ECE 286: Bayesian Machine Perception

Class 6: Variable Elimination

Florian Meyer

Electrical and Computer Engineering Department
University of California San Diego

UC San Diego
JACOBS SCHOOL OF ENGINEERING

Marginalization

- Conditional probability distribution of random variable $\boldsymbol{x} = [\boldsymbol{x}_1^{\mathrm{T}}\ \boldsymbol{x}_2^{\mathrm{T}} \ldots \boldsymbol{x}_n^{\mathrm{T}}]^{\mathrm{T}}$ given the observations $\boldsymbol{y}$

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{y})} = \frac{p(\boldsymbol{x},\boldsymbol{y})}{\sum_{\boldsymbol{x}'} p(\boldsymbol{x}=\boldsymbol{x}',\boldsymbol{y})}$$

- This is the posterior distribution of $\boldsymbol{x}$ given observations $\boldsymbol{y}$

- We are often interested in a subset $\boldsymbol{x}_1$ of all random variables $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_{\sim 1}\}$ and ``don't care'' about the remaining

$$p(\boldsymbol{x}_1|\boldsymbol{y}) = \sum_{\boldsymbol{x}_{\sim 1}} p(\boldsymbol{x}_1, \boldsymbol{x}_{\sim 1}|\boldsymbol{y})$$

- Marginalization is the process of summing out the ``don't care'' variables

Recall MMSE and MAP Estimation

- Joint density $p(\boldsymbol{x}|\boldsymbol{y}) = p(\boldsymbol{x}_1, \boldsymbol{x}_{\sim 1}|\boldsymbol{y})$ and observations $\boldsymbol{y}$

- Minimum mean-square error (MMSE) estimator for $\boldsymbol{x}_1$

$$\hat{\boldsymbol{x}}_1(\boldsymbol{y}) = \sum_{\boldsymbol{x}_1} \boldsymbol{x}_1 p(\boldsymbol{x}_1|\boldsymbol{y})$$

- Maximum a posteriori (MAP) estimator for $\boldsymbol{x}_1$

$$\hat{\boldsymbol{x}}_1(\boldsymbol{y}) = \arg\max_{\boldsymbol{x}_1} p(\boldsymbol{x}_1|\boldsymbol{y})$$

- Both MMSE and MAP rely on the marginal density $p(\boldsymbol{x}_1|\boldsymbol{y}) = \sum_{\boldsymbol{x}_{\sim 1}} p(\boldsymbol{x}_1, \boldsymbol{x}_{\sim 1}|\boldsymbol{y})$

MAP Estimation Revised

- The MAP value of a variable depends on its ``context'' – the set of variables that are jointly estimated

y_1	y_2	$P(y_1, y_2)$
0	0	0.35
0	1	0.05
1	0	0.3
1	1	0.3

- Example:
  - What is the MAP of $\boldsymbol{y} = [y_1\, y_2]^{\mathrm{T}}$?
  - What is the MAP of y_1?
- Applications:
  - Navigation: What is the most likely position, given the observations?
  - Classification: What is the most likely label, given the observations?

Complexity of Estimation

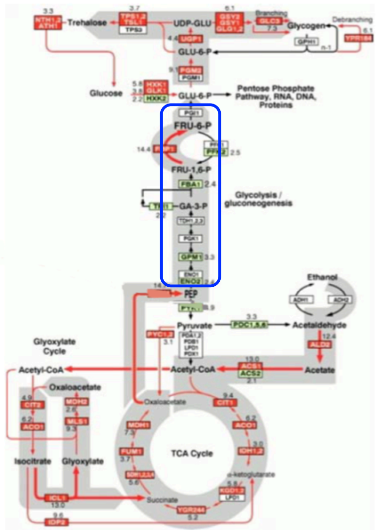
- **Theorem:** Computing $p(\boldsymbol{x}_1|\boldsymbol{y}) = \sum_{\boldsymbol{x}_{\sim 1}} p(\boldsymbol{x}_1, \boldsymbol{x}_{\sim 1}|\boldsymbol{y})$ is NP-hard

