Machine Learning
4771

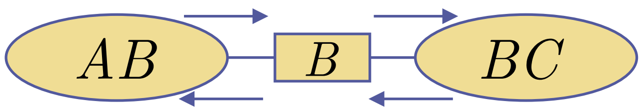
Instructor: Tony Jebara

Topic 18

- The Junction Tree Algorithm
- Collect & Distribute
- Algorithmic Complexity
- ArgMax Junction Tree Algorithm

Review: Junction Tree Algorithm

- Send message from each clique *to* its separators of what it thinks the submarginal on the separator is.
- Normalize each clique by incoming message *from* its separators so it agrees with them



$$V = \{A,B\} \quad S = \{B\} \quad W = \{B,C\}$$

If agree: $\sum_{V\backslash S} \psi_V = \phi_S = p(S) = \phi_S = \sum_{W\backslash S} \psi_W$ **...Done!**

Else: **Send message From V to W...**

$$\phi_S^* = \sum_{V\backslash S} \psi_V$$
$$\psi_W^* = \frac{\phi_S^*}{\phi_S} \psi_W$$
$$\psi_V^* = \psi_V$$

Send message From W to V...

$$\phi_S^{**} = \sum_{W\backslash S} \psi_W^*$$
$$\psi_V^{**} = \frac{\phi_S^{**}}{\phi_S^*} \psi_V^*$$
$$\psi_W^{**} = \psi_W^*$$

Now they Agree...Done!

$$\sum_{V\backslash S} \psi_V^{**} = \sum_{V\backslash S} \frac{\phi_S^{**}}{\phi_S^*} \psi_V^* = \frac{\phi_S^{**}}{\phi_S^*} \sum_{V\backslash S} \psi_V^* = \phi_S^{**} = \sum_{W\backslash S} \psi_W^{**}$$

JTA with Evidence

- Example: if *evidence* is observed, say variable A=1

Initialize as before…

$$\psi_{AB} = p(A,B) \qquad \psi_{BC} = p(C \mid B) \qquad \phi_B = 1$$

Update with slice…

$$\phi_B^* = \sum_A \psi_{AB} \delta(A=1) = \sum_A p(A,B)\delta(A=1) = p(A=1,B)$$

$$\psi_{BC}^* = \frac{\phi_S^*}{\phi_S} \psi_{BC} = \frac{p(A=1,B)}{1} p(C \mid B) = p(A=1,B,C)$$

$$\psi_{AB}^* = \psi_{AB} = p(A=1,B)$$

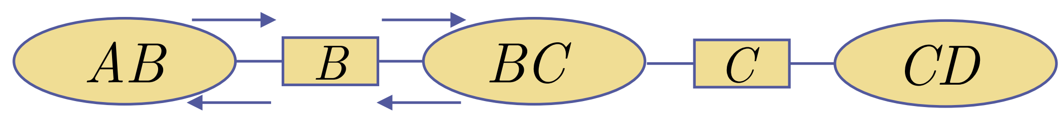
All ψ, ϕ become marginals *conditioned* on evidence

$$p(B,C \mid A=1) = \frac{\psi_{BC}^*}{\sum_{B,C} \psi_{BC}^*}$$

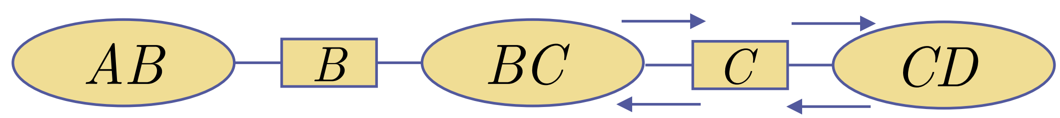
JTA with many cliques

•Problem: what if we have more than two cliques?

1) Update AB & BC



2) Update BC & CD



•Problem: AB has not heard about CD!
After BC updates, it will be inconsistent for AB

•Need to iterate the pairwise updates many times
•This will eventually converge to consistent marginals
•But, inefficient… can we do better?

