ECE521 Lecture 23
6 April 2017

Belief propagation in cyclic graphs

Mark Ebden

(with thanks to Andrew Rosenberg, Stephen Roberts and others)

Outline

- Lecture 22 continued:
  - Max-sum algorithm
  - The parallel protocol for belief propagation
  - Course evaluations
- Belief propagation in cyclic graphs:
  - Junction-tree algorithm
  - Loopy belief propagation

Belief propagation in cyclic graphs

- The message-passing we have seen so far is limited to trees (maximum one path between any two nodes)
- We can work on non-trees as well, with some care
- As before, the key is to form a graph which has the same global properties as the original problem, while allowing local representation to avoid brute-force inference
- We look first at the *junction-tree algorithm*

Belief propagation in cyclic graphs

Examples of optional readings:

- MacKay 26.4
- Bishop 8.4.7
- Murphy 20.4

The junction-tree algorithm

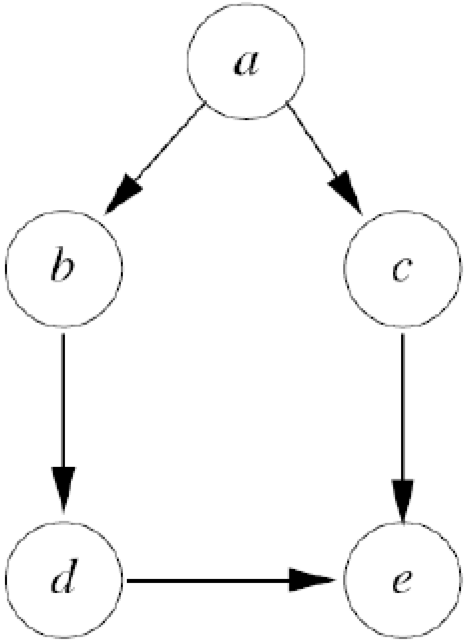
1. Construct the CPTs (conditional probability tables)
2. Convert the Bayesian network to a moralized MRF
3. Triangulate the MRF
4. Identify and link up cliques, to construct the junction tree
5. Propagate probabilities

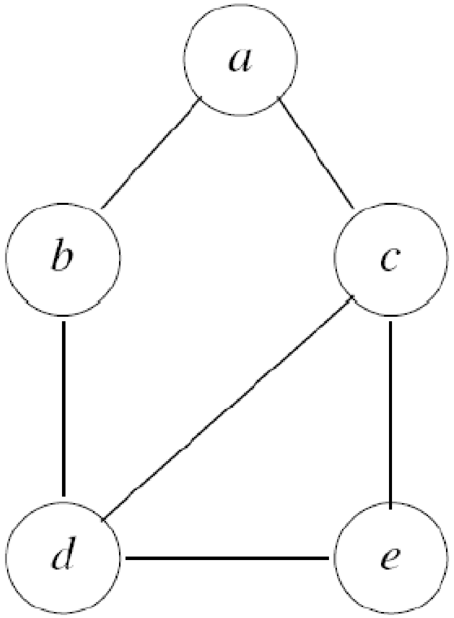
Each step can be done in polynomial time
(even #3, although finding its optimal solution is NP hard)

