Super Nodes and Junction Trees

Presented by: Mehran Kazemi
Outline

- Super Nodes
- Variable Elimination (VE)
- From VE to Junction Trees (Jtree)
- Calculating marginal probabilities in Jtrees
Super Nodes

A — C — E — G — I — K
B — D — F — H — J — L
Super Nodes

A - C - E - G - I - K
B - D - F - H - J - L
AB - CD - EF - GH - IJ - KL
Super Nodes

ABCDEFGHJKLM
Variable Elimination

- We focus on calculating $Z$
- If we know how to calculate $Z$ for a network, we can calculate all marginal probabilities.
- $P(C=\text{true} \mid G=\text{false}) = \frac{Z(\text{Network} \mid C=\text{true}, G=\text{false})}{Z(\text{Network} \mid G=\text{false})}$
Variable Elimination

- $\phi_1(C)$
- $\phi_2(C, D)$
- $\phi_3(D, I, G)$
- $\phi_4(S, I)$
- $\phi_5(H, G, J)$
- $\phi_6(G, L)$
- $\phi_7(S, L, J)$
Variable Elimination (inference)

\[ Z = \sum_J \sum_L \sum_S \sum_G \sum_H \sum_I \sum_D \sum_C \]
\[ \phi_1(C) \phi_2(C, D) \phi_3(D, I, G) \phi_4(S, I) \]
\[ \phi_5(H, G, J) \phi_6(G, L) \phi_7(S, L, J) \]

Elimination Order:
\[ \psi = <C, D, I, H, G, S, L, J> \]

\[ Z = \]
\[ \sum_J \sum_L \sum_S \phi_7(S, L, J) \sum_G \phi_6(G, L) \sum_H \phi_5(H, G, J) \]
\[ \sum_I \phi_4(S, I) \sum_D \phi_3(D, I, G) \sum_C \phi_1(C) \phi_2(C, D) \]
Variable Elimination

\[ Z = \sum_J \sum_L \sum_S \phi_7(S, L, J) \sum_G \phi_6(G, L) \sum_H \phi_5(H, G, J) \]
\[ \sum_I \phi_4(S, I) \sum_D \phi_3(D, I, G) \sum_C \phi_1(C) \phi_2(C, D) \]

\[ \phi_1(C) \quad \phi_2(C, D) \quad \lambda_1(C, D) = \phi_1(C)\phi_2(C, D) \quad \tau_1(D) = \sum_D \lambda(C, D) \]

<table>
<thead>
<tr>
<th>C</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>2</td>
</tr>
<tr>
<td>F</td>
<td>1.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>0.5</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>1</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>2</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>1.2</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>2.4</td>
</tr>
</tbody>
</table>

\[ Z = \sum_J \sum_L \sum_S \phi_7(S, L, J) \sum_G \phi_6(G, L) \sum_H \phi_5(H, G, J) \]
\[ \sum_I \phi_4(S, I) \sum_D \phi_3(D, I, G) \tau_1(D) \]
Variable Elimination (inference)

\[ Z = \sum_J \sum_L \sum_S \phi_7(S, L, J) \sum_G \phi_6(G, L) \sum_H \phi_5(H, G, J) \]

\[ \sum_I \phi_4(S, I) \sum_D \phi_3(D, I, G) \tau_1(D) \]

\[ Z = \sum_J \sum_L \sum_S \phi_7(S, L, J) \sum_G \phi_6(G, L) \sum_H \phi_5(H, G, J) \]

\[ \sum_I \phi_4(S, I) \tau_2(G, I) \]

\[ Z = \sum_J \sum_L \sum_S \phi_7(S, L, J) \sum_G \phi_6(G, L) \tau_3(S, G) \sum_H \phi_5(H, G, J) \]

\[ Z = \sum_J \sum_L \sum_S \phi_7(S, L, J) \sum_G \phi_6(G, L) \tau_3(S, G) \tau_4(G, J) \]

\[ \ldots \]

\[ Z = \sum_J \tau_{n-1}(J) \]
Variable Elimination (decoding)

- \text{Argmax}_J \text{Argmax}_L \ldots \text{Argmax}_C \phi_1(C)\phi_2(C,D)\phi_3(D,I,G)\phi_4(S,I)\phi_5(H,G,J)\phi_6(G,L)\phi_7(S,L,J)

- Elimination Order:
  \[ \psi = \langle C, D, I, H, G, S, L, J \rangle \]

- \text{Argmax}_J \ldots \text{Argmax}_C \phi_1(C)\phi_2(C,D)
Variable Elimination (decoding)

\[ \text{Argmax}_J \cdots \text{Argmax}_C \phi_1(C) \phi_2(C,D) \]

\[ \phi_1(C) \quad \phi_2(C,D) \quad \lambda_1(C,D) = \phi_1(C) \phi_2(C,D) \quad \tau_1(D) = \max_C \lambda(C,D) \]

- \( \text{Argmax}_J \cdots \tau_1(D) \)

