

ECE 175B: Probabilistic Reasoning and Graphical Models

Lecture 15: Inference on General Graphs

Florian Meyer

Electrical and Computer Engineering Department
University of California San Diego

UCSanDiego
JACOBS SCHOOL OF ENGINEERING

0

Complexity of Estimation

- **Theorem:** Computing $p(\mathbf{x}_1|\mathbf{y}) = \sum_{\mathbf{x}_{\sim 1}} p(\mathbf{x}_1, \mathbf{x}_{\sim 1}|\mathbf{y})$ is NP-hard
- Hardness implies that we cannot find a general procedure that works efficiently for arbitrary distributions $p(\mathbf{x}|\mathbf{y})$
- We can develop **provable scalable estimation methods** if $p(\mathbf{x}|\mathbf{y})$ belongs to particular families of graphical models

1

1

Elimination on Chains



- Rearranging the terms ...

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

2

2

Elimination on Chains



- Now we can perform the innermost summation

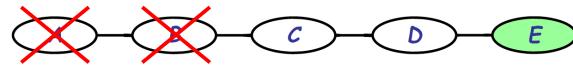
$$\begin{aligned}
 p(e) &= \sum_d \sum_c \sum_b p(c|b)p(d|c)p(e|d) \sum_a p(a)p(b|a) \\
 &= \sum_d \sum_c \sum_b p(c|b)p(d|c)p(e|d)p(b)
 \end{aligned}$$

- This summation “eliminates” one variable from our summation argument at a “local cost”

3

3

Elimination on Chains



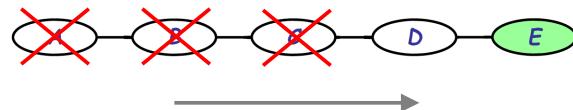
- Rearranging and then summing again, we get

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

4

4

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 $a, b, \dots \in \{1, 2, \dots, k\}$
 - Each elimination step costs $2k^2$ operations; the total complexity scales as $\mathcal{O}(nk^2)$
 - Naive evaluation that sums over joint values of $n - 1$ variables scale as $\mathcal{O}(k^n)$
- By **exploiting the structure of the graph** computational complexity is strongly reduced

5

5

General Chain Factorization



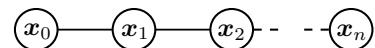
- Rearranging the terms ...

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

6

6

Example: State Space Model



- The **joint pdf** is given by

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

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

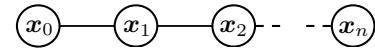
- Calculation of the **marginal posterior pdf**

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

7

7

Example: State Space Model



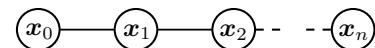
- Calculation of the **marginal posterior pdf**

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

8

8

Example: State Space Model



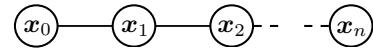
- Calculation of the **marginal posterior pdf**

$$\begin{aligned} f(\mathbf{x}_n | \mathbf{y}_{1:n}) &= \frac{1}{Z} \int \cdots \int \int f(\mathbf{y}_1 | \mathbf{x}_1) f(\mathbf{x}_2 | \mathbf{x}_1) \dots f(\mathbf{x}_n | \mathbf{x}_{n-1}) f(\mathbf{y}_n | \mathbf{x}_n) f(\mathbf{x}_1) d\mathbf{x}_1 d\mathbf{x}_2 \dots d\mathbf{x}_{n-1} \\ &= \frac{1}{Z} \int \cdots \int f(\mathbf{y}_2 | \mathbf{x}_2) \dots f(\mathbf{x}_n | \mathbf{x}_{n-1}) f(\mathbf{y}_n | \mathbf{x}_n) \underbrace{\int f(\mathbf{x}_2 | \mathbf{x}_1) f(\mathbf{y}_1 | \mathbf{x}_1) f(\mathbf{x}_1) d\mathbf{x}_1 d\mathbf{x}_2 \dots d\mathbf{x}_{n-1}}_{\propto f(\mathbf{x}_2 | \mathbf{y}_1)} \end{aligned}$$

9

9

Example: State Space Model



- Calculation of the **marginal posterior pdf**

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

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

Computational complexity is reduced from exponential to linear in time n

10

10

The Sum-Product Operation

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

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

discrete
random variables

$$\int \prod_{\phi \in \mathcal{F}} \phi d\mathbf{x}$$

continuous
random variables

where \mathcal{F} is a set of factors

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

11

11

Variable Elimination on General Graphical Models

- Write marginalization task in the form

$$p(\mathbf{x}_1 | \mathbf{z}) \propto \sum_{\mathbf{x}_n} \cdots \sum_{\mathbf{x}_3} \sum_{\mathbf{x}_2} \prod_{c \in \mathcal{C}} \psi_c(\mathbf{x}^{(c)})$$

elimination order

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

- 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 $g(\mathbf{x}_1)$ to get marginal density, i.e., $p(\mathbf{x}_1 | \mathbf{z}) = \frac{g(\mathbf{x}_1)}{\sum_{\mathbf{x}_1} g(\mathbf{x}_1)}$

