



Probabilistic Graphical Models & Probabilistic AI

Ben Lengerich

Lecture 6: Exact Inference

February 6, 2025

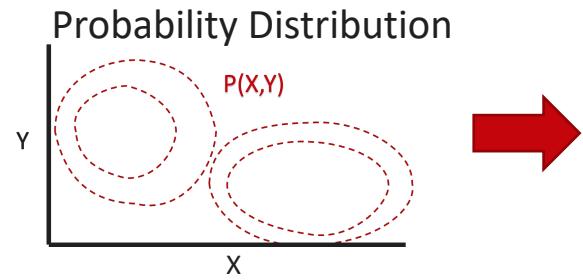
Reading: See course homepage



Logistics Reminders

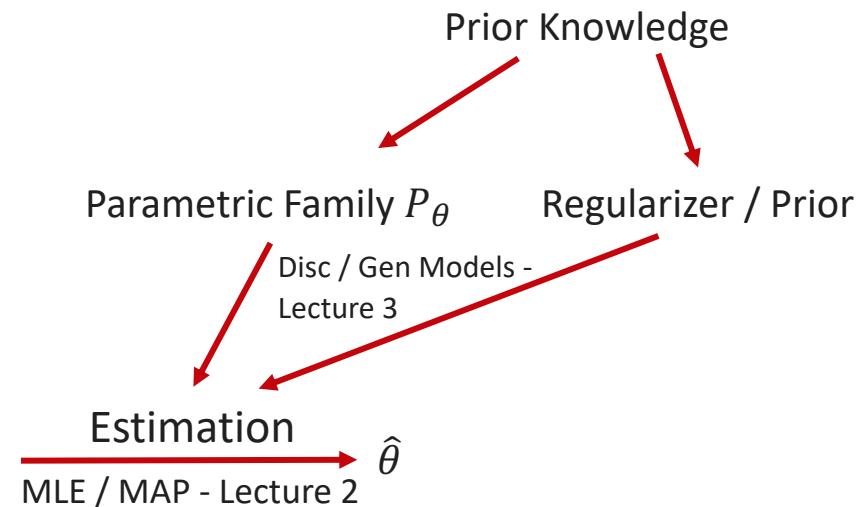
- No class Tuesday, Feb 11
- HW2 due Tuesday, Feb 11 on Canvas
- Quiz in-class Thursday, Feb 13

