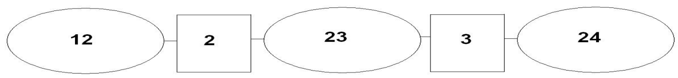

Figure 6.1. Junction Tree Example

EE 381V: Sparsity, Structures and Algorithms	**Spring 2010**

Lecture 6 — February 10

Lecturer: Sujay Sanghavi | *Scribe: Vishwas Vasuki, Nikita Sudan*

6.1 Topics covered

- Junction Tree Algorithm

6.2 Junction Tree Property

A clique tree is said to have the junction tree property if for every pair of cliques c_1 and c_2, the nodes in c_1 and c_2 appear in **all** cliques and seperators on the **unique** path between c_1 and c_2 on the clique tree. This is equivalent to saying for every node i of original graph, the cliques and seperators containing i should form a connected sub-tree of clique tree. See Figure 6.1 for an example of a tree that satisfies the junction tree property since the index "2" appears in each of the three connected nodes. Figure 6.2 on the other hand is not a junction tree.

- Clique trees with Junction property are called Junction tree.
- Note every graph has a junction tree.
- Not every graph has a junction tree (i/e/ a clique tree with the junction tree property. For e.g. the graph in Figure 6.3

Lemma 6.1. *Let C be a leaf-clique in clique tree. Let S be the unique seperator connected to it. Let R be the union of all cliques that are not C. Then the Junction property implies that $A \wedge R = \emptyset$.*

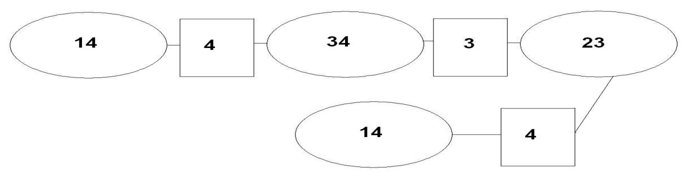

Figure 6.2. Not a Junction Tree

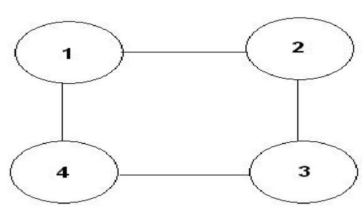

Figure 6.3. No Junction Tree possible for this graph

Proof: Suppose $v \in A \wedge v \in R$.
$\Rightarrow v \in C_{i\backslash s}$ for some $C_i \neq C \wedge v \in C$
$\Rightarrow v \in S$ by Junction Tree property.
$\Rightarrow$ contradiction

□

Proof: By Induction:
Initially: True if the Junction Tree for f contains only one clique.
Inductive Hypothesis: Suppose for all f with junction having <= $n-1$ cliques we have that, $b_c(x_c) = \sum_{x_{j\backslash c}} \tilde{f}(x) \; \forall \; C$. Now consider a Junction Tree with n cliques and let C_1 be a leaf clique and S its (unique) seperator. Let $A = C_1 \backslash S \wedge R = \bigcup_{i \neq 1}(c_i \backslash S)$ represent all other cliques. By lemma, $A \wedge R$ are disjoint. Also, the Junction tree parameter is $f(x) = f_{A,S}(x_A, x_S)\; f_{R,S}(x_R, x_S)$. Assume $m_{s \to c_1} = \prod_{c_i \neq c_1} m_{c_i \to s}(x_s)$ is constant for now i.e. for the tree $f_{R,S}$ without C_1.

Let $\tilde{b}$ be the beliefs when Junction Tree algorithm is run on this tree. Then, by the inductive assumption, $\tilde{b}_{(x_s)} = \sum_{x_R} f_{R,S}(x_R, x_S)$. Let C_2 be some clique in R that is joined to (and hence contains) S. Then $\tilde{b}(x_s) = \sum_{x_{C_2 \backslash S}} \tilde{b}(x_{C_2})$
$= [\sum_{x_{C_2} \backslash S} f_{C_2}(x_{C_2}) \prod_{s' \neq s} \tilde{m}_{s' \to C_2}(x_{s'})] \; \tilde{m}_{S \to C_2}(x_S)$
$= \tilde{m}_{C_2 \to S}(x_S) \; \tilde{m}_{S \to C_2}(x_S)$

Now, $\tilde{m}_{C_2 \to S}(x_S) = m_{C_2 \to S}(x_S)$ i.e. the message in bigger i.e. the full Junction Tree and $\tilde{m}_{S \to C_2}(x_S) = \prod_{i \neq 1,2} m_{C_i \to S}(x_S)$
Thus, $\tilde{b}(x_S) = \prod_{i \neq 1} m_{C_i \to S}(x_S)$

Combining equations 1, 2 and 3, we get:

$m_{s \to C_i(x_S)} = \sum_{x_R} f_{R,S}(x_R, x_S)$.
Therefore $b_{C_1}(x_{C_1}) = f_{A,S}(x_A, x_S) \sum_{x_R} f_{R,S}(x_R, x_S)$ i.e. $b_{C_1}(x_{C_1}) = \sum_{x_{v \backslash C_1}} f(x)$ □

Q: Can we always make a Junction Tree? If not, when?

A: No, as can be seen in Figure 6.3.

6.3 Triangulated Graphs

Every cycle in a triangulated graph has a chord. Such graphs are also called chordal graphs.

Theorem 6.2. *G has a Junction Tree ⇔ G is a triangulated graph.*

Proof: Using constructive procedure by building junction tree:
a) Choose clique nodes such that each is a maximal clique in original graph. Now, between every pair of cliques C_1 and C_2 there is a potentially empty seperator $S = C_1 \wedge C_2$. Let "weight" of this "seperator edge" between C_1 and C_2 be $w_{12} = |S| = |C_1 \wedge C_2|$.

b) Run max weight spanning tree algorithm (on clique-tree, with edge weights as above) to obtain a clique tree. This tree is a Junction tree of graph is triangulated. Consider any node k in the original graph with m clique nodes, then the number of seperators is given by

$\sum_{j=1}^{m-1} 1_{X_k \in S_j} <= [\sum_{i=1}^{M} 1_{X_k \in C_i}] - 1$
weight of tree w(T) = $\sum_{k=1}^{N} (number of seperators k appears in)$
$\sum_{k=1}^{N} (\sum_{j=1}^{m-1} 1_{k \in S_j})$
$<= \sum_{k=1}^{N} [(\sum_{i=1}^{M} 1_{kinC_i}) - 1]$
$= \sum_{i=1}^{M} (\sum_{k=1}^{N} 1_{k \in C_i}) - M$
$= \sum_{i=1}^{M} |C_i| - M$
$\Leftrightarrow$ cliques containing node k form a connected subtree in T, for every k
$\Leftrightarrow$ depth of Junction Tree $\square$