CS281A/Stat241A Lecture 19

Junction Tree Algorithm

Peter Bartlett

Announcements

- My office hours:
  Tuesday Nov 3 (today), 1-2pm, in 723 Sutardja Dai Hall.
  Thursday Nov 5, 1-2pm, in 723 Sutardja Dai Hall.
- Homework 5 due 5pm Monday, November 16.

Key ideas of this lecture

- Junction Tree Algorithm.
  - (For directed graphical models:) Moralize.
  - Triangulate.
  - Construct a junction tree.
  - Define potentials on maximal cliques.
  - Introduce evidence.
  - Propagate probabilities.

Junction Tree Algorithm

- Inference: Given
  - Graph $G = (V, \mathcal{E})$,
  - Evidence x_E, for $E \subseteq V$,
  - Set $F \subseteq V$,

  compute $p(x_F | x_E)$.

Junction Tree Algorithm

- Elimination:
  - Single set F.
  - Any G.
- Sum-product:
  - **All singleton sets** F simultaneously.
  - G a **tree**.
- **Junction tree:**
  - **All cliques** F simultaneously.
  - Any G.

Junction Tree Algorithm

1. (For directed graphical models:) Moralize.
2. Triangulate.
3. Construct a junction tree.
4. Define potentials on maximal cliques.
5. Introduce evidence.
6. Propagate probabilities.

1. Moralize

- In a directed graphical model, local conditionals are functions of a variable and its parents:

$$p(x_i|x_{\pi(i)}) = \psi_{\pi(i)\cup\{i\}}.$$

- To represent this as an undirected model, the set

$$\pi(i) \cup \{i\}$$

must be a clique.

- We consider the **moral graph**: all parents connected.

Junction Tree Algorithm

1. (For directed graphical models:) Moralize.
2. Triangulate. (e.g., run UNDIRECTEDGRAPHELIMINATE)
3. Construct a junction tree.
4. Define potentials on maximal cliques.
5. Introduce evidence.
6. Propagate probabilities.

2. Triangulate: Motivation

Theorem:
A graph G is chordal iff it has a **junction tree.**

Recall that all of the following are equivalent:

- G is chordal
- G is decomposable
- G is recursively simplicial
- G is the fixed point of UNDIRECTEDGRAPHELIMINATE
- an oriented version of G has moral graph G.
- G implies the same cond. indep. as some directed graph.

Chordal/Triangulated

Definitions:

A **cycle** for a graph $G = (V, E)$ is a vertex sequence $v_1, \ldots, v_n$ with $v_1 = v_n$ but all other vertices distinct, and $\{v_i, v_{i+1}\} \in E$.

A cycle has a **chord** if it has a pair v_i, v_j with $1 < |i - j| < n$ and $\{i, j\} \in E$.

A graph is **chordal** or **triangulated** if every cycle of length at least four has a chord.

