Enumerating the junction trees of a decomposable graph

Alun Thomas[*]
Department of Biomedical Informatics
University of Utah

Peter J Green[†]
Department of Mathematics
University of Bristol

October 4, 2007

Keywords: Chordal graphs, triangulated graphs, graphical models, Markov chain Monte Carlo methods.

Running title: Enumerating junction trees

Abstract

We derive methods for enumerating the distinct junction tree representations for any given decomposable graph. We discuss the relevance of the method to estimating conditional independence graphs of graphical models and give an algorithm that, given a junction tree, will generate uniformly at random a tree from the set of those that represent the same graph.

[*]Genetic Epidemiology, 391 Chipeta Way Suite D, Salt Lake City, UT 84108, USA. alun@genepi.med.utah.edu, +1 801 587 9303 (voice), +1 801 581 6052 (fax).

[†]Department of Mathematics, University of Bristol, Bristol BS8 1TW, UK, P.J.Green@bristol.ac.uk.

1 Introduction

Decomposable or *triangulated* or *chordal* graphs are of interest in many areas of mathematics. Our primary interest is in their role as the conditional independence graphs of decomposable graphical models and in particular, in schemes that traverse the space of decomposable graphs in order to sample from or maximize a given posterior probability distribution defined from observed multivariate data [2, 10, 9]. It is often convenient with such methods to operate not on the graph itself, but on its derived representation as a *junction tree*, which raises the prospect of discarding the underlying graph and using the junction tree exclusively. However, decomposable graphs have multiple equivalent junction tree representations and moreover the number is variable from graph to graph. Therefore, sampling the space of junction trees will over represent decomposable graphs with a large number of such representations. This can be corrected for if the number of junction trees for any particular decomposable graph can be evaluated and this is the primary motivation for the method we present here.

We begin by reviewing some definitions and standard properties of decomposable graphs and junction trees. For more complete information on these see [3] and [5], whose terminology we have adopted. We then consider the number of ways that sets of edges of a junction tree that correspond to the same clique intersection can be rearranged. These counts are then combined to give the total number of junction trees. Finally, we outline an algorithm that will take a junction tree and select an equivalent one uniformly at random from the set of all possible equivalents.

2 Definitions and preliminary results

Consider a graph $G = G(V, E)$ with vertices V and 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.

Definition 1 *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 $i = 1, 2, \ldots, c-1$*

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

This is called the *running intersection property* and is equivalent to the requirement that every cycle in G of length 4 or more 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.

Definition 2 *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, and has a weight, possibly zero, equal to the cardinality of the intersection.*

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*.

Definition 3 *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*.*

Figure 1 gives an example of a decomposable graph while Figure 2 shows one of its possible junction trees. The maximum cardinality search method of [8] will find a junction tree for a given decomposable graph in time and storage of order $|V| + |E|$.

Note that 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.

As [5] describes more fully, a junction tree for G will exist if and only if G is decomposable, and 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. Also, the junction property can be equivalently stated as the property 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.

The following result, due to [4], is a useful characterization of the junction tree. We define the *weight* of a subgraph of the junction graph to be the sum of the weights of its links.

Theorem 4 *A spanning tree of the junction graph is a junction tree if and only if it has* *maximal weight.*

From this it is clear that any tree with the cliques of G as its nodes and for which the collection of clique intersections associated with the links is equal to the collection of separators of G is a junction tree of G, since such a tree must span the junction graph and have maximal weight. Therefore, the problem of enumerating junction trees for a given graph G is equivalent to enumerating the ways that links of a given junction tree can be rearranged so that the result is also a tree, and the collection of clique intersections defined by the links of the tree is unchanged.

3 Rearranging the links for the set of separators with the same intersection

As noted above, the separators of G are not generally distinct. For example, in Figure 2 three links are associated with the clique intersection $\{17\}$ and two with the intersection $\{2, 3\}$. We now consider the effect of rearranging all the links that are associated with the same clique intersection. Let J be any junction tree of G and S one of its separators. Define T_S to be the subtree of J induced by the cliques that contain S. The junction property ensures that T_S is a single connected subtree of J.

