EE512A – Advanced Inference in Graphical Models

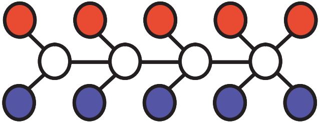
— Fall Quarter, Lecture 8 —

http://j.ee.washington.edu/~bilmes/classes/ee512a_fall_2014/

Prof. Jeff Bilmes

University of Washington, Seattle
Department of Electrical Engineering
http://melodi.ee.washington.edu/~bilmes

Oct 22nd, 2014



Announcements

- Reading assignments, posted to our canvas announcements page (`https://canvas.uw.edu/courses/914697/announcements`): `intro.pdf`, `ugms.pdf` on undirected graphical models, and `tree_inference.pdf` on trees.

Class Road Map - EE512a

- L1 (9/29): Introduction, Families, Semantics
- L2 (10/1): MRFs, elimination, Inference on Trees
- L3 (10/6): Tree inference, message passing, more general queries, non-tree)
- L4 (10/8): Non-trees, perfect elimination, triangulated graphs
- L5 (10/13): triangulated graphs, k-trees, the triangulation process/heuristics
- L6 (10/15): multiple queries, decomposable models, junction trees
- L7 (10/20): junction trees, begin intersection graphs
- L8 (10/22): intersection graphs, inference on junction trees, semirings
- L9 (10/27):
- L10 (10/29):
- L11 (11/3):
- L12 (11/5):
- L13 (11/10):
- L14 (11/12):
- L15 (11/17):
- L16 (11/19):
- L17 (11/24):
- L18 (11/26):
- L19 (12/1):
- L20 (12/3):
- Final Presentations: (12/10):

Finals Week: Dec 8th-12th, 2014.

Today

JT is the tree that all graphs can be transformed up into. Once we have the tree, we have a variety of options of message passing that generalize the message passing on trees we've already seen. Moreover, we can reduce that down to a 1-tree again if we wish.

Brief Review

- Three views of r.i.p. (c.i.p., induced sub-tree, or order-based constraint).

Brief Review

- Three views of r.i.p. (c.i.p., induced sub-tree, or order-based constraint).
- A JT is a cluster tree that satisfies the r.i.p.

Brief Review

- Three views of r.i.p. (c.i.p., induced sub-tree, or order-based constraint).
- A JT is a cluster tree that satisfies the r.i.p.
- A JT can be tree of cliques w.r.t. an o.g. iff the graph is triangulated.

+ [illegible]

Brief Review

- Three views of r.i.p. (c.i.p., induced sub-tree, or order-based constraint).
- A JT is a cluster tree that satisfies the r.i.p.
- A JT can be tree of cliques w.r.t. an o.g. iff the graph is triangulated.
- Equivalence of triangulated graphs, decomposable graphs, perfect elimination graphs, JT of cliques exists, and (soon) sub-tree graphs.

Brief Review

- Three views of r.i.p. (c.i.p., induced sub-tree, or order-based constraint).
- A JT is a cluster tree that satisfies the r.i.p.
- A JT can be tree of cliques w.r.t. an o.g. iff the graph is triangulated.
- Equivalence of triangulated graphs, decomposable graphs, perfect elimination graphs, JT of cliques exists, and (soon) sub-tree graphs.
- Inference on JTs: goal, clusters as marginals $p(x_C)$

Intersection Graphs

Definition 8.3.1 (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 $U_u \cap U_v \neq \emptyset$.

Intersection Graphs

Definition 8.3.1 (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 $\mathcal{U} = \{U_1, U_2, \ldots, U_n\}$ with $U_i \subseteq U$ — multiset, so allowed to have some i, j where $U_i = U_j$.

$n = |V(G)|$

$\iff U_u \cap U_v \neq \phi$

Intersection Graphs

Definition 8.3.1 (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 $\mathcal{U} = \{U_1, U_2, \ldots, U_n\}$ with $U_i \subseteq U$ — multiset, so allowed to have some i, j where $U_i = U_j$.

Theorem 8.3.2

Every graph is an intersection graph.

Intersection Graphs

Definition 8.3.1 (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 $\mathcal{U} = \{U_1, U_2, \ldots, U_n\}$ with $U_i \subseteq U$ — multiset, so allowed to have some i, j where $U_i = U_j$.

$U = \bigcup_{i \in [n]} U_i$

Theorem 8.3.2

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)

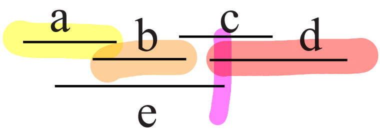
- Interval graphs are intersection graphs where the subsets are intervals/segments $[a, b]$ in $\mathbb{R}$

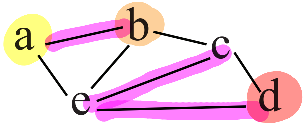
Interval Graphs (a type of intersection graph)

- 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

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





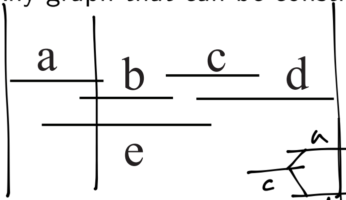
Interval Graphs (a type of intersection graph)

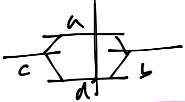
- Interval graphs are intersection graphs where the subsets are intervals/segments $[a, b]$ in $\mathbb{R}$
- Any graph that can be constructed this way is an interval graph

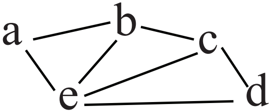
- Are all graphs interval graphs?

Interval Graphs (a type of intersection graph)

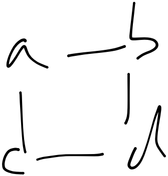
- Interval graphs are intersection graphs where the subsets are intervals/segments $[a, b]$ in $\mathbb{R}$
- Any graph that can be constructed this way is an interval graph







- Are all graphs interval graphs? 4-cycle



Interval Graphs

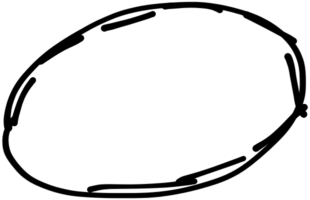
Theorem 8.3.3

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?



Interval Graphs

Theorem 8.3.3

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).

Interval Graphs

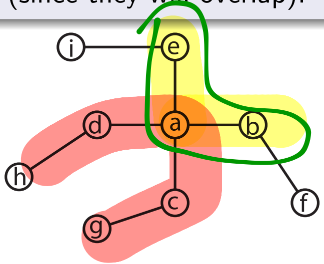
Theorem 8.3.3

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

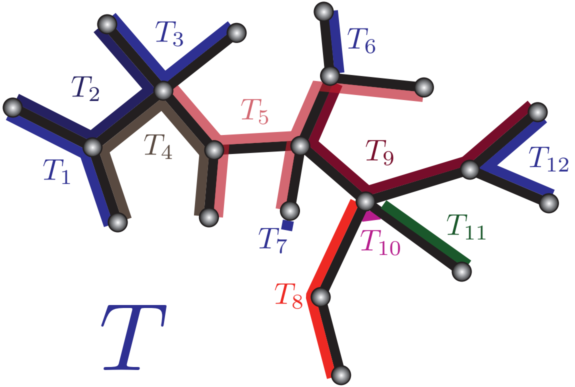
- Given underlying tree, create intersection graph, where subsets U_v for $v \in V(G)$ are (nec. connected) subtrees of some "ground" tree.

Sub-tree intersection Graphs

- Given underlying tree, create intersection graph, where subsets U_v for $v \in V(G)$ are (nec. connected) subtrees of some "ground" tree.
- Intersection exists ($U_u \cap U_v \neq \emptyset$) if there are any nodes in common amongst the two corresponding trees.

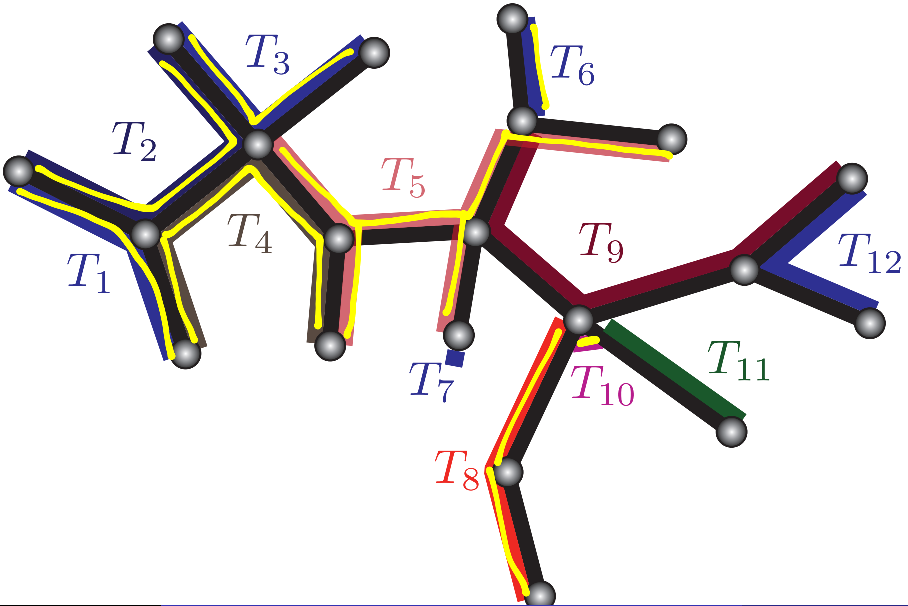
Sub-tree intersection Graphs

- Given underlying tree, create intersection graph, where subsets U_v for $v \in V(G)$ are (nec. connected) subtrees of some "ground" tree.
- Intersection exists ($U_u \cap U_v \neq \emptyset$) if there are any nodes in common amongst the two corresponding trees.

