Inference through message passing

- Recall that Markov nets factor over cliques,
  \[ P(X) = P(X_1, \ldots, X_N) = \frac{1}{Z} \prod_{i=1}^{N} \phi_i(A_i) \]
  
  - We can assign each of these factors to one of the nodes in the clique tree,
  
  \[ P(X) = \frac{1}{Z} \prod_j \phi_j(C_j) \]

Sum-product belief propagation

- Performs inference on a clique tree
- Instead of sending messages in one direction ("up" the tree), nodes send messages in all directions
  - Algorithm is almost exactly the same
  - Each node \( C_i \) sends a message to each of its neighbors \( C_j \)
  
  \[ \delta_{i \to j}(S_{i,j}) = \sum_{C_j} \psi_i(C_i) \prod_{k \in X(i) \setminus \{j\}} \delta_{k \to j}(S_{k,i}) \]
  
  - Where \( S_{i,j} = C_i \cap C_j \)
  - Note that message sent to \( j \) does not use the message sent from \( j \), to avoid double counting

Constructing a clique tree

- One way to find a clique tree is to choose a variable elimination ordering and "run" VE
- Another approach is to use a graph construction
  - If necessary, moralize to produce an undirected graph \( G \).
  - Triangulate \( G \) to produce a chordal graph \( H \). (Would like one with minimum clique size, but this is NP hard.)
  - Find maximal cliques in \( H \). (Not NP hard for chordal graphs.)
  - Construct graph with nodes corresponding to max cliques in \( H \), edges weighted according to degree of overlap. i.e. edge between \( C_1 \) and \( C_2 \) has weight \( |C_1 \cap C_2| \)
  - Find a max spanning tree on this graph to yield a clique tree.

Announcements

- A2 (still) due Thursday
### Examples

- Trees
- 2-trees
- Grids

### Tree width

- The *tree width* of a graph $G$ is equal to $m-1$, where $m$ is the size of the largest clique in the triangulated (chordal) version of $G$.

- The worst-case running time of exact inference on a Markov or Bayes network is exponential in the tree width of the graph.