10701 Recitation

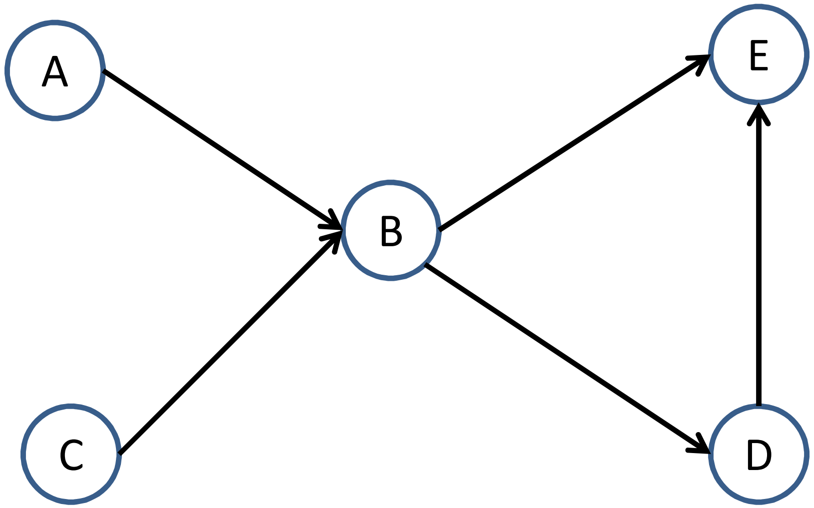
Suyash Shringarpure

11/8/2011

Plan

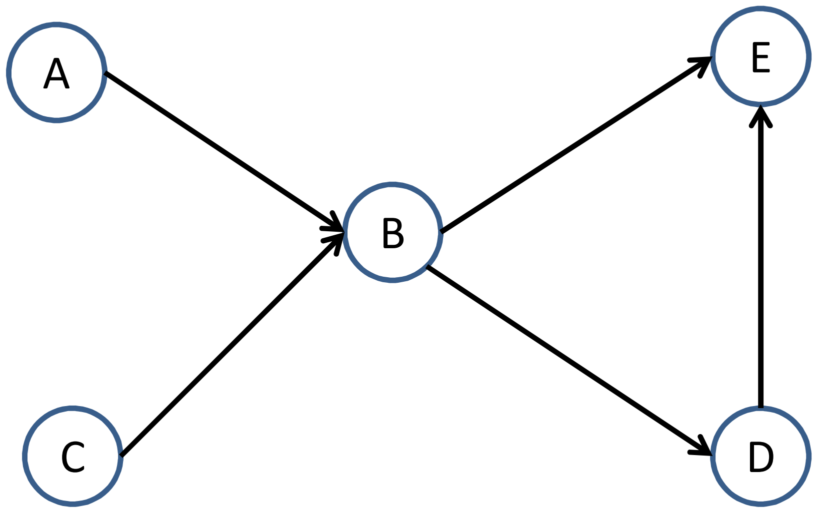
- Homework questions?
- Lecture questions?
- Bayesian networks
- D-separation
- Variable elimination
- Message passing & Junction trees
- Undirected graphs

Bayesian networks



D-separation

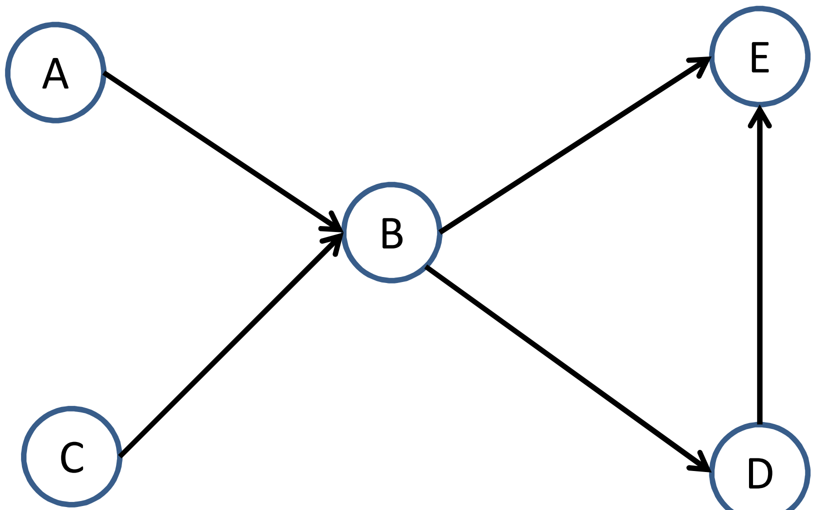
- $A \perp C$ = T/F
- $A \perp C \mid B$ = T/F
- $A \perp D \mid B$ = T/F
- $B \perp$ What | What else?



