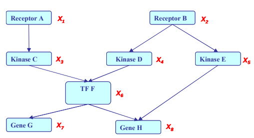
School of Computer Science
Carnegie Mellon

The Elimination Algorithm

Probabilistic Graphical Models (10-708)

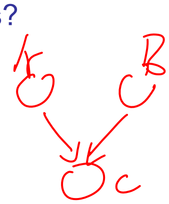
Lecture 4, Sep 26, 2007

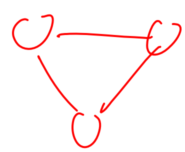


Eric Xing

Reading: J-Chap 3, KF-Chap. 8, 9

- Questions?





$A \perp B$

Probabilistic Inference

- We now have compact representations of probability distributions: **Graphical Models**
- A GM G describes a unique probability distribution P
- How do we answer **queries** about P?
- We use **inference** as a name for the process of computing answers to such queries

Query 1: Likelihood

- Most of the queries one may ask involve **evidence**
  - Evidence $\mathbf{e}$ is an assignment of values to a set $\mathbf{E}$ variables in the domain
  - Without loss of generality $\mathbf{E} = \{ X_{k+1}, \ldots, X_n \}$
- Simplest query: compute probability of evidence

$$P(\mathbf{e}) = \sum_{x_1} \cdots \sum_{x_k} P(x_1, \ldots, x_k, \mathbf{e})$$

  - this is often referred to as computing the **likelihood** of $\mathbf{e}$

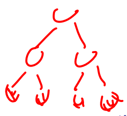
Query 2: Conditional Probability

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

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

  - this is the ***a posteriori* belief** in X, given evidence $\mathbf{e}$

- We usually query a subset $\mathbf{Y}$ of all domain variables $\mathbf{X}=\{\mathbf{Y},\mathbf{Z}\}$ and "don't care" about the remaining, $\mathbf{Z}$:

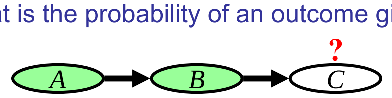


$$P(\mathbf{Y} \mid e) = \sum_{\mathbf{z}} P(\mathbf{Y},\mathbf{Z}=\mathbf{z} \mid \mathbf{e})$$

  - the process of summing out the "don't care" variables z is called **marginalization**, and the resulting $P(y|e)$ is called a **marginal** prob.

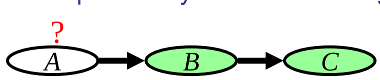
Applications of a posteriori Belief

- **Prediction**: what is the probability of an outcome given the starting condition



  - the query node is a descendent of the evidence

- **Diagnosis**: what is the probability of disease/fault given symptoms



  - the query node an ancestor of the evidence

- **Learning** under partial observation
  - fill in the unobserved values under an "EM" setting (more later)

- The directionality of information flow between variables is not restricted by the directionality of the edges in a GM
  - probabilistic inference can combine evidence form all parts of the network

Query 3: Most Probable Assignment

- In this query we want to find the **most probable joint assignment** (MPA) for *some* variables of interest
- Such reasoning is usually performed under some given evidence $\mathbf{e}$, and ignoring (the values of) other variables $\mathbf{z}$:

$$\mathrm{MPA}(\mathbf{Y} \mid \mathbf{e}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} P(\mathbf{y} \mid \mathbf{e}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} \sum_{\mathbf{z}} P(\mathbf{y}, \mathbf{z} \mid \mathbf{e})$$

  - this is the **maximum *a posteriori*** configuration of $\mathbf{y}$.

Applications of MPA

- Classification
  - find most likely label, given the evidence
- Explanation
  - what is the most likely scenario, given the evidence

Cautionary note:

- The MPA of a variable depends on its "context"---the set of variables been jointly queried
- Example:
  - MPA of X ? 1
  - MPA of (X, Y) ? 0 0

x	y	P(x,y)
0	0	0.35
0	1	0.05
1	0	0.3
1	1	0.3

Complexity of Inference

Thm:

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

- Hardness does not mean we cannot solve inference
  - It implies that we cannot find a general procedure that works efficiently for arbitrary GMs
  - For particular families of GMs, we can have provably efficient procedures

Approaches to inference

- Exact inference algorithms

$p(x|e)$

  - The elimination algorithm ✓
  - Message-passing algorithm (sum-product, belief propagation) ✓
  - The junction tree algorithms ✓
- Approximate inference techniques
  - Stochastic simulation / sampling methods ✓
  - Markov chain Monte Carlo methods ✓
  - Variational algorithms ?

