Machine Learning
4771

Instructor: Tony Jebara
Topic 17

• Triangulation Examples
• Running Intersection Property
• Building a Junction Tree
• The Junction Tree Algorithm
Triangulation Examples

- **Cycle**: A closed (simple) path, with no repeated vertices other than the starting and ending vertices.
- **Chordless Cycle**: a cycle where no two non-adjacent vertices on the cycle are joined by an edge.
- **Triangulated Graph**: a graph that contains no chordless cycle of four or more vertices (aka a Chordal Graph).
Triangulation Examples
Triangulation Examples
Triangulation Examples
Triangulation Examples
Triangulation Examples

![Graph Example 1]

![Graph Example 2]

![Graph Example 3]
Triangulation Examples

- Diagram 1: A-B-C-D-E-F
- Diagram 2: A-B-C-E-D
- Diagram 3: A-B-C-D-E-F-G-H-I
Triangulation Examples
Running Intersection Property

• Junction Tree must satisfy Running Intersection Property
• RIP: On unique path connecting clique $V$ to clique $W$, all other cliques share nodes in $V \cap W$
Running Intersection Property

• Junction Tree must satisfy Running Intersection Property
• RIP: On unique path connecting clique $V$ to clique $W$, all other cliques share nodes in $V \cap W$

HINT: Junction Tree has largest total separator cardinality

\[
|\Phi| = |\phi(B, D)| + |\phi(C, D)| = 2 + 2
\]

\[
\Phi = \phi(C, D) + \phi(D) = 2 + 1
\]
Forming the Junction Tree

•Goal: connect k cliques into a tree... \( k^{k-2} \) possibilities!
•For each, check Running Intersection Property, too slow...
•Theorem: a valid (RIP) Junction Tree connection is one that maximizes the cardinality of the separators

\[
JT^* = \arg \max_{\text{TREE STRUCTURES}} |\Phi| \\
= \arg \max_{\text{TREE STRUCTURES}} \sum_s |\phi(X_s)|
\]

•Use very fast Kruskal algorithm:
  1) Init Tree with all cliques unconnected (no edges)
  2) Compute size of separators between all pairs
  3) Connect the two cliques with the biggest separator cardinality which doesn’t create a loop in current Tree (maintains Tree structure)
  4) Stop when all nodes are connected, else goto 3
Kruskal Example

- Start with unconnected cliques (after triangulation)

<table>
<thead>
<tr>
<th></th>
<th>ACD</th>
<th>BDE</th>
<th>CDF</th>
<th>DEH</th>
<th>DFGH</th>
<th>FGHI</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACD</td>
<td>-</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>BDE</td>
<td>-</td>
<td></td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>CDF</td>
<td>-</td>
<td></td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DEH</td>
<td>-</td>
<td></td>
<td></td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DFGH</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>FGHI</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-</td>
</tr>
</tbody>
</table>
Junction Tree Probabilities

• We now have a valid Junction Tree!
• What does that mean?
• Recall probability for undirected graphs:

\[ p(X) = p(x_1, \ldots, x_M) = \frac{1}{z} \prod_{C} \psi(X_C) \]

• Can write junction tree as potentials of its cliques:

\[ p(X) = \frac{1}{z} \prod_{C} \tilde{\psi}(X_C) \]

• Alternatively: clique potentials over separator potentials:

\[ p(X) = \frac{1}{Z} \prod_{C} \psi(X_C) \prod_{S} \phi(X_S) \]

• This doesn’t change/do anything! Just less compact...
• Like *de-absorbing* smaller cliques from maximal cliques:

\[ \tilde{\psi}(A, B, D) = \frac{\psi(A, B, D)}{\phi(B, D)} \]

...gives back original formula if \( \phi(B, D) \triangleq 1 \)
Junction Tree Probabilities

• Can quickly converted directed graph into this form:

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

• Example:

By inspection, can just cut & paste CPTs as clique and separator potential functions
Junction Tree Algorithm

- Running the JTA converts clique potentials & separator potentials into marginals over their variables ...
  and does not change \( p(X) \)
- Don’t want just normalization!
  
  \[
  \sum_{A,B,D} \psi(A,B,D) \neq p(A,B,D)
  \]
- These marginals should all agree & be consistent
  
  \[
  \psi(A,B,D) \rightarrow p(A,B,D) \quad \rightarrow \sum_A p(A,B,D) = \tilde{p}(B,D)
  \]
  \[
  \phi(B,D) \rightarrow p(B,D) \quad \rightarrow p(B,D)
  \]
  \[
  \psi(B,C,D) \rightarrow p(B,C,D) \quad \rightarrow \sum_C p(B,C,D) = \tilde{p}(B,D)
  \]
- Consistency: all distributions agree on submarginals
- JTA sends messages between cliques & separators dividing each by the others marginals until consistency...
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: \[ \sum_{V \setminus S} \psi_V = \phi_s = p(S) = \phi_s = \sum_{W \setminus S} \psi_W \]

Else:

Send message From V to W...

\[ \phi_s^* = \sum_{V \setminus S} \psi_V \]
\[ \psi_w^* = \frac{\phi_s^*}{\phi_s} \psi_w \]
\[ \psi_V^* = \psi_V \]

Send message From W to V...

\[ \phi_s^{**} = \sum_{W \setminus S} \psi_w^* \]
\[ \psi_V^{**} = \frac{\phi_s^{**}}{\phi_s} \psi_V^* \]
\[ \psi_w^{**} = \psi_w^* \]

Now they Agree...Done!

\[ \sum_{V \setminus S} \psi_V^{**} = \sum_{V \setminus S} \frac{\phi_s^{**}}{\phi_s} \psi_V^* \]
\[ = \frac{\phi_s}{\phi_s^*} \sum_{V \setminus S} \psi_V^* \]
\[ = \phi_s = \sum_{W \setminus S} \psi_w^{**} \]
Junction Tree Algorithm

• When “Done”, all clique potentials are marginals and all separator potentials are submarginals!

• Note that $p(X)$ is unchanged by message passing step:

$$
\phi_S^* = \sum_{V \setminus S} \psi_V
$$

$$
\psi_W^* = \frac{\phi_S^*}{\phi_S} \psi_W
$$

$$
\psi_V^* = \psi_V
$$

$$
p(X) = \frac{1}{Z} \psi_V^* \psi_W^* = \frac{1}{Z} \psi_V \frac{\phi_S^*}{\phi_S} \psi_W = \frac{1}{Z} \psi_V \psi_W
$$

• Potentials set to conditionals (or slices) become marginals!

$$
\psi_{AB} = p(B \mid A) p(A) = p(A, B)
$$

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

$$
\psi_{BC} = p(C \mid B)
$$

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