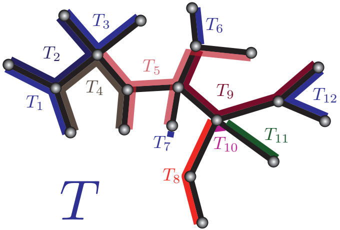


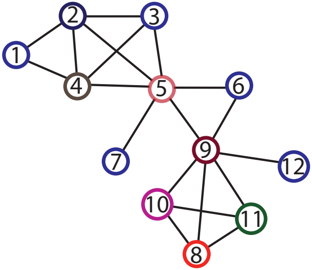
Lets zoom in a little on this

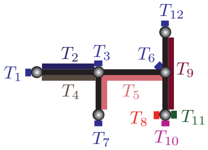
Ex: Sub-tree intersection Graph



Sub-tree intersection Graphs

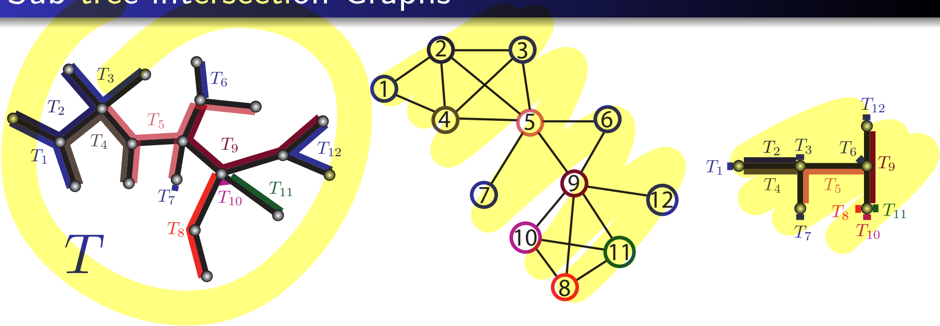






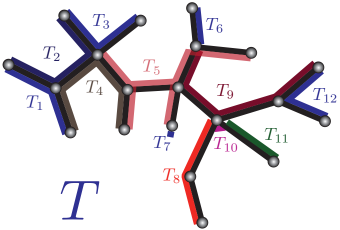
- Intersection exists if there are any nodes in common amongst the two corresponding trees.

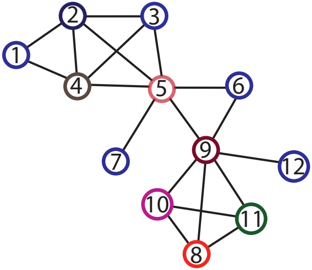
Sub-tree intersection Graphs

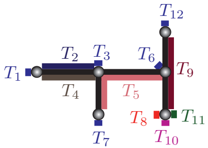


- Intersection exists if there are any nodes in common amongst the two corresponding trees.
- A sub-tree graph corresponds to more than one underlying tree (thus ground set and underlying subsets).

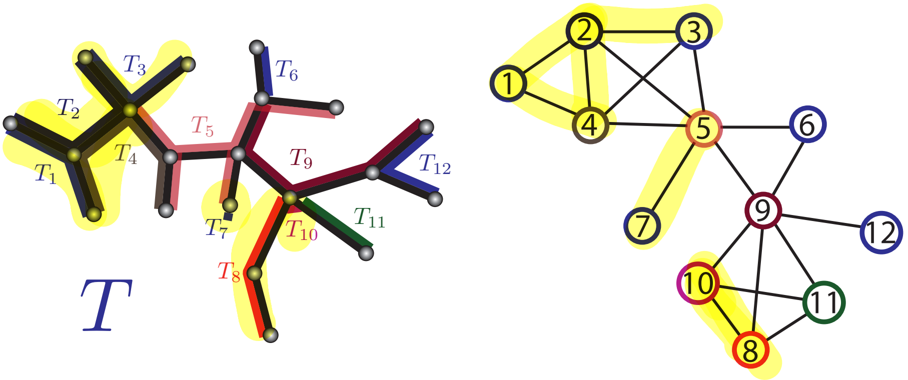
Sub-tree intersection Graphs



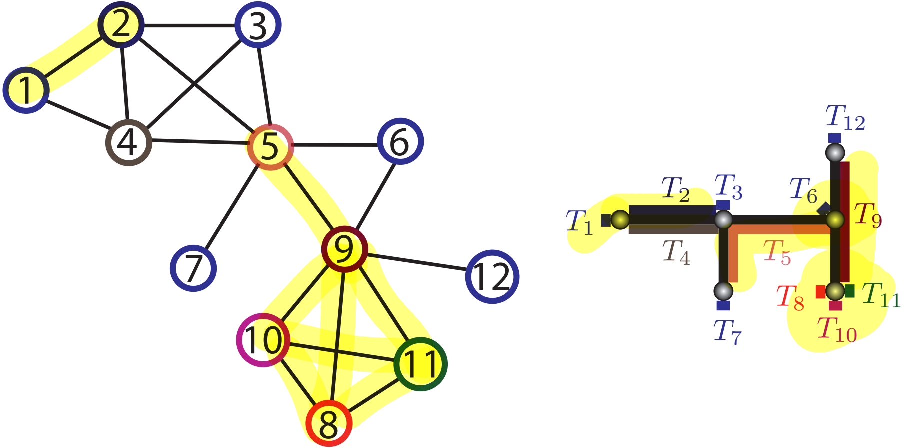




- Intersection exists if there are any nodes in common amongst the two corresponding trees.
- A sub-tree graph corresponds to more than one underlying tree (thus ground set and underlying subsets).
- What is the difference between left and right trees?

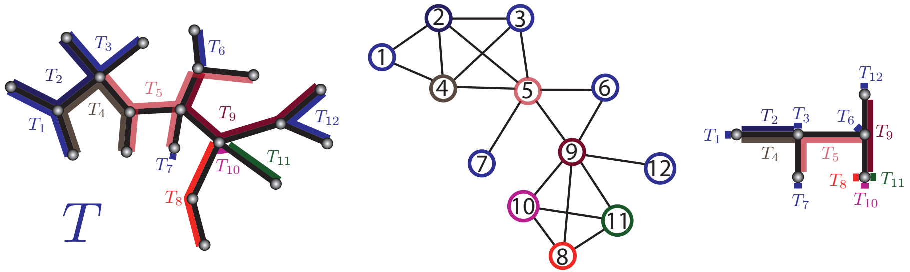


T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10} T_{11} T_{12}
T
1 2 3 4 5 6 7 8 9 10 11 12



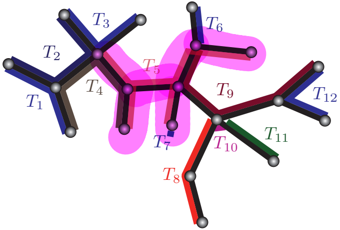
1
2
3
4
5
6
7
8
9
10
11
12
T_1
T_2
T_3
T_4
T_5
T_6
T_7
T_8
T_9
T_{10}
T_{11}
T_{12}

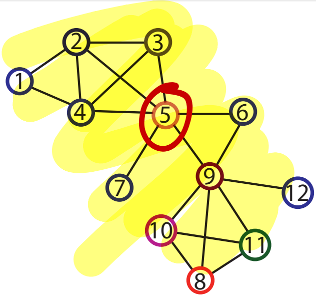
Sub-tree intersection Graphs

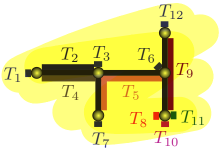


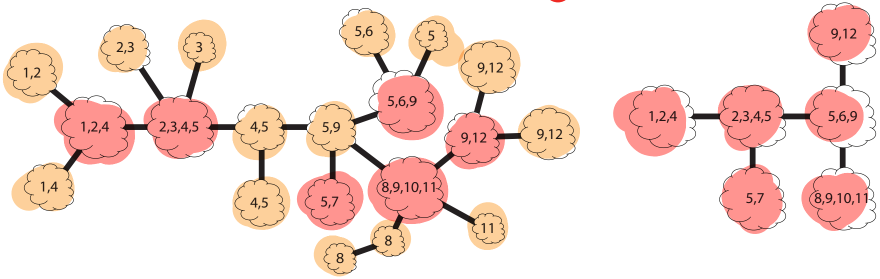
- Intersection exists if there are any nodes in common amongst the two corresponding trees.
- A sub-tree graph corresponds to more than one underlying tree (thus ground set and underlying subsets).
- What is the difference between left and right trees?
- Junction tree of cliques and maxcliques (left)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

Sub-tree intersection Graphs w. Junction Trees









Sub-tree intersection graphs

Theorem 8.3.4

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)

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?

