CS3710 Advanced Topics in AI, Lecture 6

Variable Elimination and Conditioning: Complexity Forecasts.

Collin Lynch.
collinl@cs.pitt.edu

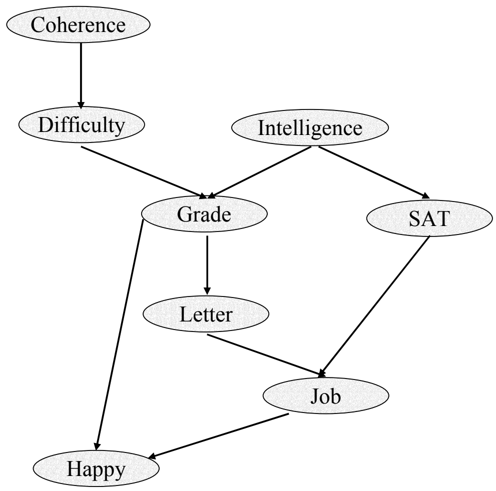
Milos Hauskrecht
milos@cs.pitt.edu

9/19/2005

Outline

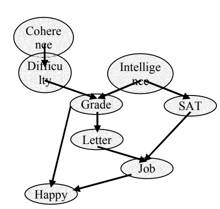
- Recap.
- Factor-Based Elimination.
- Moral Graphs and Triangulation.
- Variable Elimination.
- Induced Graph.
- Correspondence with Tree Decomposition.
- Initial Conclusions.
- Conditioning.
- Final Conclusions.

Recap (Problem).



Recap (Variable Elimination)

$$
\begin{aligned}
p(J) &= \sum_{L,S,G,H,I,D,C} \phi(c)\phi(i)\phi(d,c)\phi(g,i,d)\phi(s,i)\phi(l,g)\phi(j,l,s)\phi(h,g,j) \\
&= \sum_{L,S,G,H,I,D} \phi(i)\phi(g,i,d)\phi(s,i)\phi(l,g)\phi(j,l,s)\phi(h,g,j)\sum_{C}\phi(c)\phi(d,c) \\
&= \sum_{L,S,G,H,I,D} \phi(i)\phi(g,i,d)\phi(s,i)\phi(l,g)\phi(j,l,s)\phi(h,g,j)\tau(c) \\
&\ldots \\
&= \sum_{S,I}\phi(j,l,s)\sum_{G}\phi(l,g)\tau(s,g)\tau(g,j) \\
&= \sum_{S,I}\phi(j,l,s)\tau(l,s,j) \\
&= \sum_{I}\tau(l,j) \\
&= \tau(j)
\end{aligned}
$$



Basic messages

- Variable Elimination is not determininistic.
- The order of elimination governs the overall efficiency.
- Finding an optimal ordering is difficult.
- While different methodologies exist they are all functionally identical.
- The cost of any ordering is exponential in the number of variables that appear in the largest factor.

Factor-Based Elimination.

$$
\begin{aligned}
p(J) &= \sum_{L,S,G,H,I,D,C} \phi(c)\phi(i)\phi(d,c)\phi(g,i,d)\phi(s,i)\phi(l,g)\phi(j,l,s)\phi(h,g,j) \\
&= \sum_{L,S,G,H,I,D} \phi(i)\phi(g,i,d)\phi(s,i)\phi(l,g)\phi(j,l,s)\phi(h,g,j)\sum_{C}\phi(c)\phi(d,c) \\
&= \sum_{L,S,G,H,I,D} \phi(i)\phi(g,i,d)\phi(s,i)\phi(l,g)\phi(j,l,s)\phi(h,g,j)\tau(c) \\
&\dots \\
&= \sum_{S,I} \phi(j,l,s)\sum_{G}\phi(l,g)\tau(s,g)\tau(g,j) \\
&= \sum_{S,I} \phi(j,l,s)\tau(l,s,j) \\
&= \sum_{I} \tau(l,j) \\
&= \tau(j)
\end{aligned}
$$