Clearly, any rearrangement of the links associated with S in J must be a rearrangement among certain links of T_S since both cliques joined by such a links must contain S. For illustration, Figure 3 shows $T_{\{3\}}$, the subtree defined by the separator $\{3\}$ for the graph in Figure 1. If we now rearrange the links that are associated with S to produce a new graph, T'_S say, and replace T_S in J by T'_S to give a new graph J', J' will be a junction tree of G if and only if

- T'_S is a tree, and hence so is J', and
- T'_S has the same weight as T_S, so that J' has the same weight as J.

In fact the second condition is redundant: all cliques in T_S contain S so their intersection must also do so, and any pair of cliques whose intersection is a superset of S cannot be joined in a tree T'_S unless already joined in T_S as T'_S would then have greater weight than T_S, and J' greater weight than J thus violating the latter's maximal weight property. So we need only count the number of ways of rearranging the links of T_S associated with S such that T'_S is a tree.

Consider F_S to be the forest obtained by deleting all the links associated with S from T_S. For example, Figure 4 shows $F_{\{3\}}$, the forest obtained by deleting links associated with the separator $\{3\}$ from the tree $T_{\{3\}}$ shown in Figure 3. Define $\nu(S)$ to be the number of ways that the components of F_S can be connected into a single tree by adding the appropriate number of links. This value is given by a theorem by [6] which can be restated as follows.

Theorem 5 *The number of distinct ways that a forest of p nodes comprising q subtrees of sizes $r_1 \dots r_q$ can be connected into a single tree by adding $q-1$ links is*

$$p^{q-2} \prod_{i=1}^{q} r_i. \tag{2}$$

If the number of links associated with a given separator S is m_S we know that F_S will contain $m_S + 1$ components. Let these be of sizes $f_1, f_2, \dots f_{m_S+1}$. Let the number of nodes in T_S be t_S which is simply the number of cliques of G that contain S. Then, directly from theorem 5 we obtain the following.

Theorem 6

$$\nu(S) = t_S^{m_S - 1} \prod_{j=1}^{m_S+1} f_j. \tag{3}$$

For example, the forest in Figure 4 has 7 nodes in 4 components of sizes $1, 1, 1$ and 4. This forest, $F_{\{3\}}$, can be reconnected into a single tree by adding 3 links in $7^2 \times 1 \times 1 \times 1 \times 4 = 196$ different ways.

4 The number of junction trees for a decomposable graph

The final step in enumerating junction trees is to note that $\nu(S)$ depends only on the sizes of the components of F_S, not on their particular structure. These sizes are determined by the sets of cliques that contain separators that are supersets of S. Since the set of cliques and collection of separators are uniquely determined and independent of any particular junction tree, $\nu(S)$ is independent of J. Hence, the links associated with one particular separator can be reallocated independently of the links associated with another. Thus we obtain the following result.

Theorem 7 *Consider a decomposable graph G with separators $S_1, \ldots S_{c-1}$. Let $S_{[1]}, \ldots S_{[s]}$ be the distinct sets contained in the collection of separators. The number of junction trees of G is*

$$\mu(G) = \prod_{i=1}^{s} \nu(S_{[i]}). \tag{4}$$

As an example, the number of distinct junction trees for the graph shown in Figure 1 is 57,802,752. The calculations needed to obtain this are given in table 1.

As noted above, we can retrieve from this result the count of the number of possible conventional junction forests that a decomposable graph has. This is given simply by

$$\frac{\mu(G)}{\nu(\emptyset)},$$

which for the example is $57802752/6144 = 9408$.

5 Randomizing the junction tree

Theorem 5 is similar in style to Prüfer's constructive proof [7] of Cayley's result that there are n^{n-2} distinct labelled trees of n vertices [1]. A similar construction lets us choose uniformly at random from the set of possible trees containing a given forest as follows:

1. Label each vertex of the forest $\{i, j\}$ where $1 \leq i \leq q$ and $1 \leq j \leq r_i$, so that the first index indicates the subtree the vertex belongs to and the second reflects some ordering within the subtree. The orderings of the subtrees and of vertices within subtrees are arbitrary.