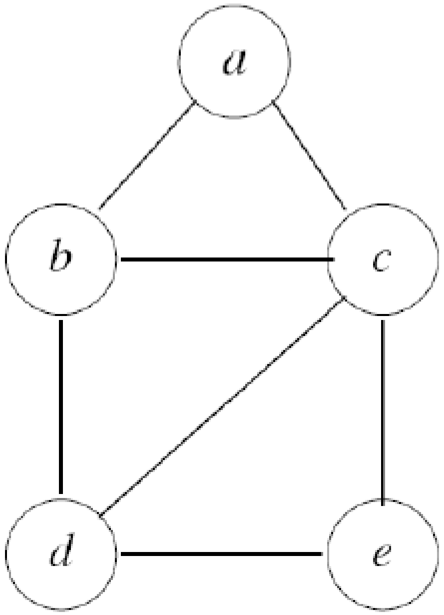
Junction-tree algorithm

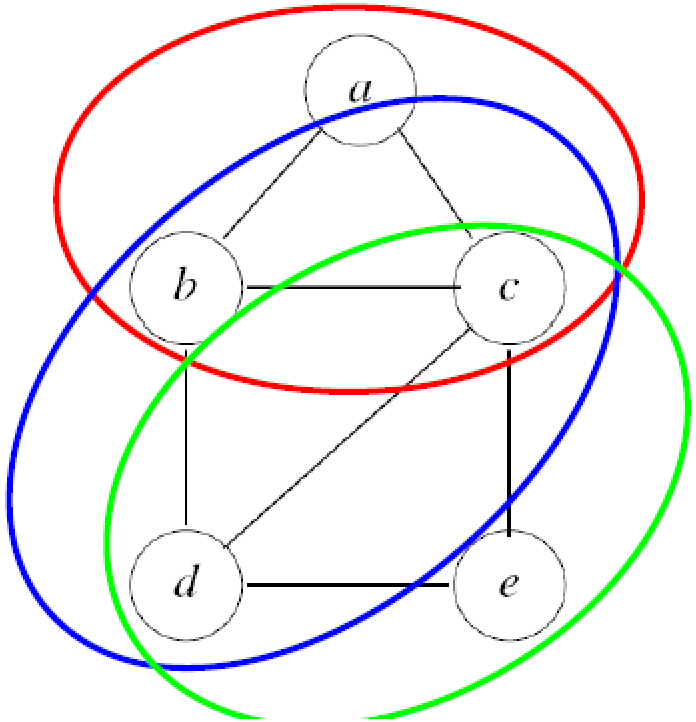
Steps 2 to 4 are known as compiling the graph:

Moralization | Triangulation | Identifying/joining cliques

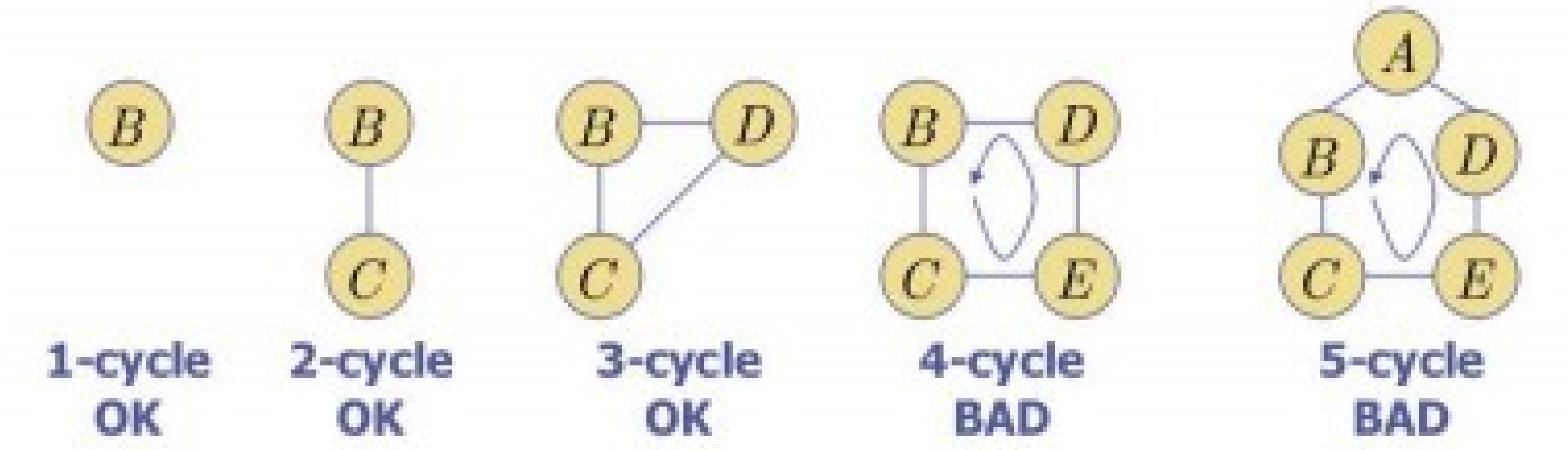






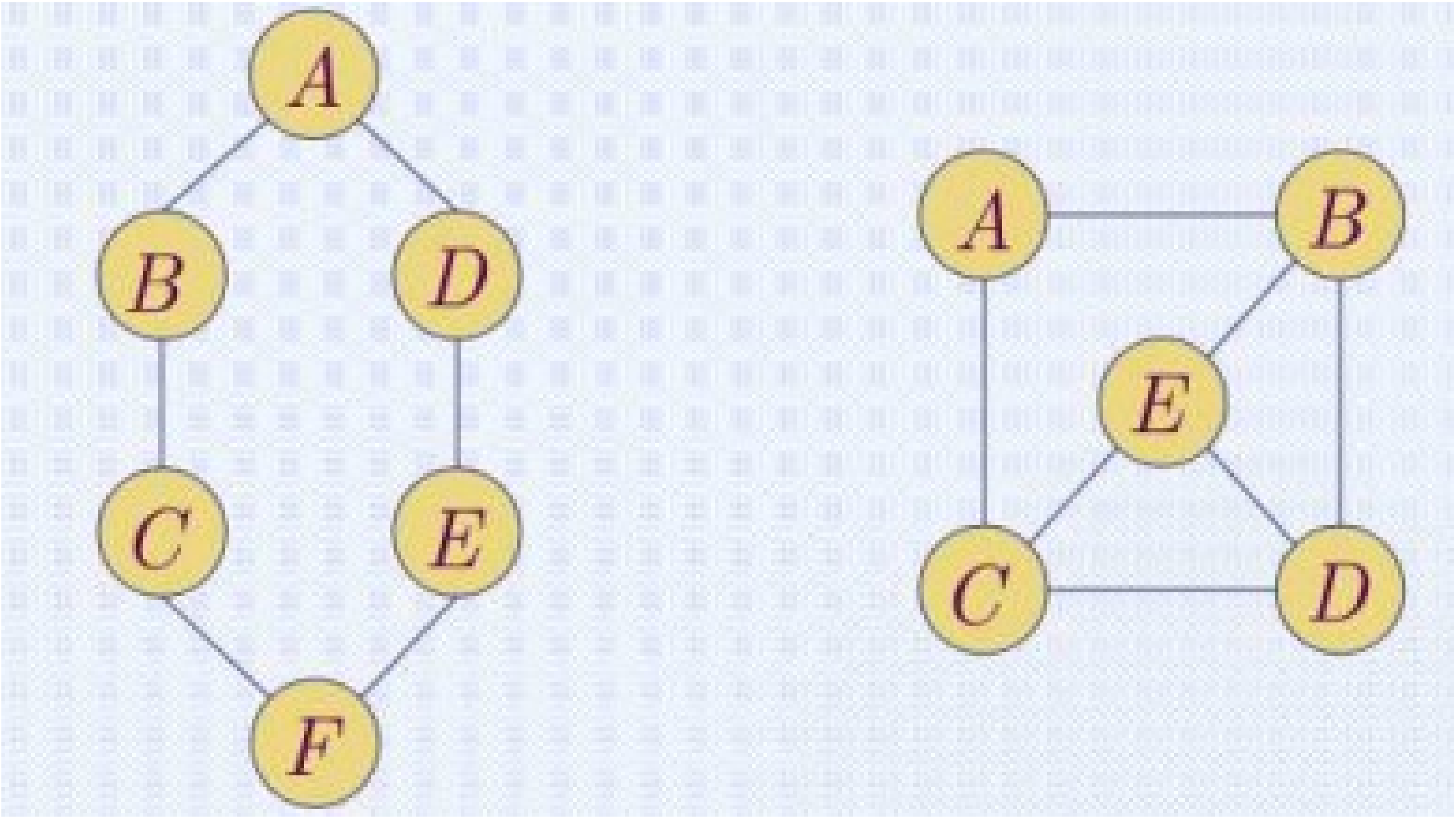


