Readings: K&F 9.2, 9.3, 9.4, 9.5

Exact Inference: Variable Elimination

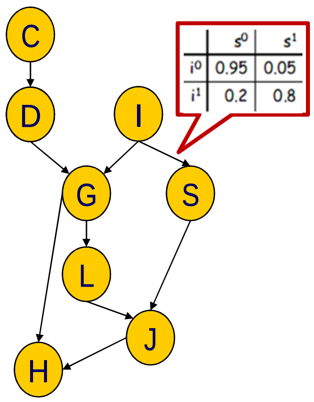
Lecture 6-7 – Apr 13/18, 2011

CSE 515, Statistical Methods, Spring 2011

Instructor: Su-In Lee

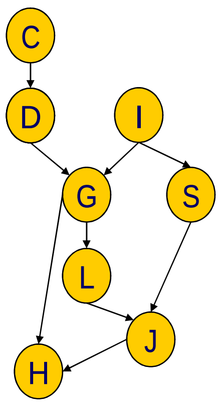
University of Washington, Seattle

Let's revisit the *Student* Network



- Binary RVs
  - **C**oherence
  - **D**ifficulty
  - **I**ntelligence
  - **G**rade
  - **S**AT
  - **L**etter
  - **J**ob
  - **H**appy
- Notations & abbreviations
  - J : a random variable
  - **X** : a set of random variables
  - Val(J) : a set of values on J
  - j : a value on J
  - |J| : size of Val(J)
  - P(j) : P(J=j)
- Assumptions
  - Local probabilistic models: table CPDs
  - Parameters and structure are given.

Inference Tasks in *Student* Network



- (Conditional) probability queries
  - $P(l^1)$ or $P(L=l^1)$
  - $P(h^0)$ or $P(H=h^0)$
  - $P(j^1)$ or $P(J=j^1)$

  - $P(j^1|i^1, d^1)$ or $P(J=j^1|I=i^0,D=d^1)$
  - $P(j^1|h^0, i^1)$ or $P(J=j^0|H=h^0, I=i^1)$
  - $P(j^1,i^0|h^0)$

    :

Query RV(s)

Evidence RV(s)

- How to compute the probabilities?
  - Use joint distribution $P(C,D,I,G,S,L,J,H)$

Naïve Approach

- Use full joint distribution $P(C,D,I,G,S,L,J,H)$
- Computing $P(J=j^1)$

$P(j^1) = P(c^0,d^0,i^0,g^0,s^0,l^0, j^1,h^0)$
$+ P(c^0,d^0,i^0,g^0,s^0,l^0, j^1,h^1)$
$+ P(c^0,d^0,i^0,g^0,s^0,l^1, j^1,h^0)$
$+ P(c^0,d^0,i^0,g^0,s^0,l^1, j^1,h^1)$
$+ P(c^0,d^0,i^0,g^0,s^1,l^0, j^1,h^0)$
$+ P(c^0,d^0,i^0,g^0,s^1,l^0, j^1,h^1)$
$+ P(c^0,d^0,i^0,g^0,s^1,l^1, j^1,h^0)$
$+ P(c^0,d^0,i^0,g^0,s^1,l^1, j^1,h^1)$
$+ P(c^0,d^0,i^0,g^1,s^0,l^0, j^1,h^0)$
$+ P(c^0,d^0,i^0,g^1,s^0,l^0, j^1,h^1)$
$+ P(c^0,d^0,i^0,g^1,s^0,l^1, j^1,h^0)$
$+ P(c^0,d^0,i^0,g^1,s^0,l^1, j^1,h^1)$
$+ P(c^0,d^0,i^0,g^1,s^1,l^0, j^1,h^0)$
$+ P(c^0,d^0,i^0,g^1,s^1,l^0, j^1,h^1)$
$+ P(c^0,d^0,i^0,g^1,s^1,l^1, j^1,h^0)$
$+ P(c^0,d^0,i^0,g^1,s^1,l^1, j^1,h^1)$
$+ \ldots$:

- Computing $P(I=i^0,J=j^1)$

$P(i^0,j^1) = P(c^0,d^0,i^0,g^0,s^0,l^0, j^1,h^0)$
$+ P(c^0,d^0,i^0,g^0,s^0,l^0, j^1,h^1)$
$+ P(c^0,d^0,i^0,g^0,s^0,l^1, j^1,h^0)$
$+ P(c^0,d^0,i^0,g^0,s^0,l^1, j^1,h^1)$
$+ P(c^0,d^0,i^0,g^0,s^1,l^0, j^1,h^0)$
$+ P(c^0,d^0,i^0,g^0,s^1,l^0, j^1,h^1)$
$+ P(c^0,d^0,i^0,g^0,s^1,l^1, j^1,h^0)$
$+ P(c^0,d^0,i^0,g^0,s^1,l^1, j^1,h^1)$
$+ P(c^0,d^0,i^0,g^1,s^0,l^0, j^1,h^0)$
$+ P(c^0,d^0,i^0,g^1,s^0,l^0, j^1,h^1)$
$+ P(c^0,d^0,i^0,g^1,s^0,l^1, j^1,h^0)$
$+ P(c^0,d^0,i^0,g^1,s^0,l^1, j^1,h^1)$
$+ P(c^0,d^0,i^0,g^1,s^1,l^0, j^1,h^0)$
$+ P(c^0,d^0,i^0,g^1,s^1,l^0, j^1,h^1)$
$+ P(c^0,d^0,i^0,g^1,s^1,l^1, j^1,h^0)$
$+ P(c^0,d^0,i^0,g^1,s^1,l^1, j^1,h^1)$
$+ \ldots$:

- Computational complexity: exponential blowup
- Exploiting the independence properties?

