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

September 18, 2008

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. In particular, we are interested in estimating decomposable graphical models from observed data using Markov chain Monte Carlo, or *MCMC*, schemes that traverse the space of decomposable graphs in order to sample from, or maximize, the posterior probability distribution defined by the data. The underlying approach is the same whether the data is continuous (Giudici & Green 1999, Jones et al. 2005) or discrete (Thomas & Camp 2004, Thomas 2005). A common feature of such schemes is that, given an incumbent decomposable graph G, we propose a new graph G' which is then accepted or rejected according to probabilities that depend on the distribution being sampled (Metropolis et al. 1953, Hastings 1970, Kirkpatrick et al. 1982). However, there are no known proposal schemes that guarantee in advance that G' will be decomposable, even if G is. Hence, it is necessary to test G' for decomposability before evaluating the usual acceptance probability. While such tests can be very quick (Giudici & Green 1999), for all practical methods for proposing a random G' of which we are aware, the probability that G' is decomposable decreases rapidly with the size of the graph, making this approach infeasible for large problems. For instance, in the genetic examples considered by Thomas (2008), involving up to 20,000 variables, the probability of proposing a decomposable G' decreases roughly as the inverse of the number of variables. Given these circumstances it would be very useful to have an alternative representation of the problem that avoids the test for decomposability. It is with this in mind that we consider what follows.

It is often convenient in graphical modelling to operate not on the graph itself, but on its derived representation as a *junction tree*. This raises the prospect of discarding the underlying conditional independence graph entirely and defining MCMC schemes on the space of junction trees. As each junction tree uniquely defines a decomposable graph, this might avoid the expensive need to propose non-decomposable models. 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-sample 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 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 Golumbic (1980) and Lauritzen (1996), whose terminology we have adopted. We then consider the number of ways that sets of links 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. A simple algorithm is then presented that will take a junction tree and select an equivalent one uniformly at random from the set of all possible equivalents. Finally, we discuss the computational complexity of our method showing that it is faster than existing algorithms, and outline potential junction tree sampling methods.

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

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

This is called the *running intersection property* 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 lexicographic search method of Tarjan & Yannakakis (1984) 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 Lauritzen (1996) 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.

As stated in Definition 2, we can consider each link of the junction graph to have a weight. Thus, any subgraph of it, and in particular any spanning tree, can also be associated with a weight defined by the sum of the weights of the links included. Jensen (1988) exploits this to give the following useful characterization of a junction tree.

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 link 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 Moon (1970) 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 (Prüfer 1918) of Cayley's result that there are n^{n-2} distinct labelled trees of n vertices (Cayley 1889). 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.

6 Computational complexity

Jensen's characterization of a junction tree as a maximal spanning tree of the junction graph means that general methods for enumerating the optimal spanning trees of a graph can also be applied to enumerating junction trees. The method of Broder & Mayr (1997) does precisely this. Recalling the notation used in section 2, the junction graph will have c nodes and $c(c-1)/2$ links. Broder and Mayr's method would require $O(M(c))$ elementary operations to enumerate its maximal spanning trees, where $M(c)$ is the number of operations needed to multiply $c \times c$ matrices. Asymptotically, the best algorithm for matrix multiplication is that of Coppersmith & Winograd (1990) which requires $O(c^{2.376})$ operations, although for realistically sized matrices the best practical methods, based on that of Strassen (1969), need $O(c^{2.807})$ operations. Letting $n = |V|$, the number of vertices in G, we note that c can be as large as n and typically grows linearly with n. Hence, Broder and Mayr's algorithm is at best an $O(n^{2.376})$ method.