Triangulation

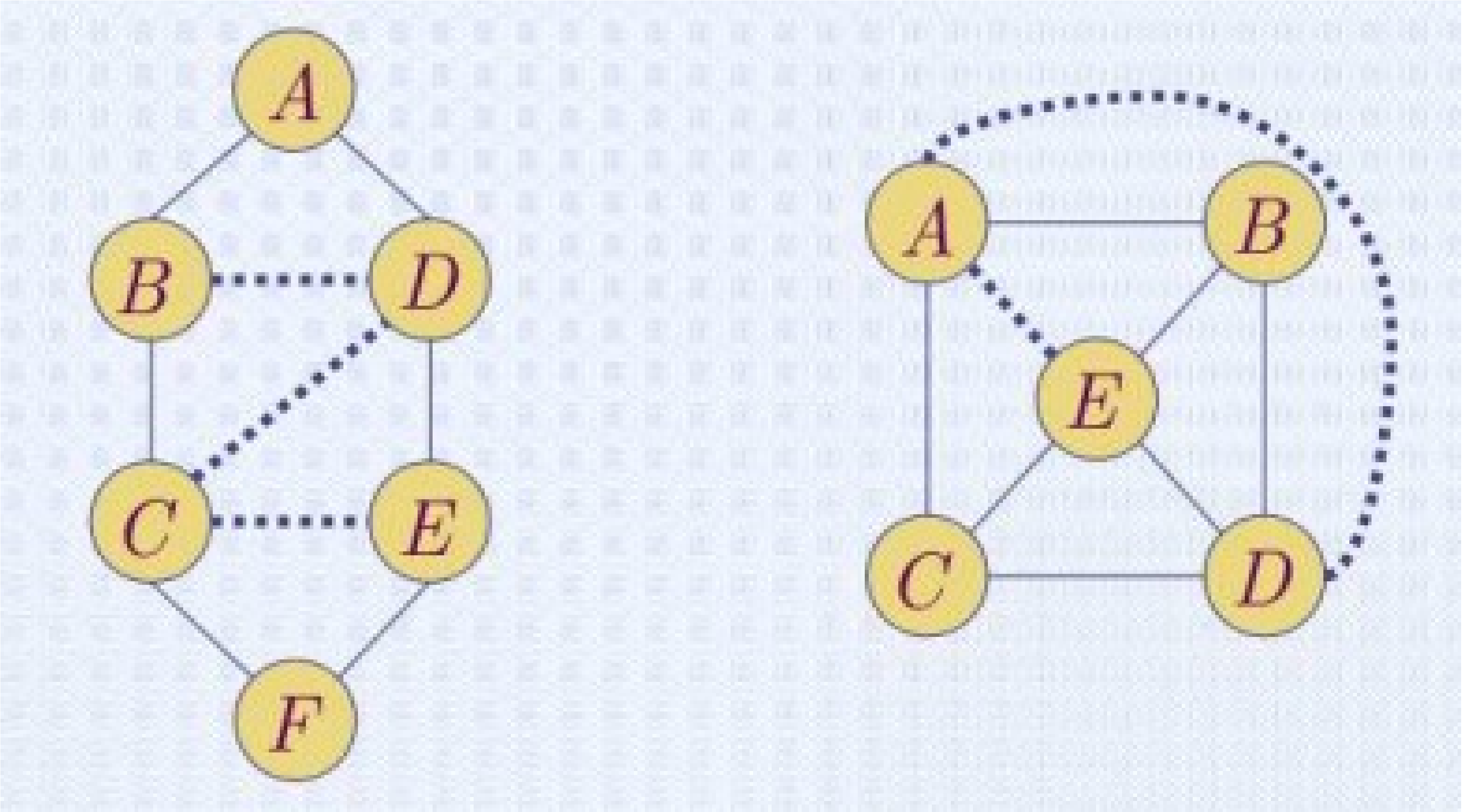


- There can be many choices for which edges to introduce
- Any triangulation is acceptable in this course

