# Lecture 11: February 23, 2015
# cs 573: Probabilistic Reasoning

Professor Nevatia

Spring 2015

# Admin

- Assignment # 3 due today; #4 to be posted today, due March 4
- Exam 1
  - March 2, class period, here (except for remote students)
  - Closed book, Closed Notes
    - Will not need to memorize long formulas
    - Calculators will not be needed
  - Style: similar to assignments + qualitative ("theory") questions
  - Content: defined by what is covered in class, including parts of this week's classes
  - Representations: Chapters 2-4 except for sections 3.4.3, 4.4.2, 4.6.2; chapter 5, focus on 5.4 (exclude 5.4.4)
  - Inference: Chapter 9, excluding 9.6; Chapter 10; Chapter 11: 11.3 (except 11.3.4 and 11.3.7)

# Review

- Last lecture:
  - Sum-Product Clique Tree Algorithm
    - Convert graphs to clique tree
    - Sum-product clique-tree calibration algorithm
- Today's objective
  - Clique-tree Belief Propagation Algorithm
  - Loopy Belief Propagation (LBP)

# Proof of Theorem 10.3

- For leaf cliques, follows from the definition itself
- For clique $\mathbf{C}_i$ that is not a leaf:

$$\sum_{\mathcal{V}_{\prec(i\to j)}} \prod_{\phi\in\mathcal{F}_{\prec(i\to j)}} \phi. \tag{10.3}$$

Let $i_1,\ldots,i_m$ be the neighboring cliques of $\boldsymbol{C}_i$ other than $\boldsymbol{C}_j$. It follows immediately from proposition 10.2 that $\mathcal{V}_{\prec(i\to j)}$ is the disjoint union of $\mathcal{V}_{\prec(i_k\to i)}$ for $k=1,\ldots,m$ and the variables $\boldsymbol{Y}_i$ eliminated at $\boldsymbol{C}_i$ itself. Similarly, $\mathcal{F}_{\prec(i\to j)}$ is the disjoint union of the $\mathcal{F}_{\prec(i_k\to i)}$ and the factors $\mathcal{F}_i$ from which $\psi_i$ was computed. Thus equation (10.3) is equal to

$$\sum_{\boldsymbol{Y}_i} \sum_{\mathcal{V}_{\prec(i_1\to i)}} \cdots \sum_{\mathcal{V}_{\prec(i_m\to i)}} \left(\prod_{\phi\in\mathcal{F}_{\prec(i_1\to i)}} \phi\right)\cdot\cdots\cdot\left(\prod_{\phi\in\mathcal{F}_{\prec(i_m\to i)}} \phi\right)\cdot\left(\prod_{\phi\in\mathcal{F}_i} \phi\right). \tag{10.4}$$

As we just showed, for each $k$, none of the variables in $\mathcal{V}_{\prec(i_k\to i)}$ appear in any of the other factors. Thus, we can use equation (9.6) and push in the summation over $\mathcal{V}_{\prec(i_k\to i)}$ in equation (10.4), and obtain:

$$\sum_{\boldsymbol{Y}_i}\left(\prod_{\phi\in\mathcal{F}_i} \phi\right)\cdot \sum_{\mathcal{V}_{\prec(i_1\to i)}}\left(\prod_{\phi\in\mathcal{F}_{\prec(i_1\to i)}} \phi\right)\cdot\cdots\cdot \sum_{\mathcal{V}_{\prec(i_m\to i)}}\left(\prod_{\phi\in\mathcal{F}_{\prec(i_m\to i)}} \phi\right). \tag{10.5}$$

Using the inductive hypothesis and the definition of $\psi_i$, this expression is equal to

$$\sum_{\boldsymbol{Y}_i} \psi_i\cdot\delta_{i_1\to i}\cdot\cdots\cdot\delta_{i_m\to i}, \tag{10.6}$$

which is precisely the operation used to compute the message $\delta_{i\to j}$. ∎

# Calibrated Tree as Distribution

- At convergence of the clique tree calibration algorithm, we have:

$$\beta_i = \psi_i \cdot \prod_{k \in \mathrm{Nb}_i} \delta_{k \to i}. \qquad \text{(by definition)}$$