A Brief Recap of our Roadmap

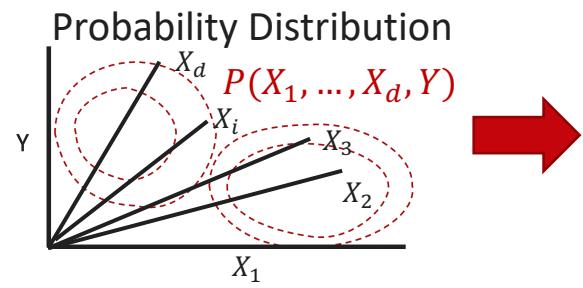


Observations

X	Y
0	0
0.1	1
0.2	1

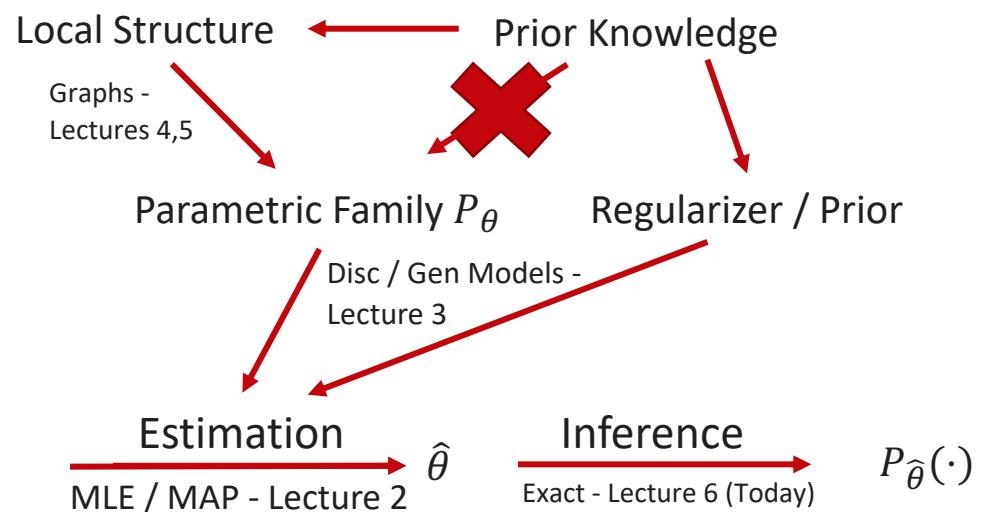


A Brief Recap of our Roadmap

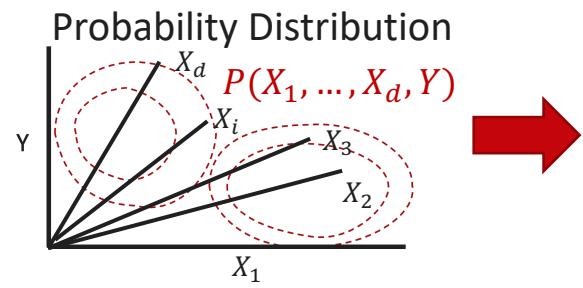


Observations

X	Y
0	0
0.1	1
0.2	1

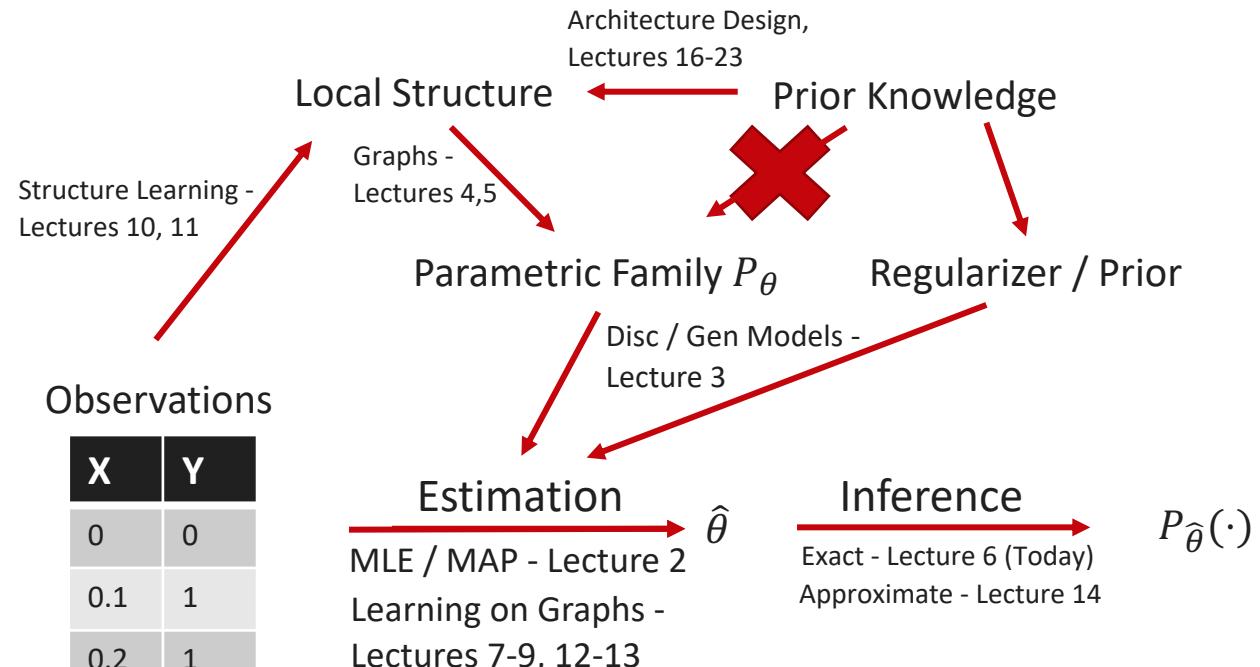


A Brief Recap of our Roadmap



Observations

X	Y
0	0
0.1	1
0.2	1





Today

- Exact Inference
 - Variable Elimination



Exact Inference



Probabilistic Inference and Learning

- We now have compact representations of probability distributions: Graphical Models (GMs)
- A GM M describes a probability distribution P_M .
- Typical tasks:
 - Task 1 (**Inference**): How do we answer queries about P_M - e.g. $P_M(X \mid Y)$?
 - Task 2 (**Learning**): How do we estimate a plausible model M from data D ?

When could “learning” be seen as a form of inference?

Bayesian Perspective:

Seeks $P_{prior}(M|D)$

Missing Data: Must impute missing data

$$P(M|D) = \int_{D_{missing}} P(D \mid D_{present}) P(M|D)$$



Example Query 1: Likelihood

- Many queries involve **evidence**
 - Evidence e is an assignment of values to a set E of variables
- Example: compute the probability of evidence e

$$P(e) = \sum_{X_1} \cdots \sum_{X_k} P(X_1, \dots, X_k, e)$$

- aka compute the likelihood of e



Example Query 2: Conditional Probability

- Often we are interested in the **conditional probability distribution** of a variable given the evidence

$$P(X | e) = \frac{P(X, e)}{P(e)} = \frac{P(X, e)}{\sum_x P(X = x, e)}$$

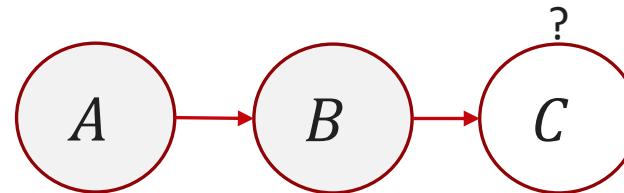
- aka compute the ***a posteriori belief*** in X given evidence e
- We usually query a subset Y of all domain variables $X = \{Y, Z\}$ and don't care about the remaining Z :

$$P(Y | e) = \sum_z P(Y, Z = z | e)$$

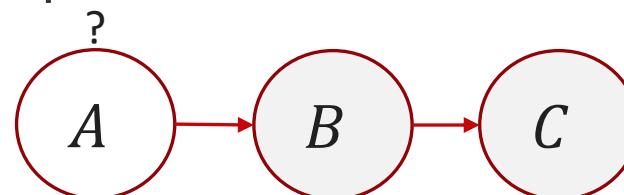
- aka ***marginalization***.

Examples of a Posteriori Belief

- **Prediction:** What's the probability of an outcome given the starting condition?



- Query node is a descendent of the evidence
- **Diagnosis:** What's the probability of an underlying disease/fault given observed symptoms?



- Query node is an ancestor of the evidence

Probabilistic inference combines evidence from all parts of the network, not just following the directionality of the edges in a GM.



Example Query 3: Most Probable Assignment

- What's the most probable assignment (MPA) for some variables of interest?
- Usually performed under some evidence e and marginalized over other variables Z :

$$\begin{aligned} MPA(Y \mid e) &= \operatorname{argmax}_y P(Y = y \mid e) \\ &= \operatorname{argmax}_y \sum_z P(Y = y, Z = z \mid e) \end{aligned}$$

- Examples:
 - Classification: $\hat{Y} = MPA(Y \mid e)$
 - Explanation: What is the most likely scenario given the evidence?



A cautionary note on MPA

- The MPA of a variable depends on the query “context” – the set of variables being jointly queried.
- Example:
 - MPA of Y_1 ?
 - MPA of (Y_1, Y_2) ?