Triangulation = adding links to break up cycles of 4 or more nodes

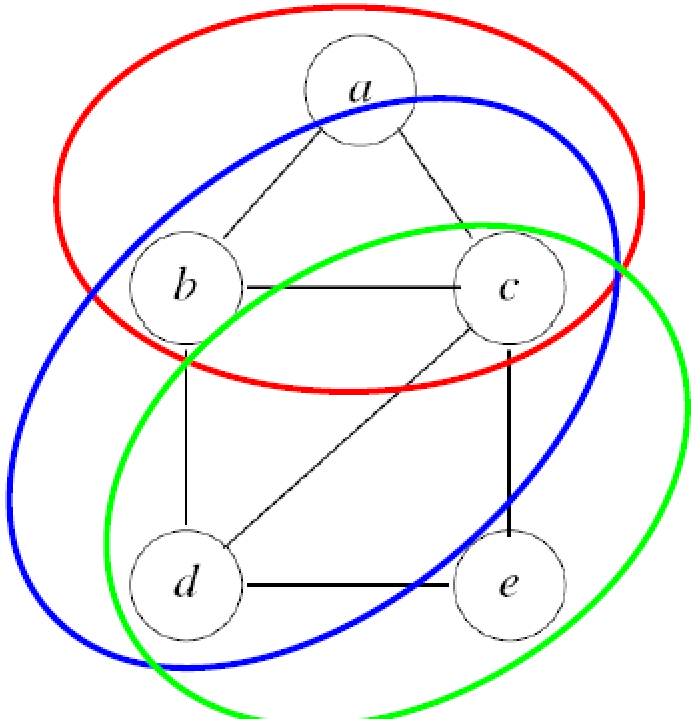


Triangulation examples



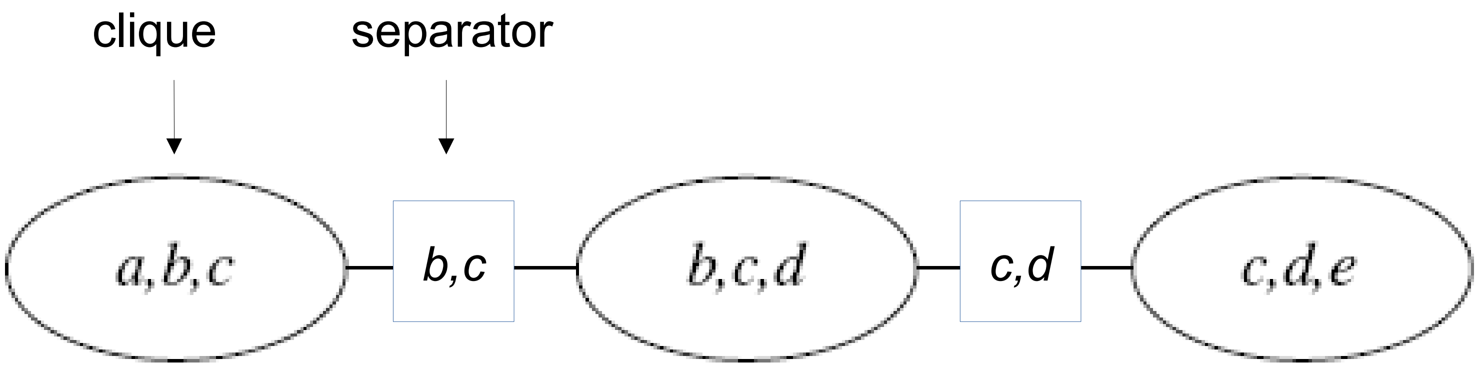
Cliques

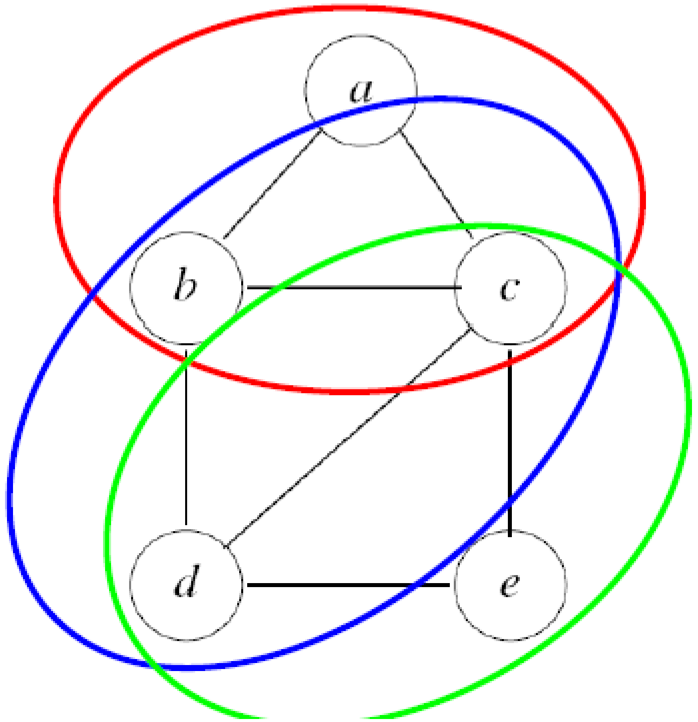
- Order the cliques by their highest vertices in the original DAG
- No node can be a descendant of a node in a lower clique
- A junction tree starts with the lowest clique, and you add progressively the predecessor clique which shares the largest number of common nodes:



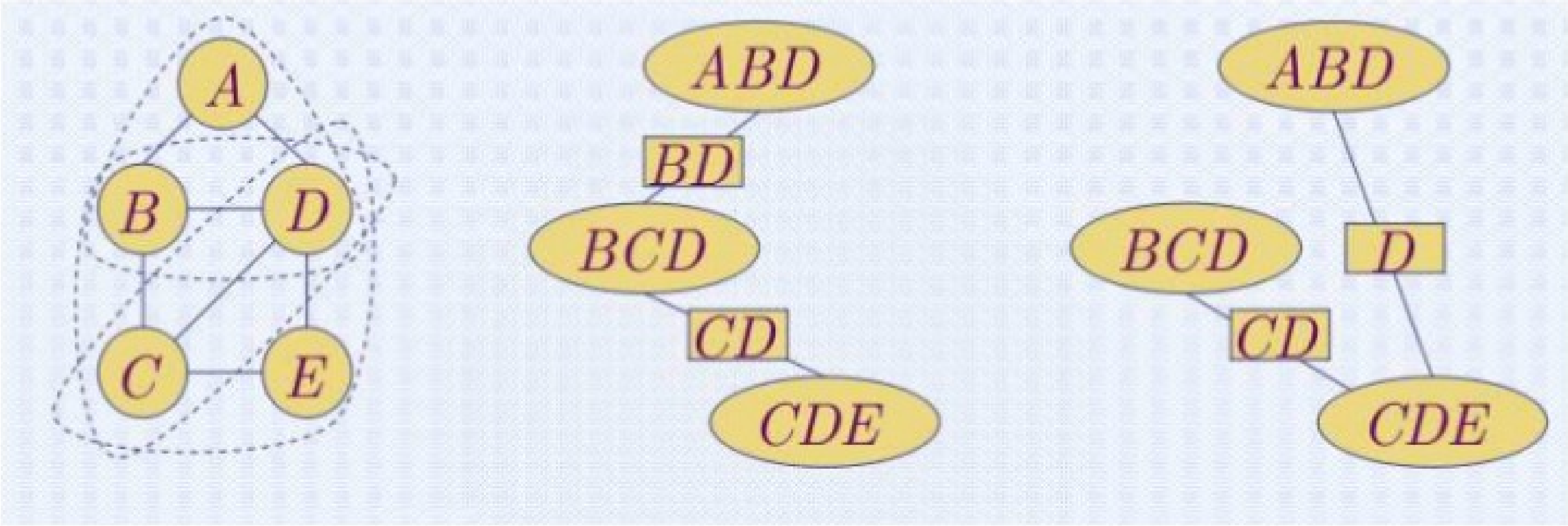
Adding separators

- Order the cliques by their highest vertices in the original DAG
- No node can be a descendant of a node in a lower clique
- A junction tree starts with the lowest clique, and you add progressively the predecessor clique which shares the largest number of common nodes:





Multiple trees can be constructed from the same graph.