2. Construct a list v containing $q - 2$ vertices each chosen at random with replacement from the set of all p vertices.

3. Construct a set w containing q vertices, one chosen at random from each subtree.

4. Find in w the vertex x with the largest first index that does not appear as a first index of any vertex in v. Since the length of v is 2 less than the size of w there must always be at least 2 such vertices.

5. Connect x to y, the vertex at the head of the list v.

6. Remove x from the set w, and delete y from the head of the list v.

7. Repeat until v is empty. At this point w contains 2 vertices. Connect them.

Given any particular junction tree representation J for a decomposable graph G we can choose uniformly at random from the set of equivalent junction trees by applying the above algorithm to each of the forests F_S defined by the distinct separators of J. Such a randomization could be incorporated into a sampling scheme on junction trees to improve mixing properties, or may even be needed to ensure that the sampling scheme determines a Markov chain that is irreducible.

6 Acknowledgments

This work was begun while the authors were participants at the Isaac Newton Institute for Mathematical Science's program on Stochastic Computation in the Biological Sciences in 2006 and we would like to thank the programme organizers, the Institute and its staff for their work and support.

This work was supported by NIH grants R21 GM070710 and R01 GM81417 to Alun Thomas.

References

[1] A Cayley. A theorem on trees. *Quarterly Journal of Mathematics*, 23:376–378, 1889.

[2] P Giudici and P J Green. Decomposable graphical Gaussian model determination. *Biometrika*, 86:785–801, 1999.

[3] M C Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, 1980.

[4] F V Jensen. Junction trees and deomposable hypergraphs. Technical report, Judex Datasystemer, Aalborg, Denmark, 1988.

[5] S L Lauritzen. *Graphical Models*. Clarendon Press, 1996.

[6] J W Moon. Enumerating labelled trees. In F Harary, editor, *Graph Theory and Theoretical Physics*. Academic Press, London, 1970.

[7] H Prüfer. Neuer beweis eines satzes uber permutationen. *Archiv fur Mathematik und Physik*, 27:142–144, 1918.

[8] R E Tarjan and M Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal of Computing*, 13:566–579, 1984.

[9] A Thomas. Characterizing allelic associations from unphased diploid data by graphical modeling. *Genetic Epidemiology*, 29:23–35, 2005.

[10] A Thomas and N J Camp. Graphical modeling of the joint distribution of alleles at associated loci. *American Journal of Human Genetics*, 74:1088–1101, 2004.

Figure 1: A decomposable graph containing 23 vertices in 4 disjoint components.

