Probabilistic Graphical Models

Exact Inference

Complexity of VE

Belief Propagation

Summary

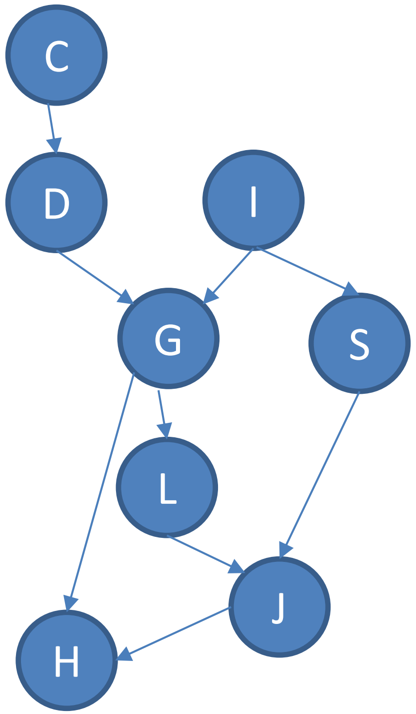
- Simple algorithm
- Works for both BNs and MNs
- Factor product and summation steps can be done in any order, subject to:
  - when Z is eliminated, all factors involving Z have been multiplied in

Daphne Koller

Variable Elimination

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

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



Complexity

$$\psi_k(\boldsymbol{X}_k) = \prod_{i=1}^{m_k} \phi_i$$

Factor product of m_k factors

$$\tau_k(\boldsymbol{X}_k \setminus \{Z\}) = \sum_Z \psi_k(\boldsymbol{X}_k)$$

Marginalization of a variable

Factor Product

$N_k = |Val(\boldsymbol{X_k})|$

$N_k\ rows$

a^1	b^1	0.5
a^1	b^2	0.8
a^2	b^1	0.1
a^2	b^2	0
a^3	b^1	0.3
a^3	b^2	0.9

b^1	c^1	0.5
b^1	c^2	0.7
b^2	c^1	0.1
b^2	c^2	0.2

a^1	b^1	c^1	$0.5 \cdot 0.5 = 0.25$
a^1	b^1	c^2	$0.5 \cdot 0.7 = 0.35$
a^1	b^2	c^1	$0.8 \cdot 0.1 = 0.08$
a^1	b^2	c^2	$0.8 \cdot 0.2 = 0.16$
a^2	b^1	c^1	$0.1 \cdot 0.5 = 0.05$
a^2	b^1	c^2	$0.1 \cdot 0.7 = 0.07$
a^2	b^2	c^1	$0 \cdot 0.1 = 0$
a^2	b^2	c^2	$0 \cdot 0.2 = 0$
a^3	b^1	c^1	$0.3 \cdot 0.5 = 0.15$
a^3	b^1	c^2	$0.3 \cdot 0.7 = 0.21$
a^3	b^2	c^1	$0.9 \cdot 0.1 = 0.09$
a^3	b^2	c^2	$0.9 \cdot 0.2 = 0.18$

$$\psi_k(\boldsymbol{X}_k) = \prod_{i=1}^{m_k} \phi_i$$

Cost: $(m_k - 1) \times N_k$

Factor Marginalization

a^1	b^1	c^1	$0.5 \cdot 0.5 = 0.25$
a^1	b^1	c^2	$0.5 \cdot 0.7 = 0.35$
a^1	b^2	c^1	$0.8 \cdot 0.1 = 0.08$
a^1	b^2	c^2	$0.8 \cdot 0.2 = 0.16$
a^2	b^1	c^1	$0.1 \cdot 0.5 = 0.05$
a^2	b^1	c^2	$0.1 \cdot 0.7 = 0.07$
a^2	b^2	c^1	$0 \cdot 0.1 = 0$
a^2	b^2	c^2	$0 \cdot 0.2 = 0$
a^3	b^1	c^1	$0.3 \cdot 0.5 = 0.15$
a^3	b^1	c^2	$0.3 \cdot 0.7 = 0.21$
a^3	b^2	c^1	$0.9 \cdot 0.1 = 0.09$
a^3	b^2	c^2	$0.9 \cdot 0.2 = 0.18$

$$\phi(a^1, c^1) = \sum_b \phi(a^1, c^1, b)$$

a^1	c^1	033
a^1	c^2	0.51
a^2	c^1	0.05
a^2	c^2	0.07
a^3	c^1	0.24
a^3	c^2	0.39

Cost: $\sim N_k$ additions

Complexity

- Complexity of Variable Elimination
- Start with m factors
  - $m \leq n$ for Bayesian networks (one for every variable)
  - can be larger for Markov networks
- At each elimination step generate 1 factor,
- At most n elimination steps
- Total number of factors: $m^{\star} \leqslant m + n$

Complexity

- $N = \max(N_k) =$ size of the largest factor
- Number of product operations:
  - $\sum_k (m_k - 1) * N_k \leq \sum_k (m_k - 1) * N =$

$$= N * \sum_k (m_k - 1) \leq N * (m + n)$$

- Number of sum operations: $n * \sum_k N_k \leq n * N$

- Linear in N and m^*

Complexity

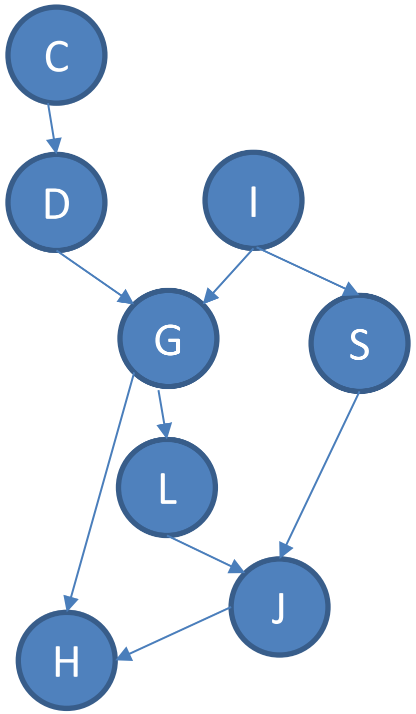
- $N = \max(N_k)$ = size of the largest factor
- Linear in N and m^*
- But!
  - $N_k = |Val(\boldsymbol{X_k})| = O(d^{r_k})$
  - $d = \max(|\mathrm{Val}(\boldsymbol{X_i})|)$ d values in their scope
  - $r_k = |\mathbf{X}_k|$ = cardinality of the scope of the k-th factor

Elimination ordering?

Step	Variable eliminated	Factors used	Variables involved	New factor
1	C	$\phi_C(C), \phi_D(D, C)$	D, C	$\tau_1(D)$
2	D	$\tau_1(D), \phi_G(G, I, D)$	G, I, D	$\tau_2(G, I)$
3	I	$\tau_2(G, I), \phi_I(I), \phi_S(S, I)$	G, I, S	$\tau_3(G, S)$
4	H	$\phi_H(H, G, J)$	H, G, J	$\tau_4(G, J)$
5	G	$\tau_3(G, S), \tau_4(G, J) \phi_L(L, G)$	G, J, L, S	$\tau_5\ (J, L, S)$
6	S	$\tau_5\ (J, L, S)\ \phi_J(J, L, S)$	J, L, S	$\tau_6\ (J, L)$
7	L	$\tau_6\ (J, L)$	J, L	$\tau_7\ (J)$

Different elimination ordering?