FBE: Trace

Step	Var	Factors Used	New Factor
1	C	$\phi_c(C), \phi_D(D,C)$	$\tau_1(D)$
2	D	$\phi_G(G,I,D), \tau_1(D)$	$\tau_2(G,I)$
3	I	$\phi_I(I), \phi_S(S,I), \tau_2(G,I)$	$\tau_3(G,S)$
4	H	$\phi_H(H,G,J)$	$\tau_4(G,J)$
5	G	$\tau_4(G,J), \tau_3(G,S), \phi_L(L,G)$	$\tau_5(J,L,S)$
6	S	$\tau_5(J,L,S), \phi_J(J,L,S)$	$\tau_6(J,L)$
7	L	$\tau_6(J,L)$	$\tau_7(J)$

FBE: Trace

Step	**Var**	**Factors Used**	**New Factor**	**Complexity**
1	C	$\phi_C(C)\phi_D(D,C)$	$\tau_1(D)$	2
2	D	$\phi_G(G,I,D)\tau_1(D)$	$\tau_2(G,I)$	3
3	I	$\phi_I(I)\phi_S(S,I)\tau_2(G,I)$	$\tau_3(G,S)$	3
4	H	$\phi_H(H,G,J)$	$\tau_4(G,J)$	3
5	G	$\tau_4(G,J)\tau_3(G,S)\phi_L(L,G)$	$\tau_5(J,L,S)$	4
6	S	$\tau_5(J,L,S)\phi_J(J,L,S)$	$\tau_6(J,L)$	3
7	L	$\tau_6(J,L)$	$\tau_7(J)$	2

Factor-based elimination

- ***Ordering Ω:***
  - A permutation of variables for elimination.
- ***Factor Φ(X,Y)→ℜ:***
  - A function mapping some set of variables to a real value.
- ***scope[Φ]***:
  - The set of variables represented in the factor.
- ***width[Ω] :***
  - The scope of the largest factor produced by Ω.

FBE: Steps

- FBE consists of a series of elimination steps.
- Each step is as follows:
  - Select a variable X from the set of variables remaining.
  - Multiply all factors τ where $X \in$ scope[τ] to produce a new factor ψ.
  - Sum X out of ψ to produce a new factor τ whose scope is ψ minus X.
- Repeat until a single factor $\tau(Y)$ remains where Y is the target variable of our inference.

Complexity

- The complexity of each elimination step is: $O(N_i k_i)$
  - Where:

$$\psi_i = \phi_1 \times \ldots \times \phi_{k_i}$$

$$N_i = | scope(\psi_i) |$$

- The complexity of the algorithm for a given ordering $\boldsymbol{\Omega}$ is: $O(nN_{\max})$
  - Where:
    - n is the initial number of factors in the graph.
    - $N_{\max} = width[\Omega]$.

FBE: Trace

Step	Var	Factors Used	New Factor
1	C	$\phi_c(C), \phi_D(D, C)$	$\tau_1(D)$
2	D	$\phi_G(G, I, D), \tau_1(D)$	$\tau_2(G, I)$
3	I	$\phi_I(I), \phi_S(S, I), \tau_2(G, I)$	$\tau_3(G, S)$
4	H	$\phi_H(H, G, J)$	$\tau_4(G, J)$
5	G	$\boxed{\tau_4(G, J), \tau_3(G, S), \phi_L(L, G)}$	$\tau_5(J, L, S)$
6	S	$\tau_5(J, L, S), \phi_J(J, L, S)$	$\tau_6(J, L)$
7	L	$\tau_6(J, L)$	$\tau_7(J)$

FBE: Trace.

