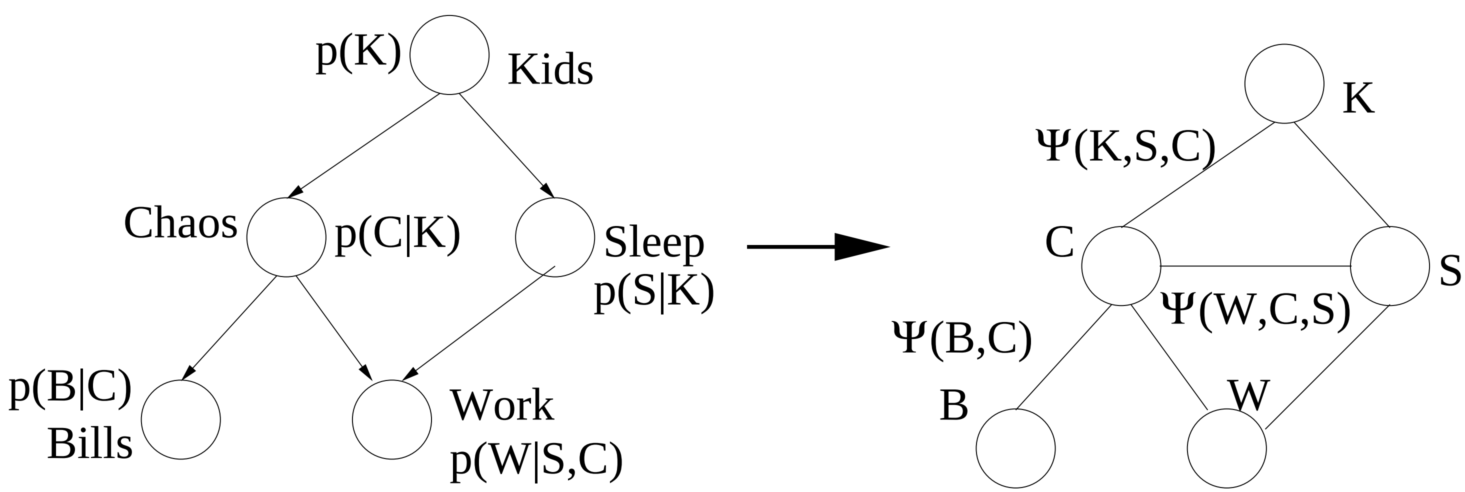
Lecture 9, 10, 11: Junction tree algorithm

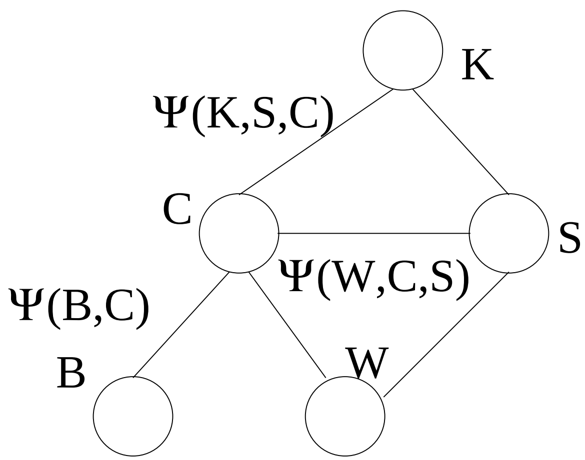
- Variable elimination revisited
- Clique trees and junction trees
- Constructing junction trees
- Parametrization
- Junction tree algorithm

Example



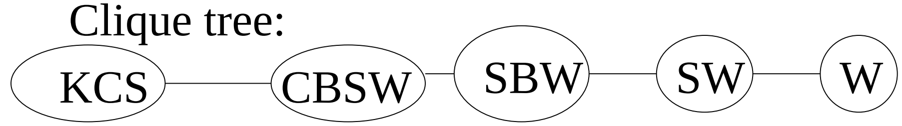
p(K)
Kids
Chaos
p(C|K)
Sleep
p(S|K)
p(B|C)
Bills
Work
p(W|S,C)
K
Ψ(K,S,C)
C
S
Ψ(W,C,S)
Ψ(B,C)
B
W

Example: Variable elimination



Var. elim. order:
K,C,B,S,W

Clique tree:



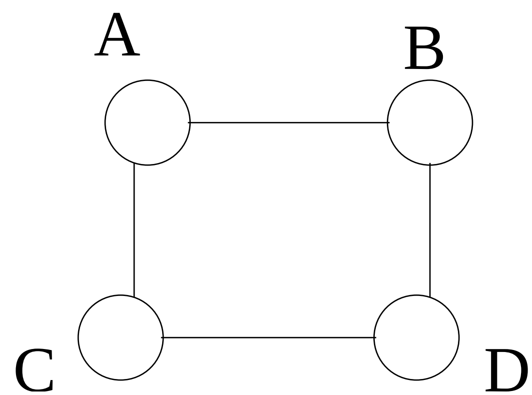
- We create a clique by connecting all the nodes that are involved in creating a factor (they would form a clique after elimination)
- The resulting structure is called a *clique tree*
- In general, a clique tree is a singly connected graph in which nodes are cliques of an underlying graph