Factorized probabilty

- $P(A,B,C,D,E) = ?$

$= P(A).P(C).P(B|A,C)$
$P(D|B).P(E|D,B)$



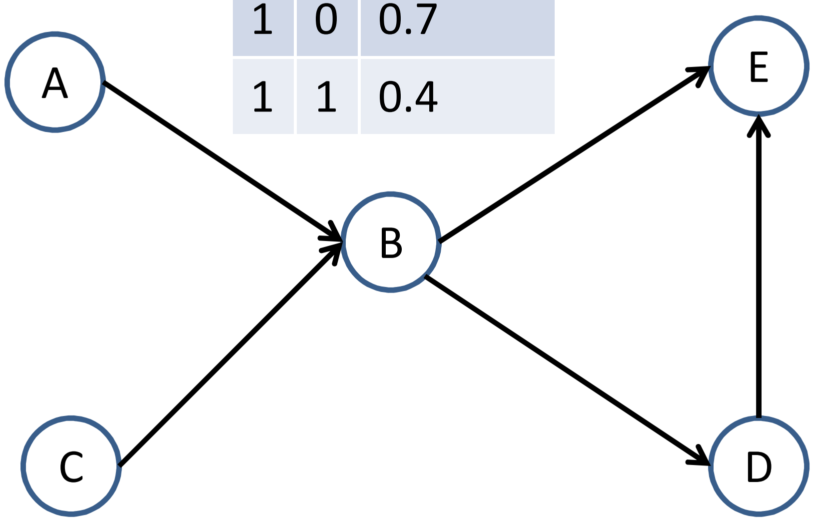
- Parameters
  - Naïve = ? $2^5 - 1 = 31$
  - Factorized = ? 12

Conditional probabilty tables

A	P(A)
0	0.2
1	0.8

A	C	P(B=1)
0	0	0.1
0	1	0.8
1	0	0.7
1	1	0.4

B	D	P(E=1)
0	0	0.7
0	1	0.5
1	0	0.1
1	1	0.4



C	P(C)
0	0.9
1	0.1

B	P(D=1)
0	0.8
1	0.1

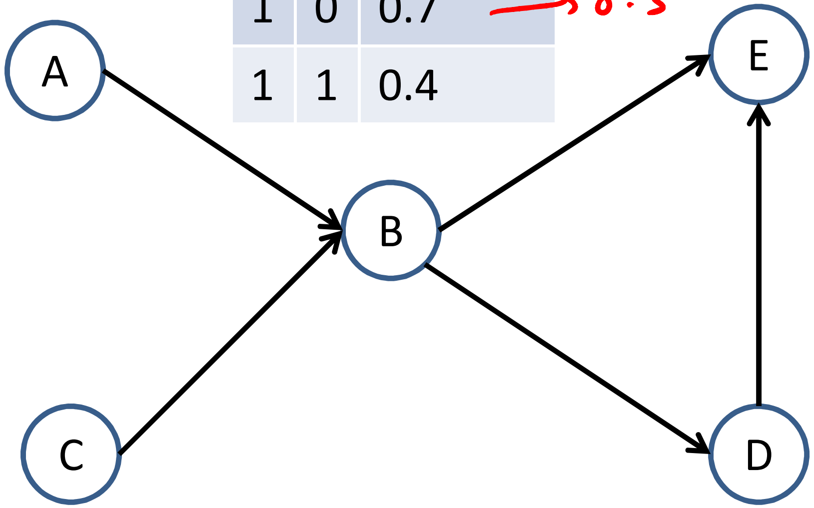
Inference problem

- What is P(E=1) ?