However, as noted above, Jensen's characterization is not the only route to obtaining a junction tree. The lexicographic search of Tarjan & Yannakakis (1984) will find a simple elimination scheme, and hence a junction tree, in time $O(n + m)$, where $m = |E|$ the number of edges in G. While $m = O(n^2)$ in the worst case, typical graphical models are sparse and the time required is closer to linear in n. The enumeration method presented here depends only on knowing a single junction tree for G. The time required to carry it out is dominated by the time needed to find each $T_{S_{[i]}}$. We note that any link L of J will be a link in $T_{S_{[i]}}$ for each i such that $S_{[i]} \subseteq L$. Finding all the $T_{S_{[i]}}$ can be done by iterating over the $c-1$ links of J, and for each link checking for inclusion of each of the s distinct separators. Since both c and s can be $O(n)$, the enumeration is an $O(n^2)$ algorithm in the worst case. Other ways of finding the $T_{S_{[i]}}$ will in practice be faster. For example, we can find $T_{S_{[i]}}$ by starting with a node that contains $S_{[i]}$ and searching outwards in J until we hit nodes that don't contain the separator. Thus, if $T_{S_{[i]}}$ is small it will be found very quickly. While it is straightforward to construct examples where this approach is also $O(n^2)$, for more typical graphs it will be considerably faster.

In summary, applying Broder and Mayr's general method to the special case of enumerating junction trees is at best an $O(n^{2.376})$ method, and more realistically $O(n^{2.807})$. By exploiting the junction property, our method improves this to a worst case of $O(n^2)$ which in practice is a very conservative upper bound.

7 Possible MCMC samplers for junction trees

Given a distribution $\pi(G)$ from which we want to sample decomposable graphs G, the methods of Metropolis et al. (1953) and Hastings (1970) allow us to construct Markov chains with $\pi(G)$ as the ergodic distribution. For example, we can propose a new graph G' by choosing two random vertices of G: if they are connected in G we disconnect them in G' and vice versa. G' is then accepted with probability $\max(1, \pi(G')/\pi(G))$, with the special case that $\pi(G')$ is defined to be 0 if G' is not a decomposable graph. Intuitively, it is easy

to see that this can be very inefficient. If we consider choosing two random vertices of G, it is quite likely that we pick two vertices that are not connected directly, but which are connected by several paths through other vertices. Adding an edge between the vertices is therefore likely to create cycles. Unless all the connecting paths are short, a cycle of length 4 or more may well be formed which prevents G' from being decomposable. Thomas (2008) shows that for modelling linkage disequilibrium between genetic loci, the acceptance rate decreases approximately as $1/n$, making the method infeasible for large numbers of variables.

The motivation for our enumeration method is that it may be possible to devise MCMC schemes that work directly on the space of junction trees with perturbations that guarantee that if J is a junction tree for $G(J)$, the proposal J' will be junction tree for some $G' = G(J')$, which ensures that G' is decomposable. Such proposals might involve merging or splitting the nodes of J. Given such proposals, it would be straightforward to construct a Markov chain with ergodic distribution

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

For each J from such a chain, we would derive a graph $G(J)$ which would be sampled with probability

$$\begin{aligned} P(G) &= \sum_{J:G(J)=G} \frac{\pi(G(J))}{\mu(G(J))} \\ &= \frac{\pi(G)}{\mu(G)} \sum_{J:G(J)=G} 1 \\ &= \pi(G) \end{aligned}$$

as required. Since this would avoid proposing non-decomposable graphs the acceptance rate should be greatly improved. This would, of course, require efficient enumeration, but note that the acceptance probability depends only on $\mu(G(J'))/\mu(G(J))$ which, given the factorization in equation 4 above, might be computable from $\mu(G(J))$ more simply than direct enumeration of $\mu(G(J'))$.

For validity, any proposal scheme on junction trees has to define an irreducible Markov chain in that space. With this in mind, moving freely between the set of junction trees for any particular decomposable graph, as is made possible by the method described in section 5 above, could be crucial. Even if irreducibility is possible without it, such a randomization would improve the mixing properties of the chain.

While some of this is speculative, and we have not yet defined a suitable perturbation scheme on junction trees, we feel that this can open up a new area of opportunities for estimating graphical models my MCMC methods. Although the space of junction trees is larger than that of decomposable graphs, it may well be more tractable and more easily traversible.