Y_1	Y_2	$P(Y_2 Y_1)$
0	0	0.35
0	1	0.05
1	0	0.3
1	1	0.3



Complexity of inference

- Computing $P(X = x \mid e)$ in a GM is **NP-hard**

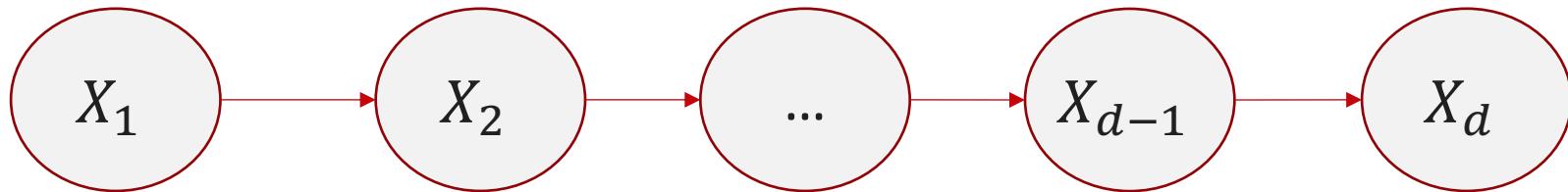
What does this mean for us?

- Inference cannot be solved in polynomial time unless P=NP.
- No general procedure that works efficiently for arbitrary GMs.
 - For families of GMs, we can have provably efficient procedures.
- Exponential worst-case performance for exact inference.
 - Motivates approximate inference.



Elimination on Chains

- Consider the following GM:



- What is the likelihood that X_d is true?

$$P(e) = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_d} P(X_1 = x_1, X_2 = x_2, \dots, X_d = x_d)$$

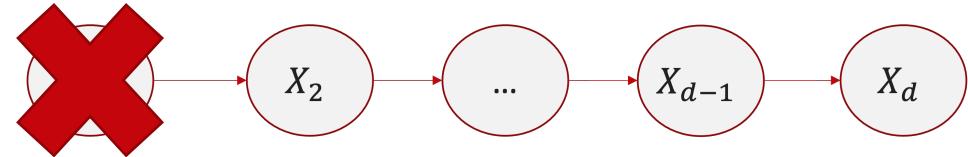
Exponential # of terms

- Leverage chain structure:

$$P(e) = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_d} P(X_1 = x_1) \prod_{i=2}^d P(X_i = x_i \mid X_{i-1} = x_{i-1})$$



Elimination on Chains



- Leverage chain structure:

$$P(e) = \sum_{x_1} \sum_{x_2} \cdots \sum_{x_d} P(X_1 = x_1) \prod_{i=2}^d P(X_i = x_i | X_{i-1} = x_{i-1})$$

- Reorder terms:

$$P(e) = \sum_{x_2} \cdots \sum_{x_d} \prod_{i=3}^d P(X_i | X_{i-1}) \sum_{x_1} P(X_1) P(X_2 | X_1)$$

- Substitute:

$$P(e) = \sum_{x_2} \cdots \sum_{x_d} \prod_{i=3}^d P(X_i | X_{i-1}) P(X_2)$$

Eliminates one variable from our summation at a local cost.

Elimination on Chains



- Continue eliminating variables:

$$P(e) = \sum_{x_3} \cdots \sum_{x_d} \prod_{i=4}^d P(X_i | X_{i-1}) \textcolor{red}{P(X_3)}$$

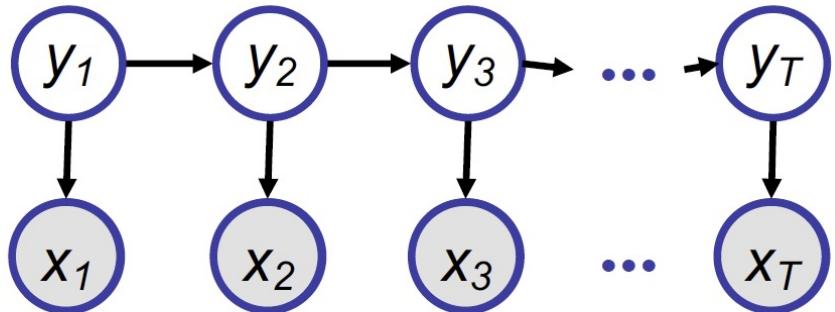
- Eliminate nodes one-by-one all the way to the end

$$P(e) = \sum_{x_{d-1}} P(X_d | X_{d-1}) \textcolor{red}{P(X_{d-1})}$$

- Complexity of this calculation:

- d steps, Each step takes $\approx |\text{Dom}(X_i)| * |\text{Dom}(X_{i-1})|$ operations
- $\rightarrow \mathcal{O}(dn^2)$ where $n = \max_i |\text{Dom}(X_i)|$
- Compare to naïve $\mathcal{O}(dn^2)$

Example: HMMs

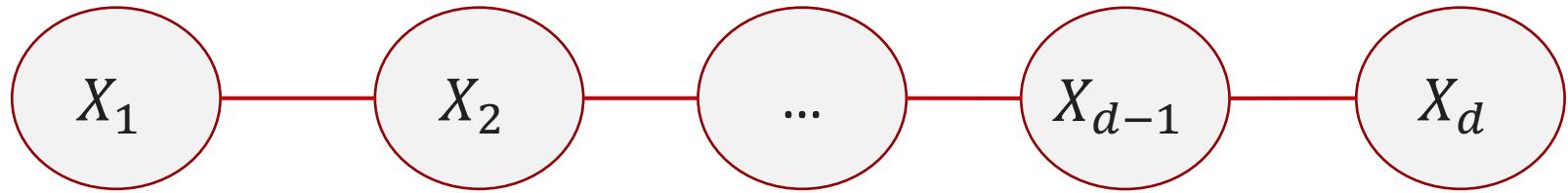


$$\begin{aligned}
 p(\mathbf{x}, \mathbf{y}) &= p(x_1, \dots, x_T, y_1, \dots, y_T) \\
 &= p(y_1) p(x_1 | y_1) p(y_2 | y_1) p(x_2 | y_2) \dots p(y_T | y_{T-1}) p(x_T | y_T)
 \end{aligned}$$

$$\begin{aligned}
 p(y_i | x_1, \dots, x_T) &= \sum_{y_1} \dots \sum_{y_{i-1}} \sum_{y_{i+1}} \dots \sum_{y_T} p(y_i, \dots, y_T, x_1, \dots, x_T) \\
 &= \sum_{y_1} \dots \sum_{y_{i-1}} \sum_{y_{i+1}} \dots \sum_{y_T} p(y_1) p(x_1 | y_1) \dots p(y_T | y_{T-1}) p(x_T | y_T)
 \end{aligned}$$