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.

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.

Inference on JTs.

- We can define an inference procedure on junction trees that corresponds to our inference procedure on trees.
- We are given $p \in \mathcal{F}(G', \mathcal{M}^{(\mathrm{f})})$, 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 $\mathcal{F}(G', \mathcal{M}^{(\mathrm{f})})$ we've solved it for the original graph.

Inference on JTs.

- We can define an inference procedure on junction trees that corresponds to our inference procedure on trees.
- We are given $p \in \mathcal{F}(G', \mathcal{M}^{(\mathrm{f})})$, 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 $\mathcal{F}(G', \mathcal{M}^{(\mathrm{f})})$ 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.

Inference on JTs.

- We can define an inference procedure on junction trees that corresponds to our inference procedure on trees.
- We are given $p \in \mathcal{F}(G', \mathcal{M}^{(\mathrm{f})})$, 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 $\mathcal{F}(G', \mathcal{M}^{(\mathrm{f})})$ 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:

$\forall c \in \mathcal{C}(G)$

$p(x_c) = \sum_{x_{V \setminus c}} p(x)$

$$p(x) = \prod_{C \in \mathcal{C}(G)} \psi_C(x_C) \tag{8.1}$$

Inference on JTs.

- Every clique $C \in \mathcal{C}(G)$ is contained in at least one clique $C' \in \mathcal{C}(G')$.

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 $C' \in \mathcal{C}(G')$.

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).

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 $S(G')$ are the set of separators in the junction tree corresponding to triangulated graph G'.

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 $S(G')$ 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

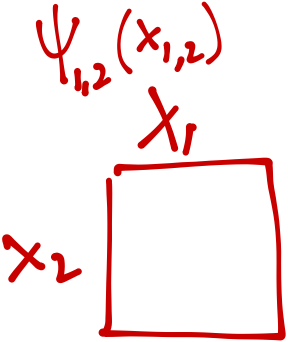
- Initialization Step: For each $C' \in \mathcal{C}(G')$, assign $\psi_{C'}(x_{C'}) = 1$.

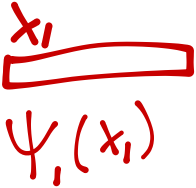
$\forall x_{c'}$

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:

$$\psi_{C'}(x_{C'}) \leftarrow \underbrace{\psi_{C'}(x_{C'})\psi_C(x_C)}_{C \subseteq C'} \tag{8.2}$$





$\psi_{12}(x_1 x_2) \cdot \psi_1(x_1)$
$= \psi'_{12}(x_1, x_2)$

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:

$$\psi_{C'}(x_{C'}) \leftarrow \psi_{C'}(x_{C'})\psi_C(x_C) \tag{8.2}$$

- Crucial: Only do this once, otherwise, we'll be double counting the clique $\psi_C(x_C)$ (i.e., a $C \in \mathcal{C}(G)$ gets assigned only one $C' \in \mathcal{C}(G')$)

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:

$$\psi_{C'}(x_{C'}) \leftarrow \psi_{C'}(x_{C'})\psi_C(x_C) \tag{8.2}$$

- Crucial: Only do this once, otherwise, we'll be double counting the clique $\psi_C(x_C)$ (i.e., a $C \in \mathcal{C}(G)$ gets assigned only one $C' \in \mathcal{C}(G')$)
- We now have the following representation of $p \in \mathcal{F}(G, \mathcal{M}^{(f)})$:

$$p(x) = \prod_{C' \in \mathcal{C}(G')} \psi_{C'}(x_{C'}) \tag{8.3}$$

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:

$$\psi_{C'}(x_{C'}) \leftarrow \psi_{C'}(x_{C'})\psi_C(x_C) \tag{8.2}$$

- Crucial: Only do this once, otherwise, we'll be double counting the clique $\psi_C(x_C)$ (i.e., a $C \in \mathcal{C}(G)$ gets assigned only one $C' \in \mathcal{C}(G')$)
- We now have the following representation of $p \in \mathcal{F}(G, \mathcal{M}^{(\mathrm{f})})$:

$$p(x) = \prod_{C' \in \mathcal{C}(G')} \psi_{C'}(x_{C'}) \tag{8.3}$$

- We also initialize all separators by doing $\phi_S(x_S) = 1 \; \forall S$.

$\forall x_S$

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:

$$\psi_{C'}(x_{C'}) \leftarrow \psi_{C'}(x_{C'})\psi_C(x_C) \tag{8.2}$$

- Crucial: Only do this once, otherwise, we'll be double counting the clique $\psi_C(x_C)$ (i.e., a $C \in \mathcal{C}(G)$ gets assigned only one $C' \in \mathcal{C}(G')$)
- We now have the following representation of $p \in \mathcal{F}(G, \mathcal{M}^{(\mathrm{f})})$:

$$p(x) = \prod_{C' \in \mathcal{C}(G')} \psi_{C'}(x_{C'}) \tag{8.3}$$

- We also initialize all separators by doing $\phi_S(x_S) = 1 \; \forall S$.
- Once this is done, we have

$$p(x) = \frac{\prod_{C' \in \mathcal{C}(G')} \psi_{C'}(x_{C'})}{\prod_{S \in \mathcal{S}(G')} \phi_S(x_S)^{d(S)-1}} \tag{8.4}$$

Maxclique marginals as the goal

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

$$p(x) = \prod_{C' \in \mathcal{C}(G')} \psi_{C'}(x_{C'}) = \frac{\prod_{C \in \mathcal{C}'} p(x_{C'})}{\prod_{S \in \mathcal{S}(G')} p(x_S)^{d(S)-1}} \tag{8.5}$$

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

Maxclique marginals as the goal

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

$$p(x) = \prod_{C' \in \mathcal{C}(G')} \psi_{C'}(x_{C'}) = \frac{\prod_{C \in \mathcal{C}'} p(x_{C'})}{\prod_{S \in \mathcal{S}(G')} p(x_S)^{d(S)-1}} \tag{8.5}$$

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

- If we set $\phi_S(x_S) = 1$ for all S, then

$$p(x) = \frac{\prod_{C \in \mathcal{C}(G')} \psi_C(x_C)}{\prod_{S \in \mathcal{S}(G')} \phi_S(x_S)^{d(S)-1}} \tag{8.6}$$

Maxclique marginals as the goal

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

$$p(x) = \prod_{C' \in \mathcal{C}(G')} \psi_{C'}(x_{C'}) = \frac{\prod_{C \in \mathcal{C}'} p(x_{C'})}{\prod_{S \in \mathcal{S}(G')} p(x_S)^{d(S)-1}} \tag{8.5}$$

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

- If we set $\phi_S(x_S) = 1$ for all S, then

$$p(x) = \frac{\prod_{C \in \mathcal{C}(G')} \psi_C(x_C)}{\prod_{S \in \mathcal{S}(G')} \phi_S(x_S)^{d(S)-1}} \tag{8.6}$$

- In Equation (8.5), we have the functions at each maxclique and at each separator equal to the **marginal distribution** over the corresponding nodes.

Maxclique marginals as the goal

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

Maxclique marginals as the goal

- 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 (??) to the representation at the right of Equation (8.5).

8.6

Maxclique marginals as the goal

- 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 (**??**) to the representation at the right of Equation (8.5).
- Can we do this using a similar message passing procedure to what we've already seen?

Maxclique marginals as the goal

- Start out (after initialization) with the expression

$$p(x) = \frac{\prod_{C' \in \mathcal{C}(G')} \psi_{C'}(x_{C'})}{\prod_{S \in \mathcal{S}(G')} \phi_S(x_S)^{d(S)-1}} \tag{8.7}$$

where $\forall S, \phi_S(x_S) = 1$, and $\psi_{C'}(x_{C'})$ is initialized as described earlier.