Step	Var	Factors Used	New Factor	Complexity
1	C	$\phi_C(C)\phi_D(D,C)$	$\tau_1(D)$	2
2	D	$\phi_G(G,I,D)\tau_1(D)$	$\tau_2(G,I)$	3
3	I	$\phi_I(I)\phi_S(S,I)\tau_2(G,I)$	$\tau_3(G,S)$	3
4	H	$\phi_H(H,G,J)$	$\tau_4(G,J)$	3
5	G	$\tau_4(G,J)\tau_3(G,S)\phi_L(L,G)$	$\tau_5(J,L,S)$	4
6	S	$\tau_5(J,L,S)\phi_J(J,L,S)$	$\tau_6(J,L)$	3
7	L	$\tau_6(J,L)$	$\tau_7(J)$	2

Total cost: $width[\Omega] = 4$

$$O(nN_{\max}) = O(8 \times 4) = O(32)$$

Ordering 2

Step	Var	Factors Used	New Factor
1	G	$\phi_G(G,I,D), \phi_L(L,G)\phi_H(H,G,J)$	$\tau_1(I,D,L,J,H)$
2	I	$\phi_I(I), \phi_S(S,I)\tau_1(I,D,L,J,H)$	$\tau_2(D,L,S,J,H)$
3	S	$\phi_J(J,L,S), \tau_2(D,L,S,J,H)$	$\tau_3(D,L,J,H)$
4	L	$\tau_3(D,L,J,H)$	$\tau_4(D,J,H)$
5	H	$\tau_4(D,J,H)$	$\tau_5(D,J)$
6	C	$\tau_5(D,J), \phi_D(D,C)$	$\tau_6(D,J)$
7	D	$\tau_6(D,J)$	$\tau_7(J)$

FBE: Trace.

Step	Var	Factors Used	New Factor	Complexity
1	C	$\phi_G(G,I,D)\phi_L(L,G)\phi_H(H,G,J)$	$\tau_1(I,D,L,J,H)$	6
2	D	$\phi_I(I)\phi_S(S,I)\tau_1(I,D,L,J,H)$	$\tau_2(D,L,S,J,H)$	6
3	I	$\phi_J(J,L,S)\tau_2(D,L,S,J,H)$	$\tau_3(D,L,J,H)$	5
4	H	$\tau_3(D,L,J,H)$	$\tau_4(D,J,H)$	4
5	G	$\tau_4(D,J,H)$	$\tau_5(D,J)$	3
6	S	$\tau_5(D,J)\phi_D(D,C)$	$\tau_6(D,J)$	3
7	L	$\tau_6(D,J)$	$\tau_7(J)$	2

Total cost: $width[\Omega] = 6$

$O(nN_{\max}) = O(8 \times 6) = O(48)$

Optimal Permutation.

Optimal Ordering Ω ':

$$\Omega' \, s.t. \forall \Omega': width\ [\Omega'] < width\ [\Omega]$$

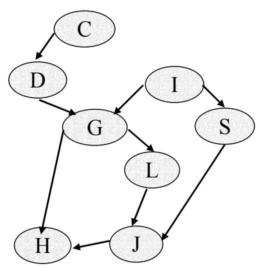
Note:

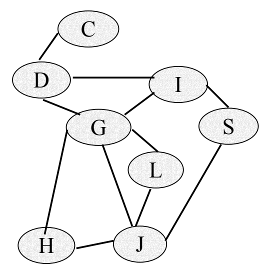
- The optimal ordering is not guaranteed to be unique.
- Nor is is guaranteed to be less than: $O(nN_{\max})$

Moral Graphs

Moral-graph H[G]: of a bayesian network over X is an undirected graph over X that contains an edge between x and y if:

- There exists a directed edge between them in G.
- They are both parents of the same node in G.



