(r)}|G, \sigma^2, \rho)}\right) = \\ [\{f(ABS)f(S)\}/\{f(AS)f(BS)\}]^{-n/2}\times\exp\left(\frac{\rho}{2\sigma^2(1-\rho)}\left\{H(ABS) + H(S) - H(AS) - H(BS)\right\}\right),$$

where now $H(D) = \sum_{r=1}^{n} (\sum_{i \in D} y_i^{(r)})^2 / (1 - \rho + v_D \rho)$ for each D.

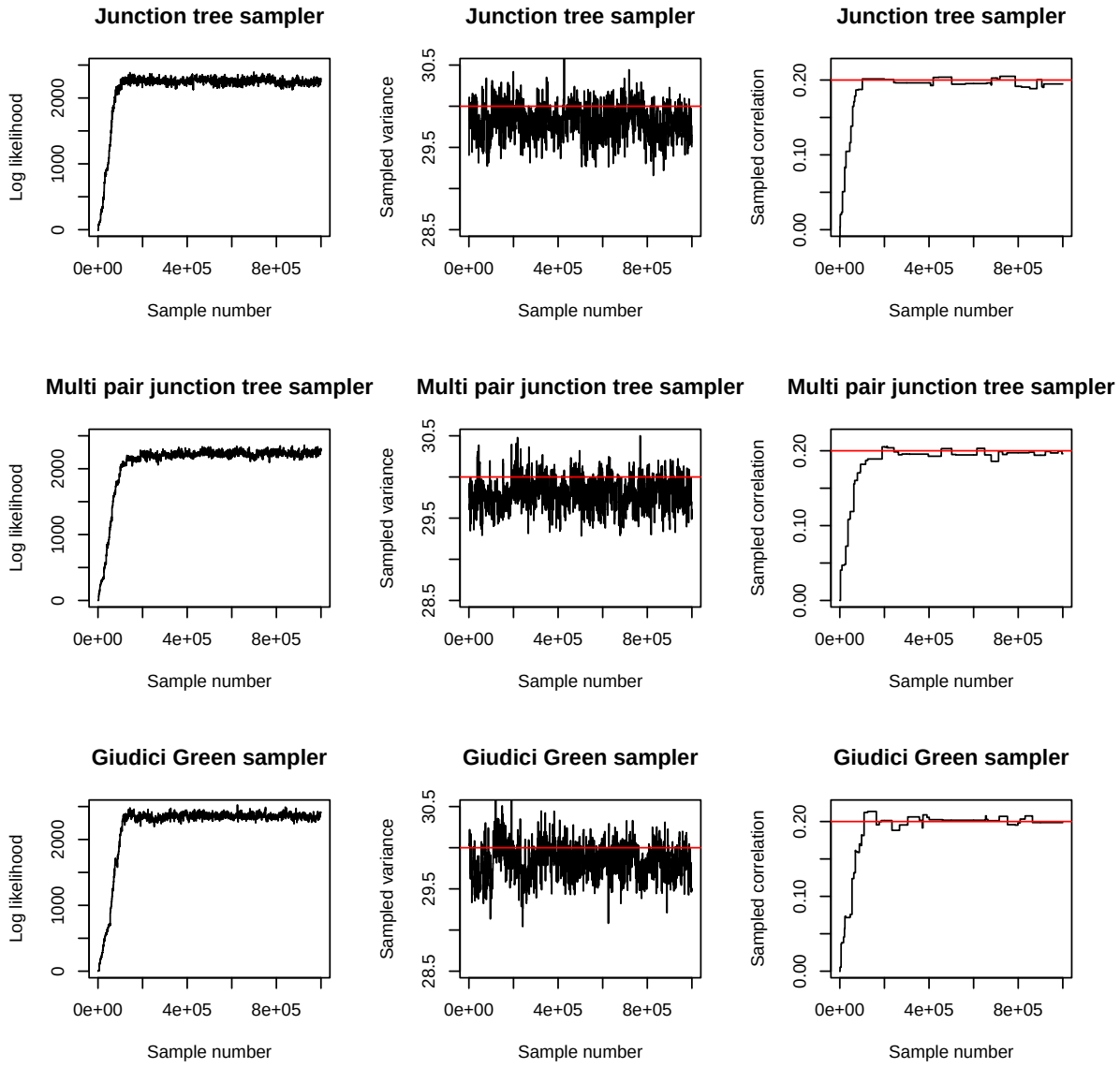

Figure 5: Log likelihoods and parameter estimates for three samplers for the GGIM model of Section 4.2, plotted by sample number. The values of the parameters used to generate the data are shown by the red horizontal lines.