- Rewrite message over sepset as:

$$\begin{aligned}
\mu_{i,j}(\boldsymbol{S}_{i,j}) &= \sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \beta_i(\boldsymbol{C}_i) \\
&= \sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \psi_i \cdot \prod_{k \in \mathrm{Nb}_i} \delta_{k \to i} \\
&= \sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \psi_i \cdot \delta_{j \to i} \prod_{k \in (\mathrm{Nb}_i - \{j\})} \delta_{k \to i} \\
&= \delta_{j \to i} \sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \psi_i \cdot \prod_{k \in (\mathrm{Nb}_i - \{j\})} \delta_{k \to i} \\
&= \delta_{j \to i} \delta_{i \to j},
\end{aligned}$$

# Calibrated Tree as Distribution

- At convergence of the clique tree calibration algorithm, we have:

$$\tilde{P}_\Phi(\mathcal{X}) = \frac{\prod_{i\in\mathcal{V}_\mathcal{T}} \beta_i(\boldsymbol{C}_i)}{\prod_{(i-j)\in\mathcal{E}_\mathcal{T}} \mu_{i,j}(\boldsymbol{S}_{i,j})}.$$

- Proof: Numerator can be written as (from definition)

$$\prod_{i\in\mathcal{V}_\mathcal{T}} \psi_i(\boldsymbol{C}_i) \prod_{k\in\mathrm{Nb}_i} \delta_{k\to i}.$$

Denominator can be written as (using eq 10.9, see previous slide)

$$\prod_{(i-j)\in\mathcal{E}_\mathcal{T}} \delta_{i\to j}\delta_{j\to i}.$$

Each δ message appears exactly once in num and denom each, so they cancel , and

$$\prod_{i\in\mathcal{V}_\mathcal{T}} \psi_i(\boldsymbol{C}_i) = \tilde{P}_\Phi.$$

# Clique Tree Measure

- Define *measure* induced by a calibrated tree, T, as:

$$Q_{\mathcal{T}} = \frac{\prod_{i \in \mathcal{V}_{\mathcal{T}}} \beta_i(C_i)}{\prod_{(i-j) \in \mathcal{E}_{\mathcal{T}}} \mu_{i,j}(S_{i,j})},$$

*where*

$$\mu_{i,j} = \sum_{C_i - S_{i,j}} \beta_i(C_i) = \sum_{C_j - S_{i,j}} \beta_j(C_j).$$

- $Q_T$ defines its own distribution function over the variables in T
- Thm 10.4

*Let $\mathcal{T}$ be a clique tree over $\Phi$, and let $\beta_i(C_i)$ be a set of calibrated potentials for $\mathcal{T}$. Then, $\tilde{P}_\Phi(\mathcal{X}) \propto Q_{\mathcal{T}}$ if and only if, for each $i \in \mathcal{V}_{\mathcal{T}}$, we have that $\beta_i(C_i) \propto \tilde{P}_\Phi(C_i)$.*

- Thm says that if we use clique tree as the representation of a distribution, clique marginals can be read-off directly
  - **initial factored representation does not include influence of other variables in the network**

# Belief Update

- An alternative, but equivalent, message passing scheme
- In previous algorithm, suppose a message is first passed from $\mathbf{C}_j$ to $\mathbf{C}_i$ ; then we wait until $\mathbf{C}_i$ gets messages from all its neighbors and is ready to send message to $\mathbf{C}_j$ .
- At this point $\mathbf{C}_i$ has all the info to compute its own potential;

$$\beta_i = \psi_i \cdot \prod_{k \in \mathrm{Nb}_i} \delta_{k \to i}.$$

- However, this potential is not used in passing message to $\mathbf{C}_j$ ; instead we must exclude the original message from $\mathbf{C}_j$ to $\mathbf{C}_i$ so we don't "double count".
- Alternately, we can just use $\beta_i$ from above but divide by $\delta_{j \to i}$ ; no need to maintain original factors
- Used in the HUGIN algorithm (a commercial package)
- Note order of updates doesn't matter; even multiple passes are OK (they reduce to passing a factor with all values equal to 1)

# Factor Division

$$\psi(\boldsymbol{X}, \boldsymbol{Y}) = \frac{\phi_1(\boldsymbol{X}, \boldsymbol{Y})}{\phi_2(\boldsymbol{Y})},$$