A	P(A)
0	0.2
1	0.8

A	C	P(B=1)	
0	0	0.1	→ 0.9
0	1	0.8	
1	0	0.7	→ 0.3
1	1	0.4	

B	D	P(E=1)
0	0	0.7
0	1	0.5
1	0	0.1
1	1	0.4



C	P(C)
0	0.9
1	0.1

B	P(D=1)
0	0.8
1	0.1

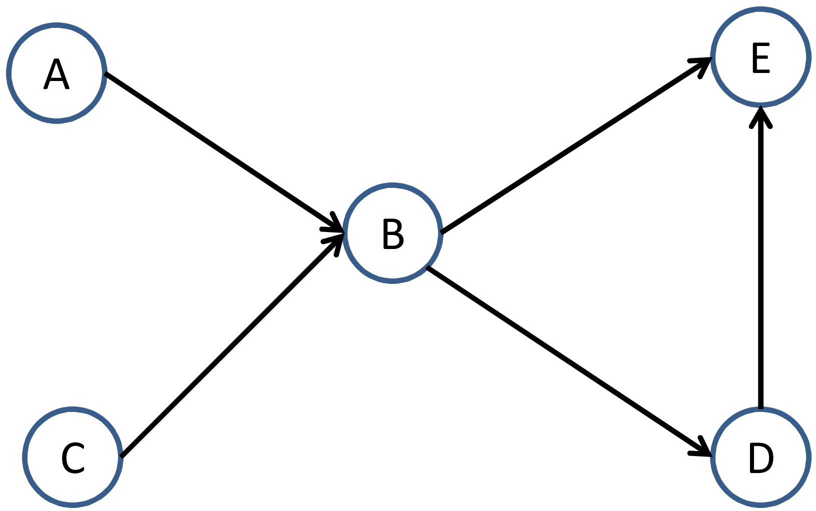
Variable elimination

- Elimination of a variable = Marginalizing out that variable from all factors that include that variable.
- For example, factors for B are?

$P(B|A,C)$

$P(E|B,D)$

$P(D|B)$



Variable elimination

- Initial factors $\{f_1..,f_n\}$ -CPT (P(Node| Parents))
- Choose an elimination order (IMPORTANT)
- To eliminate X:
  - Collect all factors $f_1..,f_k$ that contain X
  - Generate a new factor by marginalizing X
    - $g = \sum_X \prod_{j=1:k} f_j$
- Add g to the factor set and repeat

Example

- Order 1 = A,C,D,B
- $P(A,B,C,D,E)=P(A)P(C)P(B|C,A)P(D|B)P(E|D,B)$
- $P(E=1) = \sum_{A,B,C,D} P(A,B,C,D,E=1)$
  - $= \sum_{A,B,C,D} P(A)P(C)P(B|C,A)P(D|B)P(E=1|D,B)$

- According to the order, eliminate A first

Elimination steps

- $\sum_{A,B,C,D} P(A)P(C)P(B|C,A)P(D|B)P(E=1|D,B)$
  - $= \sum_{B,C,D} \underline{P(D|B)P(E=1|D,B)\ P(C)}\underline{\sum_{A}} P(A)\ P(B|C,A)$
- $\underline{m_A(B,C)} = ?$

$\sum_{A} P(A).\ P(B|C,A)$

$m_A(0,0) = P(A=0).\ P(B=0|C=0, A=0)$
$+ P(A=1)\ P(B=0|C=0, A=1)$
$\cancel{+ P(A=2)\ P(B=0|C=0, A=2)}$
$= 0.2 \times 0.9$
$+ 0.8 \times 0.7$
$= 0.18 + 0.56$
$= 0.74$

B	C	m
0	0	0.74
0	1	
1	0	
1	1	

$m_A(B,C) \Rightarrow 4$ additions

Contd.

- $\sum_{B,C,D} P(D|B)P(E=1|D,B)\ P(C)\sum_{A} P(A)\ P(B|C,A)$
  - $= \sum_{B,C,D} P(D|B)P(E=1|D,B)\ P(C)\ m_A(B,C)$
  - $= \sum_{B,D} P(D|B)P(E=1|D,B)\ \sum_{C} P(C)\ m_A(B,C)$
- $m_C(B) = ?$ $\quad \sum_{C} P(C)\, m_A(B,C)$

$m_C(0) = P(C=0) \cdot m_A(0,0) + P(C=1)\, m_A(0,1)$

B	$m_C(B)$
0	–
1	–

Contd.