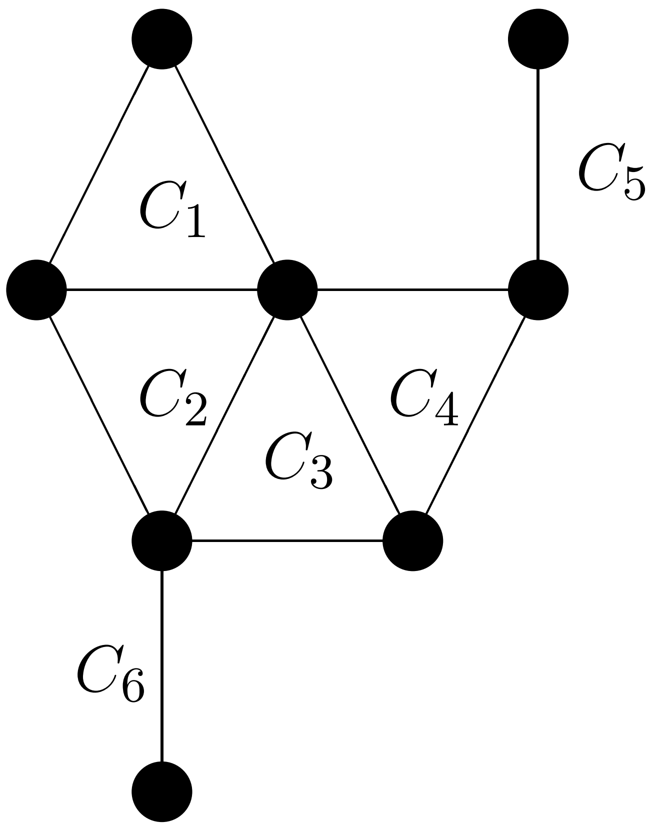
Junction Tree: Definition

A **clique tree** for a graph $G = (V, E)$ is a tree $T = (V_T, E_T)$ where

- V_T is a set of cliques of G,
- V_T contains all maximal cliques of G.

A **junction tree** for a graph G is a clique tree $T = (V_T, E_T)$ for G that has the **running intersection** property: for any cliques C_1 and C_2 in V_T, every clique on the path connecting C_1 and C_2 contains $C_1 \cap C_2$.

Junction Tree: Example



Clique Tree:

$$
\begin{array}{ccccc}
C_1 - & C_2 - & C_3 - & C_4 - & C_5 \\
 & & | & & \\
 & & C_6 & &
\end{array}
$$

2. Triangulate

Theorem:
A graph G is chordal iff it has a **junction tree.**

Chordal $\Leftrightarrow$ Recursively Simplicial, and we'll show:
Recursively Simplicial $\Rightarrow$ Junction Tree $\Rightarrow$Chordal

Recursively Simplicial $\Rightarrow$ Junction Tree

Recall: A graph G is **recursively simplicial** if it contains a simplicial vertex v (neighbors form a clique), and when v is removed, the remaining graph is recursively simplicial.

Proof idea—induction step:
Consider a recursively simplicial graph G of size $n+1$.
When we remove a simplicial vertex v, it leaves a subgraph G' with a junction tree T.
Let N be the clique in T containing v's neighbors.
Let C be a new clique of v and its neighbors.
To obtain a junction tree for G:

- If N contains only v's neighbors, replace it with C.
- Otherwise, add C with an edge to N.

Junction Tree $\Rightarrow$ Chordal

Proof idea—induction step:
Consider a junction tree T of size $n+1$.
Consider a leaf C of T and its neighbor N in T.
Remove C from T and remove $C \setminus N$ from V.
The remaining tree is a junction tree for the remaining (chordal) graph.
All cycles have a chord, since either

1. Cycle is completely in remaining graph,
2. Cycle is completely in C, or
3. Cycle is in $C \setminus N$, $C \cap N$, and $V \setminus C$
   (and $C \cap N$ is complete, so contains a chord).

Junction Tree Algorithm

1. (For directed graphical models:) Moralize.
2. Triangulate.
3. Construct a junction tree:
   Find a maximal spanning tree.
4. Define potentials on maximal cliques.
5. Introduce evidence.
6. Propagate probabilities.

Junction Tree is Maximal Spanning Tree

Define:

The **weight** of a clique tree $T = (\mathcal{C}, E)$ is

$$w(T) = \sum_{(C,C')\in E} |C \cap C'|.$$

A **maximal spanning tree** for a clique set $\mathcal{C}$ of a graph is a clique tree $T = (\mathcal{C}, E^*)$ with maximal weight over all clique trees of the form $(\mathcal{C}, E)$.

Junction Tree is Maximal Spanning Tree

Theorem: Suppose that a graph G with clique set $\mathcal{C}$ has a junction tree. Then a clique tree $(\mathcal{C}, E)$ is a junction tree iff it is a maximal spanning tree for $\mathcal{C}$.

And there are efficient greedy algorithms for finding the maximal spanning tree:

- While graph is not connected:
    - Add an edge for the biggest weight separating set that does not lead to a cycle.