- Do message passing, so that we end up with

$$p(x) = \frac{\prod_{C' \in \mathcal{C}(G')} \psi_{C'}(x_{C'})}{\prod_{S \in \mathcal{S}(G')} \phi_S(x_S)^{d(S)-1}} = \frac{\prod_{C \in \mathcal{C}'} p(x_{C'})}{\prod_{S \in \mathcal{S}(G)} p(x_S)^{d(S)-1}} \tag{8.8}$$

- Meaning, $\psi_{C'}(x_{C'}) = p(x_{C'})$ for all C' and $\phi_S(x_S) = p(x_S)$ for all S, marginals.

Marginal Agreement for Agreeable Marginals

- 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.

Marginal Agreement for Agreeable Marginals

- 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.

Marginal Agreement for Agreeable Marginals

- 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?

Marginal Agreement for Agreeable Marginals

- 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.

Marginal Agreement for Agreeable Marginals

- 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.,:

$$\sum_{x_{C'_1 \setminus S}} \psi_{C'_1}(x_{C'_1}) = \sum_{x_{C'_2 \setminus S}} \psi_{C'_2}(x_{C'_2}) \tag{8.9}$$

Marginal Agreement for Agreeable Marginals

- 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.,:

$$\sum_{x_{C'_1 \setminus S}} \psi_{C'_1}(x_{C'_1}) = \sum_{x_{C'_2 \setminus S}} \psi_{C'_2}(x_{C'_2}) \tag{8.9}$$

- Such marginal agreement is a critical idea that also lies at the heart of the approximate inference methods we'll be later covering.

Local Marginal Aggrement

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

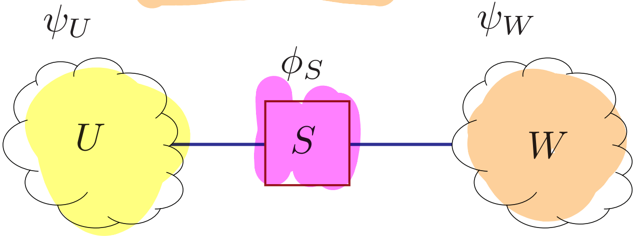
$$\sum_{x_{C_1'\setminus S}} \psi_{C_1'}(x_{C_1'}) = \sum_{x_{C_1'\setminus S}} p(x_{C_1'}) = \sum_{x_{C_2'\setminus S}} p(x_{C_2'}) = \sum_{x_{C_2'\setminus S}} \psi_{C_2'}(x_{C_2'}) \tag{8.10}$$

Local Marginal Aggrement

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

$$\sum_{x_{C'_1 \setminus S}} \psi_{C'_1}(x_{C'_1}) = \sum_{x_{C'_1 \setminus S}} p(x_{C'_1}) = \sum_{x_{C'_2 \setminus S}} p(x_{C'_2}) = \sum_{x_{C'_2 \setminus S}} \psi_{C'_2}(x_{C'_2}) \tag{8.10}$$

- Given two maxcliques U and W with separator $S = U \cap W$, and potential functions ψ_U, ψ_W, and ϕ_S, arranged in small JT as follows:



Maxclique marginals as the goal

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

Maxclique marginals as the goal

- Shorthand notation: $\phi_S^* = \sum_{U \setminus S} \psi_U$ — represents new potential over separator S obtained from ψ_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: shorthand notation

- More shorthand notation: table multiplication

$$\psi_W^* = \frac{\phi_S^*}{\phi_S} \psi_W \qquad (8.11)$$

Maxclique marginals as the goal: shorthand notation

- More shorthand notation: table multiplication

$$\psi_W^* = \frac{\phi_S^*}{\phi_S}\psi_W \tag{8.11}$$

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

$$\psi_W = \psi_W(x_W) = \psi_W(x_S, x_{W_S}), \quad \phi_S = \phi_S(x_S) \tag{8.12}$$

Maxclique marginals as the goal: shorthand notation

- More shorthand notation: table multiplication

$$\psi_W^* = \frac{\phi_S^*}{\phi_S} \psi_W \tag{8.11}$$

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

$$\psi_W = \psi_W(x_W) = \psi_W(x_S, x_{W_S}), \quad \phi_S = \phi_S(x_S) \tag{8.12}$$

and

$$\psi_W^* = \psi_W^*(x_W) = \psi_W^*(x_S, x_{W_S}), \quad \phi_S^* = \phi_S^*(x_S) \tag{8.13}$$

Maxclique marginals as the goal: shorthand notation

- More shorthand notation: table multiplication

$$\psi_W^* = \frac{\phi_S^*}{\phi_S}\psi_W \tag{8.11}$$

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

$$\psi_W = \psi_W(x_W) = \psi_W(x_S, x_{W_S}), \quad \phi_S = \phi_S(x_S) \tag{8.12}$$

and

$$\psi_W^* = \psi_W^*(x_W) = \psi_W^*(x_S, x_{W_S}), \quad \phi_S^* = \phi_S^*(x_S) \tag{8.13}$$

$\forall x_W$

so to expand everything out, we get

$$\frac{\phi_S^*}{\phi_S}\psi_W = \psi_W^* = \psi_W^*(x_S, x_{W_S}) = \frac{\phi_S^*(x_S)}{\phi_S(x_S)}\psi_W(x_S, x_{W_S}) \tag{8.14}$$

Towards Marginal Aggrement

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

$$\sum_{U \setminus S} \psi_U \neq \sum_{W \setminus S} \psi_W \quad \text{and} \quad \phi_S = 1 \tag{8.15}$$

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

Towards Marginal Aggrement

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

$$\sum_{U \setminus S} \psi_U \neq \sum_{W \setminus S} \psi_W \quad \text{and} \quad \phi_S = 1 \tag{8.15}$$

  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.

Towards Marginal Aggrement

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

$$\sum_{U \setminus S} \psi_U \neq \sum_{W \setminus S} \psi_W \quad \text{and} \quad \phi_S = 1 \tag{8.15}$$

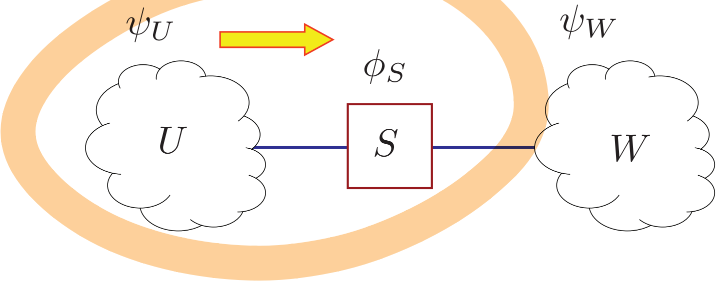
  but we still have that $p(x_U, x_W) = p(x_H, \bar{x}_E) = \psi_U \psi_W / \phi_S$.
- Note (again) that we may treat evidence $\bar{x}_E$ as additional factors contained within a clique and that any summation would only sum over corresponding evidence value, so we can avoid mentioning evidence for now.
- What we'll do: exchange information between cliques via separators to achieve consistency.

New separator potential to obtain new marginal

- **Marginalize U:**

$$\phi_S^* = \sum_{U \setminus S} \psi_U \tag{8.16}$$

which leads to a new separator potential ϕ_S^* and can be seen as a partial message, as shown in the following figure

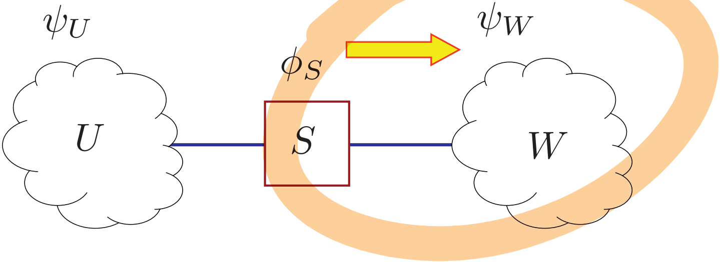


Updated W marginal based on separator

- **Rescale W:**

$$\psi_W^* = \frac{\phi_S^*}{\phi_S}\psi_W \tag{8.17}$$

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



Updated distribution unchanged

alias

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

$$\frac{\psi^*_U \psi^*_W}{\phi^*_S} = \frac{\psi_U \psi_W \cancel{\phi^*_S}}{\phi_S \cancel{\phi^*_S}} = \frac{\psi_U \psi_W}{\phi_S} \tag{8.18}$$

Updated distribution unchanged

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

$$\frac{\psi_U^* \psi_W^*}{\phi_S^*} = \frac{\psi_U \psi_W \phi_S^*}{\phi_S \phi_S^*} = \frac{\psi_U \psi_W}{\phi_S} \tag{8.18}$$

- Don't yet (nec.) have consistency since could have

$$\sum_{U\setminus S} \psi_U^* = \sum_{U\setminus S} \psi_U = \phi_S^* \neq \sum_{W\setminus S} \psi_W^* = \frac{\phi_S^*}{\phi_S} \sum_{W\setminus S} \psi_W \tag{8.19}$$

which follows because we still could have that

$$= \phi_S \neq \sum_{W\setminus S} \psi_W \tag{8.20}$$

Progress towards marginals

$$p(x) \cdot \delta(x_E = \bar{x}_E) = p(x) \cdot \prod_{e \in E} \delta(x_e = \bar{x}_e)$$

- We do at least have one marginal at ψ_W^*. This is because we started with:

$$p(x_H, \bar{x}_E) = p(x) = p(x_U, x_W) = \frac{\psi_U \psi_W}{\phi_S} \tag{8.21}$$

and

$$\psi_W^* = \frac{\phi_S^*}{\phi_S} \psi_W = \psi_W \sum_{U \setminus S} \psi_U = \sum_{x_{U \setminus S}} p(x_H, \bar{x}_E) = p(x_W) \tag{8.22}$$

is one of the marginals that we desire.