Naïve Approach

- $P(C,D,I,G,S,L,J,H) =$ $P(C)P(D|C)P(I)P(G|I,D)P(S|I)P(L|G)P(J|L,S)P(H|G,J)$
- Computing $P(J)$

$$
\begin{aligned}
P(j^1) = P(c^0)P(d^0|c^0)\,P(i^0)\,[\,& P(g^0|i^0,d^0)\,P(s^0|i^0)P(l^0|g^0)P(j^1|l^0,s^0)P(h^0|g^0,j^1) \\
&+ P(g^0|i^0,d^0)P(s^0|i^0)P(l^0|g^0)P(j^1|l^0,s^0)P(h^1|g^0,j^1) \\
&+ P(g^0|i^0,d^0)P(s^0|i^0)P(l^1|g^0)P(j^1|l^1,s^0)P(h^0|g^0,j^1) \\
&+ P(g^0|i^0,d^0)P(s^0|i^0)P(l^1|g^0)P(j^1|l^1,s^0)P(h^1|g^0,j^1) \\
&+ P(g^0|i^0,d^0)P(s^1|i^0)P(l^0|g^0)P(j^1|l^0,s^1)P(h^0|g^0,j^1) \\
&+ P(g^0|i^0,d^0)P(s^1|i^0)P(l^0|g^0)P(j^1|l^0,s^1)P(h^1|g^0,j^1) \\
&+ P(g^0|i^0,d^0)P(s^1|i^0)P(l^1|g^0)P(j^1|l^1,s^1)P(h^0|g^0,j^1) \\
&+ P(g^0|i^0,d^0)P(s^1|i^0)P(l^1|g^0)P(j^1|l^1,s^1)P(h^1|g^0,j^1) \\
&+ P(g^1|i^0,d^0)P(s^0|i^0)P(l^0|g^1)P(j^1|l^0,s^0)P(h^0|g^1,j^1) \\
&+ P(g^1|i^0,d^0)P(s^0|i^0)P(l^0|g^1)P(j^1|l^0,s^0)P(h^1|g^1,j^1) \\
&+ P(g^1|i^0,d^0)P(s^0|i^0)P(l^1|g^1)P(j^1|l^1,s^0)P(h^0|g^1,j^1) \\
&+ P(g^1|i^0,d^0)P(s^0|i^0)P(l^1|g^1)P(j^1|l^1,s^0)P(h^1|g^1,j^1) \\
&+ P(g^1|i^0,d^0)P(s^1|i^0)P(l^0|g^1)P(j^1|l^0,s^1)P(h^0|g^1,j^1) \\
&+ P(g^1|i^0,d^0)P(s^1|i^0)P(l^0|g^1)P(j^1|l^0,s^1)P(h^1|g^1,j^1) \\
&+ P(g^1|i^0,d^0)P(s^1|i^0)P(l^1|g^1)P(j^1|l^1,s^1)P(h^0|g^1,j^1) \\
&+ P(g^1|i^0,d^0)P(s^1|i^0)P(l^1|g^1)P(j^1|l^1,s^1)P(h^1|g^1,j^1)
\end{aligned}
$$

. ... .

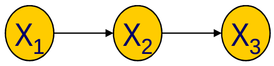
Certain terms are repeated several times

- Exploiting the structure can reduce computation.
- Let’s systematically analyze computational complexity.

Let’s start with the simplest network ...

Exact Inference Variable Elimination

- Inference in a simple chain
  - Computing $P(X_2)$



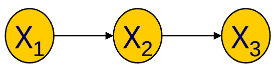
$$P(X_2) = \sum_{x_1} P(x_1, X_2)$$

All the numbers for this computation are in the CPDs of the original Bayesian network

O( ) operations

Exact Inference Variable Elimination

- Inference in a simple chain
  - Computing $P(X_2)$



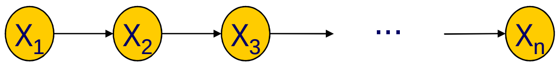
  - Computing $P(X_3)$

$$P(X_2) = \sum_{x_1} P(x_1, X_2) = \sum_{x_1} P(x_1) P(X_2 \mid x_1)$$

$$P(X_3) =$$

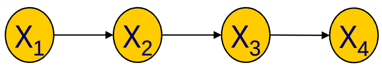
- $P(X_3|X_2)$ is a given CPD
- $P(X_2)$ was computed above
- O(|) operations

Exact Inference: Variable Elimination



- Inference in a general chain
  - Computing $P(X_n)$
    - Compute each $P(X_i)$ from $P(X_{i-1})$
    - k^2 operations for each computation for X_i (assuming $|X_i|=k$)
    - $O(nk^2)$ operations for the inference
    - Compare to k^n operations required in summing over all possible entries in the joint distribution over $X_1,\ldots X_n$
  - Inference in a general chain can be done in linear time!

Exact Inference: Variable Elimination

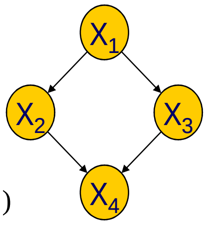


$$
\begin{aligned}
P(X_4) &= \sum_{X_1}\sum_{X_2}\sum_{X_3} P(X_1, X_2, X_3, X_4) \\
&= \sum_{X_1}\sum_{X_2}\sum_{X_3} P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_2)P(X_4 \mid X_3) \\
&= \sum_{X_3} P(X_4 \mid X_3) \sum_{X_2} P(X_3 \mid X_2) \sum_{X_1} P(X_1)P(X_2 \mid X_1) \\
&= \sum_{X_3} P(X_4 \mid X_3) \sum_{X_2} P(X_3 \mid X_2)\phi(X_2) \\
&= \sum_{X_3} P(X_4 \mid X_3)\phi(X_3) \\
&= \phi(X_4)
\end{aligned}
$$

