COS513 LECTURE 5
JUNCTION TREE ALGORITHM

D. EIS AND V. KOSTINA

The junction tree algorithm is the culmination of the way graph theory and probability combine to form graphical models. After we discuss the junction tree algorithm, we will move on to developing the models for data analysis.

1. Algorithm Outline

The junction tree algorithm has many different moving parts and we will describe each one separately and all the pieces will come together by the end.

The goal of the algorithm is to parameterize the model such that it is easy to calculate marginals.

We start with a graphical model and then do the following steps:

(1) Moralize the graph
(2) Triangulate the graph
(3) Build a junction tree
(4) Apply the message passing algorithm

When this algorithm is done we are left with the quantities needed to compute the desired marginals.

The idea behind the algorithm is to use independence properties of the graph to decompose "general probability calculations" into a linked set of local computations, along the lines of what we did in the elimination and message passing algorithms, but here we are switching from an algebraic process of inference to an inference data structure. This algorithm will work with any model that is described by a graph.

Now we return to the graph we have looked at since the beginning (Fig. 1). Remember, when we do elimination on this graph, we first moralize the graph by connecting all unconnected parents. After this we make the graph an undirected graph. After applying eliminate, we can construct the reconstituted graph (Fig. 2), which is *triangulated*, i.e. for any given cycle there is an edge between any two non-successive nodes in the cycle.

Previously we used the elimination algorithm to obtain marginals. We also employed the tree propagation algorithm to reuse intermediate factors

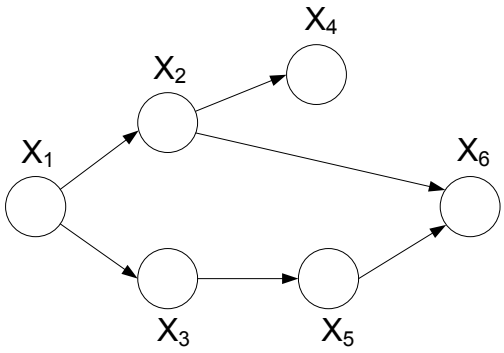

FIGURE 1. The six-node example from previous lectures.

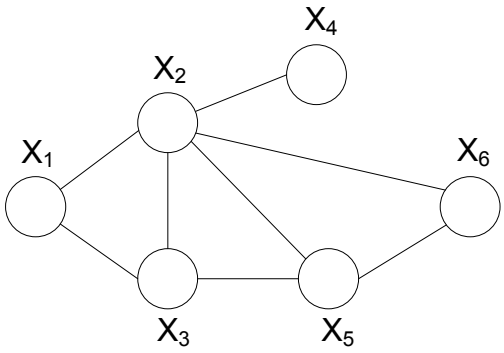

FIGURE 2. Reconstituted graph.

in a tree graph. The junction tree allows us to reuse intermediate factors in any graphical model.

The first step in moving beyond the elimination algorithm is to allocate storage for the intermediate factors. Toward this end, we form a clique tree out of the elimination cliques in the graph. Each intermediate factor is associated with one of the cliques, so the clique tree represents the storage we need to do elimination. Moreover, it illustrates the information flow of the elimination procedure - we can follow what happens in elimination by examining the clique tree. An example of a clique tree for our graph using the elimination ordering $\{6,5,4,3,2,1\}$ is shown in Fig. 3. At this point we note that the clique tree depends on the chosen ordering.