8 Acknowledgments

This work was begun while the authors were participants at the Isaac Newton Institute for Mathematical Science's programme 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.

We would also like to thank three anonymous reviewers and the journal editors for their prompt and constructive comments on how to better present and motivate this work.

This work was supported by grants NIH R21 GM070710, NIH R01 GM81417, and DOD W81XWH-07-01-0483 to Alun Thomas.

References

Broder, A. Z. & Mayr, E. W. (1997), Counting minimum weight spanning trees, *Journal of Algorithms* **24**, 171–176.

Cayley, A. (1889), A theorem on trees, *Quarterly Journal of Mathematics* **23**, 376–378.

Coppersmith, D. & Winograd, S. (1990), Matrix multiplication via arithmetic progressions, *Journal of Symbolic Computation* **9**, 251–280.

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

Golumbic, M. C. (1980), *Algorithmic Graph Theory and Perfect Graphs*, Academic Press.

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

Jensen, F. V. (1988), Junction trees and deomposable hypergraphs, Technical report, Judex Datasystemer, Aalborg, Denmark.

Jones, B., Carvalho, C., Dobra, A., anc C Carter, C. H. & West, M. (2005), Experiments in stochastic compuation for high-dimensional graphical models, *Statistical Science* **20**, 388–400.

Kirkpatrick, S., Gellatt, Jr., C. D. & Vecchi, M. P. (1982), Optimization by simmulated annealing, Technical Report RC 9353, IBM, Yorktown Heights.

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

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N. & Teller, A. H. (1953), Equations of state calculations by fast computing machines, *Journal of Chemistry and Physics* **21**, 1087–1091.

Moon, J. W. (1970), Enumerating labelled trees, *in* F. Harary, ed., Graph Theory and Theoretical Physics, Academic Press, London.

Prüfer, H. (1918), Neuer beweis eines satzes uber permutationen, *Archiv fur Mathematik und Physik* **27**, 142–144.

Strassen, V. (1969), Gaussian elimination is not optimal, **13**, 354–356.

Tarjan, R. E. & 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**, 566–579.

Thomas, A. (2005), Characterizing allelic associations from unphased diploid data by graphical modeling, *Genetic Epidemiology* **29**, 23–35.

Thomas, A. (2008), Estimation of graphical models whose conditional independence graphs are interval graphs and its application to modeling linkage disequilibrium, *Computational Statistics and Data Analysis*. In press.

Thomas, A. & Camp, N. J. (2004), Graphical modeling of the joint distribution of alleles at associated loci, *American Journal of Human Genetics* **74**, 1088–1101.