- $\sum_{B,D} P(D|B)P(E=1|D,B) \sum_{C} P(C)\, m_A(B,C)$
  - $= \sum_{B,D} P(D|B)P(E=1|D,B)\, m_C(B)$
  - $= \sum_{B} m_C(B) \sum_{D} P(D|B)P(E=1|D,B)$
- $m_D(B) = ?$ $\quad \sum_{D} P(D|B) \cdot P(E=1|D,B)$

B	$m_D(B)$
0	–
1	–

Contd.

- $\sum_B m_C(B) \sum_D P(D|B)P(E=1|D,B)$
  - $=\sum_B m_C(B)\, m_D(B)$
  - = Answer

$m_c(0)\, m_D(0) + m_c(1)\, m_D(1)$

Complexity

- Computing $m_Y(X_1,\ldots,X_n)$ $= \sum_y m'(y, x_1 \ldots x_n)$
  - Additions? $y \rightarrow |y|$, $x_i \rightarrow |x_i|$

    $(|y|-1) \prod_i |x_i|$
  - Multiplications?

    $|y| \prod_i |x_i|$ (Extra term)

- Elimination complexity depends on
  - ? Size of factor

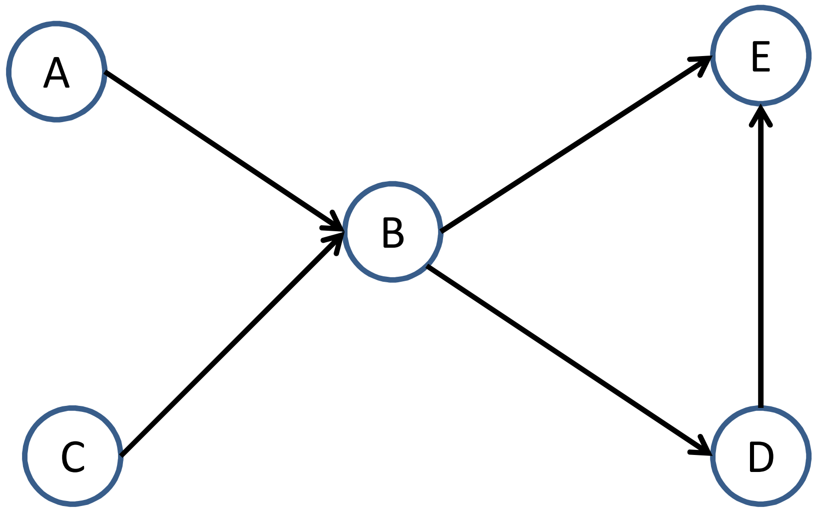
Order effects

- Worst order for the given graph?

$$= \sum_{A,C,D} P(A)\, P(C) \sum_{B} P(B \mid A, C) \times P(D \mid B) \times P(E{=}1 \mid D, B)$$