<table>
<thead>
<tr>
<th>C</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>2</td>
</tr>
<tr>
<td>F</td>
<td>1.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>0.5</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>1</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>2</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>1.2</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>2.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>1.2</td>
</tr>
<tr>
<td>F</td>
<td>2.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D</th>
<th>Val(C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
Time Complexity (Assuming binary variables)

- Let $\phi_1, \phi_2, ..., \phi_m$ be potentials containing a variable $V$.
- Let $\tau(C_1, C_2, ..., C_w) = \sum_V \phi_1 \phi_2 ... \phi_m$
- $\tau(C_1, C_2, ..., C_w)$ can be calculated in $O(2^w)$
- Let $w_1, w_2, ... w_n$ correspond to the number of variables in $\tau_1, \tau_2, ... \tau_n$ given a specific elimination order $\psi$.
- Let $\omega = \max(w_1, w_2, ..., w_n)$
- Variable elimination with elimination order $\psi$ is $O(n2^\omega)$
- $\omega$ is called the width of $\psi$. 
\[ \omega \text{ depends on the elimination order} \]

\[ Z = \sum_C \sum_D \sum_I \sum_G \sum_S \sum_L \sum_J \sum_H \]
\[ \phi_1(C) \phi_2(C, D) \phi_3(D, I, G) \phi_4(S, I) \]
\[ \phi_5(H, G, J) \phi_6(G, L) \phi_7(S, L, J) \]

**Elimination Order:**

\[ \psi = \langle G, \ldots \rangle \]

\[ Z = \sum \ldots \sum_G \phi_3(D, I, G) \phi_5(H, G, J) \phi_6(G, L) \]

\[ Z = \sum \ldots \tau(D, I, H, J, L) \]
Time Complexity

- Let \( \{\psi_1, \psi_2, ..., \psi_t\} \) represent all possible elimination orders, and \( \{\omega(\psi_1), \omega(\psi_2), ..., \omega(\psi_t)\} \) represent the widths of these elimination orders.
- Define treewidth = \( \min_{\psi \in \{\psi_1, \psi_2, ..., \psi_t\}} \omega(\psi) \)
- Variable elimination is then \( O(n2^{\text{treewidth}}) \)
- Finding a \( \psi \) with \( \omega(\psi) = \text{treewidth} \) is NP-Hard.
From VE to Junction Trees (Jtrees)

- Variable Elimination is query sensitive: we must specify the query variable in advance. Each time we run a new query, we must re-run the entire algorithm.
- The junction tree algorithms generalizes VE to avoid this; they compile the UGM into a data structure which supports simultaneous execution of queries.
From VE to Junction Trees (Jtrees)

\[ Z = \sum \ldots \sum_D \phi_3(D, I, G) \sum_C \phi_1(C)\phi_2(C, D) \]
From VE to Junction Trees (Jtrees)

\[ Z = \sum \ldots \sum_D \phi_3(D, I, G) \sum_C \phi_1(C)\phi_2(C, D) \]
From VE to Junction Trees (Jtrees)

\[ Z = \sum \ldots \sum_{D} \phi_3(D, I, G) \tau_1(D) \]

\[ \phi_1 \ast \phi_2 \]

\[ \tau_1(D) \]
From VE to Junction Trees (Jtrees)

\[ Z = \sum \ldots \sum_D \phi_3(D, I, G) \tau_1(D) \]
From VE to Junction Trees (Jtrees)

\[ Z = \sum ... \tau_2(I, G) \]

\[ \text{C,D} \quad \tau_1(D) \quad \text{D,I,G} \quad \tau_2(I, G) \]

\[ \phi_1 \ast \phi_2 \]

\[ \phi_3 \]
From VE to Junction Trees (Jtrees)

\[ Z = \sum_J \sum_L \sum_S \phi_7(S,L,J) \sum_G \phi_6(G,L) \sum_H \phi_5(H,G,J) \sum_I \phi_4(S,I) \tau_2(G,I) \]

\[ \tau_1(D) \quad \tau_2(I,G) \]

\[ \phi_1 \ast \phi_2 \quad \phi_3 \]
From VE to Junction Trees (Jtrees)

\[ Z = \sum_J \sum_L \sum_S \phi_7(S, L, J) \sum_G \phi_6(G, L) \sum_H \phi_5(H, G, J) \sum_I \phi_4(S, I) \tau_2(G, I) \]
From VE to Junction Trees (Jtrees)

\[ Z = \sum_j \sum_L \sum_S \phi_7(S, L, J) \sum_G \phi_6(G, L) \tau_3(S, G) \sum_H \phi_5(H, G, J) \]
From VE to Junction Trees (Jtrees)

\[ Z = \sum_j \sum_L \sum_S \phi_7(S, L, J) \sum_G \phi_6(G, L) \tau_3(S, G) \sum_H \phi_5(H, G, J) \]
\[ Z = \sum_j \sum_L \sum_S \phi_7(S, L, J) \sum_G \phi_6(G, L) \tau_3(S, G) \tau_4(G, J) \]
\[ Z = \sum_{j} \sum_{l} \sum_{s} \phi_7(S, L, J) \sum_{g} \phi_6(G, L) \tau_3(S, G) \tau_4(G, J) \]
From VE to Junction Trees (Jtrees)