Step	Variable eliminated
1	G
2	I
3	S
4	L
5	H
6	C
7	D



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

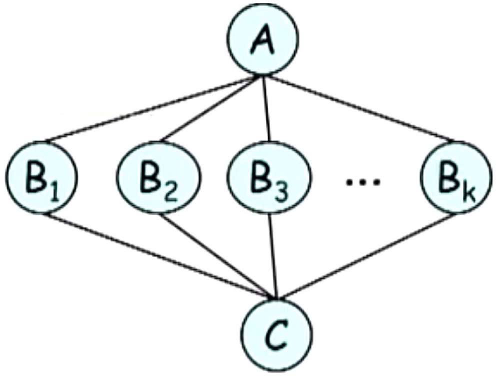
Different elimination ordering?

Step	Variable eliminated	Factors used	Variables involved	New factor
1	G	$\phi_G(G,I,D), \phi_L(L,G), \phi_H(H,G,J)$	G,I,D,L,J,H	$\tau_1(I,D,L,J,H)$
2	I	$\phi_I(I), \phi_S(S,I), \tau_1(I,D,L,S,J,H)$	S,I,D,L,J,H	$\tau_2(D,L,S,J,H)$
3	S	$\phi_J(J,L,S), \tau_2(D,L,S,J,H)$	D,L,S,J,H	$\tau_3(D,L,J,H)$
4	L	$\tau_3(D,L,J,H)$	D,L,J,H	$\tau_4(D,J,H)$
5	H	$\tau_4(D,J,H)$	D,J,H	$\tau_5(D,J)$
6	C	$\phi_C(C), \phi_D(D,C)$	D,J,C	$\tau_6(D)$
7	D	$\tau_5(D,J), \tau_6(D)$	D,J	$\tau_7(J)$

Complexity and Elimination Ordering

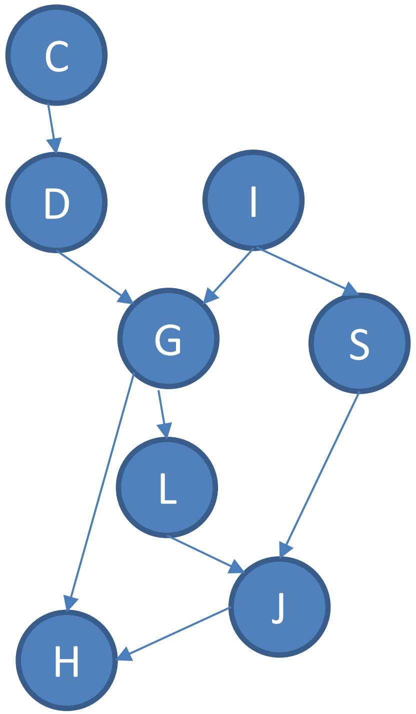
Eliminate A first:

Eliminate B_i's first:



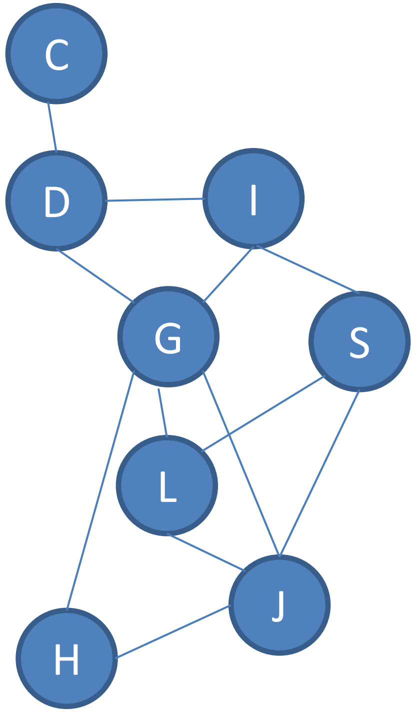
Graphical Perspective

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



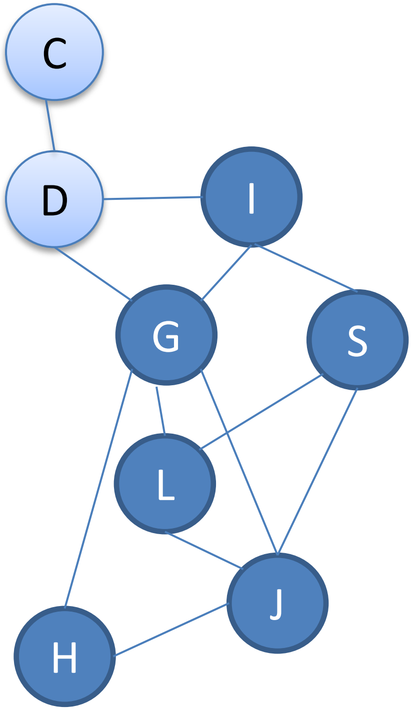
Step 0: Moralize the Graph

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



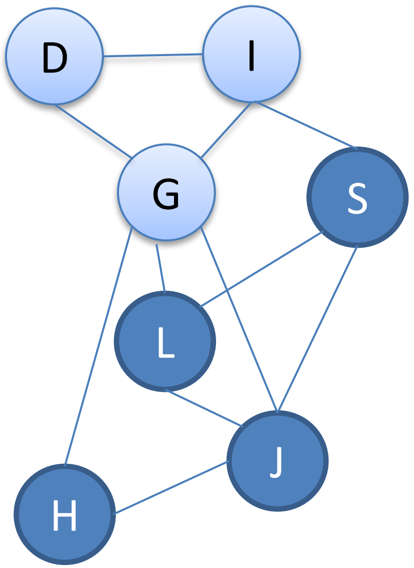
Step 1: Eliminate C

Step	Variable eliminated	Variables involved	New factor
1	C	D, C	$\tau_1(D)$
2	D	G, I, D	$\tau_2(G, I)$
3	I	G, I, S	$\tau_3(G, S)$
4	H	H, G, J	$\tau_4(G, J)$
5	G	G, J, L, S	$\tau_5\ (J, L, S)$
6	S	J, L, S	$\tau_6\ (J, L)$
7	L	J, L	$\tau_7\ (J)$



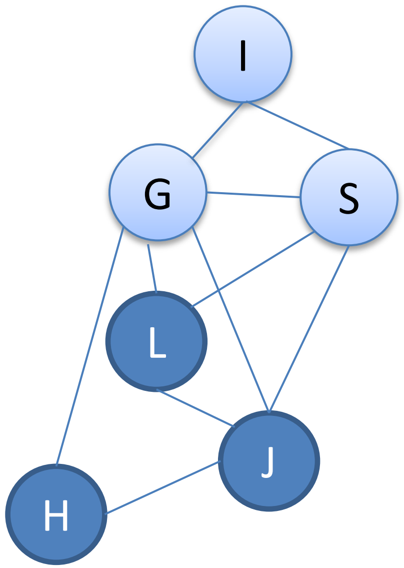
Step 2: Eliminate D

Step	Variable eliminated	Variables involved	New factor
1	C	D, C	$\tau_1(D)$
2	D	G, I, D	$\tau_2(G, I)$
3	I	G, I, S	$\tau_3(G, S)$
4	H	H, G, J	$\tau_4(G, J)$
5	G	G, J, L, S	$\tau_5\ (J, L, S)$
6	S	J, L, S	$\tau_6\ (J, L)$
7	L	J, L	$\tau_7\ (J)$



