Machine Learning
4771
Instructor: Tony Jebara
Topic 22

- Review: Moralization
- Review: Introducing Evidence
- Junction Trees
- Triangulation
- Running Intersection Property
- Building a Junction Tree
- The Junction Tree Algorithm
Review: Moralization

- More examples:

- More general graph less efficient but same inference:

\[
p(x_1) = \sum_{x_2, x_3} p(x_1, x_2, x_3) = \sum_{x_2, x_3} p(x_1 \mid x_2) p(x_2) p(x_3)
\]

\[
p(x_1) = \sum_{x_2, x_3} \frac{1}{Z} \psi(x_1, x_2, x_3) \psi(x_4, x_5) \psi(x_4, x_6) \psi(x_4, x_7)
\]
Review: Introducing Evidence

• Given moral graph, note what is observed \( X_E \rightarrow \overline{X}_E \)

\[
p(X_F | X_E = \overline{X}_E) \equiv p(X_F | \overline{X}_E)
\]

• If we know this is \textit{always} observed at \( X_E \rightarrow \overline{X}_E \), simplify...

• Reduce the probability function since those \( X_E \) fixed

• Only keep probability function over remaining nodes \( X_F \)

• Only get marginals and conditionals with subsets of \( X_F \)

\[
p(X) = \frac{1}{Z} \psi(x_1, x_2, x_3, x_4) \psi(x_4, x_5) \psi(x_4, x_6) \psi(x_4, x_7)
\]

say \( X_E = \{x_3, x_4\} \rightarrow \overline{X}_E = \{\overline{x}_3, \overline{x}_4\} \)

Replace potential functions with slices

\[
p(X_F | \overline{X}_E) \propto \frac{1}{Z} \psi(x_1, x_2, x_3 = \overline{x}_3, x_4 = \overline{x}_4) \psi(x_4 = \overline{x}_4, x_5) \psi(x_4 = \overline{x}_4, x_6) \psi(x_4 = \overline{x}_4, x_7)
\]

\[\propto \frac{1}{Z} \tilde{\psi}(x_1, x_2) \tilde{\psi}(x_5) \tilde{\psi}(x_6) \tilde{\psi}(x_7) \]

But, need to recompute different normalization \( Z \)...
Introducing Evidence

- Recall undirected separation, observing $X_E$ separates others

$$p(X_F, X_E)$$

- But, need to recompute new normalization ...

$$p(X_F \mid \overline{X_E}) \propto \frac{1}{Z} \tilde{\psi}(x_1, x_2) \tilde{\psi}(x_5) \tilde{\psi}(x_6) \tilde{\psi}(x_7)$$

$$\tilde{p}(X_F) = \frac{1}{Z} \tilde{\psi}(x_1, x_2) \tilde{\psi}(x_5) \tilde{\psi}(x_6) \tilde{\psi}(x_7)$$

- Just avoid $Z$ & normalize at the end when we are querying individual marginals and conditionals as subsets of $X_F$

$$\tilde{p}(x_2) = \frac{\sum_{x_1, x_5, x_6, x_7} \tilde{\psi}(x_1, x_2) \tilde{\psi}(x_5) \tilde{\psi}(x_6) \tilde{\psi}(x_7)}{\sum_{x_2} \sum_{x_1, x_5, x_6, x_7} \tilde{\psi}(x_1, x_2) \tilde{\psi}(x_5) \tilde{\psi}(x_6) \tilde{\psi}(x_7)}$$
Junction Trees

- Given moral graph want to build Junction Tree:
  - each node is a clique ($\psi$) of variables in moral graph
  - edges connect cliques of the potential functions
  - unique path between nodes & root node (tree)
  - between connected clique nodes, have separators ($\phi$)
  - separator nodes contain intersection of variables

$\psi(A, B, D)$
\[ p(X) = \frac{1}{2} \psi(A, B, D) \psi(B, C, D) \psi(C, D, E) \]

undirected  cliques  clique tree  junction tree
Triangulation

• Problem: imagine the following undirected graph

• Not a Tree!
• To ensure Junction Tree is a tree (no loops, etc.) before forming it must first **Triangulate** moral graph before finding the cliques...
• Triangulating gives more general graph (like moralization)
• Adds links to get rid of cycles or loops
• Triangulation: Connect nodes in moral graph such that no cycle of 4 or more nodes remains in the graph
Triangulation

- Triangulation: Connect nodes in moral graph such that no cycle of 4 or more nodes remains in graph.