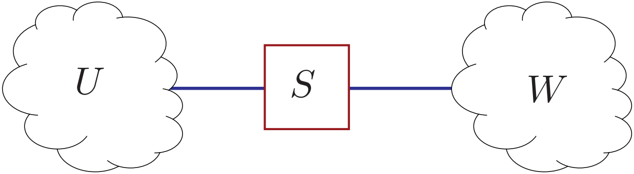
$$\sum_{x_{U \setminus S}} g(x_{U \setminus S}, x_S, x_{W \setminus S}) = \bar{g}(x_S, x_{W \setminus S}) = \bar{g}(x_W)$$

Message in a junction tree

- We see this as a message passing procedure, passing a message between two nodes in a cluster (or junction) tree.

Message in a junction tree

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



$$\psi_U \Rightarrow \phi_S^* \Rightarrow \psi_W^*$$

Send message back

- 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**:**

$$\phi_S^{**} = \sum_{W \setminus S} \psi_W^* \tag{8.23}$$

resulting in still another separator potential. And then

Update initial marginal at U

- **Rescale U:**

$$\psi_U^{**} = \frac{\phi_S^{**}}{\phi_S^{*}} \psi_U^{*} = \psi_U \tag{8.24}$$

resulting in a new potential on U.

Update initial marginal at U

- **Rescale U:**

$$\psi_U^{**} = \frac{\phi_S^{**}}{\phi_S^{*}} \psi_U^{*} \tag{8.24}$$

resulting in a new potential on U.

- Intuition: ϕ_S^{**} and ψ_U^{*} both "contain" ϕ_S^{*} so we divide it out in the computation of ψ_U^{**} so that ψ_U^{**} doesn't end up double counting ϕ_S^{*}.

Maxclique marginals as the goal

alias

- The new joint $p(x_U, x_W)$ has again not changed. Define $\psi^{**}_W = \psi^*_W$ for convenience, we get:

$$p(x) = \frac{\psi^{**}_U \psi^{**}_W}{\phi^{**}_S} = \frac{\psi_U \cancel{\phi^{**}_S} \psi_W \cancel{\phi^*_S}}{\cancel{\phi^{**}_S} \phi_S \cancel{\phi^*_S}} = \frac{\psi_U \psi_W}{\phi_S} \tag{8.25}$$

Maxclique marginals as the goal

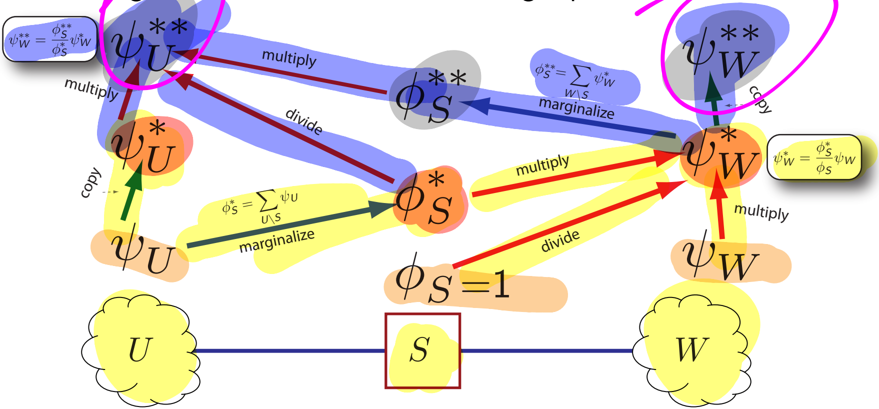
- More importantly, after backwards message, we indeed have consistency guaranteed.
- In particular, ψ_U^{**} and ψ_W^{**} are now consistent since:

ψ_V

$$\sum_{U\setminus S} \psi_U^{**} = \sum_{U\setminus S} \frac{\phi_S^{**}}{\phi_S^{*}} \psi_U^{*} = \frac{\phi_S^{**}}{\phi_S^{*}} \sum_{U\setminus S} \psi_U^{*} = \frac{\phi_S^{**}}{\cancel{\phi_S^{*}}} \boxed{\cancel{\phi_S^{*}}} = \phi_S^{**} = \sum_{W\setminus S} \psi_W^{**} \tag{8.26}$$

Forward/Backward Messages Along Cluster Tree Edge

Summarizing, forward and backwards messages proceed as follows:



Recall: $S = U \cap W$, and we initialize ψ_U and ψ_W with factors that are contained in U or W.

Marginal at U achieved

We moreover have the other marginal we want at ψ_U^{**} since:

$$\psi_U^{**} = \frac{\phi_S^{**}}{\phi_S^{*}}\psi_U = \psi_U \frac{\sum_{W \setminus S} \psi_W^{*}}{\sum_{U \setminus S} \psi_U}$$

Marginal at U achieved

We moreover have the other marginal we want at ψ_U^{**} since:

$$\psi_U^{**} = \frac{\phi_S^{**}}{\phi_S^{*}}\psi_U = \psi_U \frac{\sum_{W \setminus S} \psi_W^{*}}{\sum_{U \setminus S} \psi_U} = \psi_U \frac{\sum_{W \setminus S} \frac{\phi_S^{*}}{\phi_S}\psi_W}{\sum_{U \setminus S} \psi_U}$$

Marginal at U achieved

We moreover have the other marginal we want at ψ_U^{**} since:

$$\psi_U^{**} = \frac{\phi_S^{**}}{\phi_S^*}\psi_U = \psi_U \frac{\sum_{W \setminus S} \psi_W^*}{\sum_{U \setminus S} \psi_U} = \psi_U \frac{\sum_{W \setminus S} \frac{\phi_S^*}{\cancel{\phi_S}} \psi_W}{\sum_{U \setminus S} \psi_U}$$

$$= \psi_U \frac{\sum_{W \setminus S} \psi_W \sum_{U \setminus S} \psi_U}{\sum_{U \setminus S} \psi_U}$$

Marginal at U achieved

We moreover have the other marginal we want at ψ_U^{**} since:

$$\psi_U^{**} = \frac{\phi_S^{**}}{\phi_S^{*}}\psi_U = \psi_U \frac{\sum_{W \setminus S} \psi_W^*}{\sum_{U \setminus S} \psi_U} = \psi_U \frac{\sum_{W \setminus S} \frac{\phi_S^*}{\phi_S}\psi_W}{\sum_{U \setminus S} \psi_U}$$

$$= \psi_U \frac{\sum_{W \setminus S} \psi_W \cancel{\sum_{U \setminus S} \psi_U}}{\cancel{\sum_{U \setminus S} \psi_U}} = \psi_U \sum_{W \setminus S} \psi_W$$

$$= \sum_{W \setminus S} \psi_U \psi_W =$$

Marginal at U achieved

We moreover have the other marginal we want at ψ_U^{**} since:

$$\psi_U^{**} = \frac{\phi_S^{**}}{\phi_S^*}\psi_U = \psi_U \frac{\sum_{W \setminus S} \psi_W^*}{\sum_{U \setminus S} \psi_U} = \psi_U \frac{\sum_{W \setminus S} \frac{\phi_S^*}{\phi_S} \psi_W}{\sum_{U \setminus S} \psi_U}$$

$$= \psi_U \frac{\sum_{W \setminus S} \psi_W \sum_{U \setminus S} \psi_U}{\sum_{U \setminus S} \psi_U} = \psi_U \sum_{W \setminus S} \psi_W = \sum_{W \setminus S} p(x_U, x_W)$$

Marginal at U achieved

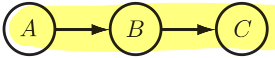
We moreover have the other marginal we want at ψ_U^{**} since:

$$\psi_U^{**} = \frac{\phi_S^{**}}{\phi_S^*}\psi_U = \psi_U \frac{\sum_{W\setminus S}\psi_W^*}{\sum_{U\setminus S}\psi_U} = \psi_U \frac{\sum_{W\setminus S}\frac{\phi_S^*}{\phi_S}\psi_W}{\sum_{U\setminus S}\psi_U}$$