Implementation and results

Using the method in Appendix 2, we simulated 1000 GGIM observations on 50 variables with $\sigma^2 = 30$ and $\rho = 0.2$. We used a second order Markov Chain graphical structure, that is, $(V^{-1})_{ij} = 0$ for all i and j such that $|i - j| > 2$. This data set is denoted by D below.

We then sampled from the joint posterior distribution of G, σ^2 and ρ given this data using three samplers: a junction tree sampler that proposes single edge connections or deletions, a junction

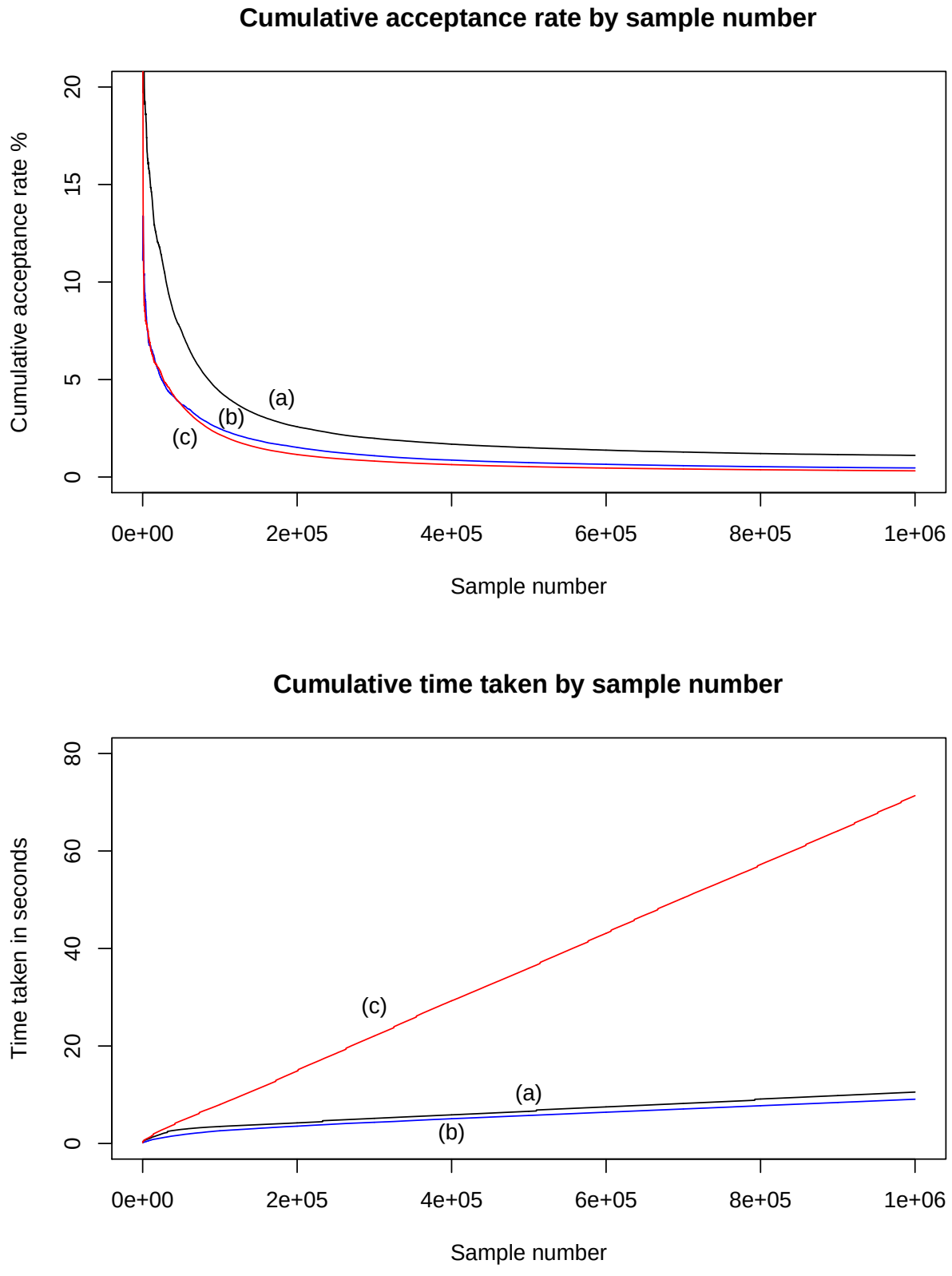

Figure 6: Cumulative acceptance rates and times taken by the three samplers for the GGIM model of Section 4.2. In each case the line closest to (a) is the single edge junction tree sampler, (b) is the multi edge junction tree sampler, and (c) is the Giudici–Green sampler.