12

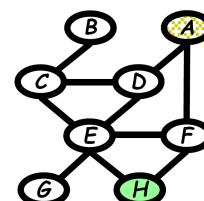
12

Example: Variable Elimination

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

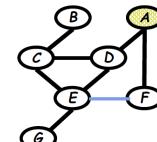
- Remaining factors:

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



- Incorporate Observation h :

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



13

13

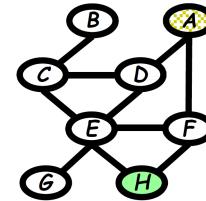
Example: Variable Elimination

- Task: Calculate $p(a|h)$

Elimination order: h, g, f, e, d, c, b

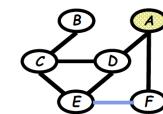
- Remaining factors:

$$p(a)p(b)p(c|b)p(d|a)p(e|c, d)p(f|a)p(g|e)m_h(e, f)$$



- Step 1: Eliminating g :

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



14

14

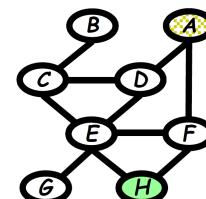
Example: Variable Elimination

- Task: Calculate $p(a|h)$

Elimination order: h, g, f, e, d, c, b

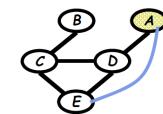
- Remaining factors:

$$\begin{aligned} & p(a)p(b)p(c|b)p(d|a)p(e|c, d)p(f|a)p(g|e)m_h(e, f) \\ & \Rightarrow p(a)p(b)p(c|b)p(d|a)p(e|c, d)p(f|a)m_h(e, f) \end{aligned}$$



- Step 2: Eliminating f :

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



15

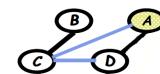
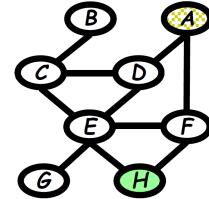
15

Example: Variable Elimination

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

- Remaining factors:

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



- Step 3: Eliminating e :

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

16

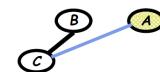
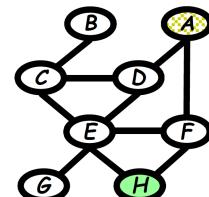
16

Example: Variable Elimination

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

- Remaining factors:

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



- Step 4: Eliminating d :

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

17

17

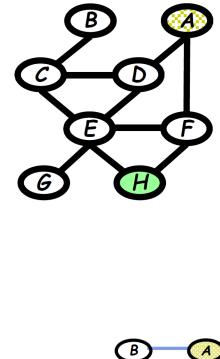
Example: Variable Elimination

- Task: Calculate $p(a|h)$

Elimination order: h, g, f, e, d, c, b

- Remaining factors:

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



- Step 5: Eliminating c :

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

18

18

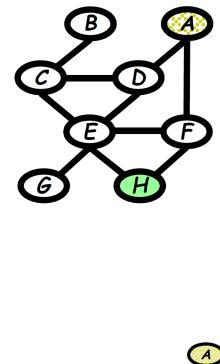
Example: Variable Elimination

- Task: Calculate $p(a|h)$

Elimination order: h, g, f, e, d, c, b

- Remaining factors:

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



- Step 6: Eliminating b :

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

19

19

Example: Variable Elimination

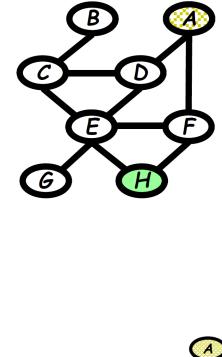
- Task: Calculate $p(a|h)$ Elimination order: h, g, f, e, d, c, b

- Remaining factors:

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

- Step 7: Normalization:

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



20

20

Complexity of Variable Elimination

- Suppose in one elimination step we compute

$$m_{\mathbf{x}}(\mathbf{y}_1, \dots, \mathbf{y}_n) = \sum_{\mathbf{x}} g(\mathbf{x}, \mathbf{y}_1, \dots, \mathbf{y}_n) \quad g(\mathbf{x}, \mathbf{y}_1, \dots, \mathbf{y}_n) = \prod_{c=1}^k \psi_c(\mathbf{x}, \mathbf{y}^{(c)})$$