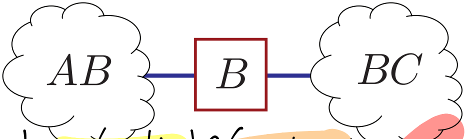
$$= \psi_U \frac{\sum_{W\setminus S}\psi_W \sum_{U\setminus S}\psi_U}{\sum_{U\setminus S}\psi_U} = \psi_U \sum_{W\setminus S}\psi_W = \sum_{W\setminus S} p(x_U, x_W)$$

$$= p(x_U)$$

BN Example: $A \to B \to C$ with evidence



A — B — C



$p(x) = p(x_C|x_B)\,p(x_B|x_A)\,p(x_A)$

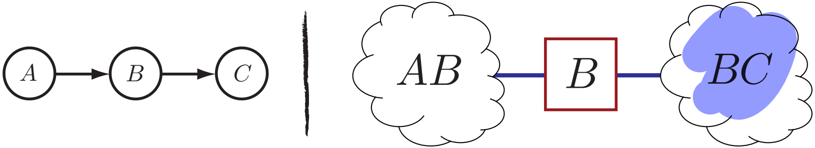
- Bayesian network, three state Markov chain.
- After moralization and triangulation (which is vacuous), we get maxclique functions $\psi_{AB}(x_A, x_B)$ and $\psi_{BC}(x_B, x_C)$.
- With evidence, we have $x_C = 1$. We initialize clique and separator functions as follows:

$$\psi_{AB}(x_A, x_B) = p(x_B|x_A)p(x_A) = p(x_A, x_B) \quad (8.27)$$

$$\psi_{BC}(x_B, x_C) = p(x_C|x_B)\delta(x_C, 1) \quad (8.28)$$

$$\phi_B(x_B) = 1 \quad (8.29)$$

BN Example: $A \to B \to C$ with evidence



- Forward (left-to-right) message:

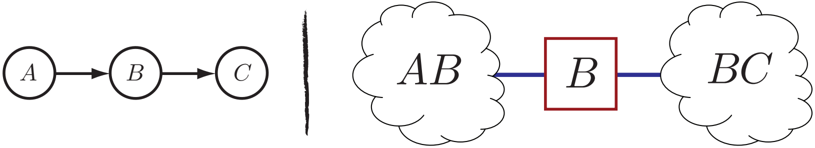
$$\phi_B^*(x_B) = \sum_{x_A} p(x_A, x_B) = p(x_B) \tag{8.30}$$

$$\psi_{BC}^*(x_B, x_C) = \frac{p(x_B)}{1} p(x_C|x_B)\delta(x_C, 1) \tag{8.31}$$

$$= p(x_B, x_C)\delta(x_C, 1) \tag{8.32}$$

$$= p(x_B, x_C = 1) \tag{8.33}$$

BN Example: $A \to B \to C$ with evidence



- Backwards (right-to-left) message

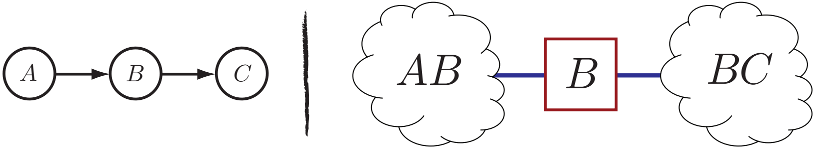
$$\phi_B^{**}(x_B) = \sum_{x_C} p(x_B, x_C)\delta(x_C, 1) = p(x_B, x_C = 1) \tag{8.34}$$

$$\psi_{AB}^{**}(x_A, x_B) = \frac{\phi_B^{**}}{\phi_B^{*}}\psi_{AB}^{*} \tag{8.35}$$

$$= \frac{p(B, C = 1)}{p(B)} p(A, B) = p(A|B)\frac{p(B)}{p(B)} p(B, C = 1) \tag{8.36}$$

$$= p(A|B, C = 1)p(B, C = 1) = p(A, B, C = 1) \tag{8.37}$$

BN Example: $A \to B \to C$ with evidence



- We are left with the maxclique functions as marginals, i.e., we have:

$$\psi^*_{BC}(x_B, x_C) = p(x_B, x_C = 1) \tag{8.38}$$

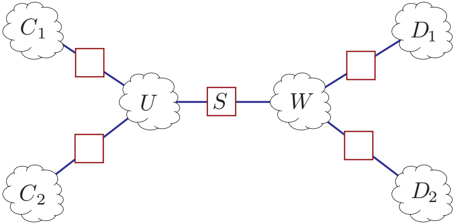
$$\psi^{**}_{AB}(x_A, x_B) = p(x_A, x_B, x_C = 1) \tag{8.39}$$

- ... from which it is easy to construct, say, maxclique conditionals, e.g., $p(x_B|x_C = 1)$, $p(x_A, x_B|C = 1)$, etc.

$\psi(x_C) = p(x_{C \setminus E}, \bar{x}_E)$

Less simple example: general tree

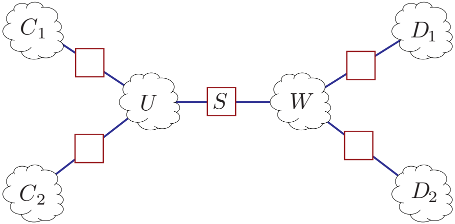
How to ensure any local consistency we achieved not ruined by later message passing steps?



E.g. once we send message $U \to W$ and then $W \to U$, we know W and U are consistent. If we next send messages $W \to D_1$ and $D_1 \to W$, then W & D_1 are consistent, but U & W might no longer be consistent.

Less simple example: general tree

How to ensure any local consistency we achieved not ruined by later message passing steps?



E.g. once we send message $U \to W$ and then $W \to U$, we know W and U are consistent. If we next send messages $W \to D_1$ and $D_1 \to W$, then W & D_1 are consistent, but U & W might no longer be consistent.

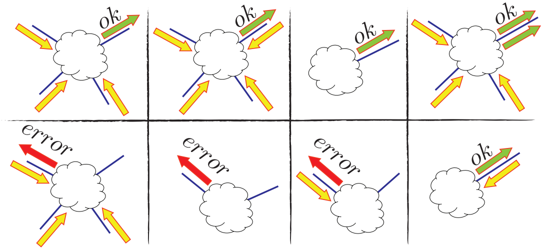
Basic problem, future messages might mess up achieved local marginal consistency.

Ensuring consistency over all marginals

We use same scheme we saw for 1-trees. I.e., recall from earlier lectures:

Definition 8.4.1 (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?

Maxclique marginals as the goal

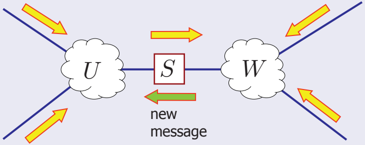
general trees

Theorem 8.4.2

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

Proof.

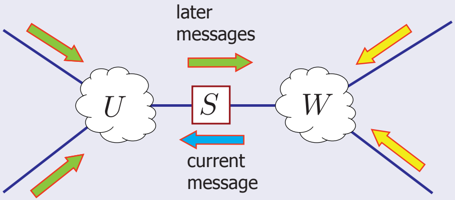
Suppose W has received a message from all other neighbors, and is sending a message to U. There are two possible cases: Case A: U already sent a message to W before, so U already received message from all other neighbors, & message renders the two consistent since neither receives any more messages.



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.



$\square$

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.

Maxclique marginals as the goal

- Above procedure works for two cliques (clique functions are marginals)

Maxclique marginals as the goal

- Above procedure works for two cliques (clique functions are marginals)
- For a general Junction Tree, when we send messages abiding MPP, we get:

Maxclique marginals as the goal

- Above procedure works for two cliques (clique functions are marginals)
- For a general Junction Tree, when we send messages abiding MPP, we get:

Theorem 8.4.3

Sending all messages along a cluster tree following message passing protocol renders the cliques locally consistent between all pairs of connected cliques in the tree.

Maxclique marginals as the goal

- Above procedure works for two cliques (clique functions are marginals)
- For a general Junction Tree, when we send messages abiding MPP, we get:

Theorem 8.4.3

Sending all messages along a cluster tree following message passing protocol renders the cliques locally consistent between all pairs of connected cliques in the tree.

- Note, we need only that it is a cluster tree. Result holds even if r.i.p. not satisfied.

Maxclique marginals as the goal

- Above procedure works for two cliques (clique functions are marginals)
- For a general Junction Tree, when we send messages abiding MPP, we get:

Theorem 8.4.3

Sending all messages along a cluster tree following message passing protocol renders the cliques locally consistent between all pairs of connected cliques in the tree.

- Note, we need only that it is a cluster tree. Result holds even if r.i.p. not satisfied.
- But we want more than this, we want to ensure that potentials over any two clusters, with common variables, agree on their common variables.