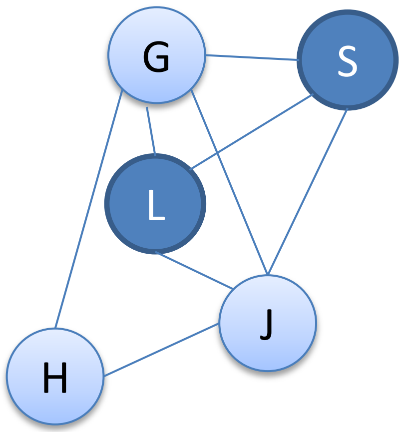
Step 3: Eliminate I

Step	Variable eliminated	Variables involved	New factor
1	C	D, C	$\tau_1(D)$
2	D	G, I, D	$\tau_2(G, I)$
3	I	G, I, S	$\tau_3(G, S)$
4	H	H, G, J	$\tau_4(G, J)$
5	G	G, J, L, S	$\tau_5\ (J, L, S)$
6	S	J, L, S	$\tau_6\ (J, L)$
7	L	J, L	$\tau_7\ (J)$



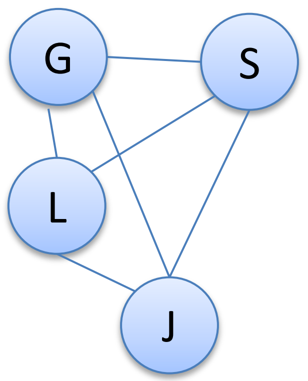
Step 4: Eliminate H

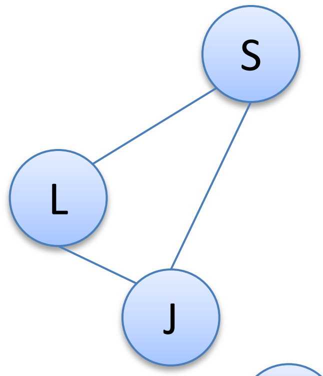
Step	Variable eliminated	Variables involved	New factor
1	C	D, C	$\tau_1(D)$
2	D	G, I, D	$\tau_2(G, I)$
3	I	G, I, S	$\tau_3(G, S)$
4	H	H, G, J	$\tau_4(G, J)$
5	G	G, J, L, S	$\tau_5\ (J, L, S)$
6	S	J, L, S	$\tau_6\ (J, L)$
7	L	J, L	$\tau_7\ (J)$

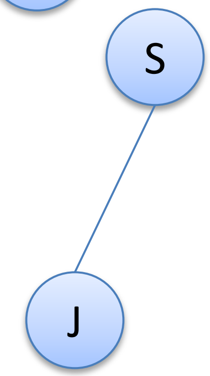


Steps 5,6,7: Eliminate G

Step	Variable eliminated	Variables involved	New factor
1	C	D, C	$\tau_1(D)$
2	D	G, I, D	$\tau_2(G, I)$
3	I	G, I, S	$\tau_3(G, S)$
4	H	H, G, J	$\tau_4(G, J)$
5	G	G, J, L, S	$\tau_5\ (J, L, S)$
6	S	J, L, S	$\tau_6\ (J, L)$
7	L	J, L	$\tau_7\ (J)$







Induced Graph of an elimination ordering

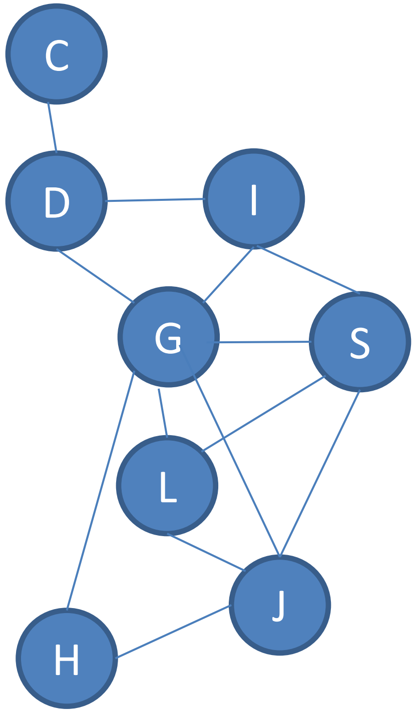
- The induced graph $I_{\Phi,a}$ over factors Φ and ordering a:
- Undirected graph
- X_i and X_j are connected if they appeared in the same factor in a run of the VE algorithm using a as the ordering

Theorem: Every factor produced during VE is a clique in the induced graph

$$\tau_1(D) = \sum_C \phi_C(C)\phi_D(C,D)$$
$$\tau_2(G,I) = \sum_D \underline{\phi_G(G,I,D)\tau_1(D)}$$
$$\tau_3(S,G) = \sum_I \phi_S(S,I)\phi_I(I)\tau_2(G,I)$$
$$\tau_4(G,J) = \sum_H \underline{\phi_H(H,G,J)}$$
$$\tau_5(L,J) = \sum_G \underline{\phi_L(L,G)\tau_3(S,G)\tau_4(G,J)}$$
$$\tau_6 = \sum_{L,S} \phi_J(J,L,S)\tau_5(L,J)$$

Induced Graph of an elimination ordering

- The induced graph $I_{\Phi,a}$ over factors Φ and ordering a:
- Undirected graph
- X_i and X_j are connected if they appeared in the same factor in a run of the VE algorithm using a as the ordering



Theorem: Every maximal clique in the graph is a factor produced during VE

Consider a maximal clique some variable is first to be eliminated

once a variable is eliminated:

no new neighbor ⇒ when eliminated it already had all the clique members as neighbors

⇒ participated in factors with all the other variables

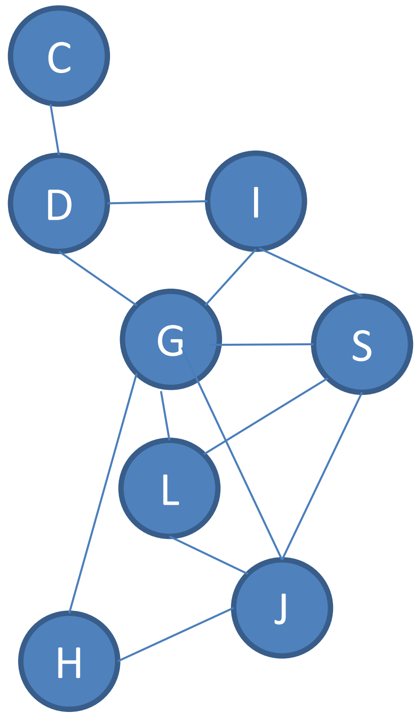
⇒ when multiplied together, we have a factor oven all of them

Complexity based on graphs

- The width of an induced graph is the number of nodes in the largest clique in the graph minus 1
- Minimal induced width of a graph K is

$$\min_a \left(width(I_{K,a}) \right)$$

- Provides a lower bound on best performance of VE to a model factorizing over K



Finding a good elimination ordering

Theorem: For a graph H, determining whether there exists an elimination ordering for H with induced width K is NP-complete

