

Machine Learning

4771

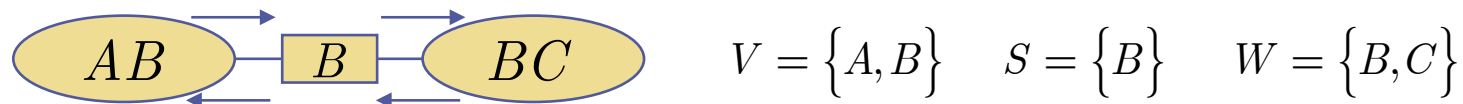
Instructor: Tony Jebara

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



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

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

$$\begin{aligned} \phi_S^* &= \sum_{V \setminus S} \psi_V \\ \psi_W^* &= \frac{\phi_S^*}{\phi_S} \psi_W \\ \psi_V^* &= \psi_V \end{aligned}$$

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

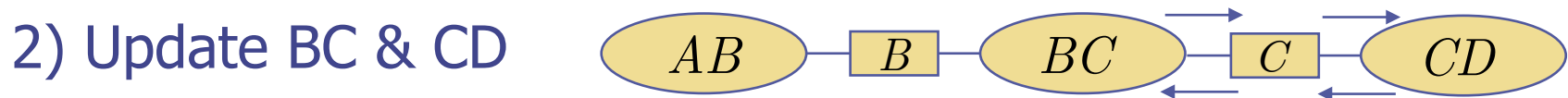
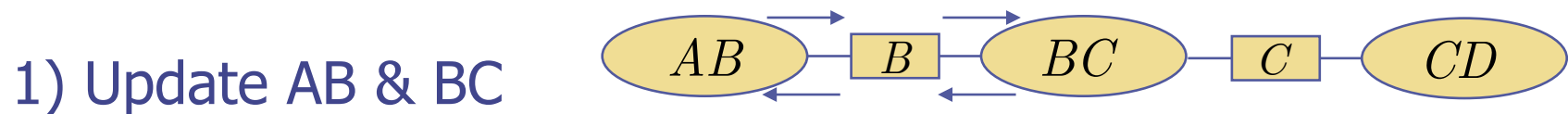
$$\begin{aligned} \phi_S^{**} &= \sum_{W \setminus S} \psi_W^* \\ \psi_V^{**} &= \frac{\phi_S^{**}}{\phi_S^*} \psi_V^* \\ \psi_W^{**} &= \psi_W^* \end{aligned}$$

**Now they
Agree...Done!**

$$\begin{aligned} \sum_{V \setminus S} \psi_V^{**} &= \sum_{V \setminus S} \frac{\phi_S^{**}}{\phi_S^*} \psi_V^* \\ &= \frac{\phi_S^{**}}{\phi_S^*} \sum_{V \setminus S} \psi_V^* \\ &= \phi_S^{**} = \sum_{W \setminus S} \psi_W^{**} \end{aligned}$$

JTA with many cliques

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



- 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

- Use tree recursion rather than iterate messages mindlessly!

initialize(DAG){ Pick root

Set all variables as: $\psi_{C_i} = p(x_i | \pi_i), \phi_S = 1$ }

collectEvidence(node) {

for each child of node {

update1(node, collectEvidence(child)); }

return(node); }

distributeEvidence(node) {

for each child of node {

update2(child, node);