0/0 =0 by defn

| | | |
|---|---|---|
| $a^1$ | $b^1$ | 0.5 |
| $a^1$ | $b^2$ | 0.2 |
| $a^2$ | $b^1$ | 0 |
| $a^2$ | $b^2$ | 0 |
| $a^3$ | $b^1$ | 0.3 |
| $a^3$ | $b^2$ | 0.45 |

| | |
|---|---|
| $a^1$ | 0.8 |
| $a^2$ | 0 |
| $a^3$ | 0.6 |

$\Rightarrow$

| | | |
|---|---|---|
| $a^1$ | $b^1$ | 0.625 |
| $a^1$ | $b^2$ | 0.25 |
| $a^2$ | $b^1$ | 0 |
| $a^2$ | $b^2$ | 0 |
| $a^3$ | $b^1$ | 0.5 |
| $a^3$ | $b^2$ | 0.75 |

# Calibration using Belief Update

**Algorithm 10.3 Calibration using belief propagation in clique tree**

**Procedure** CTree-BU-Calibrate (
  $\Phi$, // Set of factors
  $\mathcal{T}$ // Clique tree over $\Phi$
)
  Initialize-CTree
  **while** exists an uninformed clique in $\mathcal{T}$
    Select $(i\text{–}j) \in \mathcal{E}_{\mathcal{T}}$
    BU-Message$(i, j)$
  **return** $\{\beta_i\}$

**Procedure** Initialize-CTree (
)
  **for** each clique $C_i$
    $\beta_i \leftarrow \prod_{\phi \,:\, \alpha(\phi)=i} \phi$
  **for** each edge $(i\text{–}j) \in \mathcal{E}_{\mathcal{T}}$
    $\mu_{i,j} \leftarrow \mathbf{1}$

**Procedure** BU-Message (
  i, // sending clique
  j // receiving clique
)
  $\sigma_{i \to j} \leftarrow \sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \beta_i$
    // marginalize the clique over the sepset
  $\beta_j \leftarrow \beta_j \cdot \frac{\sigma_{i \to j}}{\mu_{i,j}}$
  $\mu_{i,j} \leftarrow \sigma_{i \to j}$

# Including Evidence

- As in VE, we can zero out terms in factors that are not consistent with the evidence
- In clique tree algorithms, we can also easily incorporate the evidence later. This allows incremental inference.
- $P_\phi\tilde{}(X) = \Pi_{\varphi \varepsilon \phi}\ \varphi$
- With evidence, $P_\phi'\tilde{}(X) = P_\phi\tilde{}(X, Z{=}z) = \boldsymbol{1}\{Z = z\}\ .\ \Pi_{\varphi \varepsilon \phi}\ \varphi$
- We can compute $Q_T$ first (without considering evidence) and then multiply by the evidence indicator function

$$\tilde{P}'_\Phi(\mathcal{X}) = \boldsymbol{1}\{Z = z\} \cdot \frac{\prod_{i \in \mathcal{V}_\mathcal{T}} \beta_i(\boldsymbol{C}_i)}{\prod_{(i-j) \in \mathcal{E}_\mathcal{T}} \mu_{i,j}(\boldsymbol{S}_{i,j})}.$$

- To calibrate the entire clique tree with new evidence
  - Update any one clique, say $\mathbf{C}_i$ , that contains Z
  - Pass message with updated potential to other cliques; only one pass is needed
- Retracting evidence is not easy; we must store prior to conditioning

# Queries outside a Clique

- Variables may be in more than one clique.
- Find a sub-tree that contains all the variables
- Perform VE on it

**Algorithm 10.4 Out-of-clique inference in clique tree**

**Procedure** CTree-Query (
  $\mathcal{T}$, // Clique tree over $\Phi$
  $\{\beta_i\}, \{\mu_{i,j}\}$, // Calibrated clique and sepset beliefs for $\mathcal{T}$
  $\boldsymbol{Y}$ // A query
)