Note: This NP-hardness result is distinct from the NP-hardness result of inference

- Even given the optimal ordering, inference may still be exponential

Finding a good elimination ordering

- Greedy search using heuristic cost function - At each point, eliminate node with smallest cost
- Possible cost functions:
  - min-neighbors: # neighbors in current graph
  - min-weight: weight (# values) of factor formed
  - min-fill: number of new fill edges
  - weighted min-fill: total weight of new fill edges (edge weight = product of weights of the 2 nodes)

Finding a good elimination ordering

Finding Elimination Orderings

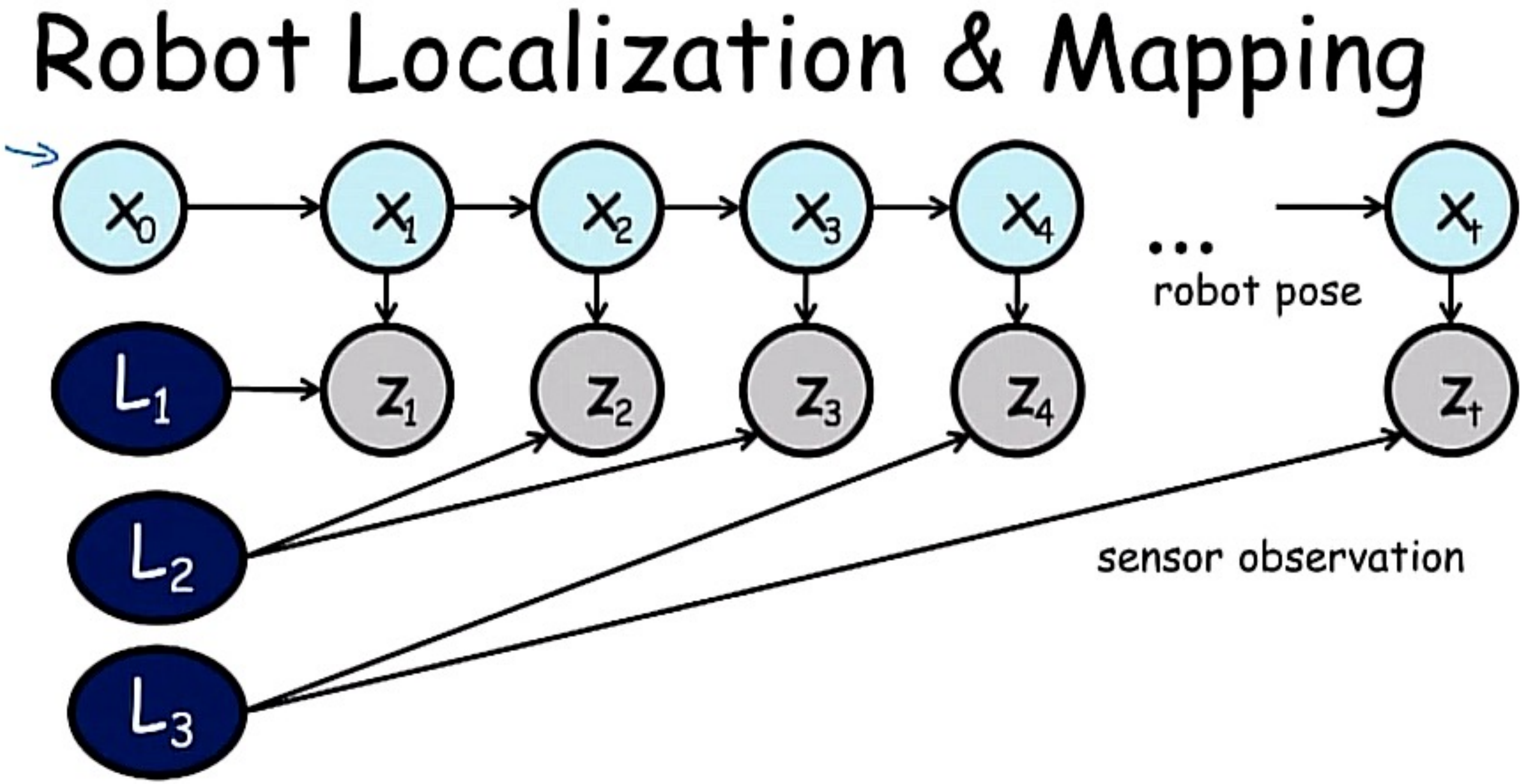
- Theorem: The induced graph is triangulated

  - No loops of length > 3 without a "bridge”

  -all cycles of four or more vertices have a chord, which is an edge that is not part of the cycle but connects two vertices of the cycle.

- Can find elimination ordering by finding a low-width triangulation of original graph H_Φ