Separator sets

- A *separator set* is the intersection of two corresponding cliques

Junction tree property

The cliques containing a particular node form a *connected subtree*



Constructing a junction tree

- Moralize the graph (if directed)
- Choose a node ordering and find the cliques generated by variable elimination. This gives a *triangulation* of the graph
- Construct a minimum spanning tree: start with no edges, add the edges that give maximum cardinality of the separator set (making sure no cycles are created)

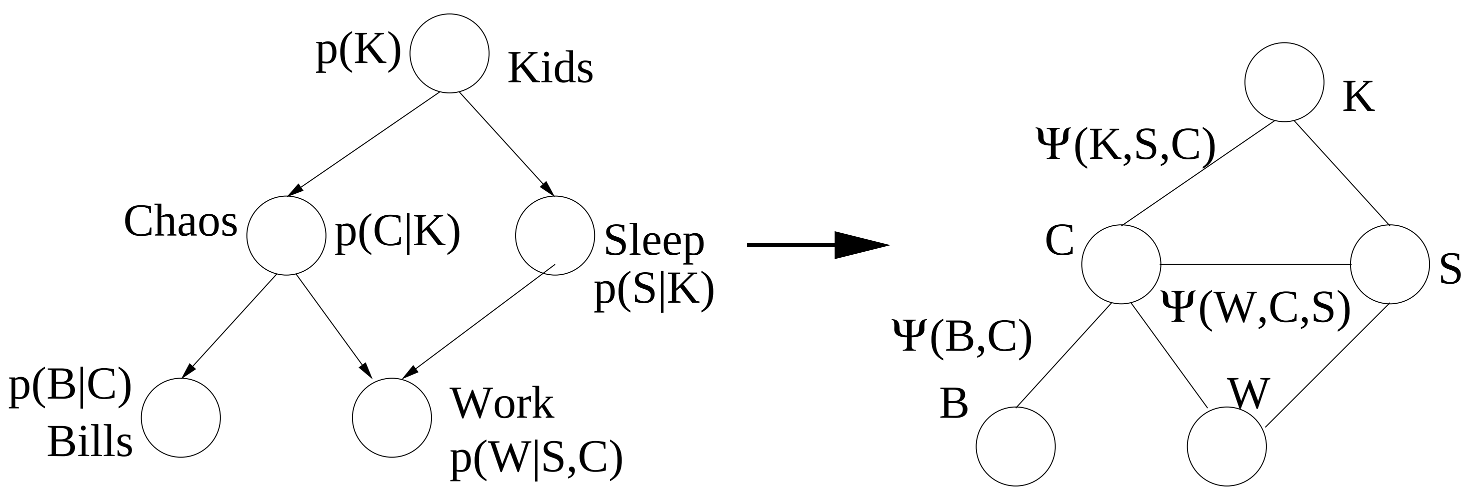
Heuristics for node ordering

- *Maximum cardinality:* Number the nodes from 1 to n, always assigning the next number to the vertex having the largest set of previously numbered neighbors. Then eliminate nodes from n to 1.
- *Minimum discrepancy:* Always eliminate the node that causes the fewest edges to be added to the induced graph
- *Minimum size/weight:* Eliminate the node that causes the smallest clique to be created (either in terms of number of nodes, or in terms of number of entries).

Junction tree algorithm

1. If the model is directed, moralize it
2. Triangulate the undirected graph, using your favorite method
3. Parameterize the undirected graph
4. Construct the junction tree, using your favorite maximum spanning tree algorithm
5. *Message passing between the nodes!*

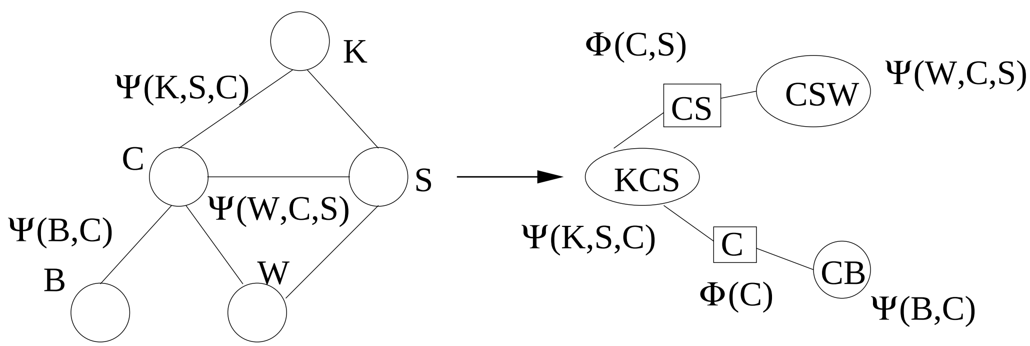
From the directed to the undirected model



$$
\begin{aligned}
\psi(K,C,S) &= p(K)p(C|K)p(S|K) \\
\psi(C,B) &= p(B|C) \\
\psi(W,C,S) &= p(W|S,C)
\end{aligned}
$$