distributeEvidence(child); }

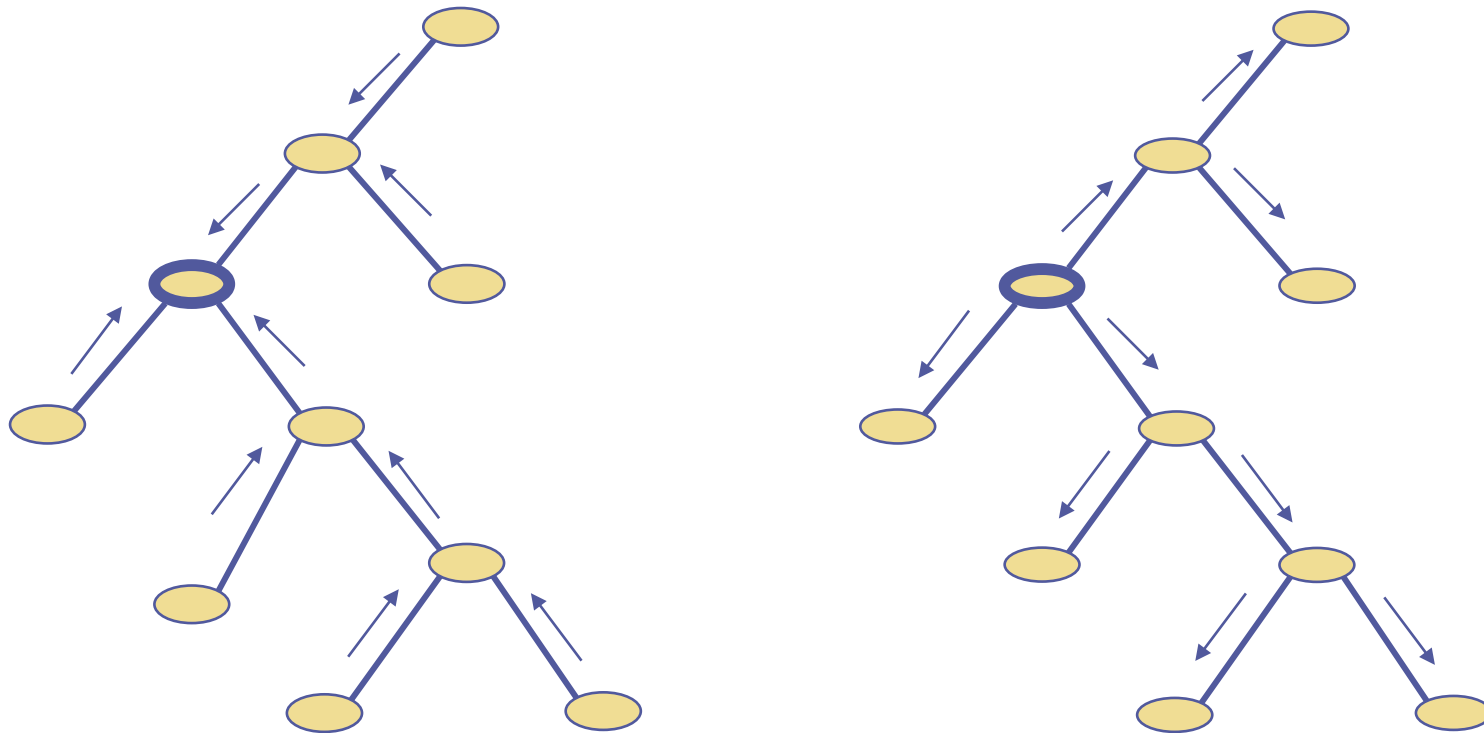
update1(node w, node v) { $\phi_{V \cap W}^* = \sum_{V \setminus (V \cap W)} \psi_V, \psi_W = \frac{\phi_{V \cap W}^*}{\phi_{V \cap W}} \psi_W$ }

update2(node w, node v) { $\phi_{V \cap W}^{} = \sum_{V \setminus (V \cap W)} \psi_V, \psi_W = \frac{\phi_{V \cap W}^{**}}{\phi_{V \cap W}^*} \psi_W$ }**

normalize() { $p(X_C) = \frac{1}{\sum_C \psi_C^{}} \psi_C^{**} \forall C, p(X_S) = \frac{1}{\sum_S \phi_S^{**}} \phi_S^{**} \forall S$ }**

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:

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)

