Our two main texts


Announcements

- Reading assignment: Read the tree_inference.pdf chapter soon to be posted on the web page.
- Read chapters 1, 5, and 6 in the MRF book.
- Read chapter 1 in the Wainwright and Jordan book.
- At our discussion board (https://catalyst.uw.edu/gopost/board/bilmes/23863/) you can find a draft copy of tree_inference.pdf and evidence.pdf

Assignments

- **Two paragraph** project pre-proposals are due Thursday 10/27, at 5:00pm via our dropbox (https://catalyst.uw.edu/collectit/dropbox/bilmes/17463).
- Homework 1 ready, due Friday, Oct 28th, 11:45pm via our dropbox (https://catalyst.uw.edu/collectit/dropbox/bilmes/17463). Any questions should be posted to our discussion board (https://catalyst.uw.edu/gopost/board/bilmes/23863/).
Class Road Map

We need to find one makeup lecture this term.

- L1 (9/28): Introduction, Families, Semantics
- LX (9/30): No class
- L2 (10/5): Trees, exact inference
- L3 (10/7): More on trees and inference.
- L4 (10/12): To tree or not to tree.
- L5 (10/14): All models lead to trees
- L6 (10/19): Decomposable, JT
- L7 (10/21): Inference on JTs
- L8 (10/26):
- L9 (10/28):
- L10 (11/2):
- L11 (11/4):
- L12 (11/9):
- LXX (11/11): Veterans Day, no class
- L13 (11/16):
- L14 (11/18):
- L15 (11/23):
- LXX (11/25): Thanksgiving, no class
- L16 (11/30):
- L17 (12/2):
- L18 (12/7):
- L19: (12/9):

Review

- We want all original graph (o.g.) clique marginals. Why?
- Finding optimal elimination order is optimal for all o.g. clique marginals.
- Def: decomposition of a graph, and factorization implication.
- Def: decomposable graph, and decomposition tree
- Thm: triangulated graph $\equiv$ decomposable graph
- Def: tree decomposition (vertex and edge cover, and induced sub-tree).
- Def: cluster graph, cluster tree, based only on o.g. nodes, not o.g. edges. Edges in cluster graph cluster tree via cluster intersection.
- Def: cluster intersection property, running intersection property, induced sub-tree property, r.i.p.
- Def: Junction tree, cluster tree with r.i.p. and edge cover.
Decomposition of $G$ and Decomposable graphs

Summarizing both:

**Definition 1 (Decomposition of $G$)**
A *decomposition* of a graph $G = (V, E)$ (if it exists) is a partition $(A, B, C)$ of $V$ such that:
- $C$ separates $A$ from $B$ in $G$.
- $C$ is a clique.

If $A$ and $B$ are both non-empty, then the decomposition is called *proper*.

**Definition 2**
A graph $G = (V, E)$ is decomposable if either: 1) $G$ is a clique, or 2) $G$ possesses a *proper* decomposition $(A, B, C)$ s.t. both subgraphs $G[A \cup C]$ and $G[B \cup C]$ are decomposable.

Note part 2. It says *possesses*. Bottom of tree might affect top.
Decomposable models

- Internal nodes in tree are complete graphs that are also separators.
- With $G$ is decomposable, what are implications for a $p \in \mathcal{F}(G, R^f)$?

\[
p(A, B, C, D, E, F, G, H, I, J, K) \\
= \frac{p(A, C, D, F)p(B, C, D, E, F, G, H, I, J, K)}{p(C, D, F)} \\
= \frac{p(A, C, D, F)}{p(C, D, F)} \left( \frac{p(B, C, G, H)p(C, D, E, F, H, I, J, K)}{p(C, H)} \right) \\
= \ldots \\
= \frac{p(A, C, D, F)p(B, G, H)p(C, B, H)p(E, I, J)p(E, I, D)p(C, K, H)p(D, K, l)p(D, K, F, C)}{p(C, D, F)p(C, H)p(B, H)p(D, I)p(E, I)p(C, K)p(D, K)}
\]

Definition 3 (tree decomposition)

Given a graph $G = (V, E)$, a tree-decomposition of a graph is a pair $(\{C_i : i \in I\}, T)$ where $T = (I, F)$ is a tree with node index set $I$, edge set $F$, and $\{C_i\}_i$ (one for each $i \in I$) is a collection of subsets of $V(G)$ such that:

1. $\bigcup_{i \in I} C_i = V$
2. for any $(u, v) \in E(G)$, there exists $i \in I$ with $u, v \in C_i$
3. for any $v \in V$, the set $\{i \in I : v \in C_i\}$ forms a connected subtree of $T$
Cluster graphs