Local implies global consistency

Theorem 8.4.4

In any JT of clusters, any configuration of cluster functions that are locally (neighbor) consistent will be globally consistent. I.e., for any clusters pair C_1, C_2 with $C_1 \cap C_2 \neq \emptyset$ we have:

$$\psi_{C_1}(x_{C_1 \cap C_2}) = \psi_{C_2}(x_{C_1 \cap C_2}) \tag{8.40}$$

for all values $x_{C_1 \cap C_2}$.

Local implies global consistency

Theorem 8.4.4

In any JT of clusters, any configuration of cluster functions that are locally (neighbor) consistent will be globally consistent. I.e., for any clusters pair C_1, C_2 *with* $C_1 \cap C_2 \neq \emptyset$ *we have:*

$$\psi_{C_1}(x_{C_1 \cap C_2}) = \psi_{C_2}(x_{C_1 \cap C_2}) \tag{8.40}$$

for all values $x_{C_1 \cap C_2}$.

Proof.

Local consistency implies that for neighboring C_1, C_2, the above equality holds. For non-neighboring C_1, C_2, cluster intersection property (r.i.p.) ensures that intersection $C_1 \cap C_2$ exists along unique path between C_1 and C_2. Each edge along that path is locally consistent. By transitivity, each distance-2 pair is consistent. Repeating this argument for any path length gives the result. □

Consistency gives Marginals

Theorem 8.4.5

Given junction tree of clusters $\mathcal{C}$ and separators $\mathcal{S}$, and given above initialization, after all messages are sent and obey MPP, cluster and separator potentials will reach the marginal state:

$$\psi_C(x_C) = p(x_C) \text{ and } \phi_S(x_S) = p(x_S) \tag{8.41}$$

Consistency gives Marginals

Theorem 8.4.5

Given junction tree of clusters $\mathcal{C}$ and separators $\mathcal{S}$, and given above initialization, after all messages are sent and obey MPP, cluster and separator potentials will reach the marginal state:

$$\psi_C(x_C) = p(x_C) \text{ and } \phi_S(x_S) = p(x_S) \tag{8.41}$$

Proof.

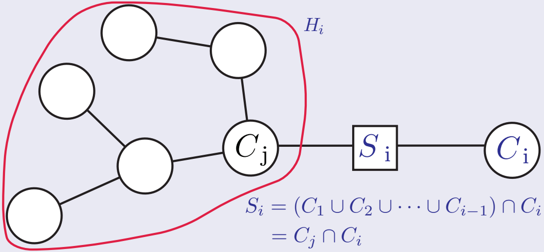
Separators are marginalizations of clusters, so ensuring clusters are marginals is sufficient for separators as marginals.

Induction: base case: One cluster is a marginal. Two clusters reach marginals (we verified above).

Assume true for $i-1$ clusters marginals, and show for i. Given JT with clusters $C_1, \ldots, C_{i-1}$ and add new cluster C_i connecting to C_j and obeying r.i.p. We have separator $S_i = C_i \cap C_j$.

Consistency gives Marginals

... proof continued.



We have (as always) $p(x) = p(x_V)$ and that

$$p(x_V) = p(x_{C_i \setminus S_i}, x_{S_i}, x_{V \setminus C_i}) = p(x_{C_i \setminus S_i} | x_{S_i}) p(x_{S_i \cup (V \setminus C_i)}) \tag{8.42}$$

due to conditional independence property of sepator S

$$X_{C_i \setminus S_i} \perp\!\!\!\perp X_{V \setminus C_i} | X_S \tag{8.43}$$

Consistency gives Marginals

... proof continued.

We have:

$$p(x_{S_i \cup (V \setminus C_i)}) = \sum_{x_{C_i \setminus S_i}} p(x_V) = \sum_{x_{C_i \setminus S_i}} \frac{\prod_{C \in \mathcal{C}} \psi_C(x_C)}{\prod_{S \in \mathcal{S}} \phi_S(x_S)^{d(S)-1}} \tag{8.44}$$

$$= \sum_{x_{C_i \setminus S_i}} \frac{\psi_{C_i}(x_{C_i}) \prod_{C \neq C_i} \psi_C(x_C)}{\phi_{S_i}(x_{S_i}) \prod_{S \in \mathcal{S}} \phi_S(x_S)^{d'(S)-1}} \tag{8.45}$$

$$= \frac{\sum_{x_{C_i \setminus S_i}} \psi_{C_i}(x_{C_i})}{\phi_{S_i}(x_{S_i})} \frac{\prod_{C \neq C_i} \psi_C(x_C)}{\prod_{S \in \mathcal{S}} \phi_S(x_S)^{d'(S)-1}} \tag{8.46}$$

$$= \frac{\prod_{C \neq C_i} \psi_C(x_C)}{\prod_{S \in \mathcal{S}} \phi_S(x_S)^{d'(S)-1}} \tag{8.47}$$

since $\sum_{x_{C_i \setminus S_i}} \psi_{C_i}(x_{C_i}) = \phi_{S_i}(x_{S_i})$ and since the only cluster containing $C_i \setminus S_i$ is C_i. $d'(S) = d(S)$ except at S_i where one less.

Consistency gives Marginals

... proof continued.

With only $i-1$ cliques, after message passing is performed, JT will have cluster functions as marginals (by induction). We need to show that $\psi_{C_i}(x_{C_i})$ is also a valid marginal. After MP, we have local and global consistency, so

$$\phi_{S_i}(x_{S_i}) = \sum_{x_{C_j \setminus S_i}} \psi_{C_j}(x_{C_j}) \tag{8.48}$$

and by induction we have that $\psi_{C_j}(x_{C_j}) = p(x_{C_j})$ giving:

$$p(x_{C_i \setminus S_i} | x_{S_i}) = \frac{p(x)}{p(x_{S_i \cup (V \setminus C_i)})} = \frac{\frac{\prod_{C \in \mathcal{C}} \psi_C(x_C)}{\prod_{S \in \mathcal{S}} \phi_S(x_S)^{d(S)-1}}}{\frac{\prod_{C \neq C_i} \psi_C(x_C)}{\prod_{S \in \mathcal{S}} \phi_S(x_S)^{d'(S)-1}}}, \tag{8.49}$$

where the first equality follows from Equation (8.42).

Consistency gives Marginals

... proof continued.

which yields

$$p(x_{C_i \setminus S_i} | x_{S_i}) = \frac{\psi_{C_i}(x_{C_i})}{\phi_{S_i}(x_{S_i})} = \frac{\psi_{C_i}(x_{C_i})}{p(x_{S_i})} \tag{8.50}$$

this then gives that:

$$\psi_{C_i}(x_{C_i}) = p(x_{C_i \setminus S_i} | x_{S_i}) p(x_{S_i}) = p(x_{C_i}) \tag{8.51}$$

a marginal as desired. □

Redundant Messages

- Once all messages have been sent according to MPP, what would happen if we send more messages?
- 1-tree formulation:

$$\mu_{i\to j}(x_j) = \sum_{x_i} \psi_{i,j}(x_i, x_j) \prod_{k\in\delta(i)\setminus\{j\}} \mu_{k\to i}(x_i) \tag{8.52}$$

- Junction-tree formulation: marginalize and rescale

$$\phi_S^{\text{new}} = \sum_{U\setminus S} \psi_U \text{ and then } \psi_W^{\text{new}} = \frac{\phi_S^{\text{new}}}{\phi_S^{\text{old}}} \psi_W \tag{8.53}$$

Redundant Messages

- Once all messages have been sent according to MPP, what would happen if we send more messages?
- 1-tree formulation:

$$\mu_{i\to j}(x_j) = \sum_{x_i} \psi_{i,j}(x_i, x_j) \prod_{k\in\delta(i)\setminus\{j\}} \mu_{k\to i}(x_i) \tag{8.52}$$

- Junction-tree formulation: marginalize and rescale

$$\phi_S^{\text{new}} = \sum_{U\setminus S} \psi_U \text{ and then } \psi_W^{\text{new}} = \frac{\phi_S^{\text{new}}}{\phi_S^{\text{old}}} \psi_W \tag{8.53}$$

- In either case, extra messages would not change functions - they're redundant, joint "state" has "converged" since $\phi_S^{\text{new}} = \phi_S^{\text{old}}$.
- all messages could run in parallel, convergence achieved once we've done D parallel steps where D is tree diameter.

Distributive Law and Other Objects

- Only one property needed for this algorithm to work, namely distributive law $ab + ac = a(b + c)$ along with factorization.
- Distributive law allows sending sums inside of factors.
- Other objects have distribute law, and in general any set of objects that is a commutative semiring will work as well

Commutative Semirings

Definition 8.5.1

A *commutative semiring* is a set K with two binary operators "$+$" and "$\cdot$" having three axioms, for all $a, b, c \in K$.

S1: "$+$" is commutative $(a + b) = (b + a)$ and associative $(a + b) + c = a + (b + c)$, and $\exists$ additive identity called "0" such that $k + 0 = k$ for all $k \in K$. I.e., $(K, +)$ is a commutative monoid.