- Hardness implies that we cannot find a general procedure that works efficiently for arbitrary distributions $p(\boldsymbol{x}|\boldsymbol{y})$

- We can develop provable scalable estimation methods if $p(\boldsymbol{x}|\boldsymbol{y})$ belongs to particular families of graphical models

Marginalization and Elimination

- A signal transduction pathway:



What is the probability that protein e is active

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

A naive summation needs to enumerate a number of terms that scales exponentially with the number of proteins

- By using the chain structure, we get, e.g.,

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

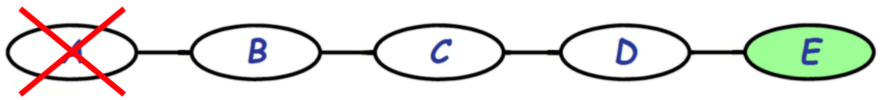
Elimination on Chains

- Rearranging the terms …

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

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

Elimination on Chains

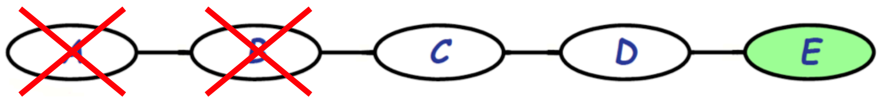


- Now we can perform the innermost summation

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

- This summation ``eliminates'' one variable from our summation argument at a ``local cost''

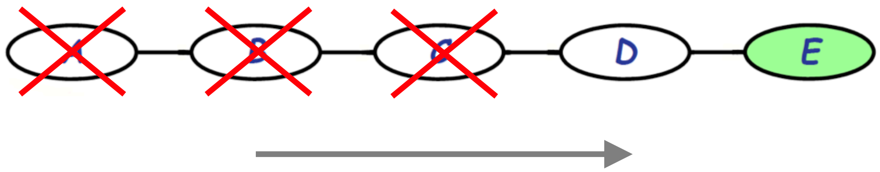
Elimination on Chains



- Rearranging and then summing again, we get

$$
\begin{aligned}
p(e) &= \sum_d \sum_c \sum_b p(c|b)\,p(d|c)\,p(e|d)\,p(b) \\
&= \sum_d \sum_c p(d|c)\,p(e|d) \sum_b p(c|b)\,p(b) \\
&= \sum_d \sum_c p(d|c)\,p(e|d)p(c)
\end{aligned}
$$

Elimination on Chains



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

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

- Complexity for number of nodes n and $a, b, ... \in \{1, 2, \ldots, k\}$
  - Each elimination step costs k^2 operations: $\mathcal{O}(nk^2)$
  - Naive evaluation that sums over joint values of $n-1$ variables: $\mathcal{O}(k^n)$

- By exploiting the structure of the graph computational complexity is strongly reduced

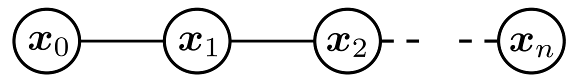
General Chain Factorization

- Rearranging the terms …

$$p(e) = \frac{1}{Z} \sum_a \sum_b \sum_c \sum_d \psi_{ab}(a,b)\,\psi_{bc}(b,c)\,\psi_{cd}(c,d)\,\psi_{de}(d,e)$$

$$= \frac{1}{Z} \sum_d \sum_c \sum_b \psi_{bc}(b,c)\,\psi_{cd}(c,d)\,\psi_{de}(d,e) \sum_a \psi_{ab}(a,b)$$

Example: State Space Model



- The joint pdf is given by

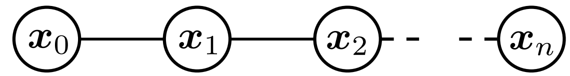
$$\begin{aligned} f(\boldsymbol{x}_{0:n}, \boldsymbol{y}_{1:n}) &= f(\boldsymbol{y}_{1:n}|\boldsymbol{x}_{0:n}) f(\boldsymbol{x}_{0:n}) \\ &= f(\boldsymbol{x}_0) f(\boldsymbol{x}_1|\boldsymbol{x}_0) f(\boldsymbol{y}_1|\boldsymbol{x}_1) f(\boldsymbol{x}_2|\boldsymbol{x}_1) f(\boldsymbol{y}_2|\boldsymbol{x}_2) \dots f(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) f(\boldsymbol{y}_n|\boldsymbol{x}_n) \end{aligned}$$