**Definition 4 (Cluster graph)**

Consider forming a new graph based on $G$ where the new graph has nodes that correspond to clusters in the original $G$, and has edges existing between two (cluster) nodes only when the corresponding clusters have a non-zero intersection. That is, let $\mathcal{C}(G) = \{ C_1, C_2, \ldots, C_{|I|} \}$ be a set of $|I|$ clusters of nodes $V(G)$, where $C_i \subseteq V(G), i \in I$. Consider a new graph $J = (I, E)$ where each node in $J$ corresponds to a set of nodes in $G$, and where edge $(i, j) \in E$ if $C_i \cap C_j \neq \emptyset$. We will also use $S_{ij} = C_i \cap C_j$ as notation.

So two cluster nodes have an edge between them iff there is non-zero intersection between the nodes.

**Cluster Trees**

If the graph is a tree, then we have what is called a cluster tree.

**Definition 5 (Cluster Tree)**

Let $\mathcal{C} = \{ C_1, C_2, \ldots, C_{|I|} \}$ be a set of node clusters of graph $G = (V, E)$. A cluster tree is a tree $T = (I, E_T)$ with vertices corresponding to clusters in $\mathcal{C}$ and edges corresponding to pairs of clusters $C_1, C_2 \in \mathcal{C}$. We can label each vertex in $i \in I$ by the set of graph nodes in the corresponding cluster in $G$, and we label each edge $(i, j) \in E_T$ by the cluster intersection, i.e., $S_{ij} = C_i \cap C_j$. 
Cluster Intersection Property (c.i.p.)

- Important: Cluster graphs and cluster trees are based only on a set of clusters of nodes of $G = (V, E)$. We haven’t, based on these definitions, yet used any of the o.g. edges of $G$.
- Edges in a cluster graph and cluster tree are not o.g. edges. Instead, they are based on if two clusters have non-empty intersection.
- We want to talk about cluster trees that have certain properties. A cluster graph might or might not have such properties.
Cluster Intersection Property (c.i.p.)

Definition 6 (Cluster Intersection Property)

We are given a cluster tree \( T = (I, \mathcal{E}_T) \), and let \( C_1, C_2 \) be any two clusters in the tree. Then the cluster intersection property states that \( C_1 \cap C_2 \subseteq C_i \) for all \( C_i \) on the (by definition, necessarily) unique path between \( C_1 \) and \( C_2 \) in the tree \( T \).

- A given cluster tree might or might not have that property.
- Example on the next few slides.

Running Intersection Property (r.i.p.)

Definition 7 (Running Intersection Property (r.i.p.))

Let \( C_1, C_2, \ldots, C_\ell \) be an ordered sequence of subsets of \( V(G) \). Then the ordering obeys the running intersection property (r.i.p.) property if for all \( i > 1 \), there exists \( j < i \) such that \( C_i \cap (\bigcup_{k<i} C_k) = C_i \cap C_j \).

- r.i.p. is defined in terms of clusters of nodes in a graph. r.i.p. holds if such an ordering can be found.
- Cluster \( j \) acts as a representative for all of \( i \)'s history.
Examples

Cluster Graph Cluster Tree Cluster Tree that violates the cluster intersection property
Another cluster Tree that violates the cluster intersection property

Induced sub-tree property (i.s.p.)

Definition 8 (Induced Sub-tree Property)

Given a cluster tree $T$ for graph $G$, the induced sub-tree property holds for $T$ if for all $v \in V$, the set of clusters $C \in \mathcal{C}$ such that $v \in C$ induces a sub-tree $T(v)$ of $T$.

Note, by definition the sub-tree is necessarily connected.
Three Properties

**Lemma 9**

The cluster intersection, running intersection, and induced sub-tree properties are identical.

---

**Tree decomposition**

Let's look again at tree decomposition, a cluster tree that satisfies (what we now know to be the) induced sub-tree property (e.g., r.i.p. and c.i.p. as well).

**Definition 10 (tree decomposition)**

Given a graph $G = (V, E)$, a tree-decomposition of a graph is a pair $(\{C_i : i \in I\}, T)$ where $T = (I, E_T)$ is a tree with node index set $I$, edge set $E_T$, and $\{C_i\}$; (one for each $i \in I$) is a collection of clusters (subsets) of $V(G)$ such that:

1. $\bigcup_{i \in I} C_i = V$
2. For any edge $(u, v) \in E(G)$, there exists $i \in I$ with $u, v \in C_i$
3. (r.i.p.) For any $v \in V$, the set $\{i \in I : v \in C_i\}$ forms a (nec. connected) subtree of $T$
**Definition 11**

Given a graph $G = (V, E)$, a junction tree corresponding to $G$ (if it exists) is a cluster tree $T = (\mathcal{C}, E_T)$ having the r.i.p. over the clusters, and where the nodes $u, v$ adjacent to every edge $(u, v) \in E(G)$ are together in at least one cluster.