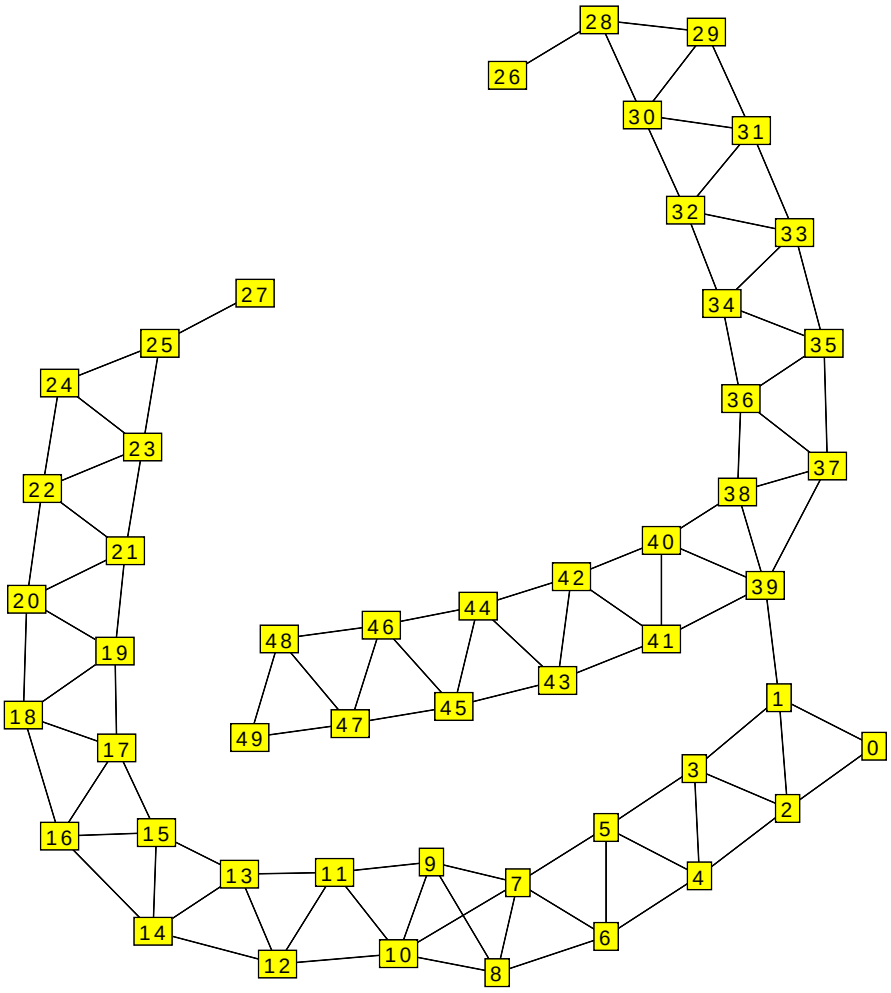

Figure 7: A graph typical of the type sampled early in their runs by all three samplers for the GGIM model of Section 4.2. The edge between variables 1 and 39 is spurious, and has to be removed before the correct edges near variables 25 and 26 can be added.

tree sampler that proposes multiple edge updates, and the Giudici–Green sampler. In each case started from the initial conditions of $\sigma^2 = 1$, $\rho = 0$ and G set to have no edges indicating complete independence between the 50 variables. We made 1,000,000 Metropolis–Hastings updates with each sampler and output values indicating the state of the chain after ever 100 iterations. The parameters σ^2 and ρ were updated as described above after each 1,000 Metropolis–Hastings steps. For the junction tree samplers we also randomized the junction tree after every 1,000 Metropolis–Hastings steps using the method given by Thomas and Green (2009b). Although the Giudici–Green sampler uses a junction tree to validate that proposals result in decomposable graphs, this test does not depend on the particular junction tree being used and so randomization was not necessary. The junction tree samplers sampled from

$$\widetilde{\pi}(J) = \frac{\pi(G|\sigma^2, \rho, D)}{\mu(J)} \tag{4}$$

so that $G(J)$ was sampled over the appropriate posterior distribution.