- After observing $\boldsymbol{y}_{1:n}$ we get $f(\boldsymbol{x}_{0:n}|\boldsymbol{y}_{1:n}) = \frac{1}{Z} f(\boldsymbol{x}_{0:n}, \boldsymbol{y}_{1:n})$ with $Z = f(\boldsymbol{y}_{1:n})$

- Calculation of the marginal posterior pdf

$$\begin{aligned} f(\boldsymbol{x}_n|\boldsymbol{y}_{1:n}) &= \frac{1}{Z} \int \cdots \int \int f(\boldsymbol{x}_{0:n}, \boldsymbol{y}_{1:n}) \mathrm{d}\boldsymbol{x}_0 \mathrm{d}\boldsymbol{x}_1 \dots \mathrm{d}\boldsymbol{x}_{n-1} \\ &= \frac{1}{Z} \int \cdots \int \int f(\boldsymbol{x}_0) f(\boldsymbol{x}_1|\boldsymbol{x}_0) f(\boldsymbol{y}_1|\boldsymbol{x}_1) f(\boldsymbol{x}_2|\boldsymbol{x}_1) \dots f(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) f(\boldsymbol{y}_n|\boldsymbol{x}_n) \mathrm{d}\boldsymbol{x}_0 \mathrm{d}\boldsymbol{x}_1 \dots \mathrm{d}\boldsymbol{x}_{n-1} \end{aligned}$$

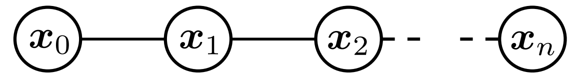
Example: State Space Model



- Calculation of the marginal posterior pdf

$$f(\boldsymbol{x}_n|\boldsymbol{y}_{1:n}) = \frac{1}{Z}\int \cdots \int f(\boldsymbol{x}_2|\boldsymbol{x}_1) f(\boldsymbol{y}_1|\boldsymbol{x}_1) \ldots f(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) f(\boldsymbol{y}_n|\boldsymbol{x}_n) \underbrace{\int f(\boldsymbol{x}_1|\boldsymbol{x}_0) f(\boldsymbol{x}_0)\,\mathrm{d}\boldsymbol{x}_0}_{f(\boldsymbol{x}_1)} \mathrm{d}\boldsymbol{x}_1 \ldots \mathrm{d}\boldsymbol{x}_{n-1}$$

Example: State Space Model

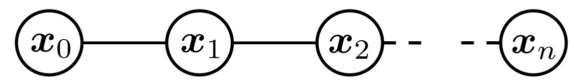


- Calculation of the marginal posterior pdf

$$f(\boldsymbol{x}_n|\boldsymbol{y}_{1:n}) = \frac{1}{Z}\int \cdots \int \int f(\boldsymbol{y}_1|\boldsymbol{x}_1) f(\boldsymbol{x}_2|\boldsymbol{x}_1) \dots f(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) f(\boldsymbol{y}_n|\boldsymbol{x}_n) f(\boldsymbol{x}_1) \mathrm{d}\boldsymbol{x}_1 \mathrm{d}\boldsymbol{x}_2 \dots \mathrm{d}\boldsymbol{x}_{n-1}$$

$$= \frac{1}{Z}\int \cdots \int f(\boldsymbol{y}_2|\boldsymbol{x}_2) \dots f(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) f(\boldsymbol{y}_n|\boldsymbol{x}_n) \underbrace{\int f(\boldsymbol{x}_2|\boldsymbol{x}_1) f(\boldsymbol{y}_1|\boldsymbol{x}_1) f(\boldsymbol{x}_1) \mathrm{d}\boldsymbol{x}_1}_{\propto f(\boldsymbol{x}_2|\boldsymbol{y}_1)} \mathrm{d}\boldsymbol{x}_2 \dots \mathrm{d}\boldsymbol{x}_{n-1}$$