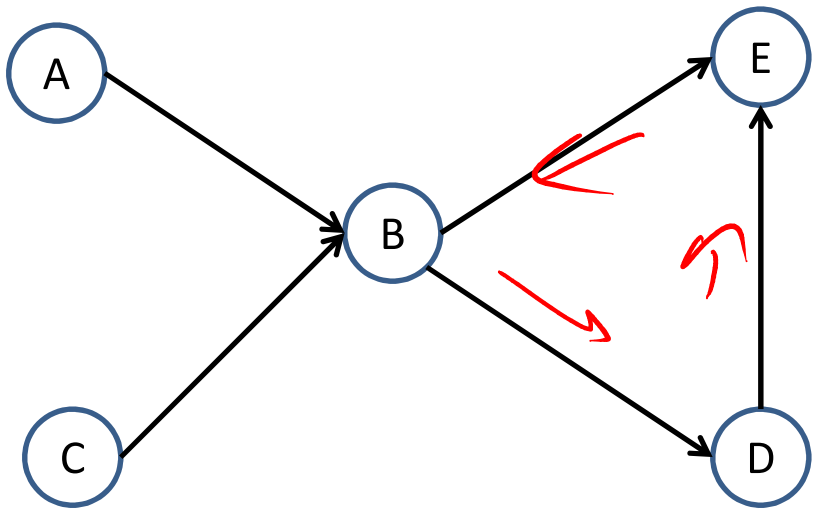
$$= m_B(A, C, D, E{=}1)$$

$$m_E(D, B) \leftarrow \sum_{E} P(E \mid D, B) \rightarrow 1$$



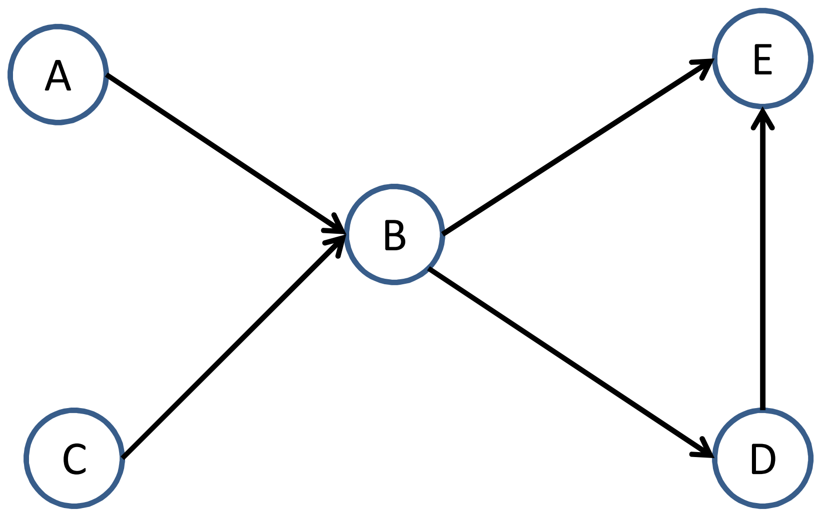
Message-passing

- A two-pass algorithm over the BN that answers multiple inference queries.
- On trees, elimination = message passing
  – Guaranteed converge and accuracy
- How about this graph?



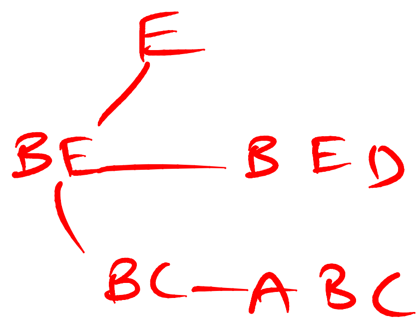
Message passing

- Loops are bad
  - Messages travel forever.
- In general,
  - Elimination = Message passing on clique trees.
- Example graph



Clique tree

- $P(A)P(C)P(B|C,A)P(D|B)P(E=1|D,B)$
  - $P(D|B)P(E=1|D,B)P(C)\ P(A)\ P(B|C,A)$
  - $P(D|B)P(E=1|D,B)P(C)\ m_A(B,C)$ - Eliminate A
  - $P(D|B)P(E=1|D,B)m_C(B)$ - Eliminate C
  - $m_C(B)\ m_D(B,E=1)$ - Eliminate D
  - $m_B(E=1)$ - Eliminate B
  - $P(E=1)$
- Cliques - E , BE , BED , BC , ABC
- Tree?



Clique tree

- Cliques - E , BE , BED , BC , ABC
- Tree?

B	E	D	f
0	0	0	
⋮	⋮	⋮	
1	1	1	

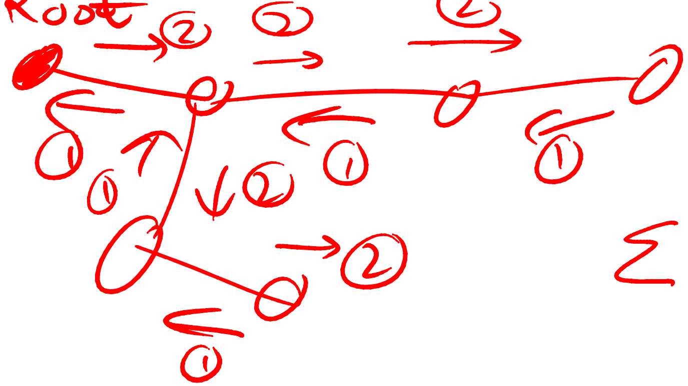
BED $\quad f(B,E,D) \rightarrow \sum_{E,D} f(B,E,D)$

