Machine Learning

4771

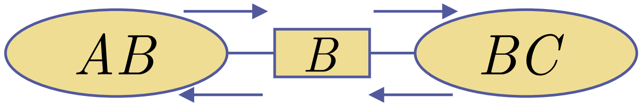
Instructor: Tony Jebara

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



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

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

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

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

Send message From W to V...

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

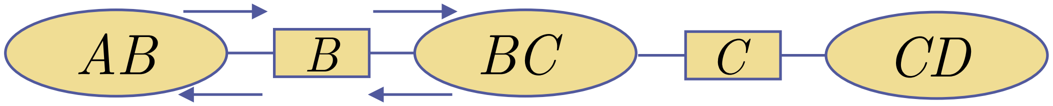
Now they Agree...Done!

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

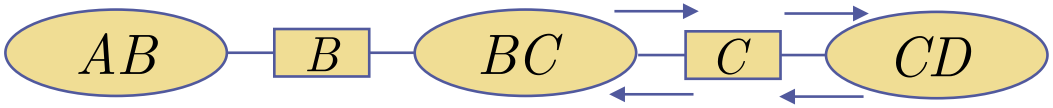
JTA with many cliques

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

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?

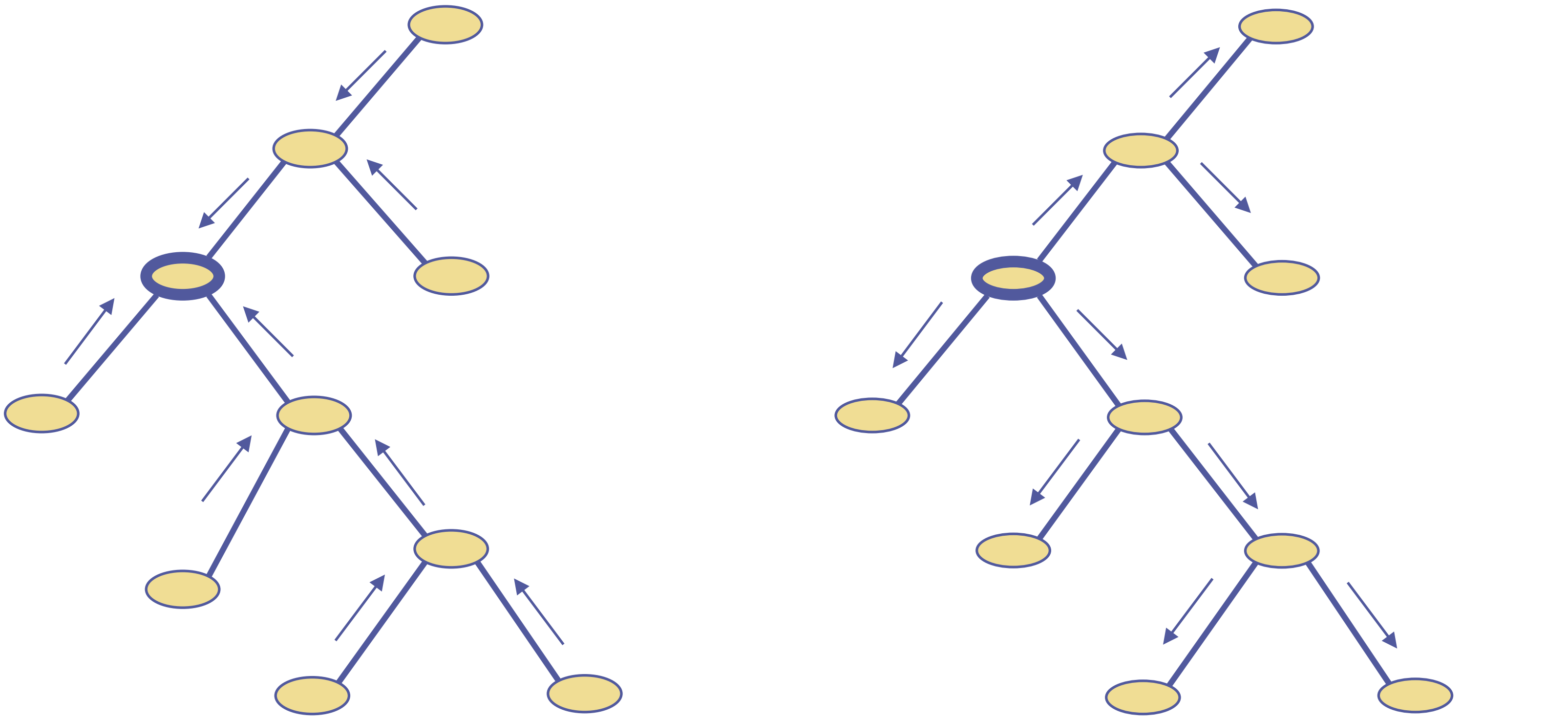
JTA: Collect & Distribute

- Use tree recursion rather than iterate messages mindlessly!