Marginalization and Elimination

- A signal transduction pathway:

$$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$$

What is the likelihood that protein E is active?

- Query: $P(e)$

$$P(e) = \underbrace{\sum_d \sum_c \sum_b \sum_a}_{} P(a,b,c,d,e)$$

a naïve summation needs to enumerate over an exponential number of terms

- By chain decomposition, we get

$$= \sum_d \sum_c \sum_b \sum_a P(a)P(b \mid a)P(c \mid b)P(d \mid c)P(e \mid d)$$

Elimination on Chains

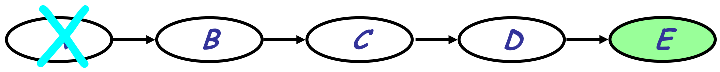
$$A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$$

- Rearranging terms ...

$$P(e) = \sum_d \sum_c \sum_b \sum_a P(a)P(b \mid a)P(c \mid b)P(d \mid c)P(e \mid d)$$

$$= \sum_d \sum_c \sum_b P(c \mid b)P(d \mid c)P(e \mid d) \sum_a P(a)P(b \mid a)$$

Elimination on Chains

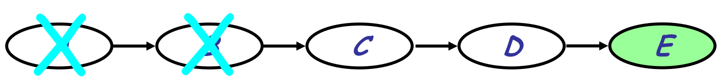


- Now we can perform innermost summation

$$P(e) = \sum_d \sum_c \sum_b P(c \mid b) P(d \mid c) P(e \mid d) \sum_a P(a) P(b \mid a)$$
$$= \sum_d \sum_c \sum_b P(c \mid b) P(d \mid c) P(e \mid d)\, p(b)$$

- This summation "eliminates" one variable from our summation argument at a "local cost".

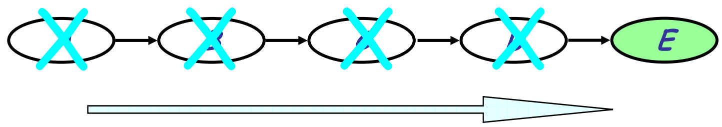
Elimination in Chains



- Rearranging and then summing again, we get

$$P(e) = \sum_d \sum_c \sum_b P(c \mid b) P(d \mid c) P(e \mid d)\, p(b)$$
$$= \sum_d \sum_c P(d \mid c) P(e \mid d) \sum_b P(c \mid b)\, p(b)$$
$$= \sum_d \sum_c P(d \mid c) P(e \mid d)\, p(c)$$

Elimination in Chains



- Eliminate nodes one by one all the way to the end, we get

$p(X_{1:5})$

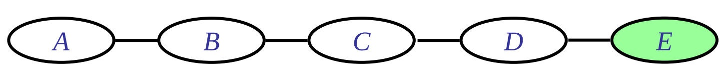
$O(n^k)$

$\phi(X_i) = \sum_{x_{i-1}} P(x_i|x_{i-1})\, p(x_{i-1})$

$$P(e) = \sum_d P(e\,|\,d)\,p(d)$$

- Complexity:
  - Each step costs $O(|Val(X_i)|*|Val(X_{i+1})|)$ operations: $O(kn^2)$
  - Compare to naïve evaluation that sums over joint values of n-1 variables $O(n^k)$

Undirected Chains



- Rearranging terms ...

$P = \frac{1}{Z} \exp(\sum \log \phi_i)$

$$P(e) = \sum_d \sum_c \sum_b \sum_a \frac{1}{Z}\phi(b,a)\phi(c,b)\phi(d,c)\phi(e,d)$$

$$= \frac{1}{Z}\sum_d \sum_c \sum_b \phi(c,b)\phi(d,c)\phi(e,d)\sum_a \phi(b,a)$$

$$= \cdots$$

The Sum-Product Operation

- In general, we can view the task at hand as that of computing the value of an expression of the form:

$$\sum_{\mathbf{z}} \prod_{\phi \in \mathcal{F}} \phi$$

where $\mathcal{F}$ is a set of factors

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

Outcome of elimination

- Let $\mathbf{X}$ be some set of variables,
  let $\mathcal{F}$ be a set of factors such that for each $\phi \in \mathcal{F}$, *Scope*$[\phi] \in \mathbf{X}$,
  let $\mathbf{Y} \subset \mathbf{X}$ be a set of query variables,
  and let $Z = \mathbf{X}-\mathbf{Y}$ be the variable to be eliminated

- The result of eliminating the variable Z is a factor

$$\tau(\mathbf{Y}) = \sum_{\mathbf{z}} \prod_{\phi \in \mathcal{F}} \phi$$

$$P(Y) = \frac{\tau(Y)}{\Sigma(Y)}$$

  - This factor does not necessarily correspond to any probability or conditional probability in this network. (example forthcoming)