S2: "$\cdot$" is also associative, commutative, and $\exists$ multiplicative identity called "1" s.t. $k \cdot 1 = k$ for all $k \in K$ ($(K, \cdot)$ is also a comm. monoid).

S3: distributive law holds: $(a \cdot b) + (a \cdot c) = a(b + c)$ for all $a, b, c \in K$.

This, and factorization w.r.t. a graph G is all that is necessary for the above message passing algorithms to work. There are many commutative semirings.

Commutative Semirings

- Additive inverse need not exist. If additive inverse exists, then we get a commutative ring ("semi-ring" since we need not have additive inverse). Note, in algebra texts, a ring often doesn't require multiplicative identity, but we assume it exists here.

Commutative Semirings

- Additive inverse need not exist. If additive inverse exists, then we get a commutative ring ("semi-ring" since we need not have additive inverse). Note, in algebra texts, a ring often doesn't require multiplicative identity, but we assume it exists here.
- Above definition does not mention $0 \cdot k = 0$, but this follows from above properties since $k \cdot k = k(k+0) = k \cdot k + k \cdot 0$ so that $k0$ must also be an additive identity, meaning that $k \cdot 0 = 0$. This is useful with evidence witih delta functions, where the delta functions multiplies by zero anything that does not obide by the evidence value.

Commutative Semirings

- Additive inverse need not exist. If additive inverse exists, then we get a commutative ring ("semi-ring" since we need not have additive inverse). Note, in algebra texts, a ring often doesn't require multiplicative identity, but we assume it exists here.
- Above definition does not mention $0 \cdot k = 0$, but this follows from above properties since $k \cdot k = k(k+0) = k \cdot k + k \cdot 0$ so that $k0$ must also be an additive identity, meaning that $k \cdot 0 = 0$. This is useful with evidence witih delta functions, where the delta functions multiplies by zero anything that does not obide by the evidence value.
- Same message passing protocol and message passing scheme on a junction tree will work to ensure that all clusters reach a state where they are the appropriate "marginals"

Commutative Semirings

- Additive inverse need not exist. If additive inverse exists, then we get a commutative ring ("semi-ring" since we need not have additive inverse). Note, in algebra texts, a ring often doesn't require multiplicative identity, but we assume it exists here.
- Above definition does not mention $0 \cdot k = 0$, but this follows from above properties since $k \cdot k = k(k+0) = k \cdot k + k \cdot 0$ so that $k0$ must also be an additive identity, meaning that $k \cdot 0 = 0$. This is useful with evidence witih delta functions, where the delta functions multiplies by zero anything that does not obide by the evidence value.
- Same message passing protocol and message passing scheme on a junction tree will work to ensure that all clusters reach a state where they are the appropriate "marginals"
- Marginals in this case dependent on ring.

Other Semi-Rings

Here, A denotes arbitrary commutative semiring, S is arbitrary finite set, Λ is arbitrary distributed lattice.

	K	"$(+,0)$"	"$(\cdot,1)$"	short name
1	A	$(+,0)$	$(\cdot,1)$	semiring
2	$A[x]$	$(+,0)$	$(\cdot,1)$	polynomial
3	$A[x,y,\dots]$	$(+,0)$	$(\cdot,1)$	polynomial
4	$[0,\infty)$	$(+,0)$	$(\cdot,1)$	sum-product
5	$(0,\infty]$	$(\min,\infty)$	$(\cdot,1)$	min-product
6	$[0,\infty)$	$(\max,0)$	$(\cdot,1)$	max-product
7	$[0,\infty)+$	$(\text{kmax},0)$	$(\cdot,1)$	k-max-product
8	$(-\infty,\infty]$	$(\min,\infty)$	$(+,0)$	min-sum
9	$[-\infty,\infty)$	$(\max,-\infty)$	$(+,0)$	max-sum
10	$\{0,1\}$	$(\text{OR},0)$	$(\text{AND},1)$	Boolean
11	2^S	$(\cup,\emptyset)$	$(\cap,S)$	Set
12	Λ	$(\vee,0)$	$(\wedge,1)$	Lattice
13	Λ	$(\wedge,1)$	$(\vee,0)$	Lattice

Example: Viterbi/MPE

- Most-probable explanation (e.g., Viterbi assignment) is just the max-product ring.
- Here, we wish to compute

$$\operatorname*{argmax}_{x_{V\setminus E}} p(x_{V\setminus E}, \bar{x}_E) \tag{8.54}$$

- After message passing with the max-product ring on a junction tree, cluster functions will reach the "max-marginal" state, where we have:

$$\psi_C(x_C) = \max_{x_{V\setminus C}} p(x_C, x_{V\setminus C}) \tag{8.55}$$

- What about a "k-max" operation (i.e., finding the k highest scoring assignments to the variables?) How would we define the operators "$+$" and "$\cdot$"?

Recap

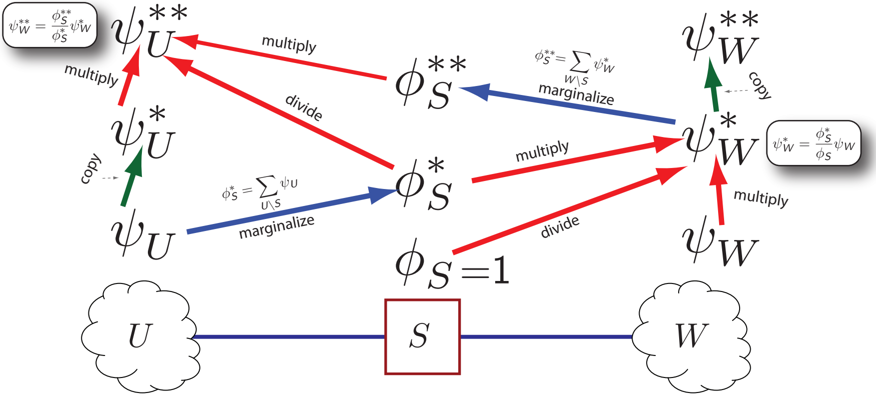
- Message passing on junction tree nodes, definition of messages, divide out old, multiply in new.

Recap

- Message passing on junction tree nodes, definition of messages, divide out old, multiply in new.
- Messages in both directions.

Forward/Backward Messages Along Cluster Tree Edge

Summarizing, forward and backwards messages proceed as follows:



Recall: $S = U \cap W$, and we initialize ψ_U and ψ_W with factors that are contained in U or W.

Recap

- Message passing on junction tree nodes, definition of messages, divide out old, multiply in new.
- Messages in both directions.
- For general tree, we have MPP like in 1-tree case.

Recap

- Message passing on junction tree nodes, definition of messages, divide out old, multiply in new.
- Messages in both directions.
- For general tree, we have MPP like in 1-tree case.
- Suff condition: locally consistent.

Recap

- Message passing on junction tree nodes, definition of messages, divide out old, multiply in new.
- Messages in both directions.
- For general tree, we have MPP like in 1-tree case.
- Suff condition: locally consistent.
- Thm: MPP renders cliques locally consistent between pairs.

Recap

- Message passing on junction tree nodes, definition of messages, divide out old, multiply in new.
- Messages in both directions.
- For general tree, we have MPP like in 1-tree case.
- Suff condition: locally consistent.
- Thm: MPP renders cliques locally consistent between pairs.
- In JT (r.i.p.) locally consistent ensures globally consistent.

Recap

- Message passing on junction tree nodes, definition of messages, divide out old, multiply in new.
- Messages in both directions.
- For general tree, we have MPP like in 1-tree case.
- Suff condition: locally consistent.
- Thm: MPP renders cliques locally consistent between pairs.
- In JT (r.i.p.) locally consistent ensures globally consistent.
- In JT (r.i.p.), running MPP gives marginals.

Recap

- Message passing on junction tree nodes, definition of messages, divide out old, multiply in new.
- Messages in both directions.
- For general tree, we have MPP like in 1-tree case.
- Suff condition: locally consistent.
- Thm: MPP renders cliques locally consistent between pairs.
- In JT (r.i.p.) locally consistent ensures globally consistent.
- In JT (r.i.p.), running MPP gives marginals.
- Commutative semiring - other algebraic objects can be used.

Recap

- Message passing on junction tree nodes, definition of messages, divide out old, multiply in new.
- Messages in both directions.
- For general tree, we have MPP like in 1-tree case.
- Suff condition: locally consistent.
- Thm: MPP renders cliques locally consistent between pairs.
- In JT (r.i.p.) locally consistent ensures globally consistent.
- In JT (r.i.p.), running MPP gives marginals.
- Commutative semiring - other algebraic objects can be used.
- Time and memory complexity is $O(Nr^{\omega+1})$ where ω is the tree-width.

Sources for Today's Lecture

- Most of this material comes from the reading handout `tree_inference.pdf`