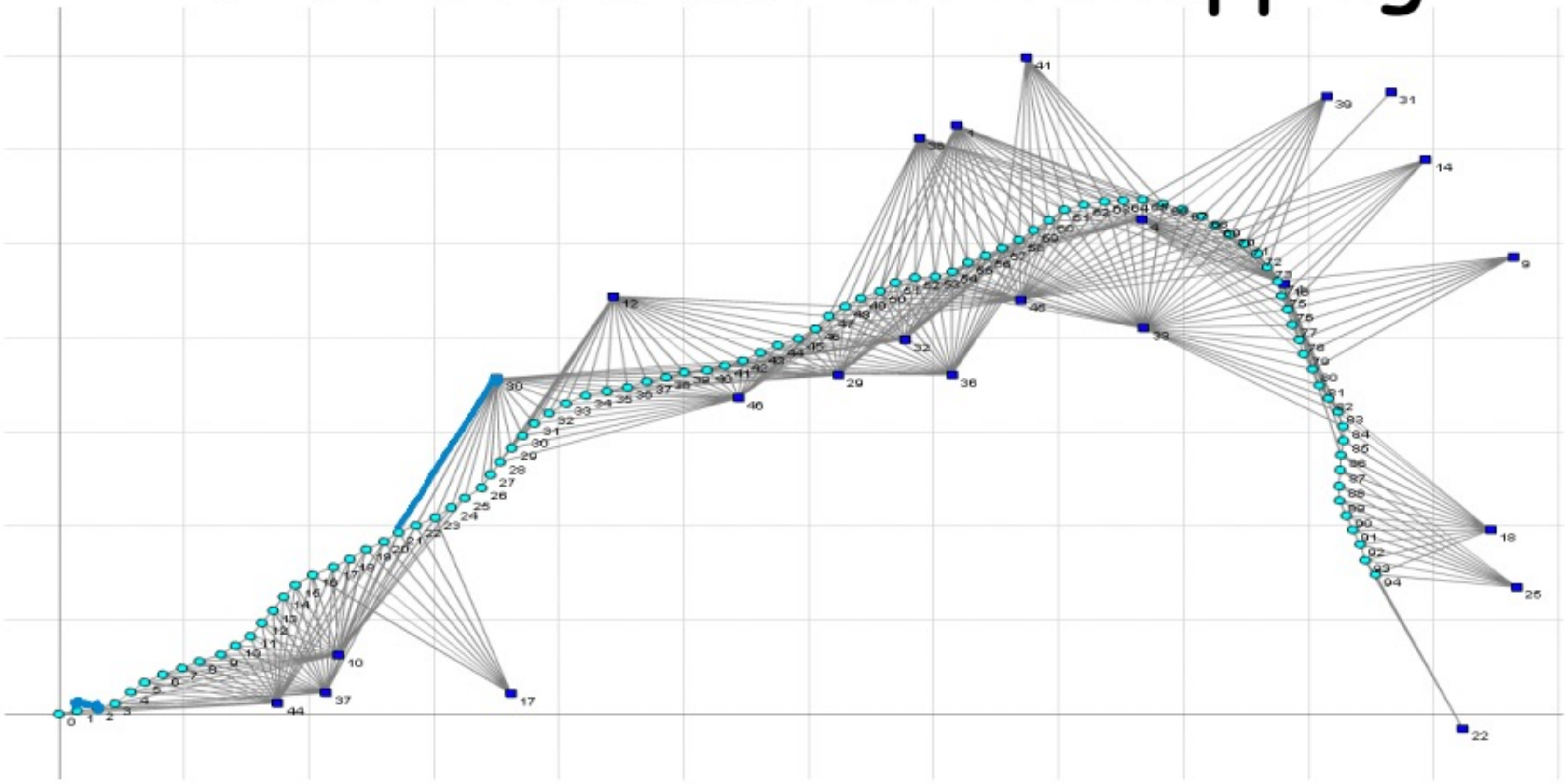
Example: Robot localization



Example: Robot localization

Square Root SAM, F. Dellaert and M. Kaess, IJRR, 2006

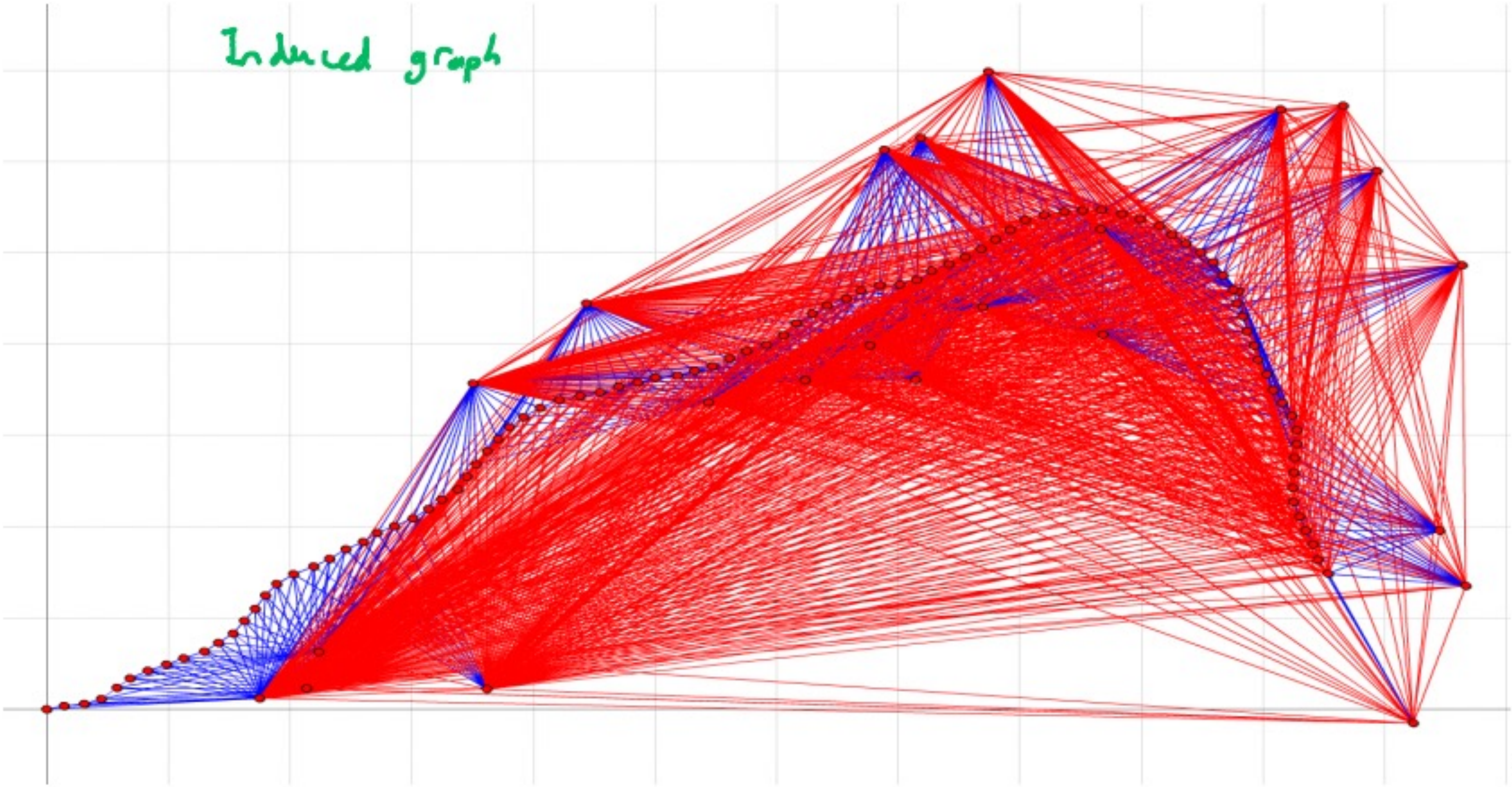
Robot Localization & Mapping



Example: Robot localization

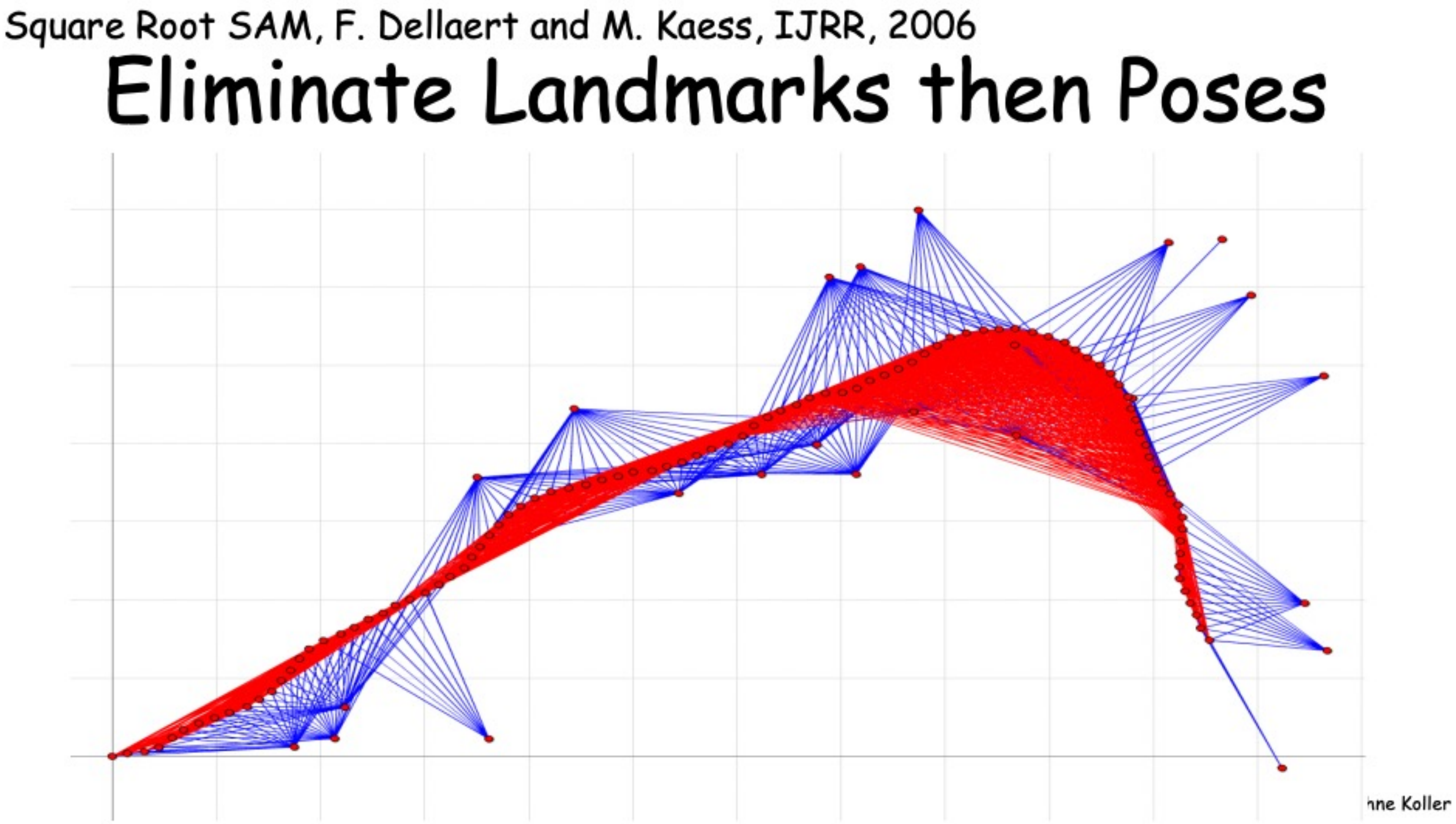
Square Root SAM, F. Dellaert and M. Kaess, IJRR, 2006

Eliminate Poses then Landmarks