We introduce the notion of a "separator set", which is just the intersection of two adjacent cliques. We show these sets in boxes separating two cliques explicitly in Fig. 3. Note that members of separation sets correspond to the parameters of the intermediate factors. For example, separation set $\{X_1\}$ corresponds to the parameter X_1 of the intermediate factor which arises

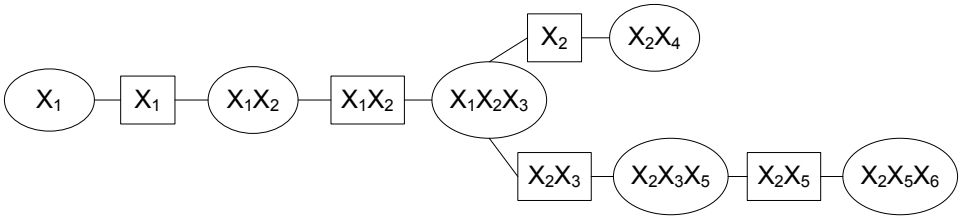

FIGURE 3. A clique tree with separator sets using elimination ordering $\{6,5,4,3,2,1\}$.

after removing $\{X_2\}$. Later we will define potentials on these separator sets.

As a preview, we note that this tree in Fig. 3 has the junction tree property: if a variable is in two cliques then it is in every clique along the path connecting the two cliques.

2. CLIQUE POTENTIALS

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be a graphical model and let $\mathcal{C}$ be a set of cliques, where a clique is a completely connected subset of the graph. Then for each $c \in \mathcal{C}$, define a clique potential $\psi_c(X_c)$. Define the joint probability as in the undirected model:

$$p(x) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \psi_c(X_c) \tag{1}$$

Define $\mathcal{C}$ to be the set of maximal cliques, meaning that no member of $\mathcal{C}$ is a subset of another member of $\mathcal{C}$.

Eventually, we will adjust the clique potentials, ψ_c's, to be marginals, so that we can answer general inferential questions about our model.

3. INITIALIZING CLIQUE POTENTIALS

The first step we do is to initialize the clique potentials.

We note now that the clique potential assigned to each clique is not necessarily the same as the clique potential in the undirected graph. In the case of the undirected graph, if the potentials that are defined on the undirected graph are all on maximal cliques then the clique potentials are just the potentials defined on the maximal cliques. If the potentials are not on maximal cliques, then we let the clique potential equal the product of potentials on subcliques, with the caveat that each potential can only be used once in these definitions, so there will be no repetitions. This ensures that the joint probability, the product of the clique potentials, is the same as the product of the potentials. For example, in Fig. 4, if the potentials are defined on the

pairwise cliques, then we can define the clique potentials as follows:

$$\psi_{ABC}(A,B,C) = \psi_{AB}(A,B)\psi_{BC}(B,C)\psi_{AC}(A,C) \tag{2}$$

$$\psi_{BCD}(B,C,D) = \psi_{CD}(C,D)\psi_{BD}(B,D) \tag{3}$$

You can confirm that you must use each potential exactly once, otherwise the joint will not be the same.

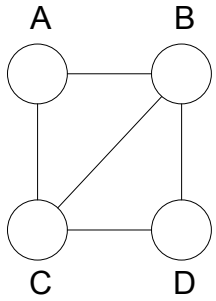

FIGURE 4. A four-node model which we assume is parameterized with pairwise potentials ψ_{AB}, ψ_{AC}, ψ_{BC}, ψ_{BD} and ψ_{CD}.

To initialize the clique potentials for directed graphs, first moralize the graph. If we do this to the graph in Fig. 1 then we get the graph shown in Fig. 2. The maximal cliques are: $\{x_2, x_5, x_6\}$, $\{x_2, x_4\}$, $\{x_1, x_3\}$, $\{x_1, x_2\}$, $\{x_3, x_5\}$. Then we define the clique potentials to be the product of conditional probabilities such that each conditional probability only involves nodes that are in the maximal clique this potential is being defined on and such that each conditional probability is assigned to exactly one clique potential. We can define the clique potentials as follows for the graph in Fig. 1:

$$\begin{aligned}
\psi_{24}(x_2, x_4) &= p(x_4|x_2) \\
\psi_{12}(x_1, x_2) &= p(x_1)p(x_2|x_1) \\
\psi_{13}(x_1, x_3) &= p(x_3|x_1) \\
\psi_{35}(x_3, x_5) &= p(x_5|x_3) \\
\psi_{256}(x_2, x_5, x_6) &= p(x_6|x_2, x_5)
\end{aligned} \tag{4}$$

To summarize, we initialize clique potentials with original undirected potentials *or* with conditional probabilities tables from the moralized graph.