Elimination on Undirected Chains



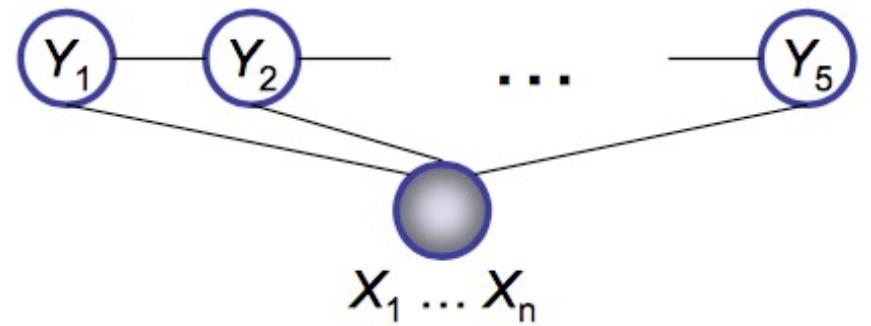
$$P(e) = \sum_{X_1} \sum_{X_2} \cdots \sum_{X_d} \frac{1}{Z} \prod_{i=2}^d \phi(X_i, X_{i-1})$$

$$= \frac{1}{Z} \sum_{X_d} \cdots \sum_{X_2} \prod_{i=3}^d \phi(X_i, X_{i-1}) \sum_{X_1} \phi(X_2, X_1)$$

$= \dots$

Example: Conditional Random Fields

$$\begin{aligned} & p(y_1, \dots, y_T | \mathbf{x}) \\ \propto & \exp \left(w_{1,2} \phi(y_1, y_2 | \mathbf{x}) + w_{2,3} \phi(y_2, y_3 | \mathbf{x}) + \dots + w_{T-1,T} \phi(y_{T-1}, y_T | \mathbf{x}) \right. \\ & \quad \left. + u_1 \phi(y_1 | \mathbf{x}) + u_2 \phi(y_2 | \mathbf{x}) + \dots + u_T \phi(y_T | \mathbf{x}) \right) \\ = & \Phi(y_1, y_2) \Phi(y_2, y_3) \dots \Phi(y_{T-1}, y_T) \end{aligned}$$





The Sum-Product Operation

- In general, we want to compute the value of an expression of the form:

$$\sum_z \prod_{\phi \in F} \phi$$

where F is a set of factors

- We call this task the **sum-product inference task**.



Variable Elimination: General form

- Write query in the form

$$P(X_1, e) = \sum_{x_d} \cdots \sum_{x_3} \sum_{x_2} \prod_i P(x_i \mid pa_i)$$

- Then iteratively:
 - Move all irrelevant terms outside of innermost sum.
 - Perform innermost sum, getting a new term.
 - Insert the new term into the product.



Outcome of elimination

- Let X be some set of variables
- Let F be a set of factors such that for each $\phi \in F, Scope[\phi] \in X$
- Let $Y \subset X$ be a set of query variables and $Z = X - Y$ be the variable to be eliminated.
- The result of eliminating Z is a factor

$$\tau(Y) = \sum_z \prod_{\phi \in F} \phi$$

- This doesn't necessarily correspond to any probability or conditional probability.



Evidence and Sum-Product

- Evidence potential

$$\delta(E_i, \bar{e}_i) = \begin{cases} 1 & \text{if } E_i \equiv \bar{e}_i \\ 0 & \text{if } E_i \neq \bar{e}_i \end{cases}$$

- Total evidence potential

$$\delta(\mathbf{E}, \bar{\mathbf{e}}) = \prod_{i \in I_E} \delta(E_i, \bar{e}_i)$$

- Introducing evidence-restricted factors:

$$\tau(Y, \bar{e}) = \sum_{z,e} \prod_{\phi \in F} \phi \cdot \delta(E, \bar{e})$$



Variable Elimination Algorithm

Procedure **Elimination**(

G , // the GM

E , // evidence

Z , // set of variables to be eliminated

X , // query variable(s)

)

1. Initialize (G)
2. Evidence (E)
3. Sum-product-Elimination (F, Z)
4. Normalization (F)



Variable Elimination Algorithm

Procedure Initialize (G, Z)

1. Let Z_1, \dots, Z_k be an ordering of Z such that $Z_i < Z_j$ iff $i < j$
2. Initialize F with the full set of factors

Procedure Evidence (E)

1. **for** each $i \in I_E$,
- $F = F \cup \delta(E_i, e_i)$

Procedure Sum-Product-Variable-Elimination (F, Z, \prec)

1. **for** $i = 1, \dots, k$
- $F \leftarrow \text{Sum-Product-Eliminate-Var}(F, Z_i)$
2. $\phi^* \leftarrow \prod_{\phi \in F} \phi$
3. **return** ϕ^*
4. Normalization (ϕ^*)

Procedure Normalization (ϕ^*)

1. $P(X|E) = \phi^*(X) / \sum_x \phi^*(X)$

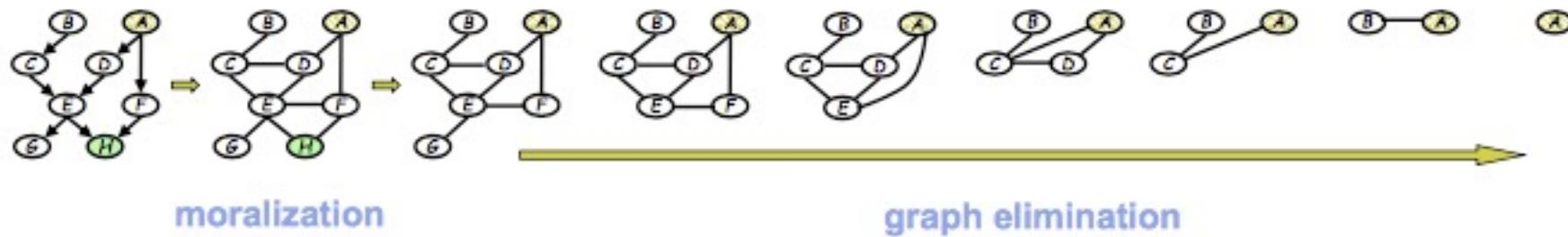
Procedure Sum-Product-Eliminate-Var (F, Z)