Note that the undirected model is already triangulated, so there is no need to do anything more at this point

From the undirected model to the junction tree



Initially:

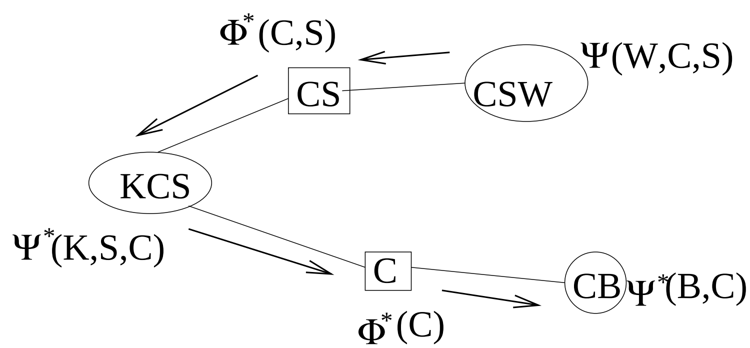
$$
\begin{aligned}
\psi(K,C,S) &= p(K)p(C|K)p(S|K) \\
\psi(C,B) &= p(B|C) \\
\psi(W,C,S) &= p(W|S,C) \\
\phi(C) &= 1 \\
\phi(C,S) &= 1
\end{aligned}
$$

This gives us correct probabilities, but we need to do a global computation!

Message passing in the junction tree

- We want to have *marginal probabilities* in all nodes and separator sets (i.e., we want all potentials to be proportional to marginals over the corresponding variables.
- Let (B, C) be the designated root (we can choose any node)

Message passing (1)



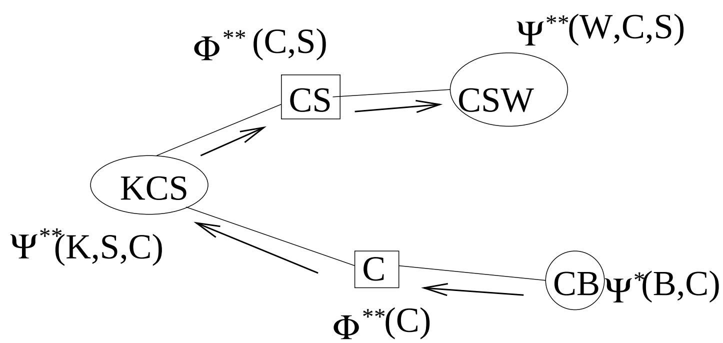
$$\phi^*(C,S) = \sum_W \psi(W,C,S) (= \sum_W p(W|C,S) = 1)$$

$$\psi^*(K,C,S) = \frac{\phi^*(C,S)}{\phi(C,S)} \psi(K,C,S) \text{(nothing for now)}$$

$$\phi^*(C) = \sum_{K,S} \psi^*(K,C,S) (= \sum_{K,S} p(K)p(C|K)p(S|K) = p(C))$$

$$\psi^*(B,C) = \frac{\phi^*(C)}{\phi(C)} \psi^*(B,C) (= p(C)p(B|C) = p(B,C))$$

Message passing (2)



$$
\begin{aligned}
\phi^{**}(C) &= \sum_B \psi(B,C)(= \sum_B p(B,C) = p(C) \\
\psi^{**}(K,C,S) &= \frac{\phi^{**}(C)}{\phi^{*}(C)} \psi^{*}(K,C,S) \text{(still nothing)} \\
\phi^{**}(C,S) &= \sum_K \psi^{**}(K)(= \sum_K p(K)p(C|K)p(S|K) = p(C,S)) \\
\psi^{*}(W,C,S) &= \frac{\phi^{**}(C,S)}{\phi^{*}(C,S)} \psi^{*}(W,C,S)(= p(C,S)p(W|C,S) = p(W,C,S))
\end{aligned}
$$

Probability propagation

- We introduce the evidence, if any
- Probabilities get propagated using the following equations (between cliques V and W with separator S):

$$\begin{aligned} \phi^*(S) &= \sum_{V-S} \psi(V) \\ \psi^*(W) &= \frac{\phi^*(S)}{\phi(S)} \psi(W) \end{aligned}$$

  These computations do not alter the joint probability distribution
- A clique node can send a message to a neighbor *after it has received messages from all other neighbors*

  This is the same protocol as the sum-product algorithm

Computational complexity

- Constructing the junction tree is done off-line, and is cheap if we are not looking for “optimal” cliques
- On-line, messages are passed on each arc exactly twice
- *But the computation here might be expensive!*