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

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)

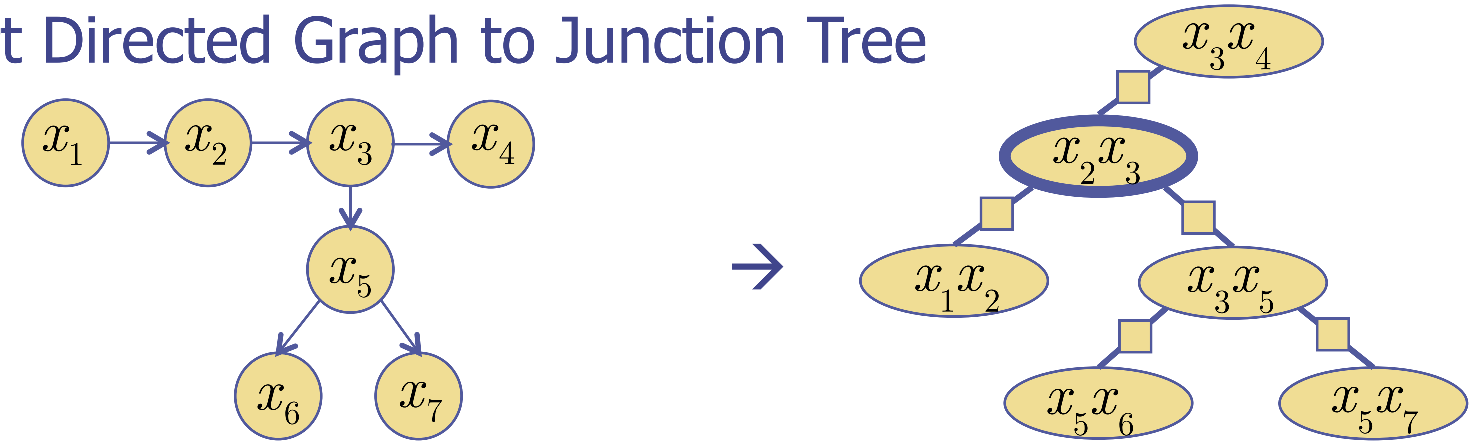
Polynomial in # of cliques

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

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

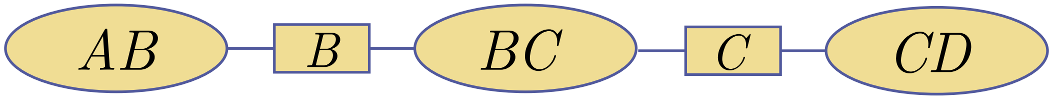
$$p(X) = \frac{1}{Z}\frac{\prod_C \psi(X_C)}{\prod_S \phi(X_S)} = \frac{1}{1}\frac{p(x_1,x_2)\,p(x_3 \mid x_2)\,p(x_4 \mid x_3)\,p(x_5 \mid x_3)\,p(x_6 \mid x_5)\,p(x_7 \mid x_5)}{1\times 1\times 1\times 1\times 1}$$

- 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: $p\left(A,B,C,D\right)=\frac{1}{Z}\psi_{AB}\psi_{BC}\psi_{CD}$

$Z=1$



$$\psi_{AB}=\begin{bmatrix}8 & 4\\3 & 1\end{bmatrix}\qquad\psi_{BC}=\begin{bmatrix}2 & 3\\1 & 1\end{bmatrix}\qquad\psi_{CD}=\begin{array}{c|c}\begin{bmatrix}1 & 4\\1 & 1\end{bmatrix} & \begin{array}{c}C=0\\C=1\end{array}\\ \begin{array}{cc}D=0 & D=1\end{array} & \end{array}$$

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

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

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

$$p\left(B\mid A=0\right)=\frac{\psi_{AB}^{**}}{\sum_{B}\psi_{AB}^{**}}\qquad p\left(B,C\mid A=0\right)=\frac{\psi_{BC}^{**}}{\sum_{B,C}\psi_{BC}^{**}}\qquad p\left(C,D\mid A=0\right)=\frac{\psi_{CD}^{**}}{\sum_{C,D}\psi_{CD}^{**}}$$

- All denominators equal the new normalizer Z'

$$Z'=p\left(A=0\right)=\sum_{B}\psi_{AB}^{**}=\sum_{B,C}\psi_{BC}^{**}=\sum_{C,D}\psi_{CD}^{**}$$

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: $p\left(X_F,\bar{X}_E\right)=p\left(x_1,\ldots,x_n,\bar{x}_{n+1},\ldots,\bar{x}_N\right)$
- 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: $X_C^*=\arg\max_C \psi\left(X_C\right)$

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:

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

- 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…

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

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