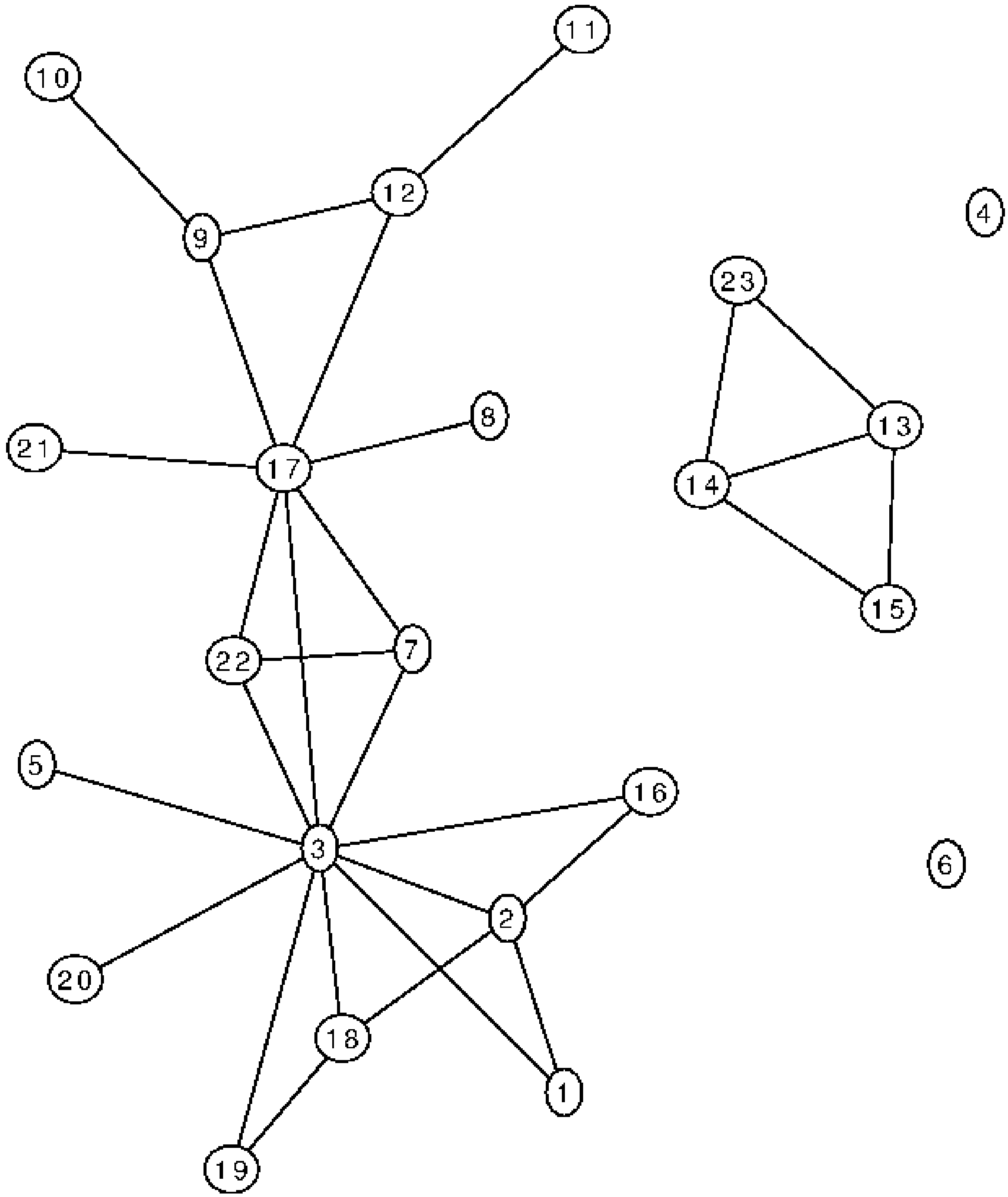

Figure 2: One possible junction tree for the graph shown in Figure 1. The 16 cliques are the vertices of the junction tree and are shown as ovals. The 15 clique separators are represented by the edges of the graph and each edge is associated with the intersection of its incident vertices. These intersections are shown as rectangles. Note that some of these intersections are empty.