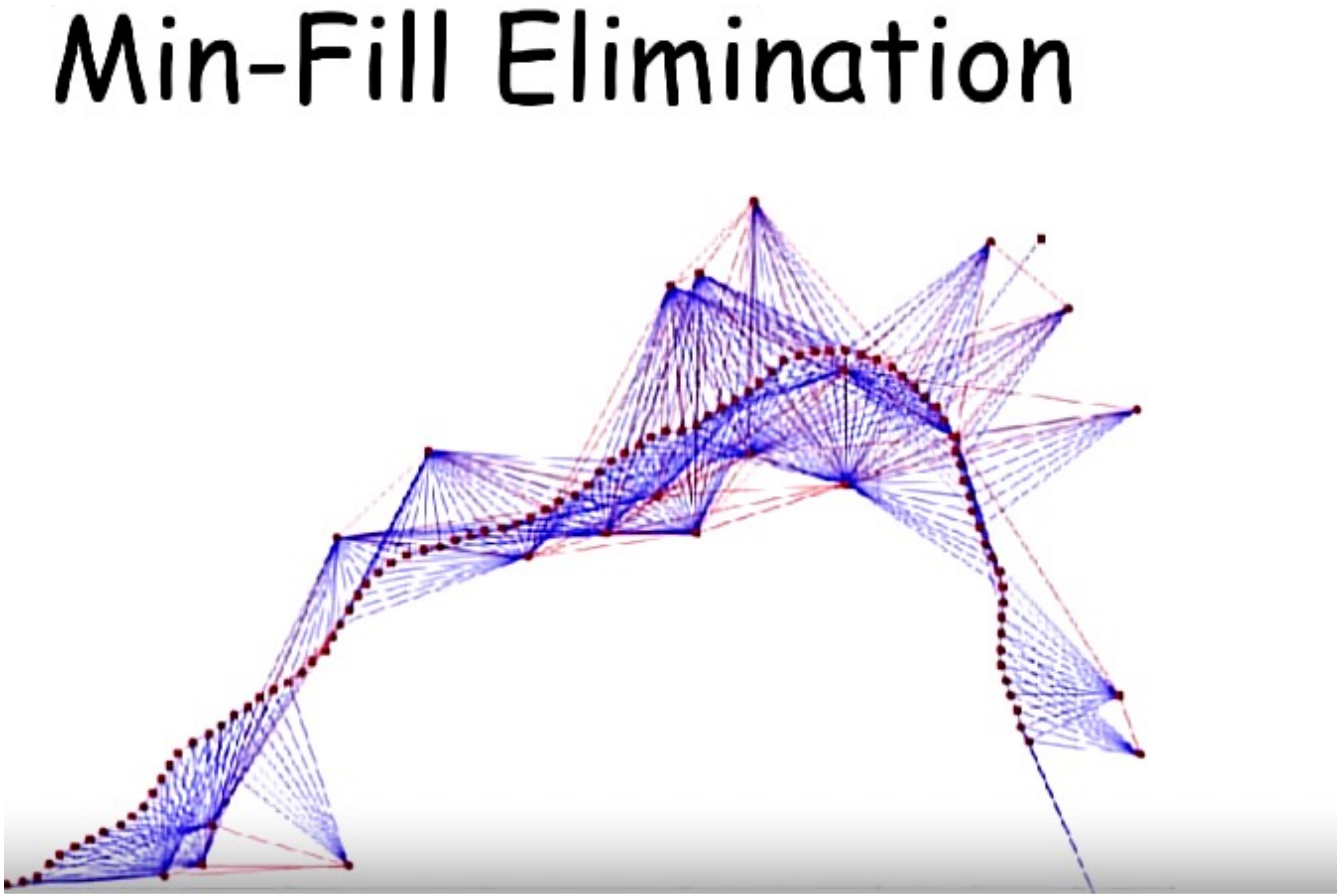


ne Koller

Example: Robot localization



Example: Robot localization



Summary

Variable elimination allows computation of marginals / conditionals

Algorithm is valid for any graphical model

Suffices to show variable elimination for MRFs, since Bayes nets → MRFs by moralization

Worst-case complexity is dependent on elimination order, and is exponential in number of variables