Maximal Spanning Tree: Proof

Consider a graph G with vertices V.
For a clique tree $(\mathcal{C}, E)$, define the separators

$$\mathcal{S} = \left\{ C \cap C' : (C, C') \in E \right\}.$$

Since T is a tree, for any $k \in V$,

$$\sum_{S \in \mathcal{S}} 1[k \in S] \leq \sum_{C \in \mathcal{C}} 1[k \in C] - 1,$$

with equality iff the subgraph of T of nodes containing k is a tree.

Maximal Spanning Tree: Proof

Thus,

$$
\begin{aligned}
w(T) &= \sum_{S \in \mathcal{S}} |S| \\
&= \sum_{S \in \mathcal{S}} \sum_{k} 1[k \in S] \\
&\leq \sum_{k} \left(\sum_{C \in \mathcal{C}} 1[k \in C] - 1 \right) \\
&= \sum_{C \in \mathcal{C}} |C| - n,
\end{aligned}
$$

with equality iff T is a junction tree.

Junction Tree Algorithm

1. (For directed graphical models:) Moralize.
2. Triangulate.
3. Construct a junction tree
4. Define potentials on maximal cliques.
5. Introduce evidence.
6. Propagate probabilities.

Define potentials on maximal cliques

To express a product of clique potentials as a product of maximal clique potentials:

$$\psi_{S_1}(x_{S_1}) \cdots \psi_{S_N}(x_{S_N}) = \prod_{C \in \mathcal{C}} \psi_C(x_C),$$

1. Set all clique potentials to 1.
2. For each potential, incorporate (multiply) it into a potential containing its variables.

Junction Tree Algorithm

1. (For directed graphical models:) Moralize.
2. Triangulate.
3. Construct a junction tree
4. Define potentials on maximal cliques.
5. Introduce evidence.
6. Propagate probabilities.

Introduce Evidence

Two equivalent approaches:

1. Introduce evidence potentials

$$\delta(x_i, \bar{x}_i) \qquad \text{for } i \text{ in } E,$$

   so that marginalizing fixes $x_E = \bar{x}_E$.

2. Take slice of each clique potential:

$$\psi_C(x_C) := \psi_C\left(x_{C\cap(V\setminus E)}, \bar{x}_{C\cap E}\right).$$

Junction Tree Algorithm

1. (For directed graphical models:) Moralize.
2. Triangulate.
3. Construct a junction tree
4. Define potentials on maximal cliques.
5. Introduce evidence.
6. Propagate probabilities.

Hugin Algorithm

Add potentials for **separator sets.**

Recall: If G has a junction tree $T = (\mathcal{C}, \mathcal{S})$, then any probability distribution that satisfies the conditional independences implied by G can be factorized as

$$p(x) = \frac{\prod_{C \in \mathcal{C}} p(x_C)}{\prod_{S \in \mathcal{S}} p(x_S)},$$

where, if $S = \{C_1, C_2\}$ then x_S denotes $x_{C_1 \cap C_2}$.

Hugin Algorithm

Represent p with potentials on separators:

$$p(x) = \frac{\prod_{C \in \mathcal{C}} \psi_C(x_C)}{\prod_{S \in \mathcal{S}} \phi_S(x_S)}$$

- This can represent any distribution that an undirected graphical model can (since we can set $\phi_S \equiv 1$).
- But nothing more (since we can incorporate each ϕ_S into one of the ψ_C's that have $S \subseteq C$).

Hugin Algorithm

1. Initialize:

$$\psi_C(x_C) = \text{appropriate clique potential}$$
$$\phi_S(x_S) = 1.$$

2. Update potentials so that
   - $p(x)$ is invariant
   - ψ_C, ϕ_S **become the marginal distributions.**

Algorithm:

- Messages are passed between cliques in the junction tree.
- A message is passed from V to adjacent vertex W once V has received messages from all its other neighbors.
- The message corresponds to the updates:

$$\phi_S^{(1)}(x_S) = \sum_{x_{V-S}} \psi_V(x_V),$$

$$\psi_W^{(1)}(x_W) = \psi_W(x_W) \frac{\phi_S^{(1)}(x_S)}{\phi_S(x_S)},$$

$$\psi_V^{(1)}(x_V) = \psi_V(x_V),$$

where $S = V \cap W$ is the separator.