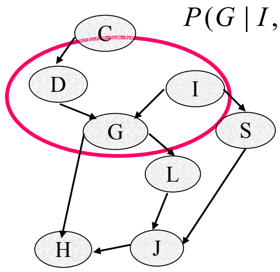


Moral Graphs

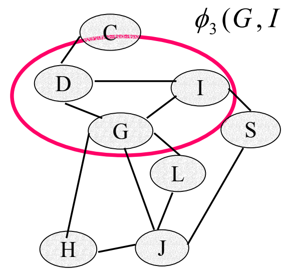
Why moralization?

$$P(C,D,G,I,S,L,J,H) =$$
$$= \quad P(C)P(D \mid C)P(G \mid I,D)P(S \mid I)P(L \mid G)P(J \mid L,S)P(H \mid G,J)$$
$$= \quad \phi_1(C)\phi_2(D,C)\phi_3(G,I,D)\phi_4(S,I)\phi_5(L,G)\phi_6(J,L,S)\phi_7(H,G,J)$$

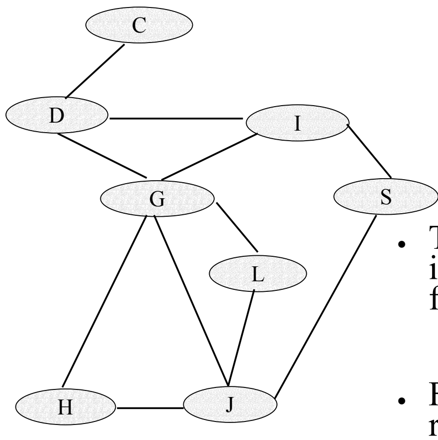
$P(G \mid I, D)$



$\phi_3(G, I, D)$



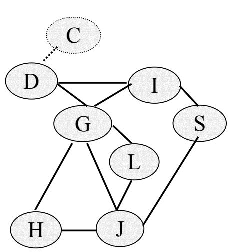
Variable Elimination



- The variable elimination algorithm is based only on the scope of each factor.
- Factors are distilled from the graph representation.
- Variable Elimination can therefore be viewed as a graph algorithm.

VE: Trace (1)

(1)



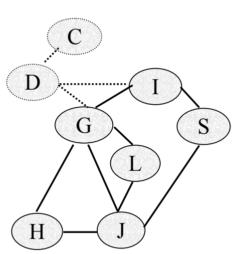
a) Multiply the factors to produce:

$$\phi(D,C) = \phi(D) \times \phi(C)$$

b) Sum over C to produce:

$$\tau(D) = \sum_{C} \phi(D,C)$$

(2)



a) Multiply the factors to produce:

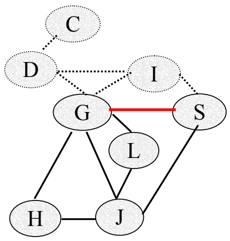
$$\phi(D,I,G) = \phi(G,I,D) \times \tau_1(D)$$

b) Sum over D to produce:

$$\tau_2(G,I) = \sum_{D} \phi(D,I,G)$$

VE: Trace (2)

(3)



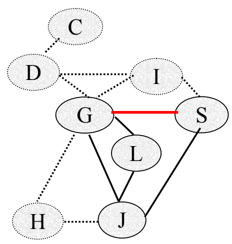
a) Multiply the factors to produce:

$$\phi(I,G,S) = \phi(I) \times \phi(S,I) \times \tau_2(G,I)$$

b) Sum over I to produce:

$$\tau_3(G,S) = \sum_I \phi(I,G,S)$$

(4)



a) Multiply the factors to produce:

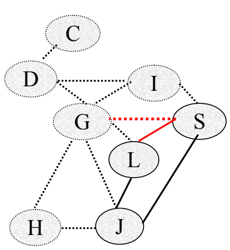
$$\phi(H,G,J) = \phi(H,G,J)$$

b) Sum over I to produce:

$$\tau_4(G,J) = \sum_H \phi(H,G,J)$$

VE: Trace (3)

(5)



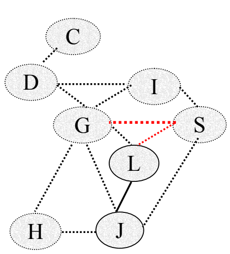
a) Multiply the factors to produce:

$$\phi(G,J,L,S) = \tau_4(G,J) \times \tau_3(G,S) \times \phi(L,G)$$

b) Sum over I to produce:

$$\tau_5(J,L,S) = \sum_G \phi(G,J,L,S)$$

(6)



a) Multiply the factors to produce:

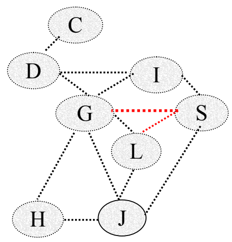
$$\phi(J,L,S) = \tau_5(J,L,S) \times \phi(J,L,S)$$

b) Sum over I to produce:

$$\tau_6(J,L) = \sum_S \phi(J,L,S)$$

VE: Trace (4)

(5)



a) Multiply the factors to produce:

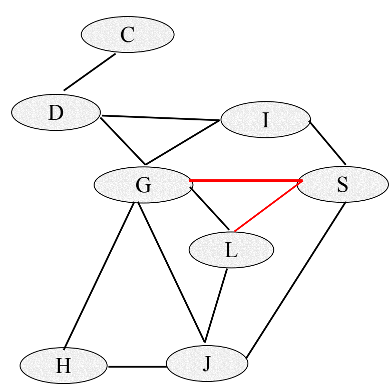
$$\phi(J,L) = \tau_6(J,L)$$

b) Sum over I to produce:

$$\tau_7(J) = \sum_L \phi(J,L)$$

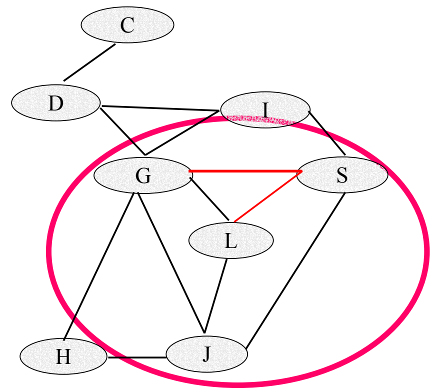
Variable elimination: Induced Graph

- Induced Graph *G′:* An undirected graph over X where y and z are connected if they both appear in some intermediate elimination factor of Ω.
- Every factor generated during Ω appears as a subclique of the graph.



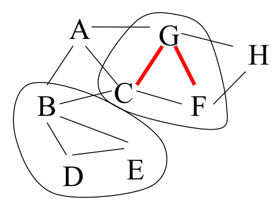
Variable elimination: Induced Graph

- Induced Graph *G'*: An undirected graph over X where y and z are connected if they both appear in some intermediate elimination factor of Ω.
- Every factor generated during Ω appears as a subclique of the graph.
- **The size of the largest clique governs the computation.**



Step	Var	Factors Used	New Factor
1	C	$\phi_c(C), \phi_D(D, C)$	$\tau_1(D)$
2	D	$\phi_G(G, I, D), \tau_1(D)$	$\tau_2(G, I)$
3	I	$\phi_I(I), \phi_S(S, I), \tau_2(G, I)$	$\tau_3(G, S)$
4	H	$\phi_H(H, G, J)$	$\tau_4(G, J)$
5	G	$\tau_4(G, J), \tau_3(G, S), \phi_L(L, G)$	$\tau_5(J, L, S)$
6	S	$\tau_5(J, L, S), \phi_J(J, L, S)$	$\tau_6(J, L)$
7	L	$\tau_6(J, L)$	$\tau_7(J)$

dence



a1	b1	0.5
a1	b2	0.2
a2	b1	0.1
a2	b2	0.3
a3	b1	0.2
a3	b2	0.4

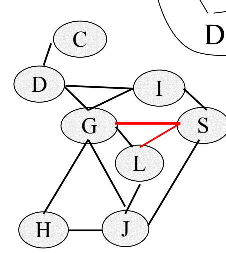
•

b1	c1	0.1
b1	c2	0.6
b2	c1	0.3
b2	c2	0.4

=

a1	b1	c1	0.5*0.
a1	b1	c2	0.51*0
a1	b2	c1	0.26*0.
a1	b2	c2	0.23*0.
a2	b1	c1	0.14*0.
a2	b1	c2	0.11*0.
a2	b2	c1	0.36*0.
a2	b2	c2	0.33*0.
a3	b1	c1	0.24*0.
a3	b1	c2	0.21*0.
a3	b2	c1	0.46*0.
a3	b2	c2	0.43*0