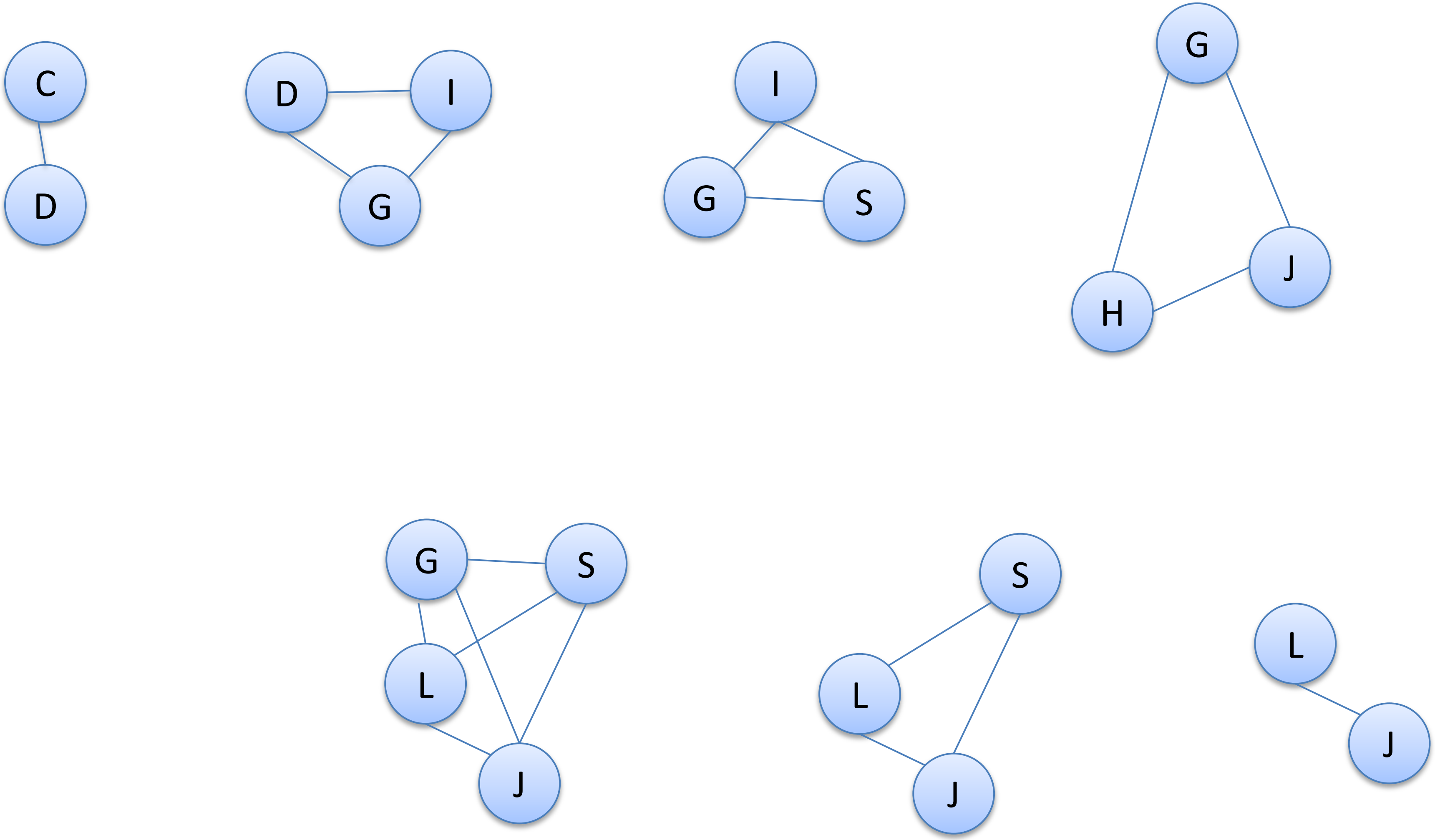
Optimal ordering = treewidth, is NP-hard to compute

Pt. 2: Message-passing inference

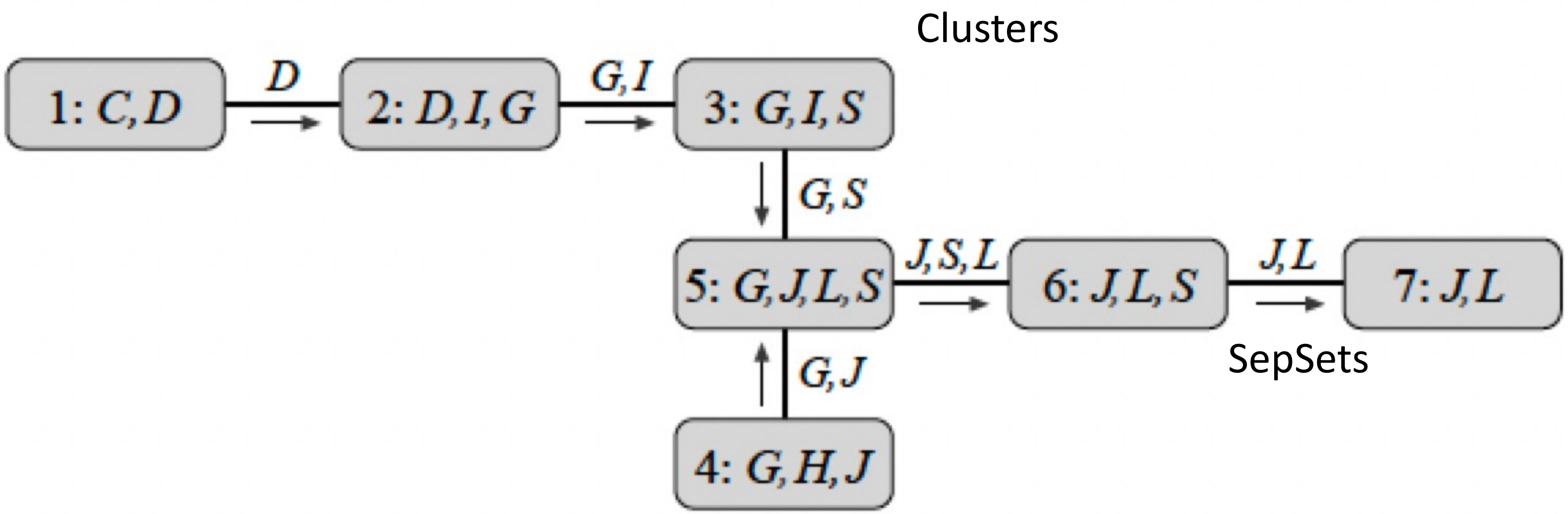
Variable Elimination

Step	Variable eliminated	Factors used	Variables involved	New factor
1	C	$\phi_C(C), \phi_D(D, C)$	D, C	$\tau_1(D)$
2	D	$\tau_1(D), \phi_G(G, I, D)$	D, I, G	$\tau_2(G, I)$
3	I	$\tau_2(G, I), \phi_I(I), \phi_S(S, I)$	G, I, S	$\tau_3(G, S)$
4	H	$\phi_H(H, G, J)$	G, H, S	$\tau_4(G, J)$
5	G	$\tau_3(G, S), \tau_4(G, J)\phi_L(L, G)$	G, J, L, S	$\tau_5\ (J, L, S)$
6	S	$\tau_5\ (J, L, S)\ \phi_J(J, L, S)$	J, L, S	$\tau_6\ (J, L)$
7	L	$\tau_6\ (J, S)$	J, L	$\tau_7\ (J)$