Pushing summations = Dynamic programming

Inference With a Loop

- Computing $P(X_4)$



$$
\begin{aligned}
P(X_4) &= \sum_{X_1}\sum_{X_2}\sum_{X_3} P(X_1, X_2, X_3, X_4) \\
&= \sum_{X_1}\sum_{X_2}\sum_{X_3} P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1)P(X_4 \mid X_2, X_3) \\
&= \sum_{X_2}\sum_{X_3} P(X_4 \mid X_2, X_3)\sum_{X_1} P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1) \\
&= \sum_{X_2}\sum_{X_3} P(X_4 \mid X_2, X_3)\phi(X_{2,3}) \\
&= \sum_{X_2} \phi(X_2, X_4) \\
&= \phi(X_4)
\end{aligned}
$$

- Differences
  - Summations are not "pushed in" as far as before.
  - The scope of ϕ includes two variables, not one.
- Depends on network structure

Efficient Inference in Bayesnets

- Properties that allow us to avoid exponential blowup in the joint distribution
  - Bayesian network structure – some subexpressions depend on a small number of variables
  - Computing these subexpressions and caching the results avoids generating them exponentially many times

Variable Elimination: Factors

- Inference algorithm defined in terms of factors
- Factors generalize the notion of CPDs
- A factor ϕ is a function from value assignments of a set of random variables **D** to real positive numbers $\Re^+$
  - The set of variables **D** is the scope of the factor
- Thus, the algorithm we describe applies both to Bayesian networks and Markov networks

Operations on Factors I: Product

- Let **X**, **Y**, **Z** be three sets of disjoint sets of RVs, and let ϕ_1(**X**,**Y**) and ϕ_2(**Y**,**Z**) be two factors
- We define the factor product $\phi_1 x \phi_2$ operation to be a factor ψ:Val(**X**,**Y**,**Z**) → $\Re$ as $\psi(\mathbf{X},\mathbf{Y},\mathbf{Z})=\phi_1(\mathbf{X},\mathbf{Y})\phi_2(\mathbf{Y},\mathbf{Z})$

X	Y	ϕ_1 [X,Y]
x^0	y^0	2
x^0	y^1	3
x^1	y^0	10
x^1	y^1	5

Y	Z	ϕ_2 [X,Y]
y^0	z^0	6
y^0	z^1	1
y^1	z^0	7
y^1	z^1	12

X	Y	Z	Ψ [X,Y,Z]
x^0	y^0	z^0	
x^0	y^0	z^1	
x^0	y^1	z^0	
x^0	y^1	z^1	
x^1	y^0	z^0	
x^1	y^0	z^1	
x^1	y^1	z^0	
x^1	y^1	z^1	

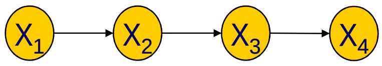
Operations on Factors II: Marginalization

- Let **X** be a set of RVs, $Y \notin \mathbf{X}$ a RV, and $\phi(\mathbf{X},Y)$ a factor
- We define the factor marginalization of Y in **X** to be a factor ψ:Val(**X**) → $\Re$ as $\psi(\mathbf{X}) = \Sigma_Y \phi(\mathbf{X},Y)$
- Also called summing out
- In a Bayesian network, summing out all variables =
- In a Markov network, summing out all variables is the

More on Factors

- For factors ϕ_1 and ϕ_2:
- Factors are commutative
  - $\phi_1 x \phi_2 = \phi_1 x \phi_2$
  - $\Sigma_{\mathbf{X}} \Sigma_{\mathbf{Y}} \phi(\mathbf{X},\mathbf{Y}) = \Sigma_{\mathbf{Y}} \Sigma_{\mathbf{X}} \phi(\mathbf{X},\mathbf{Y})$
- Products are associative
  - $(\phi_1 x \phi_2) x \phi_3 = \phi_1 x (\phi_2 x \phi_3)$
- If $\mathbf{X} \notin$ Scope$[\phi_1]$ (we used this in elimination above)
  - $\Sigma_{\mathbf{X}} \phi_1 \text{ x } \phi_2 = \phi_1 \text{ x } \Sigma_{\mathbf{X}} \phi_2$

Inference in Chain by Factors



$$
\begin{aligned}
P(X_4) &= \sum_{X_1}\sum_{X_2}\sum_{X_3} P(X_1, X_2, X_3, X_4) \\
&= \sum_{X_1}\sum_{X_2}\sum_{X_3} \phi_{X_1} \times \phi_{X_2} \times \phi_{X_3} \times \phi_{X_4} \\
&= \sum_{X_3}\sum_{X_2} \phi_{X_4} \times \phi_{X_3} \times \left(\sum_{X_1} \phi_{X_1} \times \phi_{X_2} \right) \\
&= \sum_{X_3} \phi_{X_4} \times \left(\sum_{X_2} \phi_{X_3} \times \left(\sum_{X_1} \phi_{X_1} \times \phi_{X_2} \right) \right)
\end{aligned}
$$

Scope of ϕ_{X_3} and ϕ_{X_4} does not contain X_1

Scope of ϕ_{X_4} does not contain X_2

17

Sum-Product Inference

- Let **Y** be the query RVs and **Z** be all other RVs
- We can generalize this task as that of computing the value of an expression of the form:

$$\phi(\mathbf{Y}) = \sum_{\mathbf{Z}} \prod_{\phi' \in F} \phi'$$

- Call it sum-product inference task.
- Effective computation
  - The scope of the factors is limited.
  - → "Push in" some of the summations, performing them over the product of only a subset of factors

18

Sum-Product Variable Elimination

- Algorithm
  - Given an ordering of variables $Z_1,\ldots,Z_n$,
  - Sum out the variables one at a time
  - When summing out each variable Z,
    - Multiply all the factors ϕ's that mention the variable, generating a product factor Ψ
    - Sum out the variable from the combined factor Ψ, generating a new factor *f* without the variable Z