- F , // Set of factors
 Z // Variable to be eliminated
)
1. $F' \leftarrow \{\phi \in F : Z \in \text{Scope}[\phi]\}$
 2. $F'' \leftarrow F - F'$
 3. $\psi \leftarrow \prod_{\phi \in F'} \phi$
 4. $\tau \leftarrow \sum_Z \psi$
 5. **return** $F'' \cup \{\tau\}$

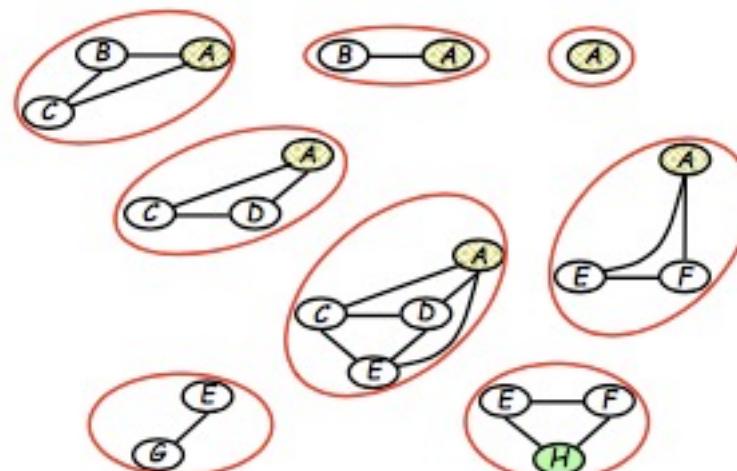
Complexity is **exponential** in number of variables in the **intermediate factor**

Understanding Variable Elimination

- A graph elimination algorithm



- Intermediate terms correspond to the **cliques** resulted from elimination



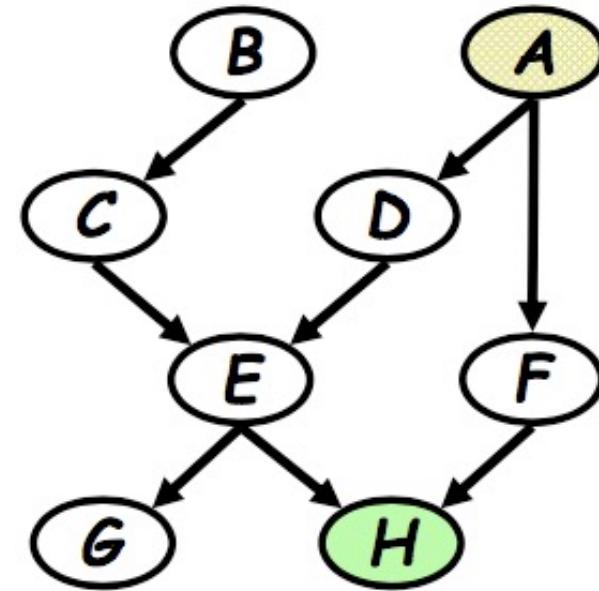
Variable Elimination: Example

Query: $P(A|h)$

- Need to eliminate: B, C, D, E, F, G, H

Initial factors:

$$P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A)P(G|E)P(H|E,F)$$



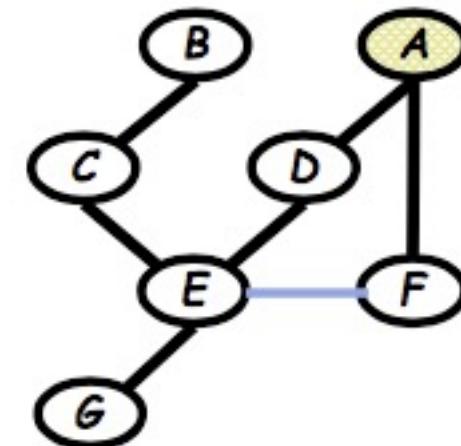
Step 1:

- **Conditioning** on evidence (fix H to h)

$$p_H(E, F) = P(H = h | E, F)$$

Same as a marginalization step

$$p_H(E, F) = \sum_{h'} P(H = h | E, F) \delta(h' = h)$$



Variable Elimination: Example, p2

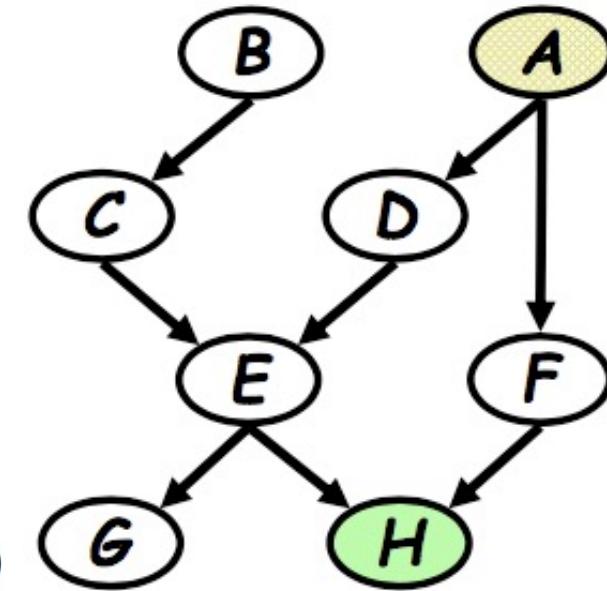
Query: $P(A|h)$

- Need to eliminate: B, C, D, E, F, G, H

Initial factors:

$$P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A)P(G|E)P(H|E,F)$$

$$\Rightarrow P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A)P(G|E)p_H(E,F)$$