Understanding Variable Elimination

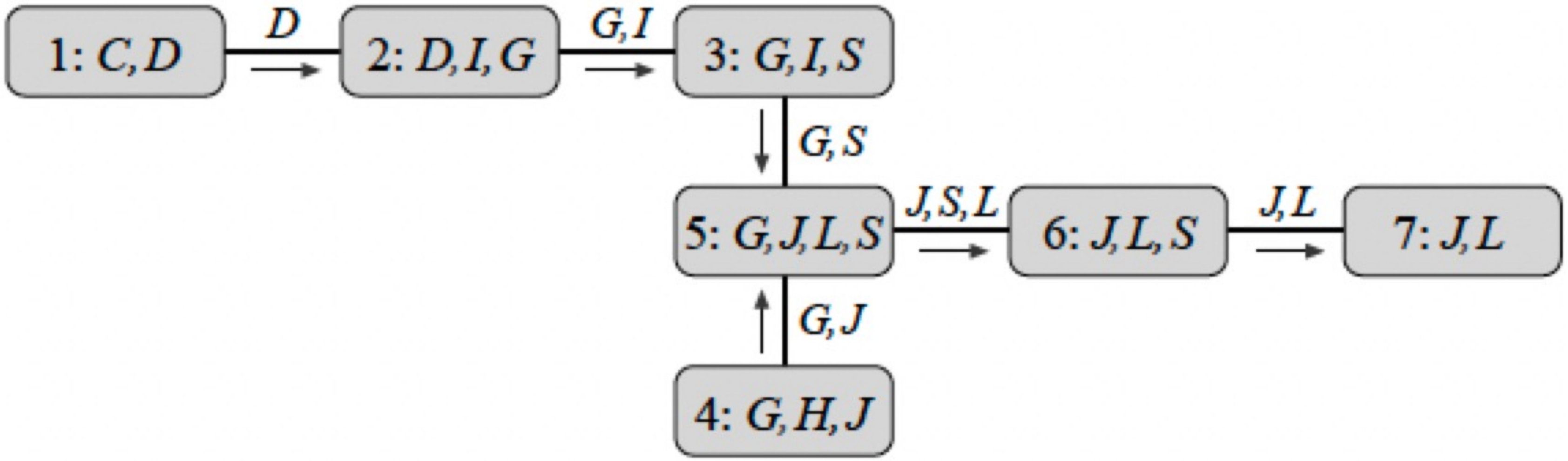


Cluster Trees



Variable Elimination as message passing

Message passing



$$\delta_{12}(D) = \sum_{D} \phi(C)\phi(C,D)$$

$$\delta_{35}(G,S) = \sum_{I} \phi(G,I,S)\delta_{35}(G,I)$$

$$\delta_{23}(G,I) = \sum_{D} \phi(D,I,G)\delta_{12}(D)$$

Message passing

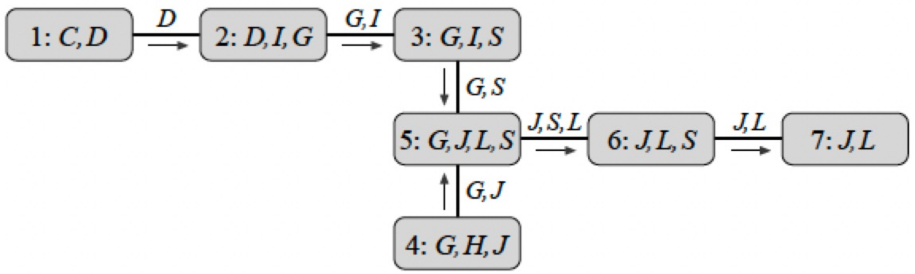
$$\delta_{12}(D) = \sum_{D} \phi(C)\phi(C,D)$$

$$\delta_{23}(G,I) = \sum_{D} \phi(D,I,G)\delta_{12}(D)$$

$$\delta_{35}(G,S) = \sum_{I} \phi(G,I,S)\delta_{35}(G,I)$$

$$\delta_{45}(G,J) = \sum_{H} \phi(G,H,J)$$

$$\delta_{56}(J,S,L) = \sum_{G} \delta_{45}(G,J)\delta_{35}(G,S)\phi(L,G)$$



$$\delta_{67}(J,L) = \sum_{S} \delta_{56}(J,S,L)$$

Clique-Tree Message Passing

1. Pick a node to be your root.
2. For each node i, initialize the potential of the node

$$\psi_i = \prod_i \phi_i$$

3. Start from a leaf and send message to all neighbors

$$\delta_{i \to j} = \sum_{\boldsymbol{C}_i - \boldsymbol{S}_{i,j}} \psi_i \cdot \prod_{k \in (\mathrm{Nb}_i - \{j\})} \delta_{k \to i}$$

4. Repeat for every node that is ready to transmit a message (i.e., has received messages from every neighbor)

Properties of Cluster Trees

Family Preservation:

For each factor $\phi_k \in \Phi$, there exists a cluster C_i s.t. Scope $[\phi_k] \subseteq \boldsymbol{C}_i$

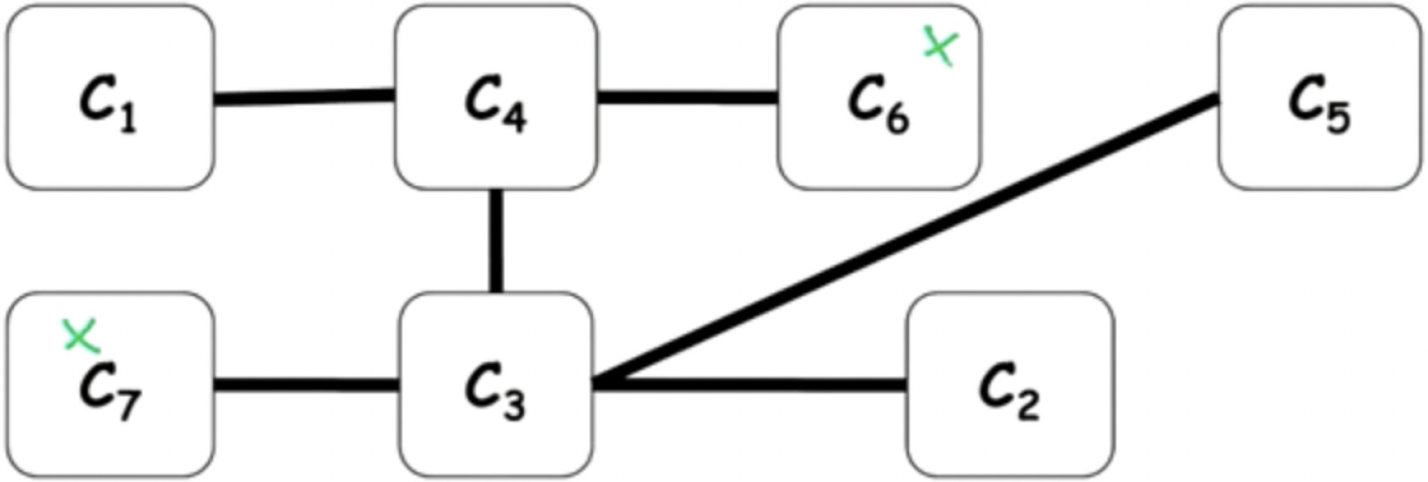
every factor has a node that can accommodate it

Running Intersection:

For each pair of clusters $C_\mathrm{i}, C_\mathrm{j}$ and variable $X \in C_i \cap C_j$, in the unique path between C_i and C_j, all clusters and sepsets contain X.

clusters that include the same variable need to communicate for consistency

Running Intersection



Which clusters need to include X?