Sum out

- Let **X** be a set of RVs, $Y \notin \mathbf{X}$ a RV, and $\phi(\mathbf{X},Y)$ a factor
- We define the factor marginalization of Y in **X** to be a factor $\psi:Val(\mathbf{X}) \rightarrow \Re$ as $\psi(\mathbf{X}) = \Sigma_Y \phi(\mathbf{X},Y)$
- Also called summing out

Page 14

Sum-Product Variable Elimination

- Theorem
  - Let **X** be a set of RVs
  - Let $\mathbf{Y} \subseteq \mathbf{X}$ be a set of query RVs
  - Let **Z**=**X**-**Y**
  - → For any ordering α over **Z**, the above algorithm returns a factor $\phi(\mathbf{Y})$ such that $\phi(\mathbf{Y}) = \sum_{\mathbf{Z}} \prod_{\phi' \in F} \phi'$
- Bayesian network query $P_G(\mathbf{Y})$
  - F consists of all CPDs in G $F = \{\phi_{X_i}\}_{i=1}^{n}$
  - Each $\phi_{X_i} = P(X_i \mid Pa(X_i))$
  - Apply variable elimination for **Z**=U-**Y** (summing out **Z**)

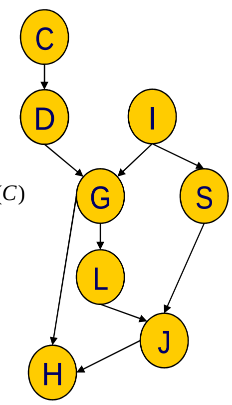
Example – Let's consider a little more complex network...

A More Complex Network

- Goal: P(J)
- Eliminate: C,D,I,H,G,S,L

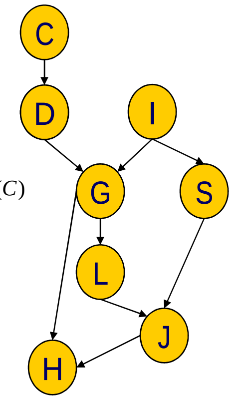
$$P(J) = \sum_{L,S,G,H,I,D,C} P(J \mid L,S)P(L \mid G)P(S \mid I)P(G \mid I,D)P(H \mid G,J)P(I)P(C \mid D)P(C)$$

$$P(J) = \sum_{L,S,G,H,I,D,C} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)\phi_D(C,D)\phi_C(C)$$



A More Complex Network

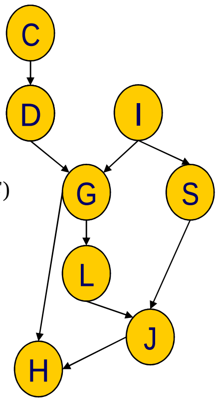
- Goal: P(J)
- Eliminate: C,D,I,H,G,S,L
- Compute: $f_1(D) = \sum_C \phi_C(C)\phi_D(C,D)$



$$P(J) = \sum_{L,S,G,H,I,D,C} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)\phi_D(C,D)\phi_C(C)$$
$$= \sum_{L,S,G,H,I,D} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)f_1(D)$$

A More Complex Network

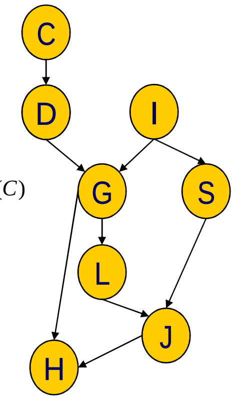
- Goal: P(J)
- Eliminate: C,D,I,H,G,S,L
- Compute: $f_2(G,I) = \sum_D \phi_G(G,I,D)f_1(D)$



$$P(J) = \sum_{L,S,G,H,I,D,C} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)\phi_D(C,D)\phi_C(C)$$
$$= \sum_{L,S,G,H,I,D} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)f_1(D)$$
$$= \sum_{L,S,G,H,I} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_H(H,G,J)\phi_I(I)f_2(G,I)$$

A More Complex Network

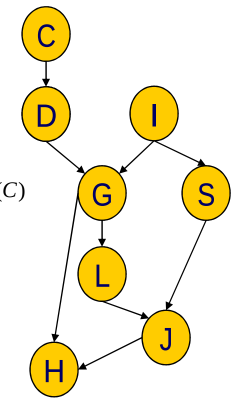
- Goal: P(J)
- Eliminate: C,D,I,H,G,S,L
- Compute: $f_3(G,S)=\sum_{I}\phi_I(I)\phi_S(S,I)f_2(G,I)$



$$P(J)=\sum_{L,S,G,H,I,D,C}\phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)\phi_D(C,D)\phi_C(C)$$
$$=\sum_{L,S,G,H,I,D}\phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)f_1(D)$$
$$=\sum_{L,S,G,H,I}\phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_H(H,G,J)\phi_I(I)f_2(G,I)$$
$$=\sum_{L,S,G,H}\phi_J(J,L,S)\phi_L(L,G)\phi_H(H,G,J)f_3(G,S)$$

A More Complex Network

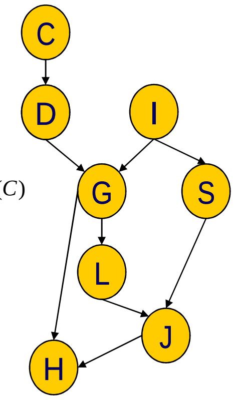
- Goal: P(J)
- Eliminate: C,D,I,H,G,S,L
- Compute: $f_4(G,J)=\sum_{H}\phi_H(H,G,J)$



$$P(J)=\sum_{L,S,G,H,I,D,C}\phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)\phi_D(C,D)\phi_C(C)$$
$$=\sum_{L,S,G,H,I,D}\phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)f_1(D)$$
$$=\sum_{L,S,G,H,I}\phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_H(H,G,J)\phi_I(I)f_2(G,I)$$
$$=\sum_{L,S,G,H}\phi_J(J,L,S)\phi_L(L,G)\phi_H(H,G,J)f_3(G,S)$$
$$=\sum_{L,S,G}\phi_J(J,L,S)\phi_L(L,G)f_3(G,S)f_4(G,J)$$