- So, junction tree (JT), for a given graph $G$, is a cluster tree that: 1) satisfies r.i.p. over the clusters, and 2) includes all edges (edge cover).
- Not all r.i.p.-satisfying cluster trees need be an edge cover.
- Clusters in JT need not be original graph cliques!!
- JT could have clusters corresponding to cliques, maxcliques, or neither of the above.
- If clusters correspond to the original graph cliques (resp. maxcliques) in $G$, it called a junction tree of cliques (resp. maxcliques).

**Lemma 12**

Given a junction tree, form a new cluster tree as follows. For each cluster $C$ in the JT, choose an order of nodes within $C$, say $c_1, c_2, \ldots, c_k$, and hang a chain of clusters off of $C$ consisting of $C \setminus \{c_1\}$ hanging from $C$, $C \setminus \{c_1, c_2\}$ hanging from $C \setminus \{c_1\}$, $C \setminus \{c_1, c_2, c_3\}$ hanging from $C \setminus \{c_1, c_2\}$, and so on. Then the resulting cluster graph is a cluster tree, and moreover it is still junction tree.

**Lemma 13**

Given a junction tree, where $(C_i, C_j)$ are neighboring clusters in the tree, we can merge these two clusters forming a new cluster $C_{ij} = C_i \cup C_j$, and where the neighbors of $C_{ij}$ are the set of neighbors of either $C_i$ and $C_j$. Then the resulting structure is still junction tree.

If we keep doing the latter, we’ll end up with one complete graph.
Examples junction trees and not

Questions to ask:

- cluster graph?
- cluster tree?
- Junction tree?
- Junction tree of cliques?
- Junction tree of maxcliques?
Examples junction trees and not

- cluster graph?
- cluster tree?
- Junction tree?
- Junction tree of cliques?
- Junction tree of maxcliques?

Prof. Jeff Bilmes  
EE12A/Fall 2011/Graphical Models – Lecture 7 - October 21th, 2011  
page 26
Examples junction trees and not

Tree of cliques for above graph. Does r.i.p. hold?

Key theorem: JT of maxcliques $\equiv$ triangulated graphs

Theorem 14

A graph $G = (V, E)$ is decomposable iff a junction tree of maxcliques for $G$ exists.

Proof.

Induction on the number of maxcliques. If $G$ has one maxclique, it is both a junction tree and decomposable. Assume true for $\leq k$ maxcliques and show it for $k + 1$. 

...
... proof continued.

A junction tree exists \(\Rightarrow\) decomposable: Let \(T\) be a junction tree of maxcliques \(C\), and let \(C_1, C_2\) be adjacent in \(T\). The edge \(C_1, C_2\) in the tree separates \(T\) into two sub-trees \(T_1\) and \(T_2\), with \(V_i\) being the nodes in \(T_i\), \(G_i = G[V_i]\) being the subgraph of \(G\) corresponding to \(T_i\), and \(C_i\) being the set of maxcliques in \(T_i\), for \(i = 1, 2\). Thus \(V(G) = V_1 \cup V_2\), and \(C = C_1 \cup C_2\). Note that \(C_1 \cap C_2 = \emptyset\). We also let \(S = V_1 \cap V_2\) which is the intersection of all the nodes in each of the two trees.

Also, the nodes in \(T_i\) are maxcliques in \(G_i\) and \(T_i\) is a junction tree for \(G_i\) since r.i.p. still holds in the subtrees of a junction tree. Therefore, by induction, \(G_i\) is decomposable. To show that \(G\) is decomposable, we need to show that: 1) \(S = V_1 \cap V_2\) is complete, and 2) that \(S\) separates \(G[V_1 \setminus S]\) from \(G[V_2 \setminus S]\).

If \(v \in S\), then for each \(G_i\) \((i = 1, 2)\), there exists a clique \(C_i'\) with \(v \in C_i'\), and the path in \(T\) joining \(C_1'\) and \(C_2'\) passes through both \(C_1\) and \(C_2\). Because of the r.i.p., we thus have that \(v \in C_1\) and \(v \in C_2\) and so \(v \in C_1 \cap C_2\). This means that \(V_1 \cap V_2 \subseteq C_1 \cap C_2\). But \(C_i \subseteq V_i\) since \(C_i\) is a clique in the corresponding tree \(T_i\). Therefore \(C_1 \cap C_2 \subseteq V_1 \cap V_2 = S\), so that \(S = C_1 \cap C_2\). This means that \(S\) contains all nodes that are common among the two subgraphs and moreover that \(S\) is complete as desired.
... proof continued.