Note that the computations of the log likelihoods under the graphs G decompose into sums of contributions, or *scores*, from the subsets of vertices that are the cliques and separators of G. The score associated with a subset of vertices depends on σ^2, ρ and the appropriate sufficient statistics, but not of G. Hence, in our implementation, after computing the score of a subset, its sufficient statistics are cached and indexed by the elements of the subset. This avoids recomputation and in the long run makes the running time of our samplers independent of the number of observations in the sample.

Figure 5 shows plots of the log likelihood of sampled states, and the sampled values of σ^2 and ρ. As can be seen, the sampling properties are similar. The variance moves to the correct range almost immediately while the correlation takes longer and requires that the current graph estimate is close to correct before it takes appropriate values.

Figure 6 shows the cumulative acceptance rates and times taken by each sampler. The acceptance rates varied between different runs, but the general pattern shown here of the single edge junction tree sampler accepting more proposals than the multi edge junction tree sampler which in turn accepts more than the Giudici–Green sampler was consistent. The running times were very consistent between runs. The greater running time for the Giudici–Green sampler is due to the necessity of searching and updating the junction tree to find proposals that result in decomposable graphs. This outweighs the time required by the junction tree methods to compute $\mu(J)$ and to perform the junction tree randomization steps. The randomization step was found to be necessary with poor graph reconstructions when it was omitted. However, its omission did not greatly affect estimation of σ^2 and ρ (data not shown).

Figure 7 shows an inappropriate graph typical of the ones that all of the samplers spend time in in the initial stages. There is, in the data, a strong, but in fact, spurious correlation between variables 1 and 39 and the corresponding edge appears in the graph. Because only decomposable graphs are sampled, the presence of this edge prevents the correct edges elsewhere in the graph from being formed. This is because adding the correct edges would make a long loop of the type that is prohibited in decomposable graphs. For the Giudici–Green sampler, getting to the more probable states requires a sequence of steps that first removes the edge between 1 and 39 and then adds one between 25 and 26, or similar. The sequence of moves required by the junction tree samplers is more complex requiring the deletion of edge 1 to 39, then the randomization of the junction tree to give one that has the clique $\{25, 27\}$ adjacent to $\{26, 28\}$ (for instance, other adjacencies will also work), and then the connection of 25 to 26, or similar. Despite the extra requirement of an appropriate junction tree configuration, all the samplers eventually make the transition into the appropriate part of the graph space. Although not shown here, the most probable graph, as sampled by all three methods, was similar to the one used to generate the data, but was missing

the edge between 24 and 26, which we put down to simple sampling error.

Also seen in figure 7 is an edge between variables 7 and 10 that was not in the generating model. Small local changes such as this appear and vanish throughout the sampling run.

Acknowledgment

This work was supported by grant NIH R01GM081417 to Alun Thomas, and by the EPSRC-funded SuSTaIn programme at the University of Bristol.

Appendices

Appendix 1: Proofs of decomposability

Here we provide proofs that the modified graphs G' in Section 2.2 are decomposable.

These proofs are constructive but indirect; we actually demonstrate that the described multiple-edge connections and disconnections can be implemented by manipulating a junction tree representing the given decomposable graph; by showing the the result is a valid junction tree we will have shown that the modified graph is decomposable. The precise manipulations to the junction tree are specified algorithmically in Sections 3.1 and 3.2, and these should be considered in parallel with Propositions 1 and 2 respectively.

It is clear that both the multiple-edge connect and disconnect moves take the current junction tree J and yield a modified graph J' that is still a tree, whose nodes are sets of vertices of G. From consideration of the algorithm specification and stated requirements about various sets of vertices being non-empty, it is clear that these nodes of J' are cliques in G'. To prove that the corresponding modified graph G' remains decomposable it is therefore sufficient to show that J' still has the junction property, and for this it is sufficient to show that for every vertex $v \in V$, the cliques containing v form a connected sub-tree of J', given that this is true of J.