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

- This requires

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

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

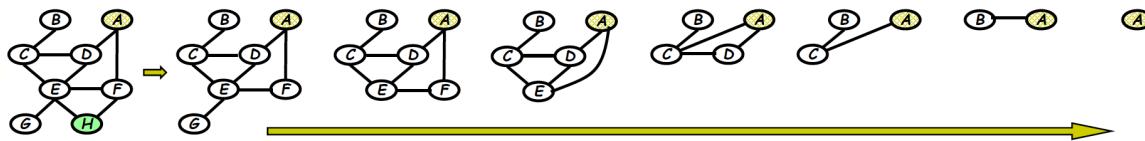
- Complexity is **exponential** in the number of variables in intermediate factors

21

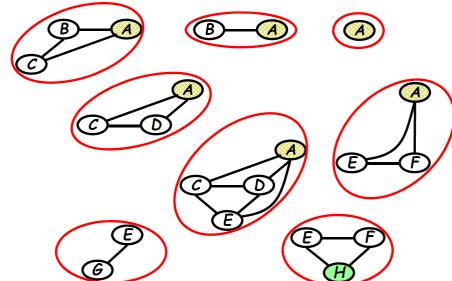
21

Understanding Variable Elimination

- A graph elimination algorithm



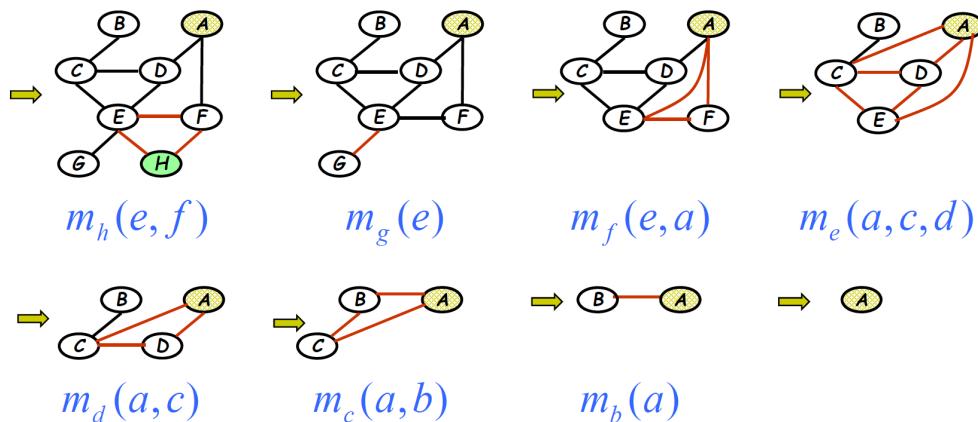
- Intermediate terms correspond to the cliques resulted from elimination



22

22

Elimination Cliques



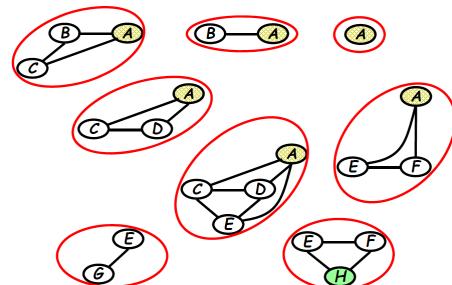
23

23

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”

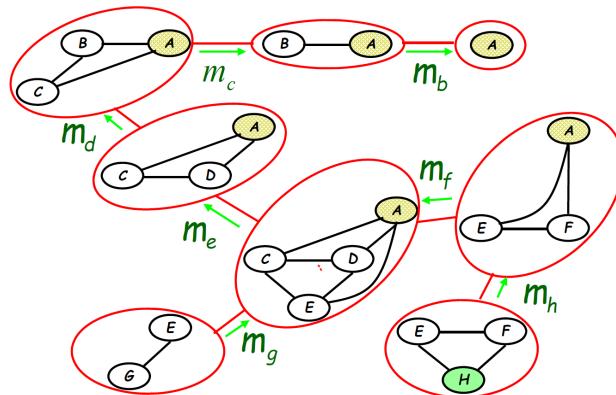
$$\begin{aligned}
 & p(a)p(b)p(c|b)p(d|a)p(e|c,d)p(f|a)p(g|e)m_h(e,f) \\
 & \Rightarrow p(a)p(b)p(c|b)p(d|a)p(e|c,d)p(f|a)m_h(e,f) \\
 & \Rightarrow p(a)p(b)p(c|b)p(d|a)p(e|c,d)m_f(a,e) \\
 & \Rightarrow p(a)p(b)p(c|b)p(d|a)m_e(a,c,d) \\
 & \Rightarrow p(a)p(b)p(c|b)m_d(a,c) \\
 & \Rightarrow p(a)p(b)m_c(a,b)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



24

24

Clique Tree and ‘‘Message Passing’’



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

25

25

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 order 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 found

26

26

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**

27

27