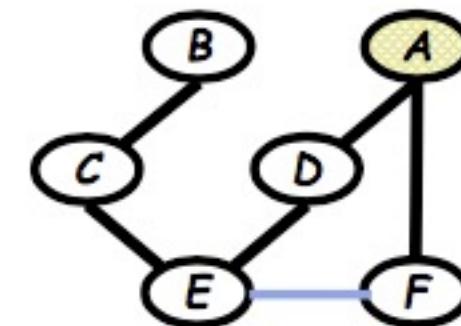


Step 2: Eliminate G

$$p_G(E) = \sum_g P(G = g | E) = 1$$

$$\Rightarrow P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A)P(G|E)p(E,F)$$

$$\Rightarrow P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A) p_H(E,F)$$



Variable Elimination: Example, p3

Query: $P(A|h)$

- Need to eliminate: B, C, D, E, F, G, H

Initial factors:

$$P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A)P(G|E)P(H|E,F)$$

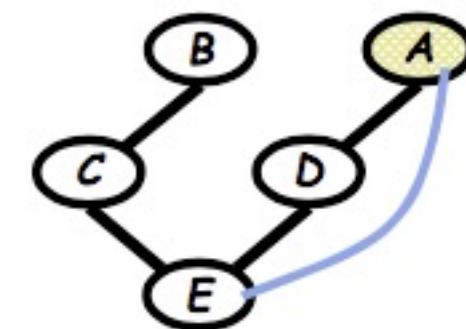
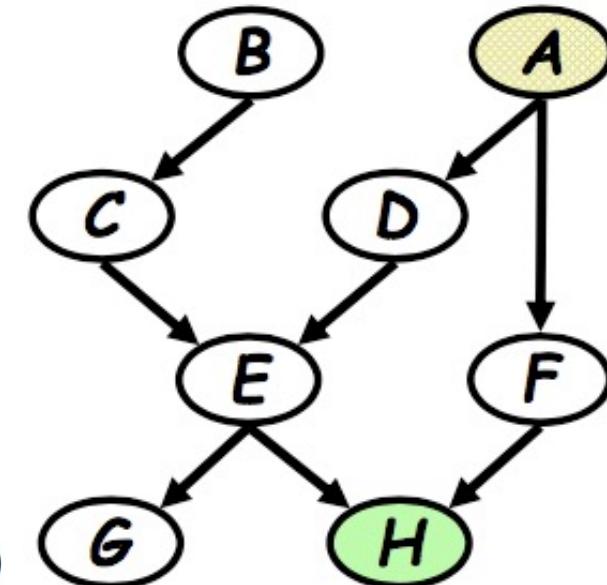
$$\Rightarrow P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A)P(G|E)p_H(E,F)$$

$$\Rightarrow P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A) p_H(E,F)$$

Step 3: Eliminate F

$$p_H(E, A) = \sum_f P(F = f|A)p_H(E, F)$$

$$\Rightarrow P(A)P(B)P(C|B)P(D|A)P(E|C,D) p_F(A,E)$$



Variable Elimination: Example, p4

Query: $P(A|h)$

- Need to eliminate: B, C, D, E, F, G, H

Initial factors:

$$P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A)P(G|E)P(H|E,F)$$

$$\Rightarrow P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A)P(G|E)p_H(E,F)$$

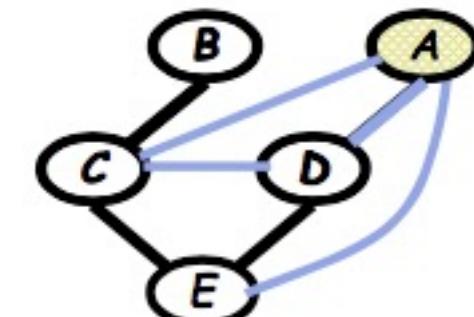
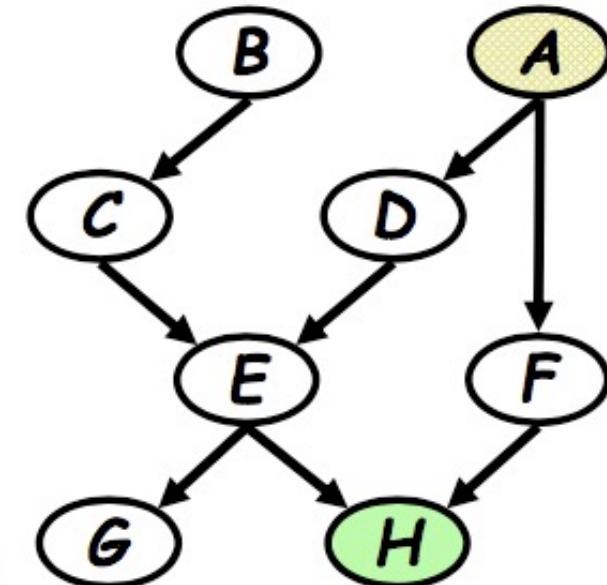
$$\Rightarrow P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A) p_H(E,F)$$

$$\Rightarrow P(A)P(B)P(C|B)P(D|A)P(E|C,D) p_F(A,E)$$

Step 4: Eliminate E

$$p_E(A, C, D) = \sum_e P(E = e | C, D) p_F(A, E)$$

$$\Rightarrow P(A)P(B)P(C|B)P(D|A)p_E(A,C,D)$$



Variable Elimination: Example, p5

Query: $P(A|h)$

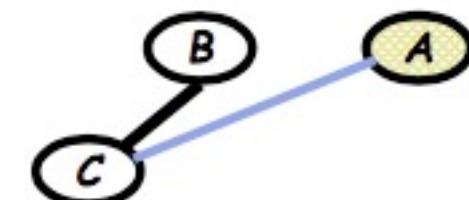
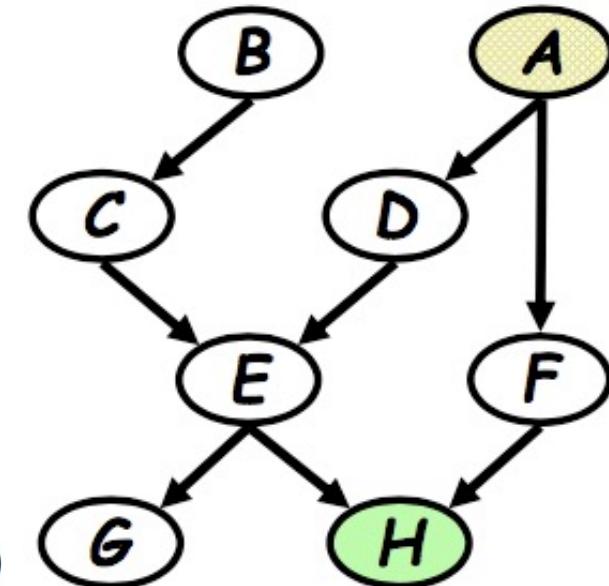
- Need to eliminate: B, C, D, E, F, G, H