Next, to show that $S$ is a separator, we take $u \in V_1 \setminus S$ and $v \in V_2 \setminus S$ (note that such choices mean $u \notin V_2$ and $v \notin V_1$ due to the commonality property of $S$). Suppose the contrary that $S$ does not separate $V_1$ from $V_2$, which means there exists a path $u, w_1, w_2, \ldots, w_k, v$ for the given $u, v$ with $w_i \notin S$ for all $i$. Therefore, there is a clique $C \in C$ containing the set $\{u, w_1\}$. We must have $C \notin C_2$ since $u \notin V_2$, which means $C \in C_1$ or $C \subseteq V_1$ implying that $w_1 \in V_1$ and moreover that $w_1 \in V_1 \setminus S$. We repeat this argument with $w_1$ taking the place of $u$ and $w_2$ taking the place of $w_1$ in the path, and so on until we end up with $v \in V_1 \setminus S$ which is a contradiction. Therefore, $S$ must separate $V_1$ from $V_2$. We have thus formed a decomposition of $G$ as $(V_1 \setminus S, S, V_2 \setminus S)$ and since $G_i$ is decomposable (by induction), we have that $G$ is decomposable.

...
... proof continued.

Let $v \in V$. If $v \not\in V_2$, then all cliques containing $v$ are in $C_1$ and those cliques form a connected tree by the junction tree property since $T_1$ is a junction tree. The same is true if $v \not\in V_1$. Otherwise, if $v \in S$ (meaning that $v \in V_1 \cap V_2$), then the cliques in $C_i$ containing $v$ are connected in $T_i$ including $C_i$ for $i = 1, 2$. But by forming $T$ by connecting $C_1$ and $C_2$, and since $v$ is arbitrary, we have retained the junction tree property. Thus, $T$ is a junction tree.

Lemma 15

A junction tree of maxcliques for graph $G = (V, E)$ exists iff a junction tree of cliques for graph $G = (V, E)$ exists.

- How can we get from one to the other?

Since decomposable is same as triangulated:

Corollary 16

A graph $G$ is triangulated iff a junction tree of cliques for $G$ exists.
How to build a junction tree

- Maximum cardinality search algorithm can do this. If graph is triangulated, it produces a list of cliques in r.i.p. order.

Maximum Cardinality Search with maxclique order

**Algorithm 1:** Maximum Cardinality Search: Determines if a graph $G$ is triangulated.

**Input:** An undirected graph $G = (V, E)$ with $n = |V|$.

**Result:** is triangulated?, if so MCS ordering $\sigma = (v_1, \ldots, v_n)$, and maxcliques in r.i.p. order.

1. $L \leftarrow \emptyset$; $i \leftarrow 1$; $C \leftarrow \emptyset$
2. while $|V \setminus L| > 0$ do
3.   Choose $v_i \in \text{argmax}_{u \in V \setminus L} |\delta(u) \cap L|$; /* $v_i$’s previously labeled neighbors has max cardinality. */
4.   $c_i \leftarrow \delta(v_i) \cap L$; /* $c_i$ is $v_i$’s neighbors in the reverse elimination order. */
5.   if $\{v_i\} \cup c_i$ is not complete in $G$ then
6.     return “not triangulated”;
7.   if $|c_i| \leq |c_{i-1}|$ then
8.     $C \leftarrow (C, \{c_{i-1} \cup \{v_{i-1}\}\})$; /* Append the next maxclique to list $C$. */
9.   if $i = n$ then
10.    $C \leftarrow (C, \{c_i \cup \{v_i\}\})$ ; /* Append the last maxclique to list $C$. */
11.   $L \leftarrow L \cup \{v_i\}$; $i \leftarrow i + 1$
12. return “triangulated”, the ordering $\sigma$, and the set of maxcliques $C$ which are in r.i.p. order.
How to build a junction tree

- Alternatively, we can construct the maxcliques in any form (say by running elimination) and find a maximal spanning tree over the edge-weighted cluster graph, where clusters correspond to maxcliques, and edge weights correspond to the size of the intersection of the two adjacent maxcliques.
- Prim’s algorithm can run in $O(|E| + |V| \log |V|)$, much better than $|V|^2$ for sparse graphs.