Analysis:

1. $p(x)$ is invariant under these updates.
2. In a tree, messages can path in both directions over every edge.
3. The potentials are *locally consistent* after messages have passed in both directions.
4. In a graph with a junction tree, local consistency implies global consistency.

Analysis:

$$p(x) = \frac{\prod_{C \in \mathcal{C}} \psi_C(x_C)}{\prod_{S \in \mathcal{S}} \phi_S(x_S)}.$$

In this ratio, only $\dfrac{\psi_V \psi_W}{\phi_S}$ changes, to

$$\frac{\psi_V^{(1)} \psi_W^{(1)}}{\phi_S^{(1)}} = \frac{\psi_V \psi_W \phi_S^{(1)}}{\phi_S^{(1)} \phi_S}$$

$$= \frac{\psi_V \psi_W}{\phi_S}.$$

Analysis:

1. $p(x)$ is invariant under these updates.
2. In a tree, messages can path in both directions over every edge.
3. The potentials are *locally consistent* after messages have passed in both directions.
4. In a graph with a junction tree, local consistency implies global consistency.

Analysis: Local Consistency

Suppose that messages pass both ways, from V to W and back:

1. a message passes from V to W:

$$\phi_S^{(1)}(x_S) = \sum_{x_{V-S}} \psi_V(x_V),$$

$$\psi_W^{(1)}(x_W) = \psi_W(x_W) \frac{\phi_S^{(1)}(x_S)}{\phi_S(x_S)},$$

$$\psi_V^{(1)}(x_V) = \psi_V(x_V),$$

Analysis: Local Consistency

2. other messages pass to W:

$$\phi_S^{(2)} = \phi_S^{(1)}$$

$$\psi_V^{(2)} = \psi_V^{(1)}$$

$$\psi_W^{(2)} = \cdots$$

Analysis: Local Consistency

3. a message passes from W to V:

$$\phi_S^{(3)}(x_S) = \sum_{x_{W-S}} \psi_W^{(2)}(x_W),$$

$$\psi_V^{(3)}(x_V) = \psi_V^{(2)}(x_V)\frac{\phi_S^{(3)}(x_S)}{\phi_S^{(2)}(x_S)},$$

$$\psi_W^{(3)}(x_V) = \psi_W(x_W).$$

Subsequently, ϕ_S, ψ_V, ψ_W remain unchanged.

Analysis: Local Consistency

After messages have passed in both directions, these potentials are *locally consistent*:

$$\sum_{x_{V\setminus S}} \psi_V^{(3)}(x_V) = \sum_{x_{W\setminus S}} \psi_W^{(3)}(x_W) = \phi_S^{(3)}(x_S).$$

c.f.:

$$\sum_{x_{V\setminus S}} p(x_V) = \sum_{x_{W\setminus S}} p(x_W) = p(x_S).$$

Analysis: Local Consistency Proof

$$\sum_{x_{V\setminus S}} \psi_V^{(3)}(x_V) = \sum_{x_{V\setminus S}} \psi_V^{(2)}(x_V) \frac{\phi_S^{(3)}(x_S)}{\phi_S^{(2)}(x_S)} \qquad (W \to V)$$

$$= \frac{\phi_S^{(3)}(x_S)}{\phi_S^{(1)}(x_S)} \sum_{x_{V\setminus S}} \psi_V(x_V) \qquad (\text{to } W)$$

$$= \phi_S^{(3)}(x_S) \qquad (V \to W)$$

$$= \sum_{x_{W\setminus S}} \psi_W^{(2)}(x_W)$$

$$= \sum_{x_{W\setminus S}} \psi_W^{(3)}(x_W).$$

Analysis:

1. $p(x)$ is invariant under these updates.
2. In a tree, messages can path in both directions over every edge.
3. The potentials are *locally consistent* after messages have passed in both directions.
4. In a graph with a junction tree, local consistency implies global consistency.