Dealing with evidence

- Conditioning as a Sum-Product Operation
  - The 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: $\delta(\mathbf{E}, \bar{\mathbf{e}}) = \prod_{i \in I_{\mathbf{E}}} \delta(E_i, \bar{e}_i)$
  - Introducing evidence --- restricted factors:

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

Inference on General GM via Variable Elimination

General idea:

- Write query in the form

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

  - this suggests an "elimination order" of latent variables to be marginalized

    hidden

- Iteratively
  - Move all irrelevant terms outside of innermost sum
  - Perform innermost sum, getting a new term
  - Insert the new term into the product
- wrap-up $P(X_1 \mid \mathbf{e}) = \dfrac{\phi(X_1, \mathbf{e})}{\sum_{x_1} \phi(X_1, \mathbf{e})}$

The elimination algorithm

Procedure Elimination (

G, // the GM

$\mathbf{E}$, // evidence

Z, // Set of variables to be eliminated

X, // query variable(s)

)

1. Initialize (G)
2. Evidence ($\mathbf{E}$)
3. Sum-Product-Elimination ($\mathcal{F}$, Z, $\prec$)
4. Normalization ($\mathcal{F}$)

The elimination algorithm

Procedure Initialize (G, Z)

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

Procedure Evidence ($\mathbf{E}$)

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

Procedure Sum-Product-Variable-Elimination ($\mathcal{F}$, Z, $\prec$)

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

E_i $\bar{e}_i$

$$\delta(E_i; \bar{e}_i) = \begin{cases} 1 \\ 0 \end{cases}$$

The elimination algorithm

Procedure Initialize (G, Z)

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

Procedure Evidence $(\mathbf{E})$

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

Procedure Sum-Product-Variable-Elimination $(\mathcal{F}, Z, \prec)$

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

Procedure Normalization (ϕ^*)

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

$\sum_{z_3} \phi(z_1)\phi(z_2)\phi(z_3) \cdot \phi(z_2, z_3)$

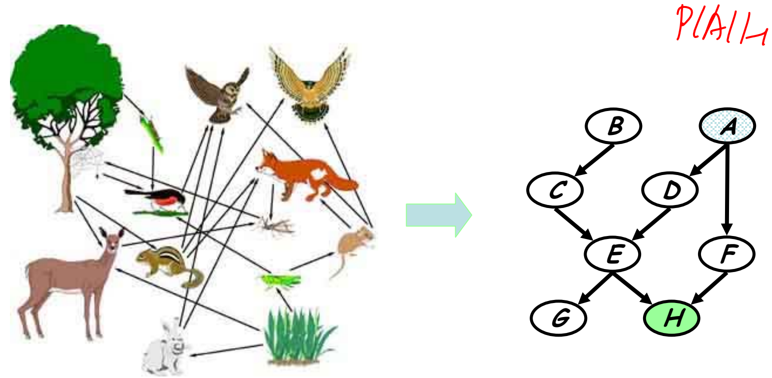
Procedure Sum-Product-Eliminate-Var (
   $\mathcal{F}$, // Set of factors
   Z // Variable to be eliminated
)

1. $\mathcal{F}' \leftarrow \{\phi \in \mathcal{F} : Z \in Scope[\phi]\}$
2. $\mathcal{F}'' \leftarrow \mathcal{F} - \mathcal{F}'$
3. $\psi \leftarrow \prod_{\phi \in \mathcal{F}'} \phi$
4. $\tau \leftarrow \sum_Z \psi$
5. **return** $\mathcal{F}'' \cup \{\tau\}$

A more complex network

A food web

$P(A|H)$



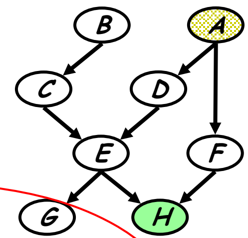
What is the probability that hawks are leaving given that the grass condition is poor?