$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad ||$

B

ABC $\quad g(A,B,C) \rightarrow \sum_{A,C} g(A,B,C)$

Root



$\sum \psi(\text{node})$

$\prod m(\cdot \to \text{node})$

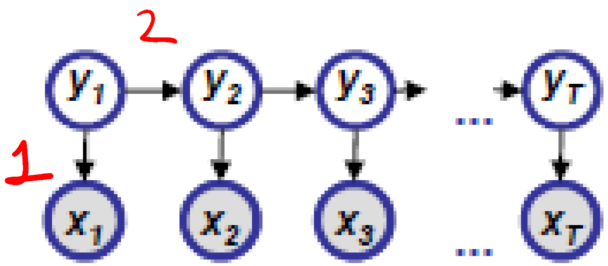
Junction trees

$\boxed{BED}$ – Clique

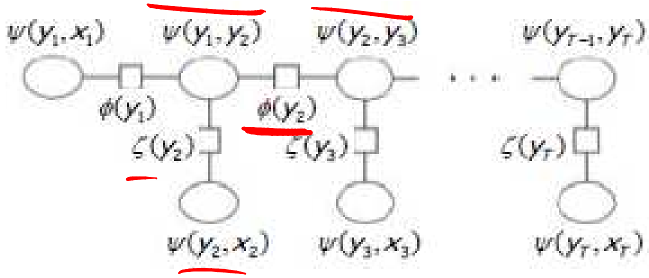
BE – Clique

- Maximal clique trees
- Edge between two cliques if they share variables
- Messages ensure consistency of marginals of common variables
- Complexity depends on clique size.
- Result: Marginals of each clique.

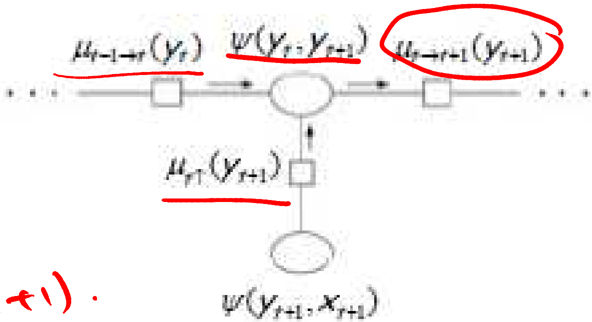
Junction tree example- HMM



$\rightarrow$



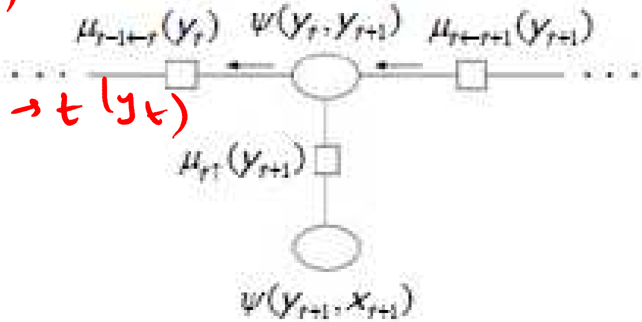
- Rightward message



$$\mu_{t\to t+1}(y_{t+1}) = \sum_{y_t} \mu_{t-1\to t}(y_t) \cdot \psi(y_t, y_{t+1}) \cdot \mu_{t\uparrow}(y_{t+1})$$

$$= b(x_{t+1}|y_{t+1}) \cdot \sum_{y_t} a(y_t, y_{t+1}) \cdot \mu_{t-1\to t}(y_t)$$

- Leftward message



= Backward algo.

Undirected graphs

- Junction tree algorithms work for undirected graphs as well
- To avoid loops, we triangularize cycles.
- If not, run loopy belief propagation
  - No guarantees
  - Works often in practice.

1 node ← 100 parents

Approx inference

- If P(x) is complex
  - Replace by simpler Q(x)
- Criterion for choosing form of Q(x)
  - Computational cost
- Criterion for choosing parameters of Q(x)
  - Minimize KL(Q||P)
- Simplest Q = ?

Q = Approximating distr.

P = ori

$$Q(x_1 \ldots x_n) = \prod_{i=1}^{n} Q_i(X_i)$$