- Aim for a junction tree with the highest *separator cardinality* (4 > 3 in the above)

- Junction trees must satisfy the **Running Intersection Property**:
  On the path connecting any two cliques, each in-between clique must include those two cliques' shared nodes

Message passing in a Junction Tree

- Letting *C(T)* be the cliques on the junction tree, and *S(T)* be the separators, our model is:

$$p(\mathbf{x}) = \frac{\prod_{c \in \mathcal{C}(T)} \psi_c(\mathbf{x}_c)}{\prod_{s \in \mathcal{S}(T)} \psi_s(\mathbf{x}_s)}$$

  where $\psi_c(\mathbf{x}_c)$ and $\psi_s(\mathbf{x}_s)$ are the clique potentials and separator potentials, respectively

- For a separator between cliques *i* and *j*, we also use the notation $\psi_s(\mathbf{x}_s) = \psi_{ij}(S_{ij})$, between clique potentials $\psi_i(C_i)$ and $\psi_j(C_j)$

- We use asterisks to indicate when a potential has been updated, e.g. $\psi_{ij}^{*}(S_{ij})$, $\psi_{ij}^{**}(S_{ij})$

Message passing in a Junction Tree

1. Initialize all separators to 1, and all $\psi_c(\mathbf{x}_c)$ to functions as we did for factor graphs

2. Starting at the bottom-most separator in the junction tree, alternate between the following two steps, climbing to the root:

$$\psi^*_{ij}(S_{ij}) = \sum_{C_i \setminus S_{ij}} \psi_i(C_i) \qquad \psi^*_j(C_j) \propto \psi_j(C_j) \frac{\psi^*_{ij}(S_{ij})}{\psi_{ij}(S_{ij})}$$

3. On the way down, alternate between these two steps:

$$\psi^{**}_{ij}(S_{ij}) = \sum_{C_i \setminus S_{ij}} \psi^*_i(C_i) \qquad \psi^{**}_j(C_j) \propto \psi^*_j(C_j) \frac{\psi^{**}_{ij}(S_{ij})}{\psi^*_{ij}(S_{ij})}$$

Loopy belief propagation

- *Treewidth* is a measurement related to the size of the largest clique
- If the treewidth is high, the junction-tree algorithm will not be quick
- An alternative is LBP: return to the factor graph approach and try applying the parallel protocol (see Lecture 22), overlooking the fact that we don’t have a tree!
- LBP gives no guarantee of convergence, but for many graphs it does converge

Questions (The three red ones are to be discussed during the lecture.)

A company you outsource your coding to has two programmers: Joab, handling 80% of projects, and someone else. For each function defined in the resulting code, there is a 70% chance it was named descriptively if Joab was the author; otherwise the chance is 90%.

1. You commission some work and find that the first two functions, `GMMfit` and `Estep`, were named descriptively.

(a) Calculate the probability that Joab was the programmer.
(b) Draw a Bayesian network with probability tables for binary variables F(irst), S(econd), and J(oab).

2. You commission a second piece of work.

(a) Use the max-sum algorithm to guess simply (yes or no) whether the first function will be named descriptively.
(b) Use the sum-product algorithm to compute the probability that the first two functions will be named descriptively.

3. Suppose now that if the nth function isn't named descriptively, the chance that the $(n+1)$st function will be named descriptively is halved. You commission a third piece of work and are interested in the probability $P(T)$ that the *third function will be named descriptively*.

(a) Draw the new Bayesian network with four binary variables and their probability tables. Don't calculate $P(T)$.
(b) Convert this Bayesian network to a junction tree. Use it to calculate $P(T)$.
(c) Draw a factor graph which, using loopy belief propagation, *might be able to* calculate $P(T)$. Don't calculate it.

Outline

- Lecture 22 continued:
  - Max-sum algorithm
  - The parallel protocol for belief propagation
  - Course evaluations
- Belief propagation in cyclic graphs:
  - Junction-tree algorithm
  - Loopy belief propagation