A More Complex Network

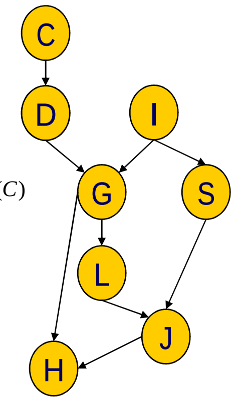
- Goal: P(J)
- Eliminate: C,D,I,H,G,S,L
- Compute: $f_5(J,L,S) = \sum_{G} \phi_L(L,G) f_3(G,S) f_4(G,J)$



$$
\begin{aligned}
P(J) &= \sum_{L,S,G,H,I,D,C} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)\phi_D(C,D)\phi_C(C) \\
&= \sum_{L,S,G,H,I,D} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I) f_1(D) \\
&= \sum_{L,S,G,H,I} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_H(H,G,J)\phi_I(I) f_2(G,I) \\
&= \sum_{L,S,G,H} \phi_J(J,L,S)\phi_L(L,G)\phi_H(H,G,J) f_3(G,S) \\
&= \sum_{L,S,G} \phi_J(J,L,S)\phi_L(L,G) f_3(G,S) f_4(G,J) \\
&= \sum_{L,S} \phi(J,L,S) f_5(J,L,S)
\end{aligned}
$$

A More Complex Network

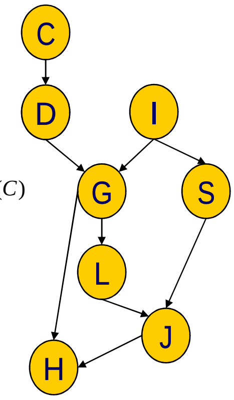
- Goal: P(J)
- Eliminate: C,D,I,H,G,S,L
- Compute: $f_6(J,L) = \sum_{S} \phi(J,L,S) f_5(J,L,S)$



$$
\begin{aligned}
P(J) &= \sum_{L,S,G,H,I,D,C} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)\phi_D(C,D)\phi_C(C) \\
&= \sum_{L,S,G,H,I,D} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I) f_1(D) \\
&= \sum_{L,S,G,H,I} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_H(H,G,J)\phi_I(I) f_2(G,I) \\
&= \sum_{L,S,G,H} \phi_J(J,L,S)\phi_L(L,G)\phi_H(H,G,J) f_3(G,S) \\
&= \sum_{L,S,G} \phi_J(J,L,S)\phi_L(L,G) f_3(G,S) f_4(G,J) \\
&= \sum_{L,S} \phi(J,L,S) f_5(J,L,S) \\
&= \sum_{L} f_6(J,L)
\end{aligned}
$$

A More Complex Network

- Goal: P(J)
- Eliminate: C,D,I,H,G,S,L
- Compute: $f_7(J) = \sum_L f_6(J,L)$

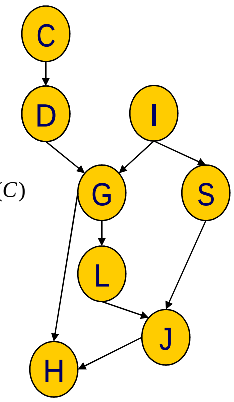


$$P(J) = \sum_{L,S,G,H,I,D,C} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)\phi_D(C,D)\phi_C(C)$$
$$= \sum_{L,S,G,H,I,D} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)f_1(D)$$
$$= \sum_{L,S,G,H,I} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_H(H,G,J)\phi_I(I)f_2(G,I)$$
$$= \sum_{L,S,G,H} \phi_J(J,L,S)\phi_L(L,G)\phi_H(H,G,J)f_3(G,S)$$
$$= \sum_{L,S,G} \phi_J(J,L,S)\phi_L(L,G)f_3(G,S)f_4(G,J)$$
$$= \sum_{L,S} \phi(J,L,S)f_5(J,L,S)$$
$$= \sum_{L} f_6(J,L)$$
$$= f_7(J)$$

29

A More Complex Network

- Goal: P(J)
- Eliminate: G,I,S,L,H,C,D (different ordering)



$$P(J) = \sum_{G,I,S,L,H,C,D} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)\phi_D(C,D)\phi_C(C)$$
$$= \sum_{I,S,L,H,C,D} \phi_J(J,L,S)\phi_S(S,I)\phi_I(I)\phi_D(C,D)\phi_C(C)f_1(I,D,L,J,H)$$
$$= \sum_{S,L,H,C,D} \phi_J(J,L,S)\phi_D(C,D)\phi_C(C)f_2(D,L,S,J,H)$$
$$= \sum_{L,H,C,D} \phi_D(C,D)\phi_C(C)f_3(D,L,J,H)$$
$$= \sum_{H,C,D} \phi_D(C,D)\phi_C(C)f_4(D,J,H)$$
$$= \sum_{C,D} \phi_D(C,D)\phi_C(C)f_5(D,J)$$
$$= \sum_{D} f_5(D,J)f_6(D,J)$$
$$= f_7(J)$$

- Note: intermediate factors tend to be large $f_1(I,D,L,J,H)$
→ ordering matters