Example: State Space Model



- Calculation of the marginal posterior pdf

$$
\begin{aligned}
f(\boldsymbol{x}_n|\boldsymbol{y}_{1:n}) &= \frac{1}{Z}\int \cdots \int \int f(\boldsymbol{y}_1|\boldsymbol{x}_1) f(\boldsymbol{x}_2|\boldsymbol{x}_1) \ldots f(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) f(\boldsymbol{y}_n|\boldsymbol{x}_n) f(\boldsymbol{x}_1) \mathrm{d}\boldsymbol{x}_1 \mathrm{d}\boldsymbol{x}_2 \ldots \mathrm{d}\boldsymbol{x}_{n-1} \\
&= \frac{1}{Z'}\int \cdots \int f(\boldsymbol{y}_2|\boldsymbol{x}_2) \ldots f(\boldsymbol{x}_n|\boldsymbol{x}_{n-1}) f(\boldsymbol{y}_n|\boldsymbol{x}_n) f(\boldsymbol{x}_2|\boldsymbol{y}_1) \mathrm{d}\boldsymbol{x}_2 \ldots \mathrm{d}\boldsymbol{x}_{n-1}
\end{aligned}
$$

Sequential Bayesian estimation is a special case of variable elimination on a chain

Computational complexity is reduced from exponential to linear in time n

The Sum-Product Operation

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

$$\sum_{\boldsymbol{x}} \prod_{\phi \in \mathcal{F}} \phi \qquad\qquad \int \prod_{\phi \in \mathcal{F}} \phi \, \mathrm{d}\boldsymbol{x}$$

discrete random variables | continuous random variables

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

- We will refer to this task as the sum-product operation

Variable Elimination on General Graphical Models

- Write marginalization task in the form

$$p(\boldsymbol{x}_1|\boldsymbol{z}) \propto \underbrace{\sum_{\boldsymbol{x}_n} \cdots \sum_{\boldsymbol{x}_3} \sum_{\boldsymbol{x}_2}}_{\text{elimination order}} \prod_{c \in \mathcal{C}} \psi_c(\boldsymbol{x}^{(c)})$$

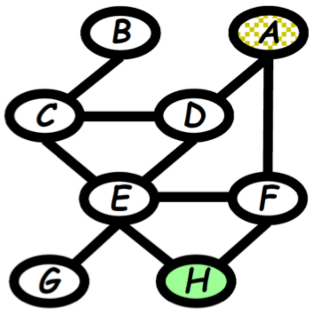
  $\mathcal{C}$ are cliques of the graph ($\boldsymbol{x}^{(c)}$ comprises certain parameter vectors $\boldsymbol{x}_i$, where each $\boldsymbol{x}_i$ can appear in several $\boldsymbol{x}^{(c)}$)

- Iteratively
  - **Move** all irrelevant terms outside the innermost sum
  - **Perform** innermost sum, getting a new term
  - **Insert** the new term into the product

- Normalize the resulting function $g(\boldsymbol{x}_1)$ to get marginal density, i.e., $p(\boldsymbol{x}_1|\boldsymbol{z}) = \dfrac{g(\boldsymbol{x}_1)}{\sum_{\boldsymbol{x}_1} g(\boldsymbol{x}_1)}$

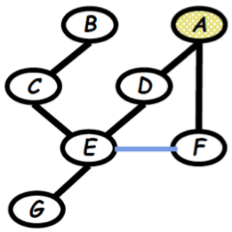
Example: Variable Elimination

- Task: Calculate $p(a|h)$ Elimination order: h, g, f, e, d, c, b



- Remaining factors:

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

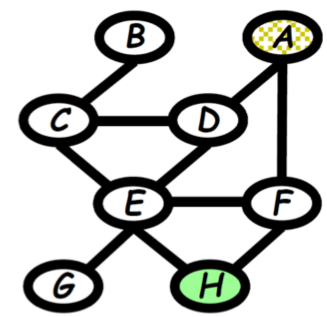


- Incorporate Observation h:

$$m_h(e,f) = p(h|e,f)$$

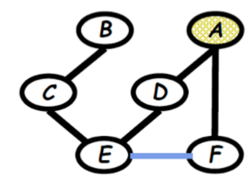
Example: Variable Elimination

- **Task:** Calculate $p(a|h)$ Elimination order: h, g, f, e, d, c, b



- Remaining factors:

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

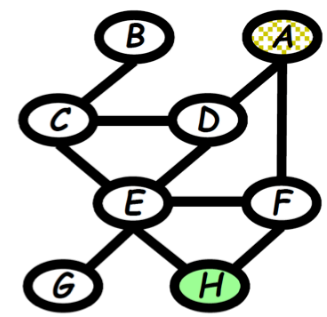


- Step 1: Eliminating g:

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

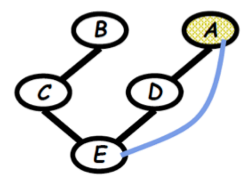
Example: Variable Elimination

- Task: Calculate $p(a|h)$ Elimination order: h, g, f, e, d, c, b



- Remaining factors:

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

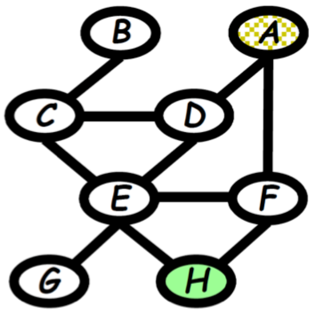


- Step 2: Eliminating f:

$$m_f(e,a) = \sum_f p(f|a)m_h(e,f)$$

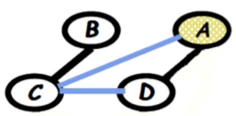
Example: Variable Elimination

- Task: Calculate $p(a|h)$ Elimination order: h, g, f, e, d, c, b



- Remaining factors:

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

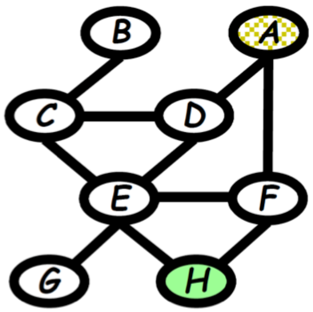


- Step 3: Eliminating e:

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

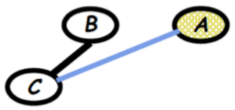
Example: Variable Elimination

- **Task:** Calculate $p(a|h)$ Elimination order: h, g, f, e, d, c, b



- Remaining factors:

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

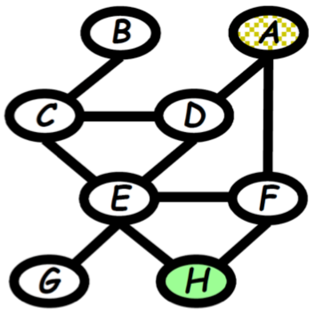


- Step 4: Eliminating d:

$$m_d(a,c) = \sum_d p(d|a)m_e(a,c,d)$$

Example: Variable Elimination

- **Task:** Calculate $p(a|h)$ **Elimination order:** h, g, f, e, d, c, b



- **Remaining factors:**

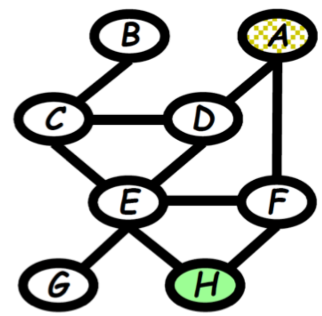
$$p(a)p(b)p(c|b)p(d|a)p(e|c,d)p(f|a)p(g|e)m_h(e,f)$$
$$\Rightarrow p(a)p(b)p(c|b)p(d|a)p(e|c,d)p(f|a)m_h(e,f)$$
$$\Rightarrow p(a)p(b)p(c|b)p(d|a)p(e|c,d)m_f(a,e)$$
$$\Rightarrow p(a)p(b)p(c|b)p(d|a)m_e(a,c,d)$$
$$\Rightarrow p(a)p(b)\underline{p(c|b)m_d(a,c)}$$

- Step 5: Eliminating c:

$$m_c(a,b) = \sum_c p(c|b)m_d(a,c)$$

Example: Variable Elimination

- **Task:** Calculate $p(a|h)$ Elimination order: h, g, f, e, d, c, b



- Remaining factors:

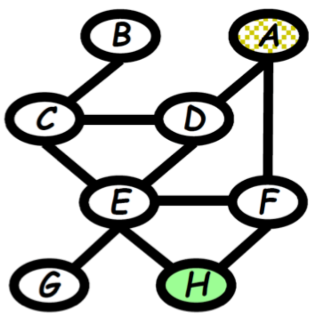
$$p(a)p(b)p(c|b)p(d|a)p(e|c,d)p(f|a)p(g|e)m_h(e,f)$$
$$\Rightarrow p(a)p(b)p(c|b)p(d|a)p(e|c,d)p(f|a)m_h(e,f)$$
$$\Rightarrow p(a)p(b)p(c|b)p(d|a)p(e|c,d)m_f(a,e)$$
$$\Rightarrow p(a)p(b)p(c|b)p(d|a)m_e(a,c,d)$$
$$\Rightarrow p(a)p(b)p(c|b)m_d(a,c)$$
$$\Rightarrow p(a)\underline{p(b)m_c(a,b)}$$

- Step 6: Eliminating b:

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

Example: Variable Elimination

- Task: Calculate $p(a|h)$ Elimination order: h, g, f, e, d, c, b



- Remaining factors:

$$
\begin{aligned}
&p(a)p(b)p(c|b)p(d|a)p(e|c,d)p(f|a)p(g|e)m_h(e,f)\\
&\quad \Rightarrow p(a)p(b)p(c|b)p(d|a)p(e|c,d)p(f|a)m_h(e,f)\\
&\quad \Rightarrow p(a)p(b)p(c|b)p(d|a)p(e|c,d)m_f(a,e)\\
&\quad \Rightarrow p(a)p(b)p(c|b)p(d|a)m_e(a,c,d)\\
&\quad \Rightarrow p(a)p(b)p(c|b)m_d(a,c)\\
&\quad \Rightarrow p(a)p(b)m_c(a,b)\\
&\quad \Rightarrow p(a)m_b(a)
\end{aligned}
$$

- Step 7: Normalization:

$$p(a|h) = \frac{p(a)m_b(a)}{\sum_{a'} p(a=a')m_b(a=a')}$$

Complexity of Variable Elimination

- Suppose in one elimination step we compute

$$m_{\boldsymbol{x}}(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) = \sum_{\boldsymbol{x}} g(\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) \qquad g(\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) = \prod_{c=1}^{k} \psi_c(\boldsymbol{x}, \boldsymbol{y}^{(c)})$$

where $\boldsymbol{x} \in \mathcal{X}, L = |\mathcal{X}|$ and $\boldsymbol{y}_i \in \mathcal{Y}_i, L_i = |\mathcal{Y}_i|$

- This requires

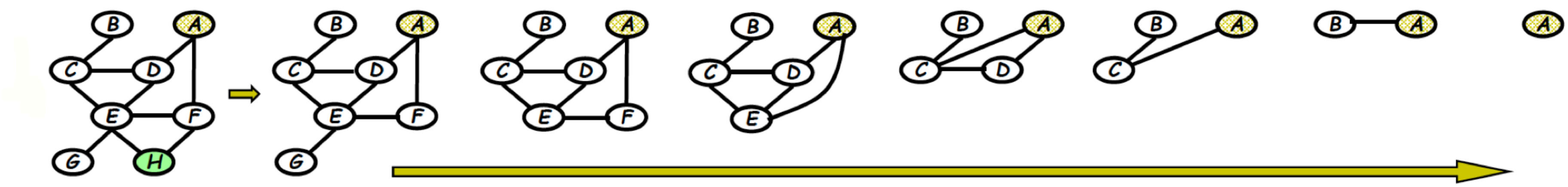
$k\,L \prod_{i=1}^{n} L_i$ multiplications (for each value of $\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ we do k multiplications)

$L \prod_{i=1}^{n} L_i$ additions (for each value of $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ we do L additions)