\[ Z = \sum_j \sum_L \sum_S \phi_7(S, L, J) \tau_5(L, S, J) \]
From VE to Junction Trees (Jtrees)

\[ Z = \sum_j \sum_L \sum_S \phi_7(S, L, J) \tau_5(L, S, J) \]
From VE to Junction Trees (Jtrees)

\[ Z = \sum_j \sum_l \tau_6(L,J) \]
From VE to Junction Trees (Jtrees)

\[ Z = \sum_j \sum_L \tau_6(L, J) \]
From VE to Junction Trees (Jtrees)

\[ Z = \sum_j \tau_7(j) \]
From VE to Junction Trees (Jtrees)

\[ Z = \sum_j \tau_7(j) \]

\[ \phi_1 \times \phi_2 \]  
\[ \tau_1(D) \]  
\[ \phi_3 \]  
\[ \tau_2(I, G) \]  
\[ \phi_4 \]  
\[ \tau_3(S, G) \]  
\[ \phi_6 \]  
\[ \tau_4(G, J) \]  
\[ \phi_5 \]  
\[ \tau_5(L, S, J) \]  
\[ \phi_7 \]  
\[ \tau_6(L, J) \]  
\[ \phi_7 \]  
\[ \tau_7(J) \]
From VE to Junction Trees (Jtrees)

\[ Z = \text{Constant} \]

\[ \phi_1 \ast \phi_2 \]

\[ \phi_3 \]

\[ \phi_4 \]

\[ \phi_5 \]

\[ \phi_6 \]

\[ \phi_7 \]
Inference in Jtrees

\[ C, D \xrightarrow{\tau_1(D)} D, I, G \xrightarrow{\tau_2(I, G)} G, S, I \xrightarrow{\tau_3(S, G)} G, L, S, J \xrightarrow{\tau_4(G, J)} H, G, J \xrightarrow{\tau_5(L, S, J)} S, L, J \xrightarrow{\tau_6(L, J)} J \xrightarrow{\tau_7(J)} \]

\[ \phi_1 \ast \phi_2 \]

\[ \phi_3 \]

\[ \phi_4 \]

\[ \phi_5 \]

\[ \phi_6 \]

\[ \phi_7 \]
Inference in Jtrees

\[
\begin{align*}
C,D & \quad \phi_1 \ast \phi_2 \\
D,I,G & \quad \tau_1(D) \quad \phi_3 \\
G,S,I & \quad \tau_2(I,G) \quad \phi_4 \\
G,L,S,J & \quad \tau_3(S,G) \quad \phi_6 \\
H,G,J & \quad \tau_4(G,J) \quad \phi_5 \\
L,J & \quad \tau_5(L,S,J) \quad \phi_7 \\
S,L,J & \quad \tau_6(L,J) \\
J & \quad \tau_7(J)
\end{align*}
\]
The joint probability of the variables in each cluster is proportional to the product of its potentials and its incoming messages.
Inference in Jtrees

\[ P(G, S, I) \propto \phi_4 \tau_2(I, G) \tau_3'(S, G) \]
Inference in Jtrees

\[ P(G, S, I) \propto \phi_4(\sum_D \phi_3 \tau_1(D)) \tau_3'(S, G) \]
Inference in Jtrees

\[ P(G, S, I) \propto \phi_4(\sum_D \phi_3 \sum_C \phi_1 \phi_2) \tau_3'(S, G) \]
Inference in Jtrees

\[ P(G, S, I) \propto \phi_4(\Sigma_D \phi_3 \Sigma_C \phi_1 \phi_2)(\Sigma_{L,J} \phi_6 \tau_4(G,J)\tau_5'(L,S,J)) \]
Inference in Jtrees

\[ P(G, S, I) \propto \phi_4(\sum_D \phi_3 \sum_C \phi_1 \phi_2)(\sum_{L, J} \phi_6(\sum_H \phi_5)\tau'_5(L, S, J)) \]
Inference in Jtrees

\[ P(G, S, I) \propto \phi_4(\sum_D \phi_3 \sum_C \phi_1 \phi_2)(\sum_{L,J} \phi_6(\sum_H \phi_5)\phi_7 \tau'_6(L,J)) \]
Inference in Jtrees

\[ P(G, S, I) \propto \phi_4(\sum_D \phi_3 \sum_C \phi_1 \phi_2)(\sum_{L,J} \phi_6(\sum_H \phi_5)\phi_7) \]
Inference in Jtrees

- For every elimination order $\psi$, we will get a different Jtree
- The time complexity of sending messages in each direction in a Jtree generated with elimination order $\psi$ is $O(n2^{\omega(\psi)})$.
- With spending twice the time of VE, we can have the probabilities for all random variables.
Demos

- Go through the GraphCuts demo
THANK YOU!