JTA: Collect & Distribute

- Trees: recursive, no need to reiterate messages mindlessly!
- Send your message only after hearing from all neighbors…

initialize(DAG){ Pick root
Set all variables as: $\psi_{C_i} = p\left(x_i \mid \pi_i\right), \phi_S = 1$ **}**

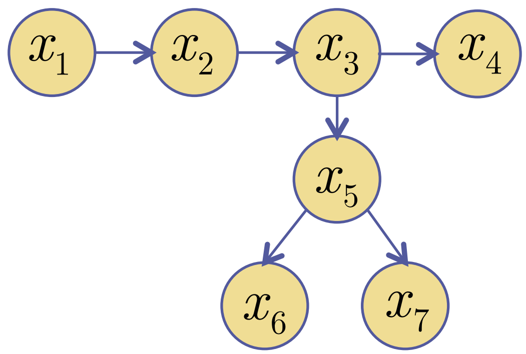
```
collectEvidence(node) {
   for each child of node {
      update(node,collectEvidence(child)); }
   return(node); }

distributeEvidence(node) {
   for each child of node {
      update(child,node);
      distributeEvidence(child); } }
```

normalize(DAG) { $p\left(X_C\right) = \frac{\psi_C}{\sum_{X_C} \psi_C}, p\left(X_S\right) = \frac{\phi_S}{\sum_{X_S} \phi_S}$ **}**

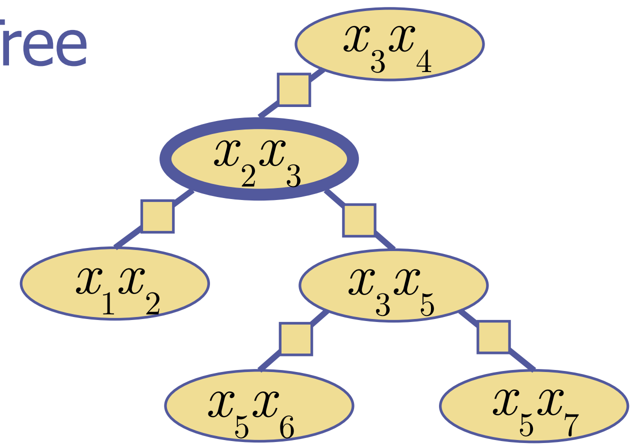
update(node ψ,evidence φ) { $\psi_C^* = \frac{\phi_S^*}{\sum_{C \setminus S} \psi_C} \psi_C$ **}**

Junction Tree Algorithm

- Convert Directed Graph to Junction Tree



$\rightarrow$



- *Initialize* separators to 1 (and Z=1) and set clique tables to the CPTs in the Directed Graph

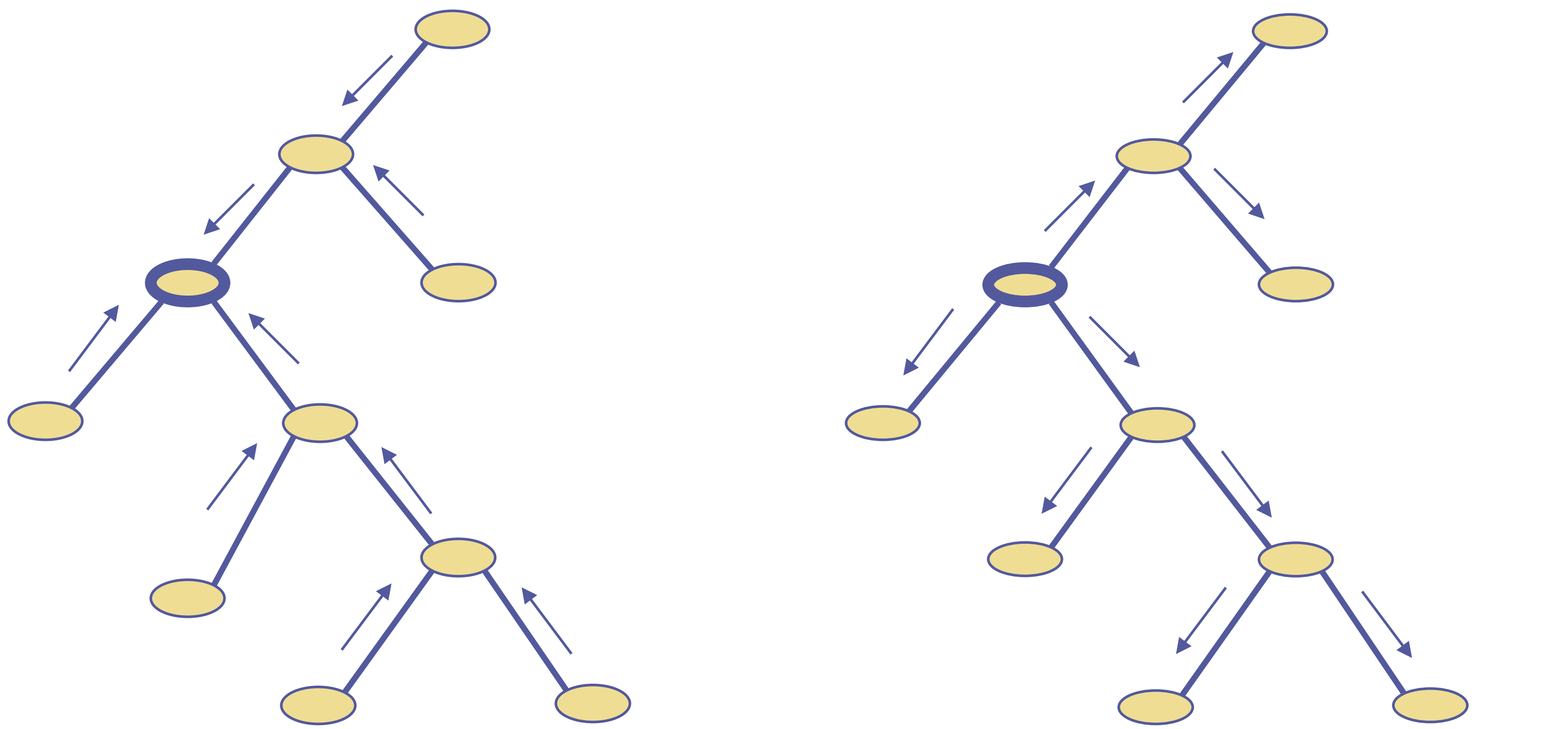
$$p(X) = p(x_1)\,p(x_2 \mid x_1)\,p(x_3 \mid x_2)\,p(x_4 \mid x_3)\,p(x_5 \mid x_3)\,p(x_6 \mid x_5)\,p(x_7 \mid x_5)$$