4. EVIDENCE

Suppose nodes are divided into subsets H and E, where E contains evidence nodes and H contains everything else. How do we compute $p(x_H|x_E)$? As it turns out, this problem is no different in principle from the calculation

of marginal probabilities from (1). The following trick will do the job: we add $\delta(x_E, \overline{x_E})$ to one clique potential containing E, where $\overline{x_E}$ is observed value. Then computing the right hand side of (1) we will get $p(x_H, \overline{x_E})$. The only nuance is that the normalization constant Z remains without δ's since it is a property of the distribution and does not depend on evidence. Let us now focus on computing clique potentials.

5. COMPUTING CLIQUE POTENTIALS

Our goal is to compute the marginal $p(x_F)$ for $F \subseteq C, C \in \mathcal{C}$. The junction tree algorithm uses information flow in the graph to adjust the clique potentials to give this marginal probability. When the algorithm terminates, we'll have $\psi_C(x_C) = p(x_C)$ (or $p(x_C, \overline{x_E})$ if we have evidence).

Let us consider the Markov chain example in Fig. 5.

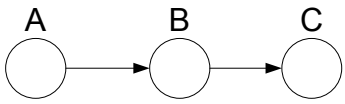

FIGURE 5. A three-node model we use to illustrate local consistency.

The clique potentials for this model are given by:

$$\begin{aligned} \psi_{AB}(A, B) &= p(A)p(B|A) = p(A, B) \\ \psi_{BC}(B, C) &= p(C|B) \end{aligned} \tag{5}$$

Clique potential ψ_{AB} already corresponds to the marginal probability $p(A, B)$. We want to make ψ_{BC} a marginal as well. Naively, we sum out A from ψ_{AB}:

$$\sum_A \psi_{AB} = p(B) \tag{6}$$

Further, multiplying ψ_{BC} by this $p(B)$ gives the new ψ_{BC}:

$$\psi_{BC} = p(C|B)p(B) = p(B, C) \tag{7}$$

Now, both clique potentials are marginals. But here is the problem: we messed up the joint (1)!

The junction tree algorithm will solve this problem. It manipulates the clique potentials to be marginals, but at the same time it does not alter the joint distribution. Moreover, it maintains both *local* and *global consistency*.

Local consistency means that the neighboring clique potentials agree when the marginal of their common node is calculated from either of them. For example, in Fig. 5 suppose we adjusted the clique potentials ψ_{AB} and ψ_{BC} to be $p(A, B)$ and $p(B, C)$. Local consistency means that when we

sum out A from ψ_{AB} it gives the same $p(B)$ as when we sum out C from ψ_{BC}. *Global consistency* means that *any* two cliques containing the same node will agree on the marginal of that node.

6. Separator sets

To maintain the joint $p(x)$ like the junction tree algorithm does, we extend our representation to include separator sets, and we define potentials on these separator sets:

$$p(x) = \frac{\prod_{c \in \mathcal{C}} \psi_c(X_c)}{\prod_{s \in \mathcal{S}} \psi_s(X_s)} \tag{8}$$

Here $\mathcal{S}$ includes all separator sets and $\psi_s(X_s)$ are separator potentials. The idea is to adjust the clique potentials to obtains marginals, and to adjust the ψ_s's to maintain the joint, $p(x)$.

For example, extended representation for the Markov chain in Fig. 5 is:

$$p(A,B,C) = p(A)p(B|A)p(C|B) = p(A,B)p(C|B) = \frac{p(A,B)p(C,B)}{p(B)},$$

where the numerator, $p(B)$, is a separator potential.

We can find this extended representation for any probability distribution.

(1) It includes all distributions in Eq. (1).
(2) It contains no others distributions.