Initial factors:

$$\begin{aligned}
 & P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A)P(G|E)P(H|E,F) \\
 \Rightarrow & P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A)P(G|E)p_H(E,F) \\
 \Rightarrow & P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A) p_H(E,F) \\
 \Rightarrow & P(A)P(B)P(C|B)P(D|A)P(E|C,D) p_F(A,E) \\
 \Rightarrow & P(A)P(B)P(C|B)P(D|A)p_E(A,C,D)
 \end{aligned}$$

Step 5: Eliminate D

$$\begin{aligned}
 p_D(A, C) &= \sum_d P(D = d|A)p_E(A, C, D) \\
 \Rightarrow & P(A)P(B)P(C|B) p_D(A,C)
 \end{aligned}$$



Variable Elimination: Example, p6

Query: $P(A|h)$

- Need to eliminate: B, C, D, E, F, G, H

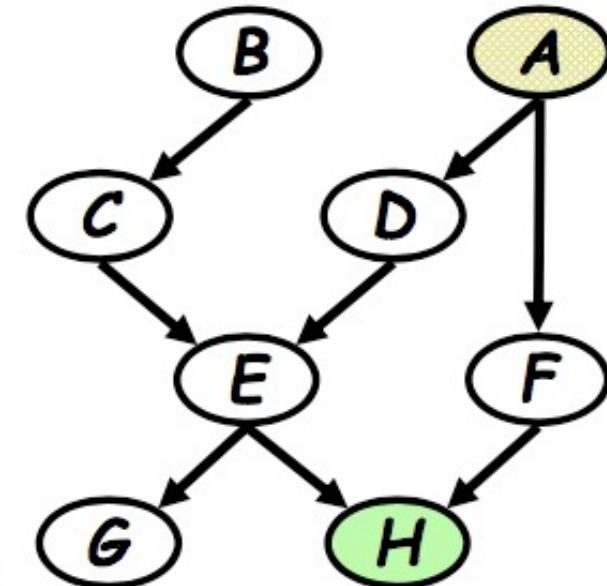
Initial factors:

$$\begin{aligned}
 & P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A)P(G|E)P(H|E,F) \\
 \Rightarrow & P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A)P(G|E)p_H(E,F) \\
 \Rightarrow & P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A) p_H(E,F) \\
 \Rightarrow & P(A)P(B)P(C|B)P(D|A)P(E|C,D) p_F(A,E) \\
 \Rightarrow & P(A)P(B)P(C|B)P(D|A)p_E(A,C,D) \\
 \Rightarrow & P(A)P(B)P(C|B) p_D(A,C)
 \end{aligned}$$

Step 6: Eliminate C

$$p_C(A, B) = \sum_c P(C = c|B)p_D(A, C)$$

$$\Rightarrow P(A)P(B)P(C|B) p_C(A,B)$$



Variable Elimination: Example, p7

Query: $P(A|h)$

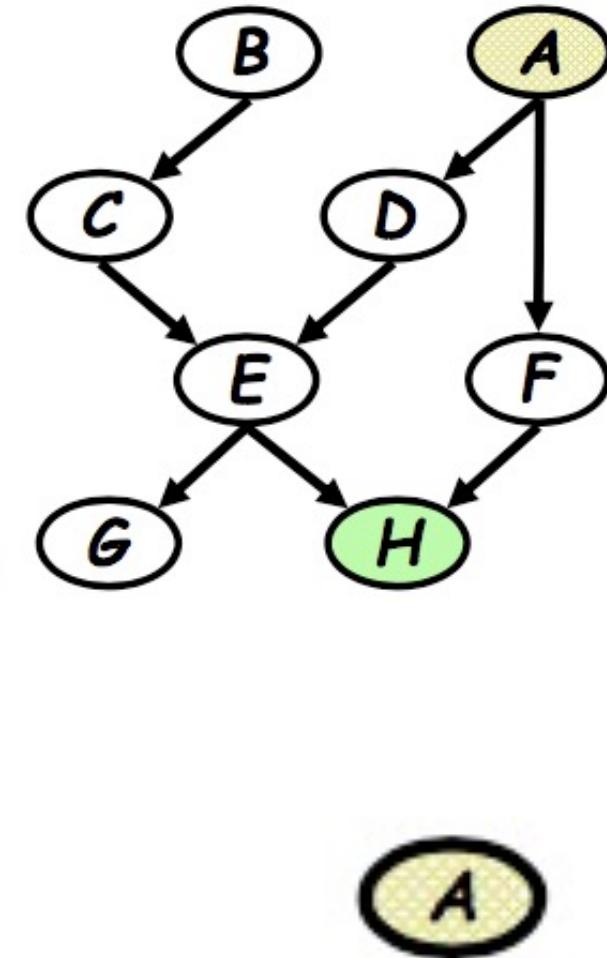
- Need to eliminate: B, C, D, E, F, G, H

Initial factors:

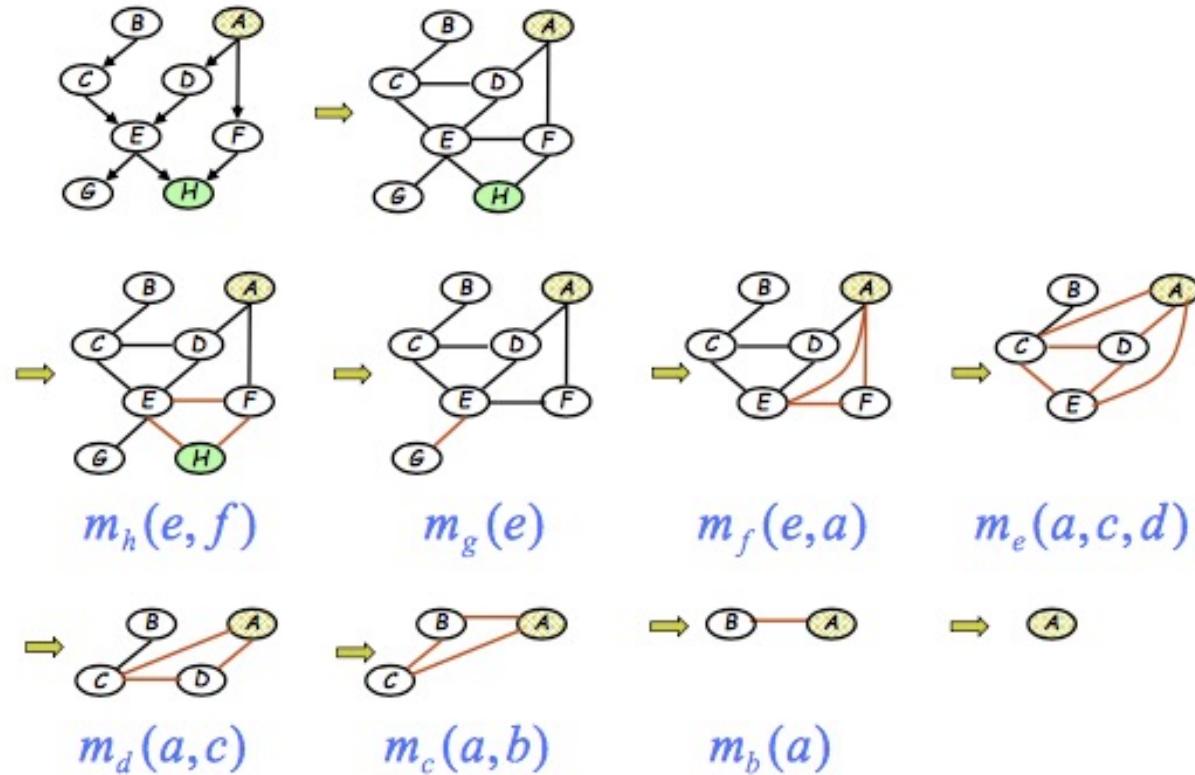
$$\begin{aligned}
 & P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A)P(G|E)P(H|E,F) \\
 \Rightarrow & P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A)P(G|E)p_H(E,F) \\
 \Rightarrow & P(A)P(B)P(C|B)P(D|A)P(E|C,D)P(F|A) p_H(E,F) \\
 \Rightarrow & P(A)P(B)P(C|B)P(D|A)P(E|C,D) p_F(A,E) \\
 \Rightarrow & P(A)P(B)P(C|B)P(D|A)p_E(A,C,D) \\
 \Rightarrow & P(A)P(B)P(C|B) p_D(A,C) \\
 \Rightarrow & P(A)P(B)p_C(A,B)
 \end{aligned}$$

Step 7: Eliminate B

$$\begin{aligned}
 p_B(A) &= \sum_b P(B = b|A)p_C(A,B) \\
 \Rightarrow P(A)p_B(A)
 \end{aligned}$$



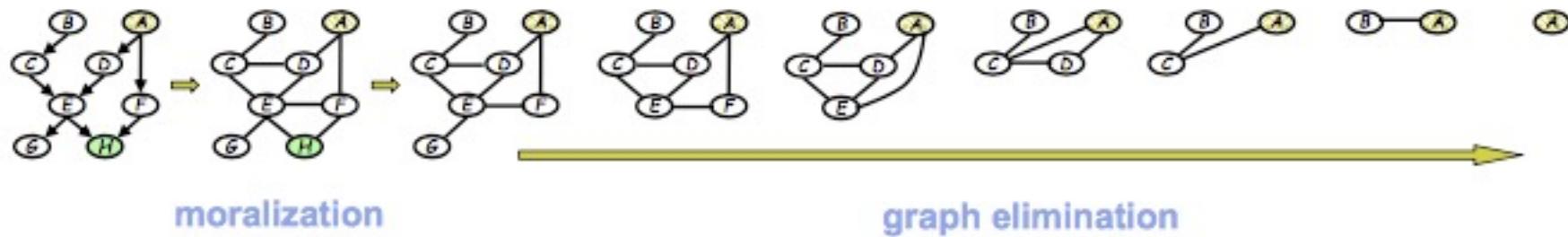
Elimination Cliques



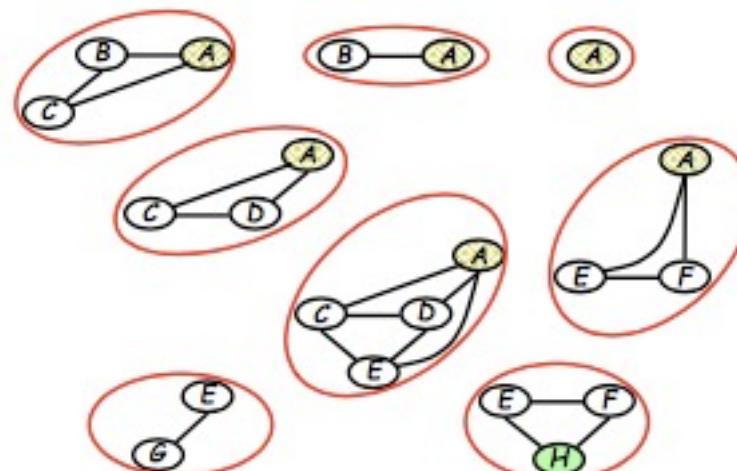
Algebraic elimination = Graphical Elimination

Understanding Variable Elimination

- A graph elimination algorithm



- Intermediate terms correspond to the **cliques** resulted from elimination



Questions?