**Theorem 17**

A tree of maxcliques $T$ is a junction tree iff it is a maximum spanning tree on the maxclique graph, with edge weights set according to the cardinality of the separator between the two maxcliques.

- Note: graph must be triangulated. I.e., maximum spanning tree of a cluster graph where the clusters are maxcliques but the graph is not triangulated will clearly not produce a junction tree.

Other aspects of JTs

- There can be multiple JTs for a given triangulated graph (e.g., consider any graph where $d(S) \geq 3$ for some separator $S$).
- JTs are not binary decomposition trees (BDTs), but they are related. Leaf nodes of BDTs correspond to nodes in a JT of maxcliques. Non-leaf nodes in a BDTs may correspond to edges in a JT. Therefore, edges in a JT may correspond to all minimal separators in triangulated graph $G'$.
- Set of maxcliques is unique in a triangulated graph. Set of minimal separators is unique in a triangulated graph.
- Again, JT can be over not just maxcliques. JT can exist over all cliques, or over some cliques (if they contain all maxcliques)
- Different JTs always have same set of nodes and separators, just different configurations.
Intersection Graphs

- We’re next going to look at seemingly very different way to view triangulated graphs and junction trees, based on intersection graph theory.
- We’ll see that triangulated graphs are identical to a type of intersection graph, where the underlying object is a tree (furthering our connection to trees).
- first, lets talk a bit about terminology.

Edge Clique Covers

- Set cover - sets must cover the ground/universal set (ground set cover)
- Vertex cover - vertices must cover the edges (edge vertex cover)
- Edge cover - edges must cover the vertices (vertex edge cover)
- clique cover - cliques cover the edges (edge clique cover)
- The nodes of a junction tree of cliques (or maxcliques) constitute an edge clique cover for triangulated graph $G'$ — start with set of nodes $V = \bigcup_{C \in C} C$. Add edge between $u, v \in V$ if exists a $C \in C$ such that $u, v \in C$.
- Going from $G'$ to JT and back to the graph yields the same graph.
Definition 18 (Intersection Graph)

An intersection graph is a graph $G = (V, E)$ where each vertex $v \in V(G)$ corresponds to a set $U_v$ and each edge $(u, v) \in E(G)$ exists only if $U_u \cap U_v \neq \emptyset$.

- some underlying set of objects $U$ and a multiset of subsets of $U$ of the form $U = \{U_1, U_2, \ldots, U_n\}$ with $U_i \subseteq U$ — might have some $i, j$ where $U_i = U_j$.

Theorem 19

Every graph is an intersection graph.

This can be seen informally by consider an arbitrary graph, create a $U_i$ for every node, and construct the subsets so that the edges will exist when taking intersection.

Interval Graphs (a type of intersection graph)

- Interval graphs are intersection graphs where the subsets are intervals/segments $[a, b]$ in $\mathbb{R}$

- Any graph that can be constructed this way is an interval graph

$$\begin{align*}
a &- b &- c &- d \\
e &
\end{align*}$$

- Are all graphs interval graphs? 4-cycle
Theorem 20

All Interval Graphs are triangulated.

proof sketch.

Given interval graph $G = (V, E)$, consider any cycle $u, w_1, w_2, \ldots, w_k, v, u \in V(G)$. Cycle must go (w.l.o.g.) forward and then backwards along the line in order to connect back to $u$, so there must be a chord between some non-adjacent nodes (since they will overlap).

Are all triangulated graphs interval graphs? No, consider spider graph (elongated star graph).

Sub-tree intersection Graphs

- Given underlying tree, create intersection graph, where subsets are (nec. connected) subtrees of some “ground” tree.
- Intersection exists if there are any nodes in common amongst the two corresponding trees.
Intersection exists if there are any nodes in common amongst the two corresponding trees.

A sub-tree graph corresponds to more than one underlying tree (thus ground set and underlying subsets).

What is the difference between left and right trees?

Junction tree of cliques and maxcliques vs. junction tree of just maxcliques.
Sub-tree intersection Graphs

- Intersection exists if there are any nodes in common amongst the two corresponding trees.
- A sub-tree graph corresponds to more than one underlying tree (thus ground set and underlying subsets).
- What is the difference between left and right trees?
- Junction tree of cliques and maxcliques vs. junction tree of just maxcliques.
Sub-tree intersection graphs

Theorem 21

A graph $G = (V, E)$ is triangulated iff it corresponds to a sub-tree graph (i.e., an intersection graph on subtrees of some tree).

proof sketch.