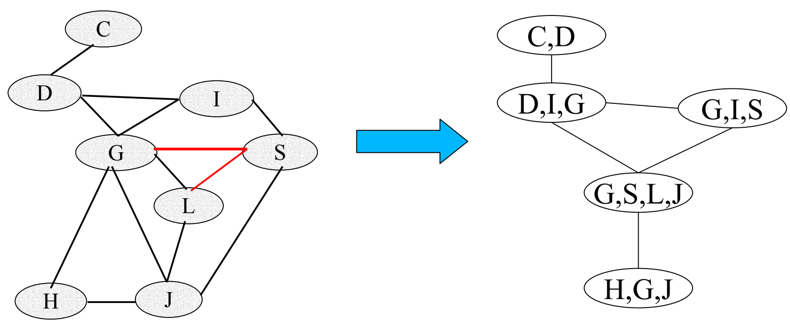
4



- *Tree-Width = best Max Factor-Width -1 = best Max Clique-Size -1*
- the tree-width defined by transformation of the undirected graph to the best factor tree is the same

Induced Graph to Tree Decomposition

- For each Clique in the induced graph:
  - Collapse the clique into a compound factor node containing the member variables.
  - Draw a link between each pair of nodes that share a member variable.

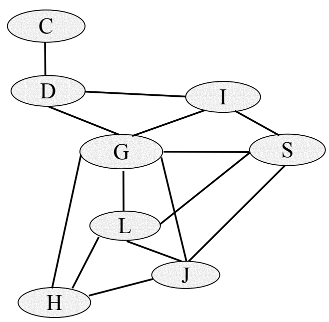


The Bad News

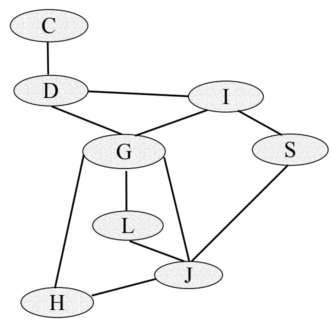
- Deciding whether or not there exists an ordering Ω s.t. W*idth(Ω) < N is NP-Hard.*
  - *No matter what method is used.*
- At best the inference is still exponential in the treewidth of the factor graph

Chordal graphs and Triangulations

Chordal Graph: an undirected graph *G* whose minimum cycle contains 3 verticies.



Chordal.



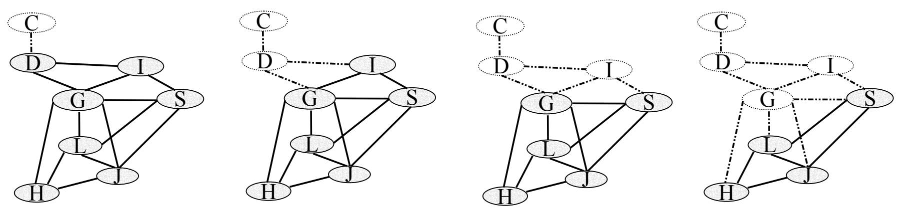
Not Chordal.

Triangulation

- The process of converting a graph G into a chordal graph is called Triangulation.
- The induced graph is:
  1) Guaranteed to be chordal.
  2) Not guaranteed to be optimal.
- There exist exact algorithms for minimal chordal graphs, and heuristic methods with a guaranteed upper bound.