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

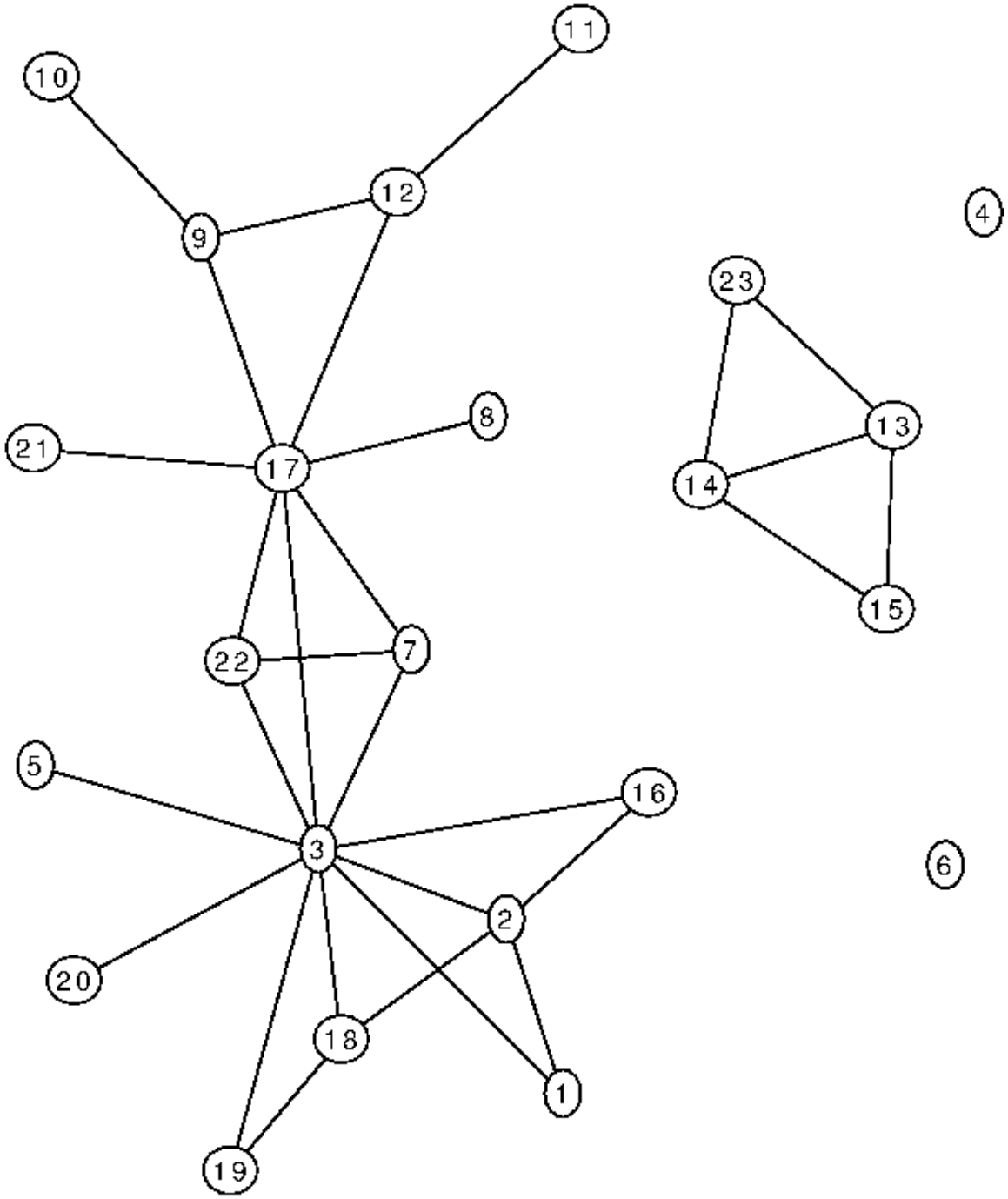

Figure 2: One possible junction tree for the graph shown in Figure 1. The 16 cliques are the vertices of the junction tree and are shown as ovals. The 15 clique separators are represented by the edges of the graph and each edge is associated with the intersection of its incident vertices. These intersections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