Example: Variable Elimination

- Query: $P(A \mid h)$
  - Need to eliminate: B,C,D,E,F,G,H
- Initial factors:

  F ⇐ Factor ∪ Evidence

  $$P(a)P(b)P(c \mid b)P(d \mid a)P(e \mid c,d)P(f \mid a)P(g \mid e)P(h \mid e,f)$$
- Choose an elimination order: H,G,F,E,D,C,B



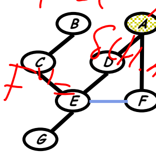
- Step 1:
  - **Conditioning** (fix the evidence node (i.e., h) on its observed value (i.e., $\tilde{h}$)):

  $$m_h(e,f) = p(h = \tilde{h} \mid e,f)$$

  - This step is isomorphic to a marginalization step:

  $$m_h(e,f) = \sum_h p(h \mid e,f)\delta(h = \tilde{h})$$

F' = (δ(h, h̃))

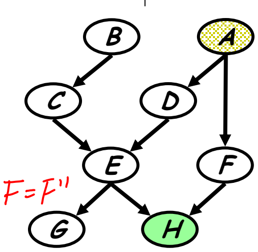


Example: Variable Elimination

- Query: $P(B \mid h)$
  - Need to eliminate: B,C,D,E,F,G
- Initial factors:

  $$P(a)P(b)P(c \mid b)P(d \mid a)P(e \mid c,d)P(f \mid a)P(g \mid e)P(h \mid e,f)$$
  $$\Rightarrow P(a)P(b)P(c \mid b)P(d \mid a)P(e \mid c,d)P(f \mid a)P(g \mid e)m_h(e,f) \quad F = F''$$

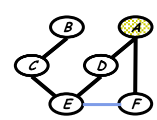
  → F'



- Step 2: Eliminate G
  - compute

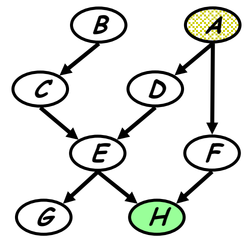
  $$m_g(e) = \sum_g p(g \mid e) = 1$$

  $$\Rightarrow P(a)P(b)P(c \mid b)P(d \mid a)P(e \mid c,d)P(f \mid a)m_g(e)m_h(e,f)$$
  $$= P(a)P(b)P(c \mid b)P(d \mid a)P(e \mid c,d)P(f \mid a)\underline{m_h(e,f)} \quad F''$$



Example: Variable Elimination

- Query: $P(B \mid h)$
  - Need to eliminate: B, C, D, E, F



- Initial factors:

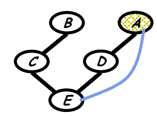
$$P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$$
$$\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f)$$
$$\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)\underline{P(f|a)m_h(e,f)}$$

$\mathcal{F}'$ $\quad\quad$ $\mathcal{F}'' = \mathcal{F} \setminus \mathcal{F}'$

- Step 3: Eliminate F
  - compute

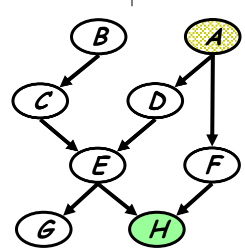
$$m_f(e,a) = \sum_f p(f \mid a) m_h(e,f) \quad \underline{\sum_f \prod}$$

$$\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)\underline{m_f(a,e)}$$



Example: Variable Elimination

- Query: $P(B \mid h)$
  - Need to eliminate: B, C, D, E



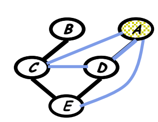
- Initial factors:

$$P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$$
$$\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f)$$
$$\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_h(e,f)$$
$$\Rightarrow P(a)P(b)P(c|b)P(d|a)\underline{P(e|c,d)m_f(a,e)}$$

- Step 4: Eliminate E
  - compute

$$m_e(a,c,d) = \sum_e p(e|c,d) m_f(a,e)$$

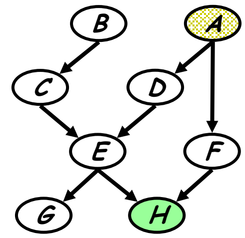
$$\Rightarrow P(a)P(b)P(c|b)P(d|a)\underline{m_e(a,c,d)}$$



Example: Variable Elimination

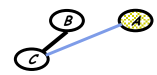
- Query: *P(B | h)*
  - Need to eliminate: *B,C,D*
- Initial factors:

$$P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$$
$$\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f)$$
$$\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_h(e,f)$$
$$\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)m_f(a,e)$$
$$\Rightarrow P(a)P(b)P(c|b)\underline{P(d|a)m_e(a,c,d)}$$