1-cycle OK

2-cycle OK

3-cycle OK

4-cycle BAD

5-cycle BAD

- So, *add links*, but many possible choices...
- HINT: Try to keep largest clique size small
  - (makes junction tree algorithm more efficient)
- Sub-optimal triangulations of moral graph are Polynomial
- Triangulation that minimizes largest clique size is NP
- But, OK to use a suboptimal triangulation (slower JTA...)
Triangulation

- Triangulation: Connect nodes in moral graph such that no cycle of 4 or more nodes remains in graph

- So, *add links*, but many possible choices...
- HINT: Try to keep largest clique size small
  (makes junction tree algorithm more efficient)
- Sub-optimal triangulations of moral graph are Polynomial
- Triangulation that minimizes largest clique size is NP
- But, OK to use a suboptimal triangulation (slower JTA...)
Triangulation Examples

- **Chordless**: A cycle is chordless if no two non-adjacent vertices on the cycle are joined by an edge.
- **Triangulated**: A graph is triangulated if it has no chordless cycles:

![Diagram of chordless and triangulated examples]
Triangulation Examples

- **Chordless**: A cycle is chordless if no two non-adjacent vertices on the cycle are joined by an edge.
- **Triangulated**: A graph is triangulated if it has no chordless cycles:
Triangulation Examples

- **Chordless**: A cycle is chordless if no two non-adjacent vertices on the cycle are joined by an edge.
- **Triangulated**: A graph is triangulated if it has no chordless cycles:
Triangulation Examples

- **Chordless**: A cycle is chordless if no two non-adjacent vertices on the cycle are joined by an edge.
- **Triangulated**: A graph is triangulated if it has no chordless cycles:
Triangulation Examples

- **Chordless**: A cycle is chordless if no two non-adjacent vertices on the cycle are joined by an edge.
- **Triangulated**: A graph is triangulated if it has no chordless cycles:
Triangulation Examples

- **Chordless**: A cycle is chordless if no two non-adjacent vertices on the cycle are joined by an edge.
- **Triangulated**: A graph is triangulated if it has no chordless cycles:
Triangulation Examples

- **Chordless**: A cycle is chordless if no two non-adjacent vertices on the cycle are joined by an edge.
- **Triangulated**: A graph is triangulated if it has no chordless cycles:
Triangulation Examples

- **Chordless**: A cycle is chordless if no two non-adjacent vertices on the cycle are joined by an edge.
- **Triangulated**: A graph is triangulated if it has no chordless cycles:
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

$$|\Phi| = |\phi(B, C)| + |\phi(C, D)| = 2 + 2$$
$$\Phi = |\phi(C, D)| + |\phi(D)| = 2 + 1$$
Forming the Junction Tree

• Now need to connect the cliques into a Junction Tree
• But, must ensure Running Intersection Property
• Theorem: a valid (RIP) Junction Tree connection is one that maximizes the cardinality of the separators

\[ JT^* = \max_{\text{tree structures}} |\Phi| \]

\[ = \max_{\text{tree structures}} \sum_S |\phi(X_S)| \]

• Use Kruskal’s 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)
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:
  \[ p(X) = \frac{1}{Z} \prod_c \frac{\psi(X_c)}{\prod_s \phi(X_s)} \]

• 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 \( \phi(B, D) \equiv 1 \)
Junction Tree Probabilities

- Can quickly convert 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

- Note: Here, division by zero can occur, but numerator will also be 0 so treat 0/0 = 0 since, for example:
Junction Tree Algorithm

- Running the JTA converts clique potentials & separator potentials into marginals over their variables
  \[
  \psi(A, B, D) \rightarrow p(A, B, D) \\
  \phi(B, D) \rightarrow p(B, D) \\
  \psi(B, C, D) \rightarrow p(B, C, D)
  \]
- Don’t want just normalization!
  \[
  \frac{\psi(A, B, D)}{\sum_{A,B,D} \psi(A, B, D)} \neq p(A, B, D)
  \]
- These marginals should all agree & be consistent
  \[
  \psi(A, B, D) \rightarrow p(A, B, D) \rightarrow \sum_A p(A, B, D) = \tilde{p}(B, D) \\
  \phi(B, D) \rightarrow p(B, D) \rightarrow p(B, D) \\
  \psi(B, C, D) \rightarrow p(B, C, D) \rightarrow \sum_C p(B, C, D) = \tilde{p}(B, D)
  \]
- 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: \[ \sum_{V \backslash S} \psi_V = \phi_s = p(S) = \phi_s = \sum_{W \backslash S} \psi_W \]

...Done!

Else: Send message From V to W...

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

Send message From W to V...

\[ \phi_s^{**} = \sum_{W \backslash S} \psi_w^* \]
\[ \psi_v^{**} = \frac{\phi_s^{**}}{\phi_s} \psi_v^* \]
\[ \psi_w^{**} = \psi_w \]

Now they Agree...Done!

\[ \sum_{V \backslash S} \psi_V^{**} = \sum_{V \backslash S} \frac{\phi_s^{**}}{\phi_s} \psi_V^* \]
\[ = \frac{\phi_s^{**}}{\phi_s} \sum_{V \backslash S} \psi_V^* \]
\[ = \phi_s^{**} = \sum_{W \backslash S} \psi_w^{**} \]
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:

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

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

$$
\psi_V^* = \psi_V
$$

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

- Example: if potentials are poorly initialized... get corrected!

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

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

$$
\phi_B = 1
$$

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

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