Proof of Proposition 1. We consider the 4 cases (a), (b), (c), (d) in turn, in each case considering the possibilities that v is in X, Y, S or $V \setminus (X \cup Y \cup S)$. In case (a), the cliques in J containing v for $v \in X$ are XS and possibly others forming a sub-tree including XS; in J', XS is replaced by XYS, with the same adjacencies, and this new clique still contains such v. For $v \in Y$, the argument is identical; for $v \in S$, the adjacent cliques XS and YS containing v are merged into XYS whose adjacencies combine those of XS and YS, so adjacencies among all ciques containing v are preserved. For $v \in V \setminus (X \cup Y \cup S)$, there is no change to the cliques containing v or their adjacencies. In case (b) the only change to J is that vertices in X are added into the clique YS, which is adjacent to XS in J so the connected sub-tree property is maintained. Case (c) is similar. Finally, in case (d), the change in J' is that an additional clique XYS is inserted between XS and YS: since this is the union of these two cliques, this change cannot affect the connectedness of the sub-trees containing any vertex. □

Proof of Proposition 2. The arguments about validity of the multiple-edge disconnections proceed along similar lines. Vertices outside $X \cup Y \cup S$ are not affected by the changes to J. In case (a), the requirement to connect cliques in $\mathcal{N}_X$ to XS and those in $\mathcal{N}_Y$ to YS, described in Section 3.2(a), ensures connectedness of the sub-trees containing vertices in $X \cup Y$, while those vertices in S are included in all of the new parts of the junction tree. In case (b) and (c) we are removing vertices (in X and Y respectively) from the clique XYS; but by assumption XS (respectively YS) is the only adjacent clique intersecting X (respectively Y), so all adjacencies are maintained. Finally in case (d), we remove the clique XYS and make XS and YS adjacent. This cannot break any adjacencies. □

Appendix 2: Sampling data from the graphical Gaussian intra-class model

Suppose that $y \sim N(0, V_G(\sigma^2, \rho))$. Then if $\mathcal{C}$ and $\mathcal{S}$ denote the sets of cliques and separators of G, we have the well-known clique-separator factorisation (2).

We can easily exploit this to sample from the distribution. It follows that for any clique C and separator S such that $S \subset C$,

$$p(y_{C\setminus S}|y_S, G, \sigma^2, \rho) = \frac{p(y_C|G, \sigma^2, \rho)}{p(y_S|G, \sigma^2, \rho)}$$

and after some algebra we find this can be written

$$y_{C\setminus S}|y_S, G, \sigma^2, \rho \sim N\left(\frac{\rho}{1-\rho+v_S\rho}(\sum_{i\in S} y_i)1_{C\setminus S}, (1-\rho)\sigma^2(I_{C\setminus S} + \frac{\rho}{1-\rho+v_S\rho}J_{C\setminus S})\right),$$

where $1_{C\setminus S}$ is a vector of 1's appropriately indexed.

This can be used to simulate a draw from $N(0, V_G(\sigma^2, \rho))$ by scanning through a junction tree, according to a perfect numbering.

```
ricnorm<-function(n,p,mu,a,b)
{
# simulate a sample of size n from the p-variate normal with mean mu
# and variance matrix aI+bJ
z<-matrix(rnorm(n*p),n,p)
mu+sqrt(a)*z+(sqrt(a+p*b)-sqrt(a))*apply(z,1,mean)
}

rggim<-function (n,jt,sigma2,rho)
{
# simulate a sample of size n from the GGIM model on the decomposable
# graph represented by the assumed-perfectly-numbered junction tree jt
# and stated parameters \sigma^2 and \rho
vs<-unique(sort(unlist(jt$cliq))); v<-length(vs)
if(any(vs!=(1:v))) stop('invalid vertices')
y<-matrix(0,n,v)
c<-jt$cliq[[1]]
y[,c]<-ricnorm(n,length(c),0,sigma2*(1-rho),sigma2*rho)
for(j in 2:length(jt$cliq))
{
c<-jt$cliq[[j]]; s<-jt$sep[[j]]; cprev<-jt$cliq[[jt$prev[[j]]]]
if(!(all(s%in%c)&all(s%in%cprev))) stop('invalid jt')
cms<-c[!(c%in%s)]
z<-apply(y[,s,drop=FALSE],1,sum)
y[,cms]<-ricnorm(n,length(cms),(rho/(1-rho+length(s)*rho))*z,sigma2*(1-rho),
sigma2*rho*(1-rho)/(1-rho+length(s)*rho))
}
y
}

> str(jt)
List of 3
```

```
 $ cliq:List of 3
  ..$ : int [1:3] 1 2 3
  ..$ : int [1:2] 3 4
  ..$ : num [1:2] 2 5
 $ sep :List of 3
  ..$ : NULL
  ..$ : num 3
  ..$ : num 2
 $ prev:List of 3
  ..$ : NULL
  ..$ : num 1
  ..$ : num 1
> var(rggim(100000,jt,30,.2))
          [,1]      [,2]      [,3]       [,4]       [,5]
[1,] 29.931173  5.994112  5.991068  1.2074432  1.2351501
[2,]  5.994112 30.051968  6.213571  1.2461824  6.0986495
[3,]  5.991068  6.213571 30.032293  6.0735334  1.3400007
[4,]  1.207443  1.246182  6.073533 30.1098215  0.1746564
[5,]  1.235150  6.098650  1.340001  0.1746564 30.1919158
```