Analysis: Local implies Global

Local consistency:
For all adjacent cliques V, W with separator S,

$$\sum_{x_{V\setminus S}} \psi_V^{(3)}(x_V) = \sum_{x_{W\setminus S}} \psi_W^{(3)}(x_W) = \phi_S^{(3)}(x_S).$$

Global consistency:
For all cliques C,

$$\psi_C(x_C) = \sum_{x_{C^c}} \frac{\prod_C \psi_C(x_C)}{\prod_S \phi_S(x_S)} = p(x_C)$$

Local implies Global: Proof

Induction on number of cliques.
Trivial for one clique.
Assume true for all junction trees with n cliques.
Consider a tree T_W of size $n+1$.
Fix a leaf B, attached by separator R.
Define

$$\begin{aligned} W &= \text{all variables} \\ N &= B \setminus R \\ V &= W \setminus N \\ T_V &= \text{junction tree without } B. \end{aligned}$$

Local implies Global: Proof

The junction tree T_V:

- Has only variables V (from the junction tree property).
- Satisfies local consistency.
- Hence satisfies global consistency for V:
  for all $A \in T_V$,

$$\psi_A(x_A) = \sum_{x_{V \setminus A}} \frac{\prod_{C \in \mathcal{C}_V} \psi_C(x_C)}{\prod_{S \in \mathcal{S}_V} \phi_S(x_S)}.$$

Local implies Global: Proof

- But viewing this A in the larger T_W, we have

$$\sum_{x_{W\setminus A}} \frac{\prod_{C\in\mathcal{C}_W}\psi_C(x_C)}{\prod_{S\in\mathcal{S}_W}\phi_S(x_S)} = \sum_{x_{V\setminus A}} \sum_{x_N} \frac{\psi_B(x_B)}{\phi_R(x_R)} \frac{\prod_{C\in\mathcal{C}_V}\psi_C(x_C)}{\prod_{S\in\mathcal{S}_V}\phi_S(x_S)}$$

$$= \sum_{x_{V\setminus A}} \frac{\prod_{C\in\mathcal{C}_V}\psi_C(x_C)}{\prod_{S\in\mathcal{S}_V}\phi_S(x_S)}$$

$$= \psi_A(x_A).$$

Local implies Global: Proof

And for the new clique B:
Suppose A is the neighbor of B in T_W.

$$\sum_{x_{W \setminus B}} \frac{\prod_{C \in \mathcal{C}_W} \psi_C(x_C)}{\prod_{S \in \mathcal{S}_W} \phi_S(x_S)} = \sum_{x_{A \setminus R}} \sum_{x_{V \setminus A}} \frac{\psi_B(x_B)}{\phi_R(x_R)} \frac{\prod_{C \in \mathcal{C}_V} \psi_C(x_C)}{\prod_{S \in \mathcal{S}_V} \phi_S(x_S)}$$

$$= \frac{\psi_B(x_B)}{\phi_R(x_R)} \sum_{x_{A \setminus R}} \left(\sum_{x_{V \setminus A}} \frac{\prod_{C \in \mathcal{C}_V} \psi_C(x_C)}{\prod_{S \in \mathcal{S}_V} \phi_S(x_S)} \right)$$

$$= \psi_B(x_B) \frac{\sum_{x_{A \setminus R}} \psi_A(x_A)}{\phi_R(x_R)}$$

$$= \psi_B(x_B).$$

Junction Tree Algorithm

1. (For directed graphical models:) Moralize.
2. Triangulate.
3. Construct a junction tree
4. Define potentials on maximal cliques.
5. Introduce evidence.
6. Propagate probabilities.

Junction Tree Algorithm: Computation

- Size of largest maximal clique determines run-time.
- If variables are categorical and potentials are represented as tables, marginalizing takes time exponential in clique size.
- Finding a triangulation to minimize the size of the largest clique is NP-hard.

Key ideas of this lecture

- Junction Tree Algorithm.
  - (For directed graphical models:) Moralize.
  - Triangulate.
  - Construct a junction tree.
  - Define potentials on maximal cliques.
  - Introduce evidence.
  - Propagate probabilities.