BBN inference using junction trees

Mihai Rotaru
CS 3750

Problem

- Given a BBN over U (set of variables)
  - Want to compute P(V)
  - Want to compute P(V|E)
- Computing P(V) using the joint distribution P(U) is exponential
  - Requires 2^{n-1} sums and $n*2^{n-1}$ products if |U|=n and every variable has 2 values

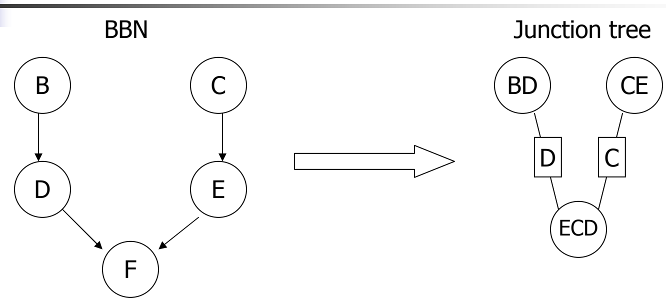
$$P(V) = \sum_{U \setminus V} P(U) = \sum_{U \setminus V} \prod_i P(X_i | Pa(X_i))$$

Solution

- Compute joint over partitions of U
  - $U = \bigcup_i C_i$
  - C_i small subset of U (typically made of a variable and its parents) - clusters
  - C_i not necessary disjoint
  - Know $P(C_i)$
- To compute P(X)
  - Need far less operations

$$P(X) = \sum_{C_i \setminus X} P(C_i)$$

Example



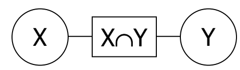
$$P(U) = \sum P(F|D,E)P(D|B)P(E|C)P(B)P(C)$$

$$P(D,E,F) = P(F|D,E)\sum_{B,C} P(D|B)P(E|C)P(B)P(C) =$$

$$= P(F|D,E)\underbrace{\sum_{B} P(D|B)P(B)}_{\text{Groups D and B}}\underbrace{\sum_{C} P(E|C)P(C)}_{\text{Groups E and C}} = P(F|D,E)\sum_{B} P(B,D)\sum_{C} P(C,E)$$

Junction tree

- Undirected tree; each node is a cluster of variables C_i; each edge XY labeled with X∩Y (separation set – sepset)



- Graphical properties
  - All clusters on path between X and Y must contain X∩Y
  - For each variable V, there is a cluster that contains both V and Pa(V)
- Numerical component
  - Each cluster and sepset has a belief potential t.

Junction tree - continued

- Numerical properties
  - Local consistency – C cluster, S neighboring sepset

$$\sum_{C \setminus S} t_C = t_S$$

  - Encodes the original joint distribution

$$P(U) = \frac{\prod_i t_{C_i}}{\prod_j t_{S_j}}$$

- Can be shown that for such a structure

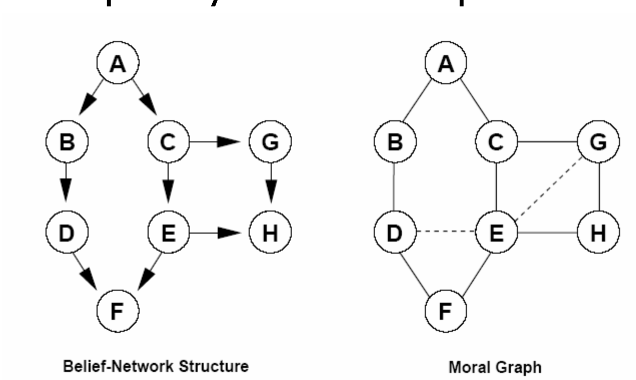
$$P(C) = t_C$$
$$\Downarrow$$
$$P(V) = \sum_{C \setminus V} t_C$$

Building the junction tree

- Has 4 steps
  - Build moral graph out of BBN graph
  - Triangulate the moral graph
  - Identify cliques (clusters)
  - Build the joint tree by connecting the clusters
- 2nd and 4th step are nondeterministic

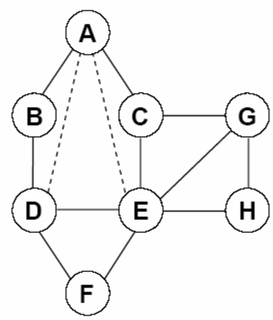
Step 1 – Moral graph

- Drop the directions
- Completely connect all parents of every



Step 2 – Triangulate

- Add edges until the graph is triangulated (i.e. nodes from any cycle are completely connected)



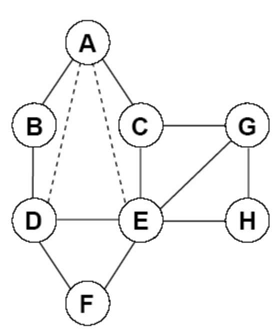
Triangulated Graph

Eliminated Vertex	Induced Cluster	Edges Added
H	EGH	none
G	CEG	none
F	DEF	none
C	ACE	(A,E)
B	ABD	(A,D)
D	ADE	none
E	AE	none
A	A	none

Elimination Ordering

Step 3 – Identify cliques

- Use the step 2 to get all the cliques



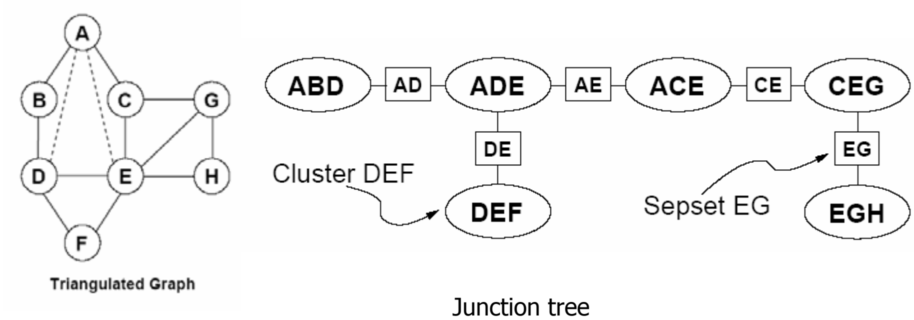
Triangulated Graph

Eliminated Vertex	Induced Cluster	Edges Added
H	EGH	none
G	CEG	none
F	DEF	none
C	ACE	(A,E)
B	ABD	(A,D)
D	ADE	none
E	AE	none
A	A	none

Elimination Ordering

Step 4 – build joint tree

- See algorithm in the paper



Building the numeric component

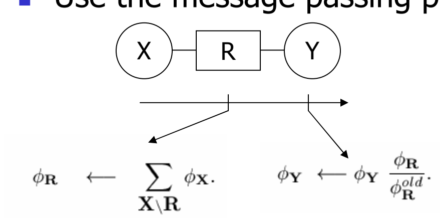
- 2 steps
  - Initialization
  - Global propagation

Step 1 - Initialization

- Set t_C to 1 for every cluster and sepset
- For every variable V choose a cluster C that contains V and Pa(V) and do
  - $t_C \leftarrow t_C * P(V|Pa(V))$
- Remark: the second numeric property holds now

Step 2 – Global propagation

- Use the message passing procedure



- Second property still holds and X is consistent with S
- Moreover, if Y was consistent with S before the message it will still be consistent after the message

Step 2 – Global propagation (cont)

- Choose a cluster X
- Do message passing towards X (Collect evidence)
- Do message passing away from X (Distribute evidence)

- After two passes the structure satisfies both numeric properties
- We need a tree so that propagation can finish

Handling evidence

- What about P(V|E)
- Cut part of the belief potential by setting to 0 where the value differs from the evidence value
- Apply same algorithm