- Step 5: Eliminate *D*
  - compute $m_d(a,c) = \sum_d p(d|a)m_e(a,c,d)$

$$\Rightarrow P(a)P(b)\underline{P(c|d)m_d(a,c)}$$

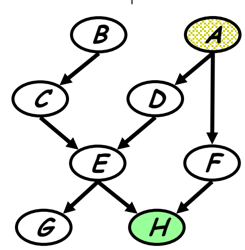


Eric Xing 29

Example: Variable Elimination

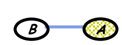
- Query: *P(B | h)*
  - Need to eliminate: *B,C*
- Initial factors:

$$P(a)P(b)P(c|d)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$$
$$\Rightarrow P(a)P(b)P(c|d)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f)$$
$$\Rightarrow P(a)P(b)P(c|d)P(d|a)P(e|c,d)P(f|a)m_h(e,f)$$
$$\Rightarrow P(a)P(b)P(c|d)P(d|a)P(e|c,d)m_f(a,e)$$
$$\Rightarrow P(a)P(b)P(c|d)P(d|a)m_e(a,c,d)$$
$$\Rightarrow P(a)P(b)\underline{P(c|d)m_d(a,c)}$$



- Step 6: Eliminate *C*
  - compute $m_c(a,b) = \sum_c p(c|b)m_d(a,c)$

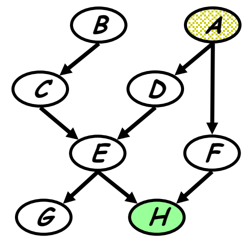
$$\Rightarrow P(a)P(b)\underline{P(c|d)m_d(a,c)}$$



Eric Xing 30

Example: Variable Elimination

- Query: *P(B | h)*
  - Need to eliminate: *B*
- Initial factors:



$$P(a)P(b)P(c|d)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$$
$$\Rightarrow P(a)P(b)P(c|d)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f)$$
$$\Rightarrow P(a)P(b)P(c|d)P(d|a)P(e|c,d)P(f|a)m_h(e,f)$$
$$\Rightarrow P(a)P(b)P(c|d)P(d|a)P(e|c,d)m_f(a,e)$$
$$\Rightarrow P(a)P(b)P(c|d)P(d|a)m_e(a,c,d)$$
$$\Rightarrow P(a)P(b)P(c|d)m_d(a,c)$$
$$\Rightarrow P(a)\underline{P(b)m_c(a,b)}$$

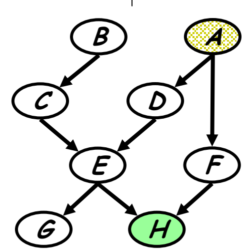
- Step 7: Eliminate *B*

  - compute $$m_b(a) = \sum_b p(b)m_c(a,b)$$

$$\Rightarrow P(a)\underline{m_b(a)}$$

Example: Variable Elimination

- Query: *P(B | h)*
  - Need to eliminate: *B*
- Initial factors:



$$P(a)P(b)P(c|d)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$$
$$\Rightarrow P(a)P(b)P(c|d)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f)$$
$$\Rightarrow P(a)P(b)P(c|d)P(d|a)P(e|c,d)P(f|a)m_h(e,f)$$
$$\Rightarrow P(a)P(b)P(c|d)P(d|a)P(e|c,d)m_f(a,e)$$
$$\Rightarrow P(a)P(b)P(c|d)P(d|a)m_e(a,c,d)$$
$$\Rightarrow P(a)P(b)P(c|d)m_d(a,c)$$
$$\Rightarrow P(a)P(b)m_c(a,b)$$
$$\Rightarrow P(a)m_b(a)$$

- Step 8: Wrap-up $$p(a,\tilde{h}) = p(a)m_b(a), \quad p(\tilde{h}) = \sum_a p(a)m_b(a)$$

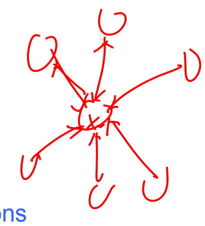
$$\Rightarrow P(a|\tilde{h}) = \frac{p(a)m_b(a)}{\sum_a p(a)m_b(a)}$$

Complexity of variable elimination

- Suppose in one elimination step we compute

$$m_x(y_1,\dots,y_k) = \sum_x m'_x(x, y_1,\dots,y_k)$$

$$m'_x(x, y_1,\dots,y_k) = \prod_{i=1}^{k} m_i(x, \mathbf{y}_{c_i})$$



This requires

- $k \bullet |\mathrm{Val}(X)| \bullet \prod_i |\mathrm{Val}(\mathbf{Y}_{C_i})|$ multiplications
  - For each value for $x, y_1, \dots, y_k$, we do k multiplications

$\phi(x\, \iota_i \quad)$

- $|\mathrm{Val}(X)| \bullet \prod_i |\mathrm{Val}(\mathbf{Y}_{C_i})|$ additions
  - For each value of $y_1, \dots, y_k$, we do $|Val(X)|$ additions

Complexity is exponential in number of variables in the intermediate factor