We see that any sub-tree graph is such that nodes in the tree correspond to cliques in $G$, and by the nature of how the graph is constructed (subtrees of some underlying tree), the tree corresponds to a cluster tree that satisfies the induced subtree property. Therefore, any sub-tree graph corresponds to a junction tree, and any corresponding graph $G$ is triangulated.
Sub-tree intersection graphs

- All interval graphs are sub-tree intersection graphs (underlying tree is a chain, subtrees are sub-chains)
- Are all sub-tree intersection graphs interval graphs?
- So sub-tree intersection graphs capture the “tree-like” nature of triangulated graphs.
- Triangulated graphs are also called hyper-trees (specific type of hyper-graph, where edges are generalized to be clusters of nodes rather than 2 nodes in a normal graph). In hyper-tree, the unique “max-edge” path between any two nodes property is generalized.

Inference on JTs.

- We can define an inference procedure on junction trees that corresponds to our inference procedure on trees.
- We are given $p \in \mathcal{F}(G', R^{(f)})$, where $G'$ is triangulated. It might be naturally triangulated, might be an MRF for which we’ve found a good elimination order, or might even have come from a triangulated moralized Bayesian network. In either case, if we solve inference for the family $\mathcal{F}(G', R^{(f)})$ we’ve solved it for the original graph.
- Let $G$ be the original graph with cliques $\mathcal{C}(G)$, and let $\mathcal{C}(G')$ be the cliques of the triangulated graph.
- We know we have factorization:

$$p(x) = \prod_{C \in \mathcal{C}(G)} \psi_C(x_C)$$

(1)
Inference on JTs.

- Every clique \( C \in \mathcal{C}(G) \) is contained in at least one clique \( C' \in \mathcal{C}(G') \).
- Therefore, each factor \( \psi_C(x_C) \) for \( C \in \mathcal{C}(G) \) can be assigned to a new factor \( \psi_{C'}(x_{C'}) \) for some \( C' \in \mathcal{C}(G') \).
- Given that we have a junction tree of maxcliques, we are going to allocate "storage" for maxclique potentials \( \psi_{C'}(x_{C'}) \) for all \( C' \in \mathcal{C}(G') \) (equivalently all nodes in the junction tree).
- We are also going to allocate storage for all separators in the junction tree. That is, we will have a function \( \phi_S(x_S) \) for all \( S \in \mathcal{S}(G') \) where \( \mathcal{S}(G') \) are the set of separators in the junction tree corresponding to triangulated graph \( G' \).
- We need to know how to initialize these separators.

Inference on JTs - table initialization

- Initialization Step: For each \( C' \in \mathcal{C}(G') \), assign \( \psi_{C'}(x_{C'}) = 1 \).
- For each clique \( C \in \mathcal{C}(G) \), find one \( C' \in \mathcal{C}(G') \) such that \( C \subseteq C' \), and update \( \psi_{C'}(x_{C'}) \) as follows:
  \[
  \psi_{C'}(x_{C'}) \leftarrow \psi_{C'}(x_{C'}) \psi_C(x_C)
  \]
  (2)
- Crucial: Only do this once, otherwise, we’ll be double counting the clique \( \psi_C(x_C) \).
- We now have the following representation of \( p \in \mathcal{F}(G, R^{(f)}) \):
  \[
  p(x) = \prod_{C' \in \mathcal{C}(G')} \psi_{C'}(x_{C'})
  \]
  (3)
- We also initialize all separators by doing \( \phi_S(x_S) = 1 \ \forall S \).
- Once this is done, we have
  \[
  p(x) = \frac{\prod_{C' \in \mathcal{C}(G')} \psi_{C'}(x_{C'})}{\prod_{S \in \mathcal{S}(G')} \phi_S(x_S)^{d(S)-1}}
  \]
  (4)
Maxclique marginals as the goal

- Since $G'$ is triangulated, and is decomposable, we know it is possible to represent $p$ as:

$$p(x) = \prod_{C' \in \mathcal{C}(G')} \psi_{C'}(x_{C'}) = \frac{\prod_{C \in C'} p(x_{C'})}{\prod_{S \in S(G)} p(x_S)^{d(S)-1}} \quad (5)$$

where $d(S)$ is the shattering coefficient of separator $S$.

- In Equation 5, we have the functions at each maxclique and at each separator equal to the **marginal distribution** over the corresponding nodes.

- With the marginals, we can easily compute any desired original-graph clique marginal for any $C \in \mathcal{C}(G)$.

- Our goal is to efficiently go from the representation at Equation 4 to the representation at the right of Equation 5.

- Can we do this using a similar message passing procedure to what we’ve already seen?

We do this using a junction tree (which we know to exist over the cliques and/or maxcliques of $G'$). So form a junction tree.