30

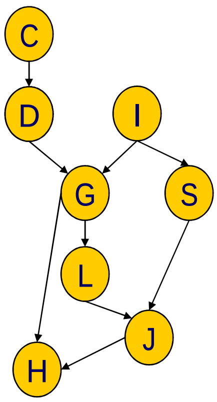
Inference With Evidence

- Computing P(**Y**|**E**=**e**)
- Let **Y** be the query RVs
- Let **E** be the evidence RVs and **e** their assignment
- Let **Z** be all other RVs (U-**Y**-**E**)
- The general inference task is

$$\frac{\phi(\mathbf{Y},\mathbf{e})}{\phi(\mathbf{e})} = \frac{\sum_{\mathbf{Z}} \prod_{X \in U} \phi_{X|\mathbf{E}=\mathbf{e}}}{\sum_{\mathbf{Y},\mathbf{Z}} \prod_{X \in U} \phi_{X|\mathbf{E}=\mathbf{e}}}$$

Inference With Evidence

- Goal: P(J|H=h,I=i)
- Eliminate: C,D,G,S,L
- Below, compute f(J,H=h,I=i)



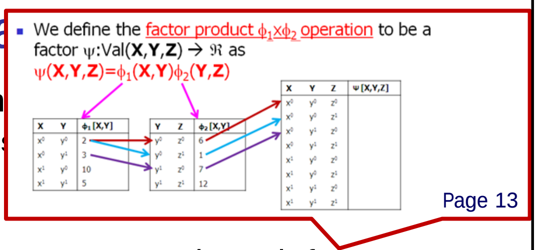
$$P(J,h,i) = \sum_{L,S,G,D,C} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,i)\phi_G(G,i,D)\phi_H(h,G,J)\phi_I(i)\phi_D(C,D)\phi_C(C)$$
$$= \sum_{L,S,G,D} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,i)\phi_G(G,i,D)\phi_H(h,G,J)\phi_I(i)f_1(D)$$
$$= \sum_{L,S,G} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,i)\phi_H(h,G,J)\phi_I(i)f_2(G,i)$$
$$= \sum_{L,S} \phi_J(J,L,S)\phi_S(S,i)f_3(L,J)$$
$$= \sum_{L} f_4(L,J)$$
$$= f_5(J)$$

- Differences
  - Less number of variables to be eliminated (H and I are excluded)
  - Scope of factors tend to be smaller.

What’s the complexity of variable elimination?

Complexity of Va

- Variable elimination con
  - Generating the factors f ‘s
  - Summing out



- Generating the factor $f_i=\phi_1$ x,…,x ϕ_{k_i} through factor product operation
  - Let $\mathbf{X_i}$ be the scope of f_i
  - Each entry requires k_i multiplications to generate
  - → Generating factor f_i is
- Summing out
  - Addition operations, at most $|Val(X_i)|$
- Per factor: O(kN) where $N=\max_i|Val(\mathbf{X_i})|$, $k=\max_i k_i$

Complexity of Variable Elimination

- Start with n factors (n=number of variables)
- Generate exactly one factor at each iteration
  → there are at most 2n factors

- Generating factors (Say, $N=\max_i|Val(\mathbf{X}_i)|$)
  - At most $\Sigma_i|Val(\mathbf{X}_i)|k_i \leq N\Sigma_i k_i \leq N\cdot 2n$ (since each factor is multiplied in exactly once and there are 2n factors)
- Summing out
  - $\Sigma_i|Val(\mathbf{X}_i)| \leq N\cdot n$ (since we have n summing outs to do)

- Total work is linear in N and n, where $N=\max_i|Val(\mathbf{X}_i)|$
- Exponential blowup can be in N_i which for factor i can be v^m if factor i has m variables with v values each

- Interpretation: maximum scope size is important.

35

Factors and Undirected Graphs

- The algorithm does not care whether the graph that generated the factors is directed or undirected.
  - The algorithm's input is a set of factors, and the only relevant aspect to the computational is the scope of the factors.

- Let's view the algorithm as operating on an undirected graph H.
  - For Bayesian networks, we consider the moralized Markov network of the original BNs.

- How does the network structure change in each variable elimination step?

36

VE as Graph Transformation

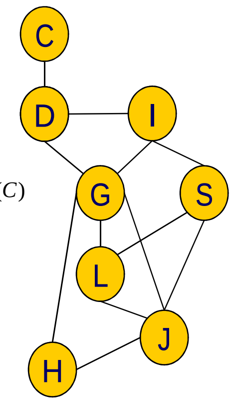
- At each step we are computing $f_i = \sum_{X_i} \prod_j f_j(\mathbf{Z}_j)$
- Plot a graph where there is an undirected edge X—Y if variables X and Y appear in the same factor
- Note: this is the Markov network of the probability on the variables that were not eliminated yet

37

VE as Graph Transformation

- Goal: P(J)
- Eliminate: C,D,I,H,G,S,L

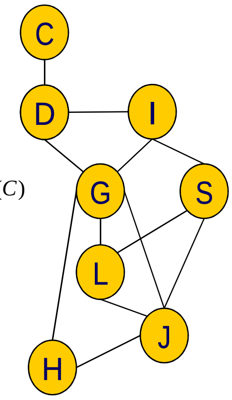
$$P(J) = \sum_{L,S,G,H,I,D,C} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)\phi_D(C,D)\phi_C(C)$$



38

VE as Graph Transformation

- Goal: P(J)
- Eliminate: C,D,I,H,G,S,L
- Compute: $f_1(D) = \sum_C \phi_C(C)\phi_D(C,D)$

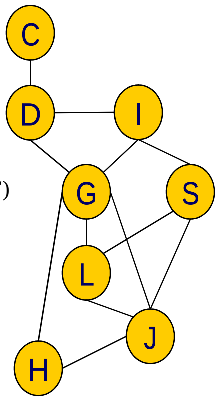


$$P(J) = \sum_{L,S,G,H,I,D,C} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)\phi_D(C,D)\phi_C(C)$$
$$= \sum_{L,S,G,H,I,D} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)f_1(D)$$