Appendix 3: MCMC updating of σ^2 and ρ

It is clear from (3) that the inverse Gamma distribution is conditionally conjugate for σ^2 in this model, thus if *a priori* $\sigma^{-2} \sim \text{Gamma}(\alpha, \beta)$ then the posterior full conditional for σ^2 is

$$\sigma^{-2}|\rho, G, y \sim \text{Gamma}(\alpha + nv/2, \beta + Q/(2(1-\rho))),$$

where $Q = \sum_{r=1}^{n} (y^{(r)})^T y^{(r)} - \rho \sum_{C \in \mathcal{C}} H(C) + \rho \sum_{S \in \mathcal{S}} H(S)$. Thus there is a straightforward Gibbs sampler update for σ^2.

On the other hand, for ρ we must use a Metropolis–Hastings update as the full conditional is non-standard for any prior. In view of the constraint on ρ, we suggest a symmetric additive (random-walk Metropolis) proposal on the logistic-like transform $g(\rho) = \log((\rho+1/(v-1))/(1-\rho))$. Thus we set $\rho^\star = g^{-1}(g(\rho) + z) = 1 - (v/(v-1))/(\exp(g(\rho) + z) + 1) = 1 - (v/(v-1))/(e^z((\rho + 1/(v-1))/(1-\rho)) + 1)$, where the innovation z has any distribution symmetric about 0.

The acceptance probability for detailed balance with respect to the posterior distribution will be

$$\alpha = \min\left\{1, \frac{p(\rho^\star)p(y|G, \sigma^2, \rho^\star)g'(\rho)}{p(\rho)p(y|G, \sigma^2, \rho)g'(\rho^\star)}\right\}.$$

Note that since $g'(\rho) = (v/(v-1))/\{(\rho + 1/(v-1))(1-\rho)\}$, this becomes

$$\alpha = \min\left\{1, \frac{p(\rho^\star)p(y|G, \sigma^2, \rho^\star)(\rho^\star + 1/(v-1))(1-\rho^\star)}{p(\rho)p(y|G, \sigma^2, \rho)(\rho + 1/(v-1))(1-\rho)}\right\}.$$

The distribution $p(y|G, \sigma^2, \rho)$ is given in (3), and while there is some cancellation, this is still quite a cumbersome calculation.

References

Abel, H. J. and Thomas, A. (2011). Accuracy and computational efficiency of a graphical modeling approach to linkage disequilibrium estimation. *Statistical Applications in Genetics and Molecular Biology*, **10**. Article 5.

Besag, J., Green, P. J., Higdon, D., and Mengersen, K. (1995). Bayesian computation and stochastic systems (with discussion). *Statistical Science*, **10**, 3–66.

Cayley, A. (1889). A theorem on trees. *Quarterly Journal of Mathematics*, **23**, 376–8.

Frydenberg, M. and Lauritzen, S. L. (1989). Decomposition of maximum likelihood in mixed interaction models. *Biometrika*, **76**, 539–55.

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

Grone, R., Johnson, C. R., Sá, E. M., and Wolkowicz, H. (1984). Positive definite completions of partial Hermitian matrices. *Linear Algebra and its Applications*, **58**, 109–24.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, (1), 97–109.

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

Peskun, P. H. (1973). Optimum Monte-Carlo sampling using Markov chains. *Biometrika*, **60**, (3), 607–12.

Tarjan, R. E. and Yannakakis, M. (1984). Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal of Computing*, **13**, 566–79.

Thomas, A. and Green, P. J. (2009a). Enumerating the decomposable neighbours of a decomposable graph under a simple perturbation scheme. *Computational Statistics and Data Analysis*, **53**, 1232–8.

Thomas, A. and Green, P. J. (2009b). Enumerating the junction trees of a decomposable graph. *Journal of Computational and Graphical Statistics*, **18**, 930–40.