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

Lecture 6 — Feb 10

Lecturer: Sujay Sanghavi *Scribe: vishvAs vAsuki, Nikita Sudan*

6.1 Topics covered

- Inference using Clique trees.
- Junction trees

  Relationship with triangulated graphs.

6.2 Inference on clique trees

6.2.1 Clique trees

We studied clique trees in the previous lecture. In a clique tree, the nodes are maximal cliques of the origninal graph, together with separator nodes. An example is shown in Figure 6.2.

6.2.2 Marginalization

An example

Consider the graphical model in Figure 6.1, and its clique tree, shown in Figure 6.2. We want to do marginalization by passing messages in the clique tree, to evaluate $Pr(x_2, x_3)$.

The following messages are passed.

$$m_{21\to 2}(x_2) = \sum_{x_1} f_{21}(x_2, x_1)$$

$$m_{24\to 2}(x_2) = \sum_{x_2} f_{24}(x_2, x_4)$$

The node 2 then combines these messages:

$$m_{2\to 23}(x_2) = m_{21\to 2}(x_2)m_{24\to 2}(x_2)$$

Clique 23 generates the belief:

$$b_{23}(x_2, x_3) = f_{23}(x_2, x_3)m_{2\to 23}(x_2)$$

If $Pr(x) = f_{12}f_{23}f_{24}$, then it is easy to see that $b_{23}(x_2, x_3) = Pr(x_2, x_3)$.

Update rules and beliefs in general

Let c be a clique, and let it be connected to separator s. Then, the update rules are as follows:

$$m_{c\to s}(x_s) = \sum_{x_{c\setminus s}} f_c(x_c) \prod_{s'\subset c,\ s'\neq s} m_{s'\to c}(x_{s'})$$

$$m_{s\to c}(x_s) = \prod_{s\subset c,\ c'\neq c} m_{c'\to s}(x_s)$$

The beliefs are as follows:

$$b_c(x_c) = f_c(x_c) \prod_{s\subset c} m_{s\to c}(x_s)$$

6.2.3 Consistency and correctness

Each variable i can appear in multiple cliques (say c, c') of the clique tree. So, unlike in the case of sum products, we should check for consistency: the marginal of a variable, as predicted by each clique should be the same, independent of the clique. So, if $b_i(x_i) = \sum_{x_{c\setminus i}} b_c(x_c)$ and $b'_i(x_i) = \sum_{x_{c'\setminus i}} b'_c(x'_c)$, we want that $\forall i : b_i(x_i) = b'_i(x_i)$.

Clique trees where inference fails

Consider the graph in figure 6.3 and its clique tree, shown in figure 6.4. Let us find $b_1(x_1) = \sum_{x_2} b_{12}(x_1, x_2)$ and $b'_1(x_1) = \sum_{x_4} b_{14}(x_1, x_4)$.

Below, we use a_1 in the place of x_1 when we compute $m_{14\to 4}(x_4)$ to emphasize the fact that x_1 appears again, later in the computation.

$$\begin{aligned}
m_{4\to 34}(x_4) &= m_{14\to 4}(x_4) = \sum_{a_1} f_{14}(a_1, x_4) \\
m_{34\to 3}(x_3) &= m_{3\to 23}(x_3) = \sum_{x_4, a_1} f_{14}(a_1, x_4) f_{34}(x_3, x_4) \\
m_{23\to 2}(x_2) &= m_{2\to 12}(x_2) = \sum_{x_3, x_4, a_1} f_{14}(a_1, x_4) f_{34}(x_3, x_4) f_{23}(x_2, x_3) \\
b_{12}(x_1, x_2) &= f_{12}(x_1, x_2) \sum_{x_3, x_4, a_1} f_{14}(a_1, x_4) f_{34}(x_3, x_4) f_{23}(x_2, x_3)
\end{aligned}$$

By symmetry, we have:

$$b_{14}(x_1, x_4) = f_{14}(x_1, x_4) \sum_{x_3, x_2, a_1} f_{14}(a_1, x_2) f_{32}(x_3, x_2) f_{43}(x_4, x_3)$$

.

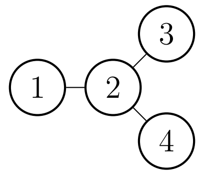

Figure 6.1. An undirected graphical model

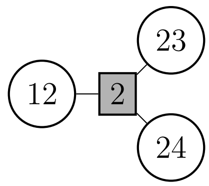

Figure 6.2. Clique Tree corresponding to graph in Figure 6.1. This has the junction tree property.