39

VE as Graph Transformation

- Goal: P(J)
- Eliminate: C,D,I,H,G,S,L
- Compute: $f_2(G,I) = \sum_D \phi_G(G,I,D)f_1(D)$

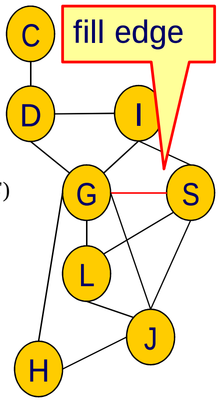


$$P(J) = \sum_{L,S,G,H,I,D,C} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)\phi_D(C,D)\phi_C(C)$$
$$= \sum_{L,S,G,H,I,D} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)f_1(D)$$
$$= \sum_{L,S,G,H,I} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_H(H,G,J)\phi_I(I)f_2(G,I)$$

40

VE as Graph Transformation

- Goal: P(J)
- Eliminate: C,D,I,H,G,S,L
- Compute: $f_3(G,S)=\sum_I \phi_I(I)\phi_S(S,I)f_2(G,I)$

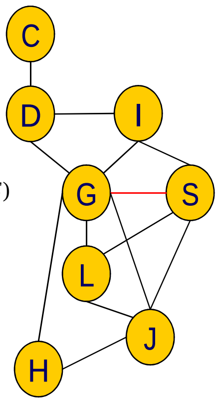


$$P(J)=\sum_{L,S,G,H,I,D,C}\phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)\phi_D(C,D)\phi_C(C)$$
$$=\sum_{L,S,G,H,I,D}\phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)f_1(D)$$
$$=\sum_{L,S,G,H,I}\phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_H(H,G,J)\phi_I(I)f_2(G,I)$$
$$=\sum_{L,S,G,H}\phi_J(J,L,S)\phi_L(L,G)\phi_H(H,G,J)f_3(G,S)$$

41

VE as Graph Transformation

- Goal: P(J)
- Eliminate: C,D,I,H,G,S,L
- Compute: $f_4(G,J)=\sum_H \phi_H(H,G,J)$

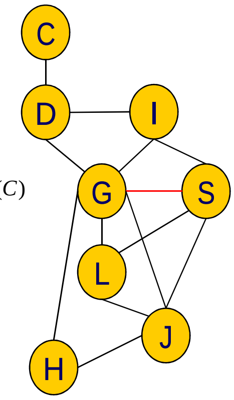


$$P(J)=\sum_{L,S,G,H,I,D,C}\phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)\phi_D(C,D)\phi_C(C)$$
$$=\sum_{L,S,G,H,I,D}\phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)f_1(D)$$
$$=\sum_{L,S,G,H,I}\phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_H(H,G,J)\phi_I(I)f_2(G,I)$$
$$=\sum_{L,S,G,H}\phi_J(J,L,S)\phi_L(L,G)\phi_H(H,G,J)f_3(G,S)$$
$$=\sum_{L,S,G}\phi_J(J,L,S)\phi_L(L,G)f_3(G,S)f_4(G,J)$$

42

VE as Graph Transformation

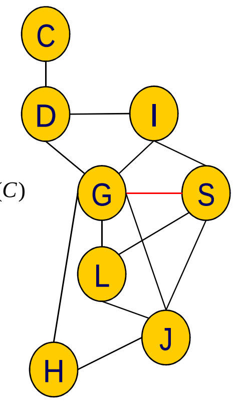
- Goal: P(J)
- Eliminate: C,D,I,H,G,S,L
- Compute: $f_5(J,L,S)=\sum_G \phi_L(L,G)\,f_3(G,S)\,f_4(G,J)$



$$P(J)=\sum_{L,S,G,H,I,D,C}\phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)\phi_D(C,D)\phi_C(C)$$
$$=\sum_{L,S,G,H,I,D}\phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)f_1(D)$$
$$=\sum_{L,S,G,H,I}\phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_H(H,G,J)\phi_I(I)f_2(G,I)$$
$$=\sum_{L,S,G,H}\phi_J(J,L,S)\phi_L(L,G)\phi_H(H,G,J)f_3(G,S)$$
$$=\sum_{L,S,G}\phi_J(J,L,S)\phi_L(L,G)f_3(G,S)\,f_4(G,J)$$
$$=\sum_{L,S}\phi(J,L,S)\,f_5(J,L,S)$$

VE as Graph Transformation

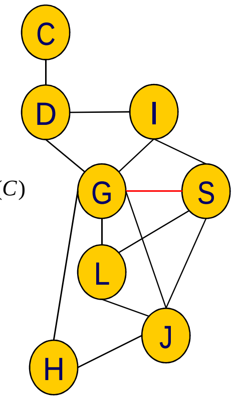
- Goal: P(J)
- Eliminate: C,D,I,H,G,S,L
- Compute: $f_6(J,L)=\sum_S \phi(J,L,S)\,f_5(J,L,S)$



$$P(J)=\sum_{L,S,G,H,I,D,C}\phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)\phi_D(C,D)\phi_C(C)$$
$$=\sum_{L,S,G,H,I,D}\phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)f_1(D)$$
$$=\sum_{L,S,G,H,I}\phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_H(H,G,J)\phi_I(I)f_2(G,I)$$
$$=\sum_{L,S,G,H}\phi_J(J,L,S)\phi_L(L,G)\phi_H(H,G,J)f_3(G,S)$$
$$=\sum_{L,S,G}\phi_J(J,L,S)\phi_L(L,G)f_3(G,S)\,f_4(G,J)$$
$$=\sum_{L,S}\phi(J,L,S)\,f_5(J,L,S)$$
$$=\sum_{L}f_6(J,L)$$