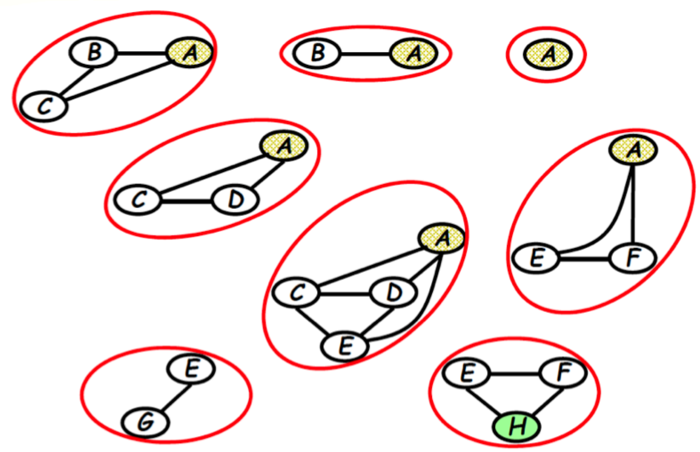
- Complexity is exponential in the number of variables in intermediate factors

Understanding Variable Elimination

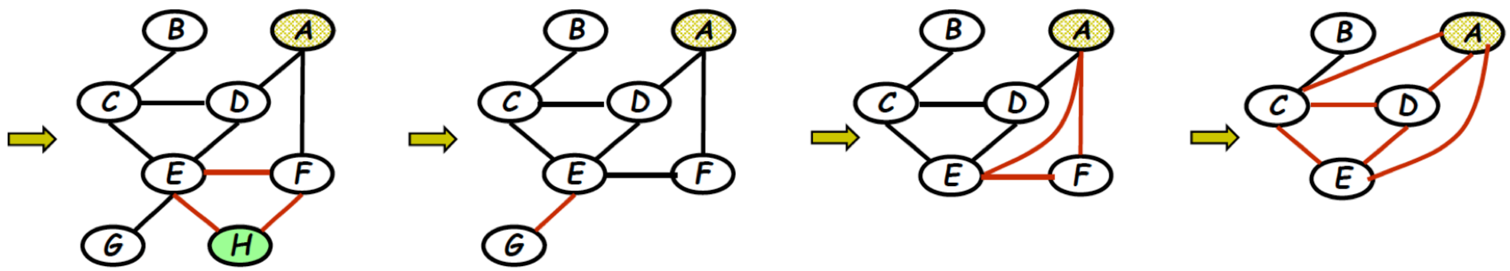
- A graph elimination algorithm



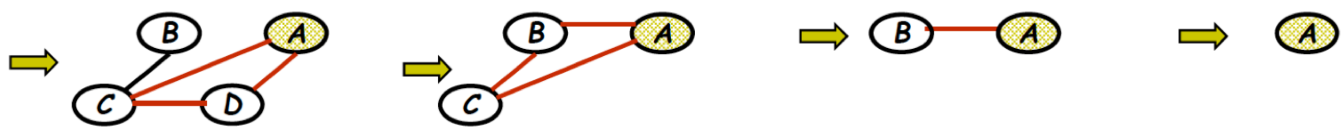
- Intermediate terms correspond to the cliques resulted from elimination