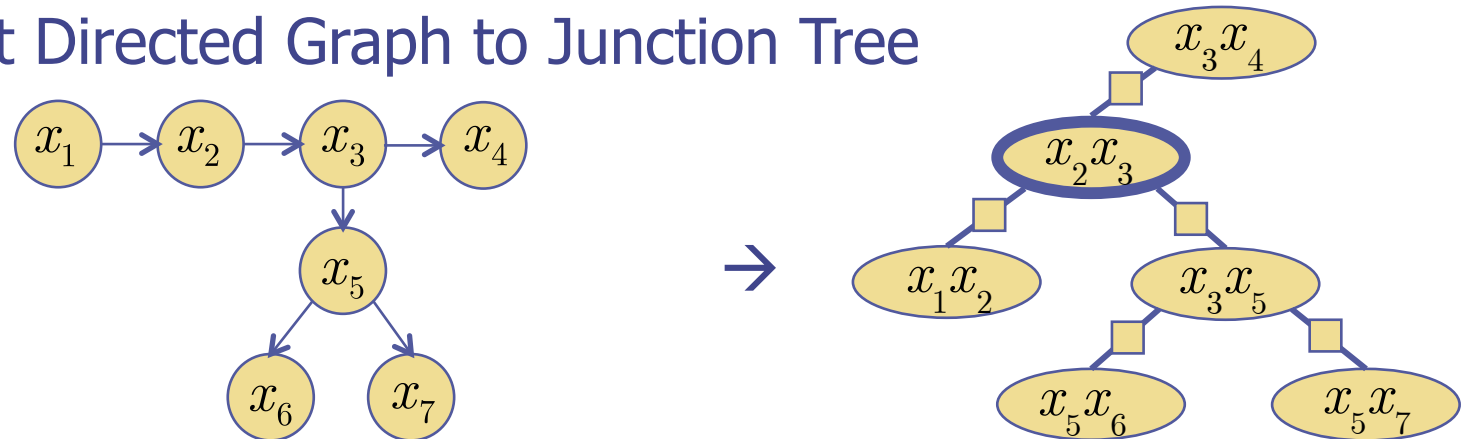
Polynomial in # of cliques

5) Junction Tree Algorithm (Init,Collect,Distribute,Normalize)

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

Junction Tree Algorithm

- Convert Directed Graph to Junction Tree



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

$$p(X) = p(x_1)p(x_2 | x_1)p(x_3 | x_2)p(x_4 | x_3)p(x_5 | x_3)p(x_6 | x_5)p(x_7 | x_5)$$

$$p(X) = \frac{1}{Z} \frac{\prod_C \psi(X_C)}{\prod_S \phi(X_S)} = \frac{1}{1} \frac{p(x_1, x_2)p(x_3 | x_2)p(x_4 | x_3)p(x_5 | x_3)p(x_6 | x_5)p(x_7 | x_5)}{1 \times 1 \times 1 \times 1 \times 1}$$

- Run *Collect, Distribute, Normalize*
- Get valid marginals from all ψ, ϕ tables

JTA with Extra Evidence

- If extra *evidence* is observed, must slice tables accordingly
- Example: $p(A, B, C, D) = \frac{1}{Z} \psi_{AB} \psi_{BC} \psi_{CD}$

$Z = 1$

$$\psi_{AB} = \begin{bmatrix} 8 & 4 \\ 3 & 1 \end{bmatrix} \begin{matrix} A=0 \\ A=1 \end{matrix} \quad \psi_{BC} = \begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix} \begin{matrix} B=0 \\ B=1 \end{matrix} \quad \psi_{CD} = \begin{bmatrix} 1 & 4 \\ 1 & 1 \end{bmatrix} \begin{matrix} C=0 \\ C=1 \end{matrix}$$

$B=0 \quad B=1$ $C=0 \quad C=1$ $D=0 \quad D=1$

- You are given *evidence*: $A=0$. Replace table with slices...

$$\psi_{AB} \rightarrow \begin{bmatrix} 8 & 4 \end{bmatrix} \quad \psi_{BC} = \begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix} \quad \psi_{CD} = \begin{bmatrix} 1 & 4 \\ 1 & 1 \end{bmatrix}$$

- JTA now gives ψ, ϕ as marginals *conditioned* on evidence

$$p(B | A = 0) = \frac{\psi_{AB}^{**}}{\sum_B \psi_{AB}^{**}} \quad p(B, C | A = 0) = \frac{\psi_{BC}^{**}}{\sum_{B,C} \psi_{BC}^{**}} \quad p(C, D | A = 0) = \frac{\psi_{CD}^{**}}{\sum_{C,D} \psi_{CD}^{**}}$$

- All denominators equal the new normalizer Z'

$$Z' = p(EVIDENCE) = \sum_B \psi_{AB}^{**} = \sum_{B,C} \psi_{BC}^{**} = \sum_{C,D} \psi_{CD}^{**}$$

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: $p(X_F, \bar{X}_E) = p(x_1, \dots, x_n, \bar{x}_{n+1}, \dots, \bar{x}_N)$
- Most likely (highest p) X_F ? $X_F^* = \arg \max_{X_F} p(X_F, \bar{X}_E)$
- What is most likely state of patient with fever & headache?