$$p(X) = \frac{1}{Z}\frac{\prod_C \psi(X_C)}{\prod_S \phi(X_S)}$$

$$= \frac{1}{1}\frac{p(x_1, x_2)\,p(x_3 \mid x_2)\,p(x_4 \mid x_3)\,p(x_5 \mid x_3)\,p(x_6 \mid x_5)\,p(x_7 \mid x_5)}{1 \times 1 \times 1 \times 1 \times 1}$$

Junction Tree Algorithm

•JTA: 1)*Initialize* 2)*Collect* 3)*Distribute* 4)*Normalize*



•Note: leaves do not change their ψ during *collect*
•Note: the first cliques *collect* changes are parents of leaves
•Note: root does not change its ψ during *distribute*

Algorithmic Complexity

- The 5 steps of JTA are all efficient:

OFFLINE

1) Moralization

Polynomial in # of nodes

2) Introduce Evidence (fixed or constant)

Polynomial in # of nodes (convert pdf to slices)

3) Triangulate (Tarjan & Yannakakis 1984)

Suboptimal=Polynomial, Optimal=NP

4) Construct Junction Tree (Kruskal)

Polynomial in # of cliques

ONLINE (for each query, new evidence, etc.)

5) Propagate Probabilities (Junction Tree Algorithm)

Polynomial (linear) in # of cliques, *Exponential* in Clique Cardinality

ArgMax Junction Tree Algorithm

- We can also use JTA for finding the max not the sum over the joint to get argmax of marginals & conditionals
- Say have some evidence: $p\left(X_F, \bar{X}_E\right) = p\left(x_1, \ldots, x_n, \bar{x}_{n+1}, \ldots, \bar{x}_N\right)$
- Most likely (highest p) X_F? $X_F^* = \arg\max_{X_F} p\left(X_F, \bar{X}_E\right)$
- What is most likely state of patient with fever & headache?

$$\begin{aligned} p_F^* &= \max_{x_2,x_3,x_4,x_5} p\left(x_1 = 1, x_2, x_3, x_4, x_5, x_6 = 1\right) \\ &= \max_{x_2} p\left(x_2 \mid x_1 = 1\right) p\left(x_1 = 1\right) \max_{x_3} p\left(x_3 \mid x_1 = 1\right) \\ &\quad \max_{x_4} p\left(x_4 \mid x_2\right) \max_{x_5} p\left(x_5 \mid x_3\right) p\left(x_6 = 1 \mid x_2, x_5\right) \end{aligned}$$

- Solution: update in JTA uses max instead of sum:

$$\phi_S^* = \max_{V \setminus S} \psi_V \qquad \psi_W^* = \frac{\phi_S^*}{\phi_S} \psi_W \qquad \psi_V^* = \psi_V$$

- Final potentials aren't marginals: $\psi\left(X_C\right) = \max_{U \setminus C} p\left(X\right)$
- Highest value in potential is most likely: $X_C^* = \arg\max_C \psi\left(X_C\right)$