Goal (again) is for the clique and separator functions to equal marginals.

What must be true of clique functions if they are marginals? They must (at least) agree with what they have in common.

Consider pair of neighboring cliques in a JT. Given maxclique $C'_1$ and $C'_2$ of $\mathcal{C}$, with $S = C'_1 \cap C'_2$, they must agree, i.e.,:

$$\sum_{x_{C'_1 \backslash S}} \psi_{C'_1}(x_{C'_1}) = \sum_{x_{C'_2 \backslash S}} \psi_{C'_2}(x_{C'_2}) \quad (6)$$
Maxclique marginals as the goal

- This is a necessary condition for the clique/separator functions to be marginals because

\[ \sum_{x_{C_1} \setminus S} \psi_{C_1}(x_{C_1}) = \sum_{x_{C_1} \setminus S} p(x_{C_1}) = \sum_{x_{C_2} \setminus S} p(x_{C_2}) = \sum_{x_{C_2} \setminus S} \psi_{C_2}(x_{C_2}) \]  

(7)

- Given two maxcliques \( U \) and \( W \) with separator \( S = U \cap W \), and potential functions \( \psi_U \), \( \psi_W \), and \( \phi_S \), arranged in small JT as follows:

\[ \psi_U \quad \phi_S \quad \psi_W \]

\[ U \quad S \quad W \]

Maxclique marginals as the goal

- Shorthand notation: \( \phi_S^* = \sum_{U \setminus S} \psi_U \) — represents new potential over separator \( S \) obtained from \( \psi_U \) where all but \( S \) has been marginalized away.

- Thus,

\[ \sum_{U \setminus S} \psi_U \triangleq \sum_{x_{U \setminus S}} \psi_U(x_U) = \sum_{x_{U \setminus S}} \psi_U(x_{U \setminus S}, x_S) = \phi_S^*(x_S) \]

which is a function only of \( x_S \).
Maxclique marginals as the goal

- More shorthand notation: table multiplication

\[ \psi^*_W = \frac{\phi^*_S}{\phi_S} \psi_W \]  \hspace{1cm} (8)

- Let \( W_S = W \setminus S \), so that \( W = S \cup W_S \). then

\[ \psi_W = \psi_W(x_W) = \psi_W(x_S, x_{W_S}), \quad \phi_S = \phi_S(x_S) \]  \hspace{1cm} (9)

and

\[ \psi^*_W = \psi^*_W(x_W) = \psi^*_W(x_S, x_{W_S}), \quad \phi^*_S = \phi^*_S(x_S) \]  \hspace{1cm} (10)

so to expand everything out, we get

\[ \psi^*_W = \psi^*_W(x_S, x_{W_S}) = \frac{\phi^*_S(x_S)}{\phi_S(x_S)} \psi_W(x_S, x_{W_S}) \]  \hspace{1cm} (11)

Suppose, JT potentials start out inconsistent. i.e.,

\[ \sum_{U \setminus S} \psi_U \neq \sum_{W \setminus S} \psi_W \quad \text{and} \quad \phi_S = 1 \]  \hspace{1cm} (12)

but we still have that \( p(x_U, x_W) = p(x_H, \bar{x}_E) = \psi_U \psi_W / \phi_S \).

- Note (again) that we may treat evidence \( \bar{x}_E \) as additional factors contained within a clique and that any summation would only sum over corresponding evidence value, so we can avoid mentioning evidence for now.

- What we’ll do: exchange information between cliques via separators to achieve consistency.
Maxclique marginals as the goal

- **Marginalize** $U$:

  \[
  \phi^*_S = \sum_{U \setminus S} \psi_U
  \]

  which leads to a new separator potential $\phi^*_S$ and can be seen as a partial message, as shown in the following figure.

- **Rescale** $W$:

  \[
  \psi^*_W = \frac{\phi^*_S}{\phi_S} \psi_W
  \]

  This produces a new potential on $W$ based on the updated separator potential at $S$. This can also be seen as a partial message.
Maxclique marginals as the goal

- After these ops, joint has not changed: define $\psi^*_U = \psi_U$ for convenience, we get:

$$\frac{\psi^*_U \psi^*_W}{\phi^*_S} = \frac{\psi_U \psi_W \phi^*_S}{\phi_S \phi^*_S} = \frac{\psi_U \psi_W}{\phi_S}$$  \hspace{1cm} (15)