$$\begin{aligned}
 p_F^* &= \max_{x_2, x_3, x_4, x_5} p(x_1 = 1, x_2, x_3, x_4, x_5, x_6 = 1) \\
 &= \max_{x_2} p(x_2 | x_1 = 1) p(x_1 = 1) \max_{x_3} p(x_3 | x_1 = 1) \\
 &\quad \max_{x_4} p(x_4 | x_2) \max_{x_5} p(x_5 | x_3) p(x_6 = 1 | x_2, x_5)
 \end{aligned}$$
- Solution: replace sum with max inside JTA update code

$$\phi_{V \cap W}^* = \max_{V \setminus (V \cap W)} \psi_V, \quad \psi_W = \frac{\phi_{V \cap W}^*}{\phi_{V \cap W}} \psi_W \quad \phi_{V \cap W}^{**} = \max_{V \setminus (V \cap W)} \psi_V, \quad \psi_W = \frac{\phi_{V \cap W}^{**}}{\phi_{V \cap W}^*} \psi_W$$
- Final potentials are *max marginals*: $\psi^{**}(X_C) = \max_{U \setminus C} p(X)$
- Highest value in potential is most likely: $X_C^* = \arg \max_C \psi^{**}(X_C)$

ArgMax Junction Tree Algorithm

- Why do I need the ArgMax junction tree algorithm?
- Can't I just compute marginals using the Sum algorithm and then find the highest value in each marginal???
- No!! Here's a counter-example:

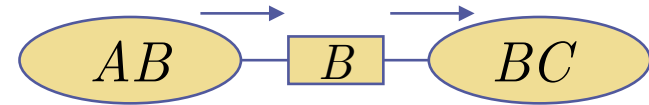
$$p(x_1, x_2) = \begin{array}{c} x_1 \quad x_1 \quad x_1 \\ A \quad B \quad C \\ x_2 = 0 \quad \left[\begin{array}{ccc} .14 & .05 & .27 \end{array} \right] \\ x_2 = 1 \quad \left[\begin{array}{ccc} .24 & .20 & .10 \end{array} \right] \end{array}$$

- Most likely is $x_1^* = C$ and $x_2^* = 0$
- But the sub-marginals $p(x_1)$ and $p(x_2)$ do not reveal this...

$$p(x_1) = \begin{array}{c} A \quad B \quad C \\ \left[\begin{array}{ccc} 0.38 & 0.25 & 0.37 \end{array} \right] \end{array} \quad p(x_2) = \begin{array}{c} x_2 = 0 \quad \left[\begin{array}{c} .46 \end{array} \right] \\ x_2 = 1 \quad \left[\begin{array}{c} .54 \end{array} \right] \end{array}$$

- The marginals would *falsely* imply that is $x_1^* = A$ and $x_2^* = 1$

Example



- Note that products are element-wise
- Let us send a regular JTA message from AB to BC

$$\psi_{AB} = \begin{bmatrix} 8 & 4 \\ 3 & 1 \end{bmatrix} \begin{matrix} A=0 \\ A=1 \\ B=0 & B=1 \end{matrix} \quad \phi_B = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{matrix} B=0 \\ B=1 \end{matrix} \quad \psi_{BC} = \begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix} \begin{matrix} B=0 \\ B=1 \\ C=0 & C=1 \end{matrix}$$

$$\phi_B^* = \sum_A \psi_{AB} = \sum_A \begin{bmatrix} 8 & 4 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 11 & 5 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 11 \\ 5 \end{bmatrix} \begin{matrix} B=0 \\ B=1 \end{matrix}$$

$$\psi_{BC}^* = \frac{\phi_B^*}{\phi_B} \psi_{BC} = \begin{bmatrix} 11 \\ 5 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 22 & 33 \\ 5 & 5 \end{bmatrix} \begin{matrix} B=0 \\ B=1 \\ C=0 & C=1 \end{matrix}$$

- If argmax JTA, just change the separator update to:

$$\phi_B^* = \max_A \psi_{AB} = \max_A \begin{bmatrix} 8 & 4 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 8 & 4 \\ 3 & 1 \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \end{bmatrix} \begin{matrix} B=0 \\ B=1 \end{matrix}$$