Elimination Cliques



$m_h(e, f)$ $m_g(e)$ $m_f(e, a)$ $m_e(a, c, d)$

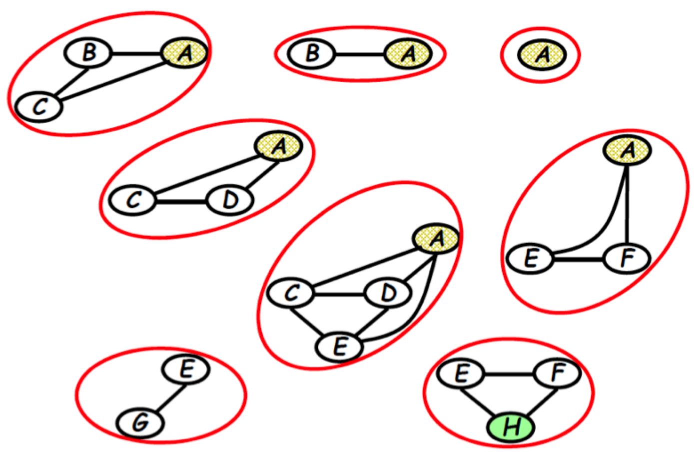


$m_d(a, c)$ $m_c(a, b)$ $m_b(a)$

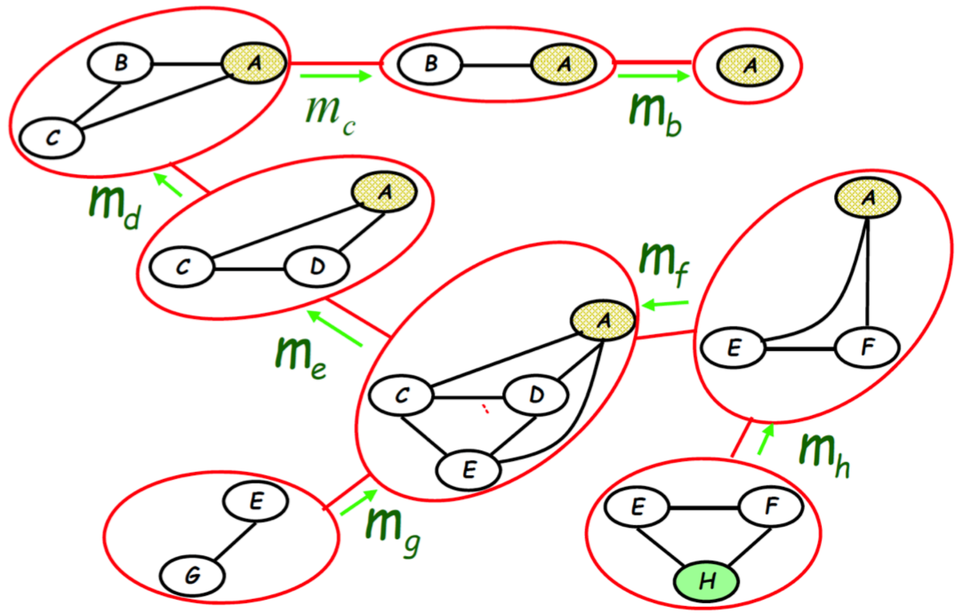
Graph Elimination

- **Graph Elimination:** Variable elimination can be interpreted as local operations performed on the graph where
  - every summation removed a node from the graph and connected its neighbors
  - the resulting intermediate terms correspond to ``elimination cliques''

$$
\begin{aligned}
& p(a)p(b)p(c|b)p(d|a)p(e|c,d)p(f|a)p(g|e)m_h(e,f) \\
& \quad \Rightarrow p(a)p(b)p(c|b)p(d|a)p(e|c,d)p(f|a)m_h(e,f) \\
& \quad \Rightarrow p(a)p(b)p(c|b)p(d|a)p(e|c,d)m_f(a,e) \\
& \quad \Rightarrow p(a)p(b)p(c|b)p(d|a)m_e(a,c,d) \\
& \quad \Rightarrow p(a)p(b)p(c|b)m_d(a,c) \\
& \quad \Rightarrow p(a)p(b)m_c(a,b) \\
& \quad \Rightarrow p(a)m_b(a)
\end{aligned}
$$



Clique Tree and ``Message Passing''



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

Complexity

- The total computational complexity is determined by the size of the largest elimination clique:
  - The largest elimination clique can be determined by purely graph-theoretic reasoning
  - ``Good'' elimination orderings lead to small cliques and hence to a reduced computational complexity
  - Finding the best elimination ordering for a graph is an NP-hard problem
  - For many graphs ``obvious'' optimal or near-optimal elimination orders can be foun

Example

- **Star:**

- **Tree:**

Summary

- The graph elimination algorithm is based on the sum-product operation which is the key algorithmic operation underlying estimation

- The computational complexity of variable elimination algorithm
  - is reduced from an exponential scaling in the total number of nodes to an exponential scaling in the number of nodes of the largest elimination clique
  - depends on the elimination order
  - can be analyzed by purely graph-theoretic considerations