Observe that $b_1(x_1) = \sum_{x_2} b_{12}(x_1, x_2)$ and $b_1'(x_1) = \sum_{x_4} b_{14}(x_1, x_4)$ are not equal in general. To see this, consider the case where $f_{14}(x_1, x_4)$ is a constant with respect to x_1, but $f_{12}(x_1, x_2)$ depends on x_1.

6.2.4 Summary and fixes

Suppose we altered the figure in Figure 6.3 to get the Figure 6.5. Then, the junction tree would be Figure 6.6. Now, $Pr(x) = f_{124}(x) f_{234}(x)$, and we do not encounter the problem of a variable being 'lost' in the process of being marginalized out, and being 'found' later in computations of some other clique in the clique tree. Next, we study, the property that distinguishes the cases in Figure 6.3 and Figure 6.6.

6.3 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 6.2 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 6.4 on the other hand is not a junction tree.

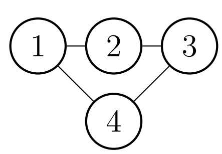

Figure 6.3. There exists no Junction Tree possible for this graph.

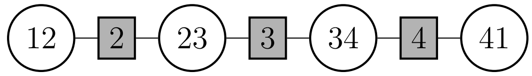

Figure 6.4. Clique tree corresponding to graph in figure 6.3. This does not have the junction tree property.

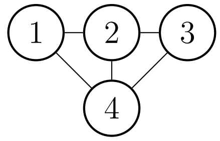

Figure 6.5. The graph in Figure 6.3 altered to be chordal.

- 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 $A \wedge R = \emptyset$.*

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

$\square$

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 $f(x) = f_{A,S}(x_A, x_S)\ f_{R,S}(x_R, x_S)$.

$$m_{s \to c_1} = \prod_{c_i \neq c_1} m_{c_i \to s}(x_s) \tag{6.1}$$

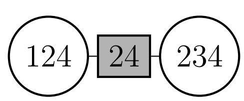

Figure 6.6. Clique tree corresponding to graph in figure 6.5. This has the junction tree property.

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,

$$\tilde{b}_{(x_s)} = \sum_{x_R} f_{R,S}(x_R, x_S). \tag{6.2}$$

Let C_2 be some clique in R that is joined to (and hence contains) S. Then,

$$\begin{aligned} \tilde{b}(x_s) &= \sum_{x_{C_2 \setminus S}} \tilde{b}(x_{C_2}) \\ &= [\sum_{x_{C_2 \setminus S}} f_{C_2}(x_{C_2}) \prod_{s' \neq s} \tilde{m}_{s' \to C_2}(x_{s'})] \tilde{m}_{S \to C_2}(x_S) \\ &= \tilde{m}_{C_2 \to S}(x_S) \tilde{m}_{S \to C_2}(x_S) \end{aligned}$$

Now, $\tilde{m}_{C_2 \to S}(x_S) = m_{C_2 \to S}(x_S)$ i.e. the message in bigger for the full Junction Tree and

$$\begin{aligned} \tilde{m}_{S \to C_2}(x_S) &= \prod_{i \neq 1,2} m_{C_i \to S}(x_S) \\ \therefore \tilde{b}(x_S) &= \prod_{i \neq 1} m_{C_i \to S}(x_S) \end{aligned} \tag{6.3}$$

Combining equations 6.1, 6.2 and 6.3, we get:

$$\begin{aligned} m_{s \to C_i(x_S)} &= \sum_{x_R} f_{R,S}(x_R, x_S) \\ \therefore b_{C_1}(x_{C_1}) = f_{A,S}(x_A, x_S) \sum_{x_R} f_{R,S}(x_R, x_S) &= \sum_{x_{v \setminus C_1}} f(x) \end{aligned}$$

□

Q: Can we always make a Junction Tree? If not, when?
A: No, as can be seen in Figure 6.3.

6.4 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 $\Leftrightarrow$ 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,

$$
\begin{aligned}
num\_seperators = by \sum_{j=1}^{m-1} 1_{X_k \in S_j} &<= [\sum_{i=1}^{M} 1_{X_k \in C_i}] - 1 \\
weight\_tree = \sum_{k=1}^{N} &(num\_seperators) \\
&\sum_{k=1}^{N}(\sum_{j=1}^{m-1} 1_{k \in S_j}) \\
&<= \sum_{k=1}^{N}[(\sum_{i=1}^{M} 1_{kinC_i}) - 1] \\
&= \sum_{i=1}^{M}(\sum_{k=1}^{N} 1_{k \in C_i}) - M \\
&= \sum_{i=1}^{M} |C_i| - M
\end{aligned}
$$

The above is equivalent to the cliques containing node k which form a connected subtree in T, for every k. This is nothing but the depth of the Junction Tree. $\square$