VE as Graph Transformation

- Goal: P(J)
- Eliminate: C,D,I,H,G,S,L
- Compute: $f_7(J) = \sum_L f_6(J,L)$

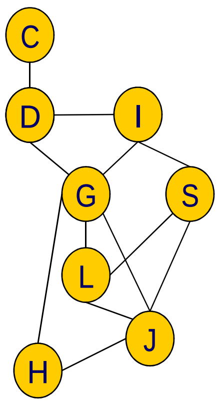


$$
\begin{aligned}
P(J) &= \sum_{L,S,G,H,I,D,C} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)\phi_D(C,D)\phi_C(C) \\
&= \sum_{L,S,G,H,I,D} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_G(G,I,D)\phi_H(H,G,J)\phi_I(I)f_1(D) \\
&= \sum_{L,S,G,H,I} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,I)\phi_H(H,G,J)\phi_I(I)f_2(G,I) \\
&= \sum_{L,S,G,H} \phi_J(J,L,S)\phi_L(L,G)\phi_H(H,G,J)f_3(G,S) \\
&= \sum_{L,S,G} \phi_J(J,L,S)\phi_L(L,G)f_3(G,S)f_4(G,J) \\
&= \sum_{L,S} \phi(J,L,S)f_5(J,L,S) \\
&= \sum_{L} f_6(J,L) \\
&= f_7(J)
\end{aligned}
$$

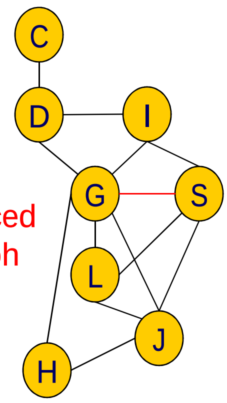
The Induced Graph

- The induced graph $I_{F,\alpha}$ over factors F and ordering α:
  - Union of all of the graphs resulting from the different steps of the variable elimination algorithm.
  - X_i and X_j are connected if they appeared in the same factor throughout the VE algorithm using α as the ordering

Original graph



Induced graph



The Induced Graph

- The induced graph $I_{F,\alpha}$ over factors F and ordering α:
  - Undirected
  - X_i and X_j are connected if they appeared in the same factor throughout the VE algorithm using α as the ordering
- The width of an induced graph width($I_{K,\alpha}$) is the number of nodes in the largest clique in the graph minus 1
  - Minimal induced width of a graph K is $\min_\alpha$width($I_{K,\alpha}$)
  - Minimal induced width provides a lower bound on best performance by applying VE to a model that factorized on K
- How can we compute the minimal induced width of the graph, and the elimination ordering achieving that width?
  - No easy way to answer this question.

47

The Induced Graph

- Finding the optimal ordering is NP-hard
- Hopeless? No, heuristic techniques can find good elimination orderings
- Greedy search using heuristic cost function
  - We eliminate variables one at a time in a greedy way, so that each step tends to lead to a small blowup in size.
  - At each point, find the node with smallest cost
  - Possible costs: number of neighbors in current graph, neighbors of neighbors, number of filling edges

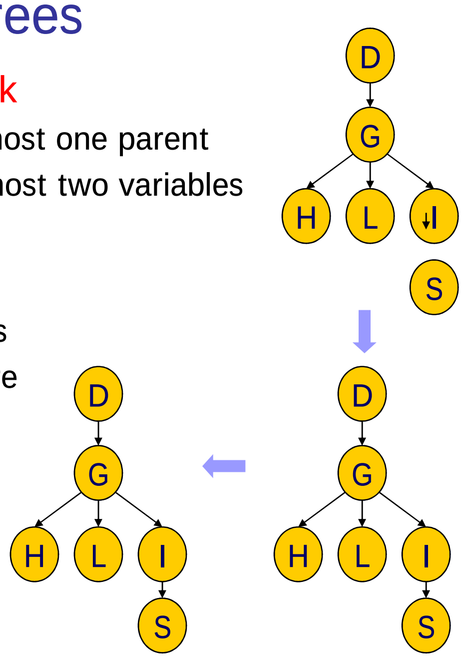
48

Inference should be efficient for certain kinds of graphs …

49

Elimination On Trees

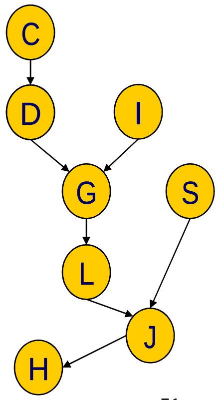
- Tree Bayesian network
  - Each variable has at most one parent
  - All factors involve at most two variables
- Elimination
  - Eliminate leaf variables
  - Maintains tree structure
  - Induced width = 1



50

Elimination on PolyTrees

- PolyTree Bayesian network
  - At most one path between any two variables
- Theorem: inference is linear in the network representation size



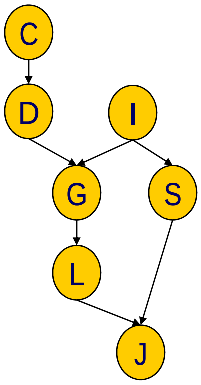
51

For a fixed graph structure, is there any way to reduce the induced width?

52

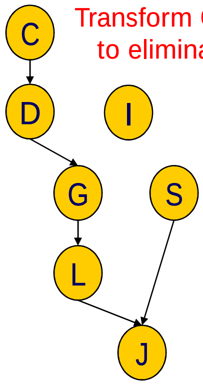
Inference By Conditioning

- General idea
  - Enumerate the possible values of a variable
  - Apply Variable Elimination in a simplified network
  - Aggregate the results



(G ⊥ S | I)

Observe I



Transform CPDs of G and S to eliminate I as parent