(More) Clique trees

CS B553
Spring 2013

Announcements

- A2 (still) due Thursday

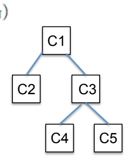
Inference through message passing

- Recall that Markov nets factor over cliques,

$$P(\mathbf{X}) = P(X_1, ..., X_N) = \frac{1}{Z}\phi_1(\mathbf{A_1}) \cdot \phi_2(\mathbf{A_2}) \cdot ... \cdot \phi_N(\mathbf{A_N})$$

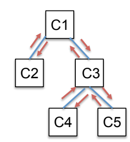
  - We can assign each of these factors to one of the nodes in the clique tree,

$$P(\mathbf{X}) = \frac{1}{Z}\prod_j \psi_j(\mathbf{C_j})$$



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,



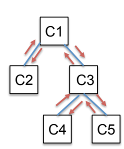
$$\delta_{i\to j}(\mathbf{S_{i,j}}) = \sum_{\mathbf{C_i}-\mathbf{C_j}} \psi_i(\mathbf{C_i}) \prod_{k\in\mathcal{N}(i)-\{j\}} \delta_{k\to i}(\mathbf{S_{k,i}})$$

  - 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

Sum-product belief propagation

- Once all messages have been exchanged, each node can compute its marginal,

$$P(\mathbf{C_i}) = \frac{1}{Z}\psi_i(\mathbf{C_i}) \prod_{k\in\mathcal{N}(i)} \delta_{k\to i}(\mathbf{S_{k,i}})$$



- Running time of sum-product BP?

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 C1 and C2 has weight $|C1 \cap C2|$
  - **Find a max spanning tree** on this graph to yield a clique tree.

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.**