Chordal Graphs: Good News

- If the moralized graph of our original network is chordal:
  - There exists an elimination ordering that adds no edges.
  - The minimal induced width of the graph is the size of the largest clique - 1.



Chordal Graphs: Bad News

- Given a minimum triangulation for a graph *G*, we can carry out the variable-elimination algorithm in the minimum possible time.

- However, finding the minimal triangulation is NP-Hard.
  - Time is exponential in terms of the largest clique (factor) in *G*.

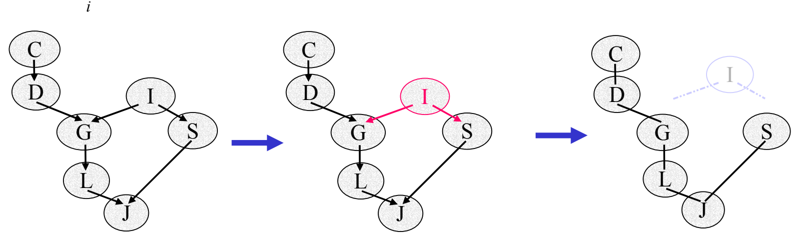
Initial Conclusions

- We cannot escape **exponential costs in the treewidth.**
- But in many graphs the treewidth is much smaller than the total number of variables
- Still a problem: Finding the optimal decomposition is hard
  - But, paying the cost up front may be worth it.
  - Triangulate once, query many times.
  - Real cost savings if not a bounded one.

Conditioning

- Up until now we have ignored some independences
- Assume the Student network from Koller and Friedman

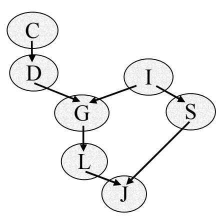
$$\begin{aligned}
P(J) &= \sum_{c,d,i,g,s,l} P(C=c, D=d, I=i, G=g, S=s, L=l, J) \\
&= \sum_{c,d,i,g,s,l} P(I=i) P(C=c, D=d, G=g, S=s, L=l, J \mid I=i) \\
&= \sum_{i} P(I=i) \sum_{c,d,g,s,l} P(C=c, D=d, G=g, S=s, L=l, J \mid I=i) \\
&= \sum_{i} P(I=i) P(J \mid I=i)
\end{aligned}$$

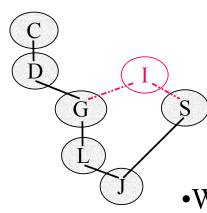


Conditioning

Concept:

- Fix some variable I s.t. I=i and remove it from the graph.
- Recalculate neighboring probabilities.
- Execute VE with the reduced graph.





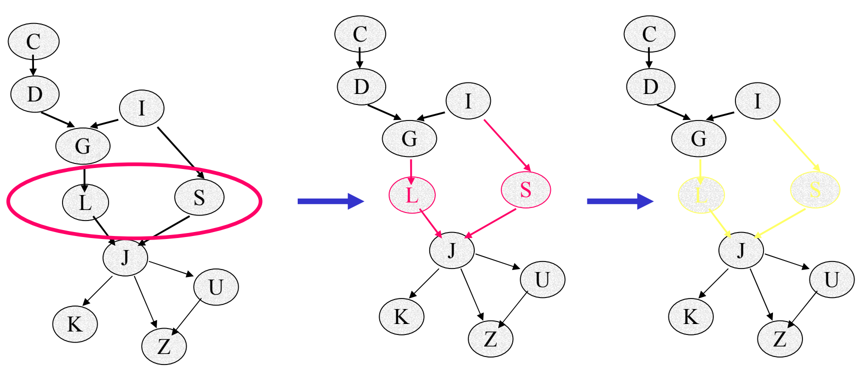
•We need to do a separate inference for every value i of I

•But the structure of the network is much simpler

Conditioning

General conditioning works for sets of variables.

Assume:



Conditioning conclusions

- Conditioning on some variables may simplify the structure of the remaining network
  - The network may break into small pieces.
- Variable elimination may be easier to perform on the new network