Let $\mathcal{T}'$ be a subtree of $\mathcal{T}$ such that $\boldsymbol{Y} \subseteq \mathit{Scope}[\mathcal{T}']$
Select a clique $r \in \mathcal{V}_{\mathcal{T}'}$ to be the root
$\Phi \leftarrow \beta_r$
**for** each $i \in \mathcal{V}'_{\mathcal{T}}$
  $\phi \leftarrow \frac{\beta_i}{\mu_{i,p_r(i)}}$
  $\Phi \leftarrow \Phi \cup \{\phi\}$
$\boldsymbol{Z} \leftarrow \mathit{Scope}[\mathcal{T}'] - \boldsymbol{Y}$
Let $\prec$ be some ordering over $\boldsymbol{Z}$
**return** Sum-Product-VE($\Phi, \boldsymbol{Z}, \prec$)

# VE vs Clique Tree Algorithms

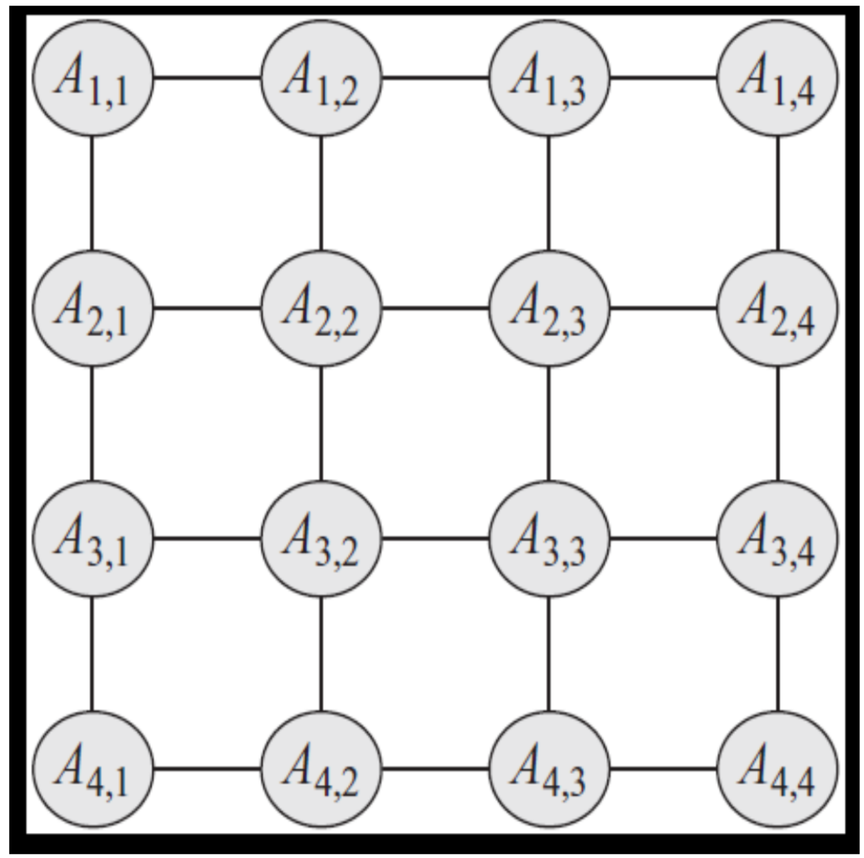
- Clique tree algorithms give distribution over all cliques in a single computation. VE would require multiple iterations.
- Clique tree: harder to take advantage of specific query, *e.g*. a particular evidence variable, as clique structure is pre-determined

# More Issues in Probabilistic Inference

- Clique-tree sum-product and belief propagation algorithms apply to arbitrarily complex BNs and MNs; still many difficulties remain
  - Nonetheless, exact inference remains NP-hard in worst case
  - Example: loops in graphs may result in an exponential number of cliques
- CPTs may be large for multi-valued variables with many parents
- In temporal reasoning, number of variables can grow large, though the number of parameters may remain small
- Continuous variables:
  - Large CPTs if we discretize
  - In some special cases (Gaussian distributions), closed form solutions can be obtained
- Need for approximate solutions
- Next: Loopy Belief Propagation

# Approximate Inference in Graphical Models

- Exact inference is NP-hard, may not be practical for large networks
  - Some factors may become exponentially large
- Consider a Grid MRF
- Can we convert to a clique tree
  - Size of maximal clique?
- Need for approximation



# Next Class

- Read sections 11.3, all of chapter 7 of the KF book