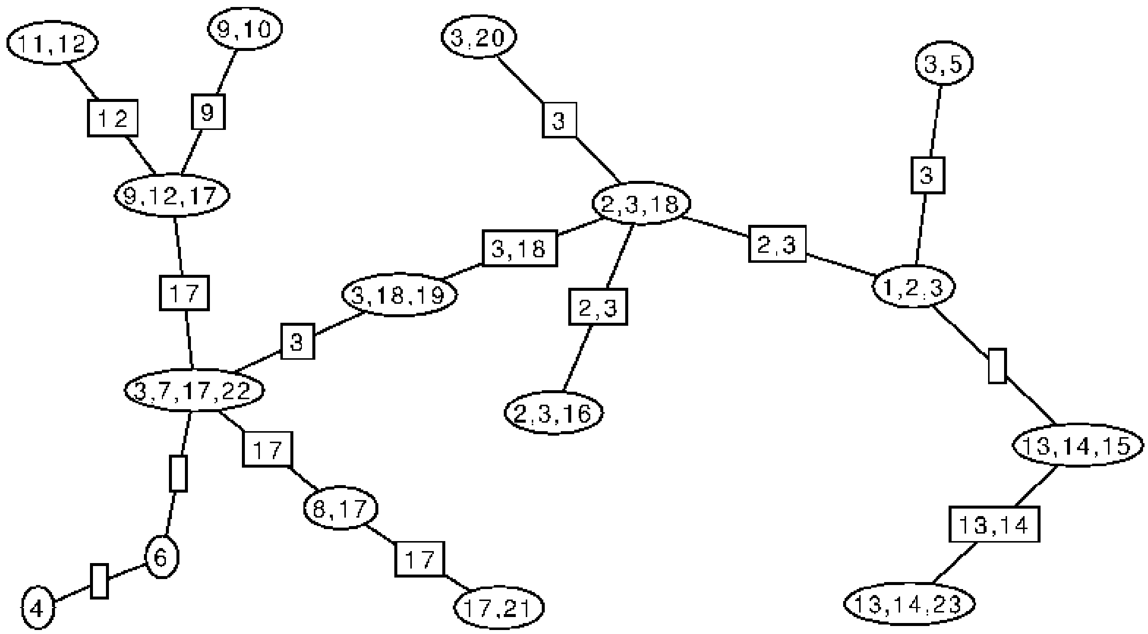

Table 1: The computations that enumerate the distinct junction trees for the decomposable graph given in Figure 1.

Separator S	m_S	t_S	$\{f_S\}$	$\nu(S)$
$\emptyset$	3	16	1, 1, 2, 12	6144
$\{13, 14\}$	1	2	1, 1	1
$\{3\}$	3	7	1, 1, 1, 4	196
$\{2, 3\}$	2	3	1, 1, 1	3
$\{3, 18\}$	1	2	1, 1	1
$\{9\}$	1	2	1, 1	1
$\{12\}$	1	2	1, 1	1
$\{17\}$	3	4	1, 1, 1, 1	16

$$\mu(G) = 6144 \times 1 \times 196 \times 3 \times 1 \times 1 \times 1 \times 16 = 57802752$$

Figure 3: $T_{\{3\}}$, the connected subtree of the junction graph shown in Figure 2 induced by the cliques that contain the separator $\{3\}$.

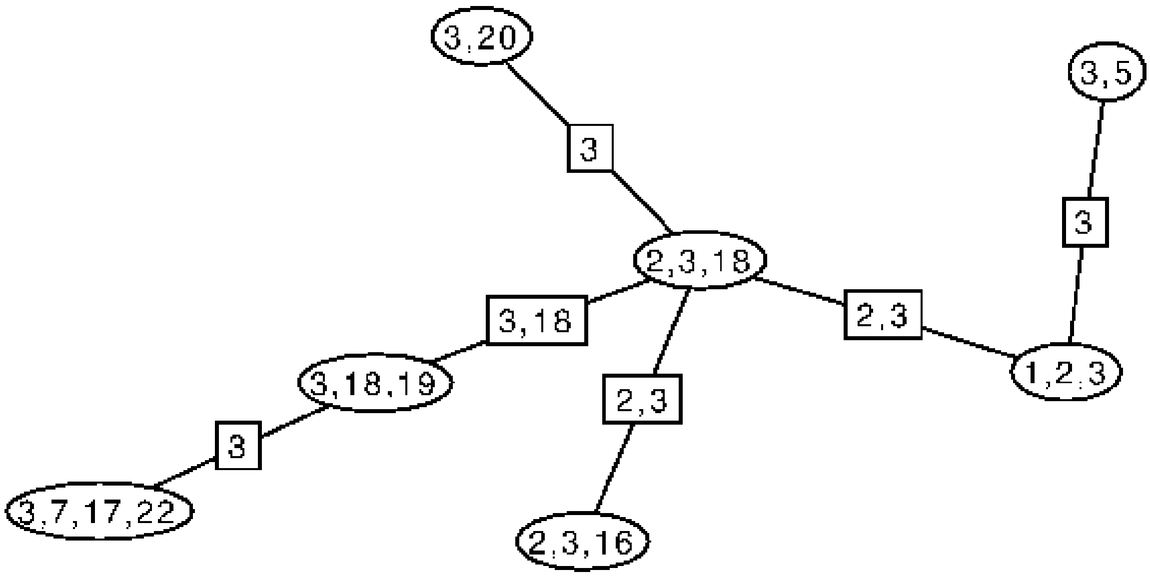

Figure 4: $F_{\{3\}}$, the forest obtained by from the tree shown in Figure 3 by deleting edges associated with the separator $\{3\}$.