- Don’t yet (nec.) have consistency since

$$\sum_{U \setminus S} \psi^*_U = \sum_{U \setminus S} \psi_U = \phi^*_S \neq \sum_{W \setminus S} \psi^*_W = \frac{\phi^*_S}{\phi_S} \sum_{W \setminus S} \psi_W$$  \hspace{1cm} (16)

which follows because

$$\phi_S \neq \sum_{W \setminus S} \psi_W$$  \hspace{1cm} (17)

We do at least have one marginal at $\psi^*_W$. This is because we started with:

$$p(x_U, x_W) = \psi_U \psi_W \frac{\phi^*_S}{\phi_S}$$  \hspace{1cm} (18)

and

$$\psi^*_W = \frac{\phi^*_S}{\phi_S} \psi_W = \psi_W \sum_{U \setminus S} \psi_U = \sum_{x_U \setminus S} p(x_H, \bar{x}_E) = p(x_W)$$  \hspace{1cm} (19)

is one of the marginals that we desire.
Maxclique marginals as the goal

- We see this as a message passing procedure, passing a message between two nodes in a cluster tree.
- Message from cluster $U$ through $S$ and to $W$ is the message directly from $U$ to $W$ (but done in two steps).

![Diagram of message passing]

$\psi_U \rightarrow \phi^*_S \rightarrow \psi^*_W$

What if we were to do the same set of operations in reverse, i.e., send a message from $W$ back to $U$ using the new state of the potential functions. I.e., we first

**Marginalize $W$:**

$$\phi^{**}_S = \sum_{W \setminus S} \psi^*_W \quad (20)$$

resulting in still another separator potential. And then
Maxclique marginals as the goal

- Rescale $U$:
  \[
  \psi_{**}^U = \frac{\phi_{**}^S}{\phi_{**}^S} \psi_{**}^U
  \]
  resulting in a new potential on $U$.

- The new joint $p(x_U, x_W)$ has again not changed. Define $\psi_{**}^W = \psi_{**}^W$ for convenience, we get:
  \[
  \frac{\psi_{**}^U \psi_{**}^W}{\phi_{**}^S} = \frac{\psi_{**}^U \phi_{**}^S \psi_{**}^W \phi_{**}^S}{\phi_{**}^S \phi_{**}^S \phi_{**}^S} = \frac{\psi_{**}^U \psi_{**}^W}{\phi_{**}^S}
  \]
Maxclique marginals as the goal

More importantly, after backwards message, we have consistency. In particular, $\psi^*_U$ and $\psi^*_W$ are now consistent since:

$$\sum_{U \setminus S} \psi^*_U = \sum_{U \setminus S} \frac{\phi^*_S \psi^*_U}{\phi^*_S} = \frac{\phi^*_S}{\phi^*_S} \sum_{U \setminus S} \psi^*_U = \frac{\phi^*_S}{\phi^*_S} \phi^*_S = \phi^*_S = \sum_{W \setminus S} \psi^*_W \tag{23}$$
Maxclique marginals as the goal

We moreover have the other marginal we want at $\psi_{**}^*$ since:
\[
\psi_{**}^* = \frac{\phi_{**}^*}{\phi_{**}^*} \psi = \psi U \frac{\sum W \psi W}{\sum U \phi U} = \psi U \frac{\sum W \phi_{**}^* \psi W}{\sum U \phi U} = \psi U \frac{\sum W \psi W}{\sum U \psi U} = \psi U \sum W \psi W = \sum p(x_U, x_W) = p(x_U)
\]
Maxclique marginals as the goal
general trees

We use same scheme we saw for 1-trees. I.e.,

**Definition 22 (Message passing protocol)**

A clique can send a message to a neighboring cluster in a JT **only** after it has received messages from all of its *other* neighbors.

We already know collect/distribute evidence is a simple algorithm that obeys MPP (designate root, and do bottom up messages and then top-down messages). Does this achieve consistency?

**Theorem 23**

The message passing protocol renders the cliques locally consistent between all pairs of connected cliques in the tree.

**Proof.**

Suppose $W$ has received a message from all other neighbors, and is sending a message to $U$. There are two possible cases: Case A: $U$ already sent a message to $W$ before, so $U$ already received message from all other neighbors, & message renders the two consistent since neither receives any more messages.
Maxclique marginals as the goal
general trees

proof continued.

Case B: $U$ has not yet sent a message to $W$, so $W$ sends to $U$ & waits. Later, $U$ will have received message from all other neighbors & will send message back to $W$, but this will contain appropriate update from $W$.

Another way we can see it: If we abide by the message passing protocol, the potential functions will just be scaled by a constant, and we’ll get back to the same case that we were before with two cliques.
Most of this material comes from the tree_inference.pdf handout posted at our discussion board (https://catalyst.uw.edu/gopost/board/bilmes/23863/).