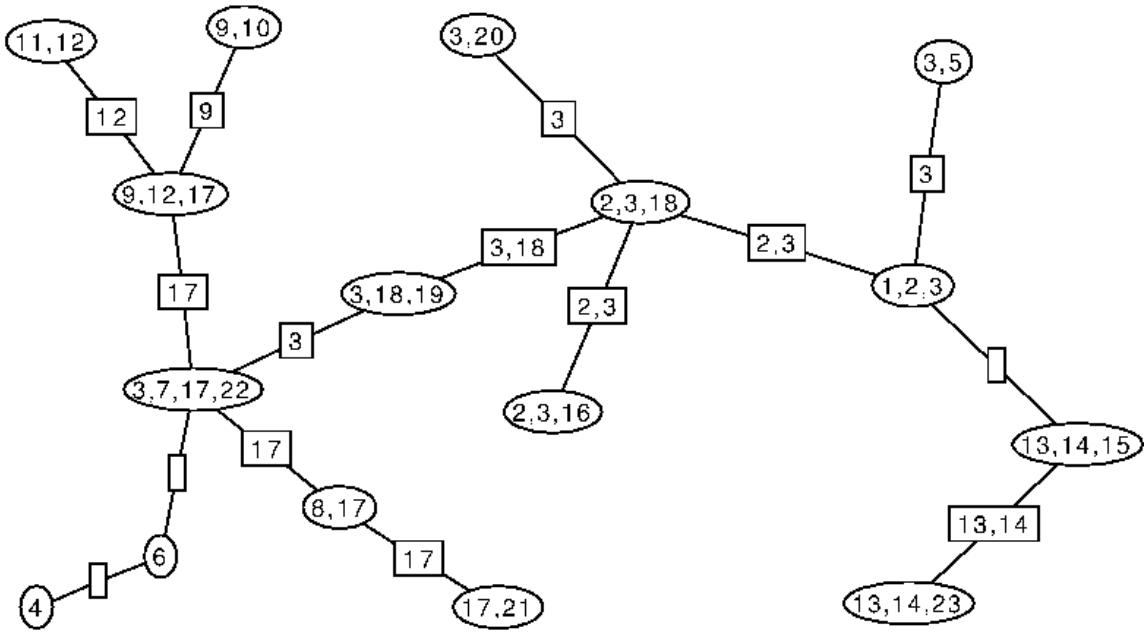

Figure 3: $T_{\{3\}}$, the connected subtree of the junction graph shown in Figure 2 induced by the cliques that contain the separator $\{3\}$.

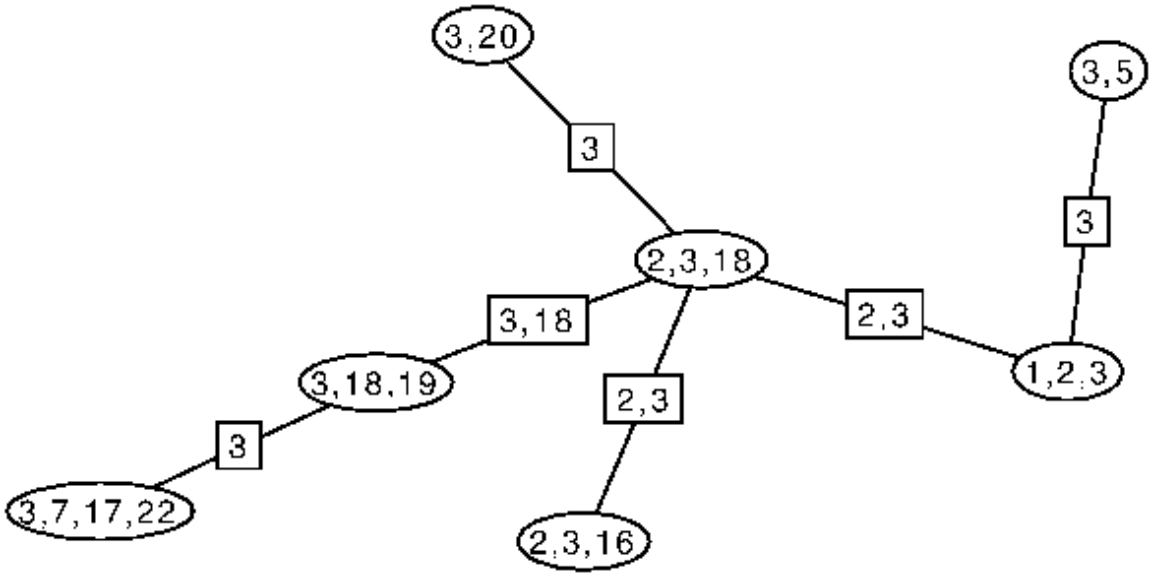

Figure 4: $F_{\{3\}}$, the forest obtained by from the tree shown in Figure 3 by deleting edges associated with the separator $\{3\}$.