ECE/ML/CS/ISYE 8803

Exact Inference in Graphical Models

Module 4: Variable Elimination

Faramarz Fekri

Center for Signal and Information Processing

Overview

- Various form of Queries in inference
- Complexity of Inference
- Exact Inference Methods
  - Variable Elimination,
    - Examples (Chain networks, HMM, CRF)
  - Message passing (Sum-Product, belief propagation)
  - Example
  - Elimination order
  - Inference with evidence

Read Chapter 9 of K&F

Inference in Graphical Models

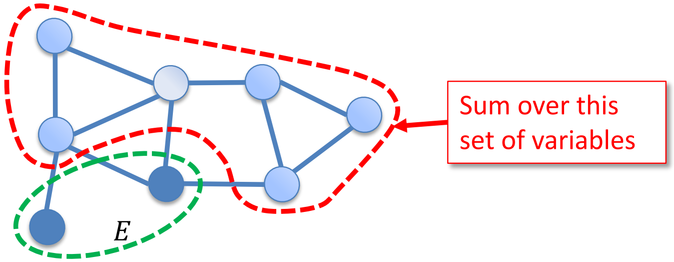
- Graphical models give compact representations of probabilistic distributions (i.e., reduces n-way tables to much smaller tables)
- Graphical model contains all information required to answer any query about the distribution
- Inference is the process of answering such queries
- The directionality of information flow between variables is not restricted by the directionality of the edges in a GM
- Inference combines evidence from all parts of the Graphical model

Queries (I):

1. Likelihood query:

- Compute probability (=likelihood) of the evidence
- Evidence: subset of variables **E** and an assignment **e**
- Task: compute P(**E**=**e**):

$$P(\boldsymbol{E}=\boldsymbol{e}) = \sum_{z\in \boldsymbol{U}-\boldsymbol{E}} P(\boldsymbol{Z}=\boldsymbol{z}, \boldsymbol{E}=\boldsymbol{e})$$

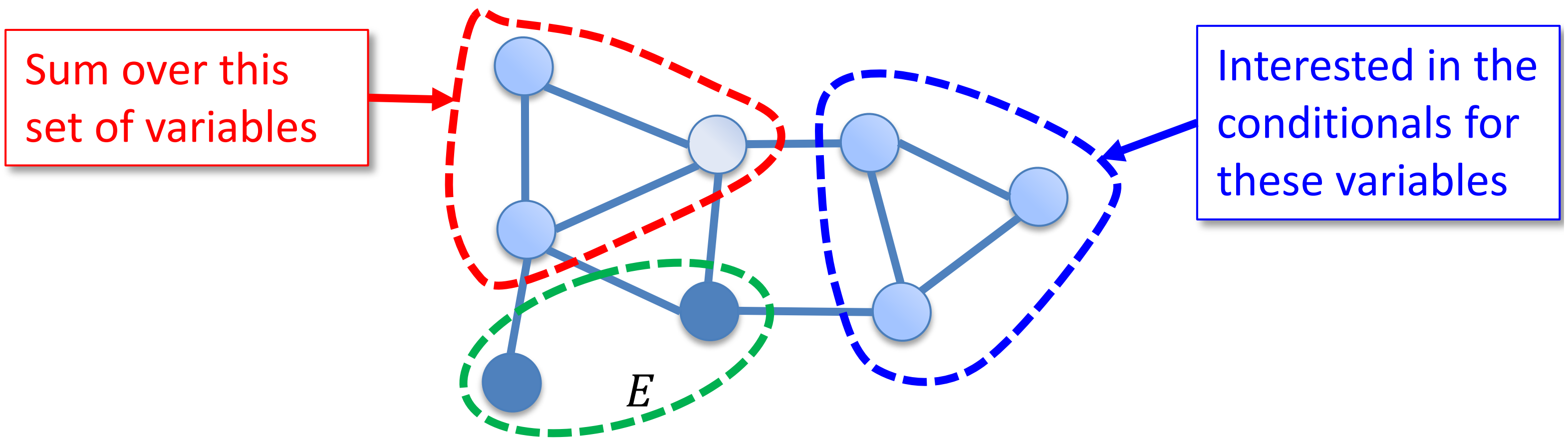


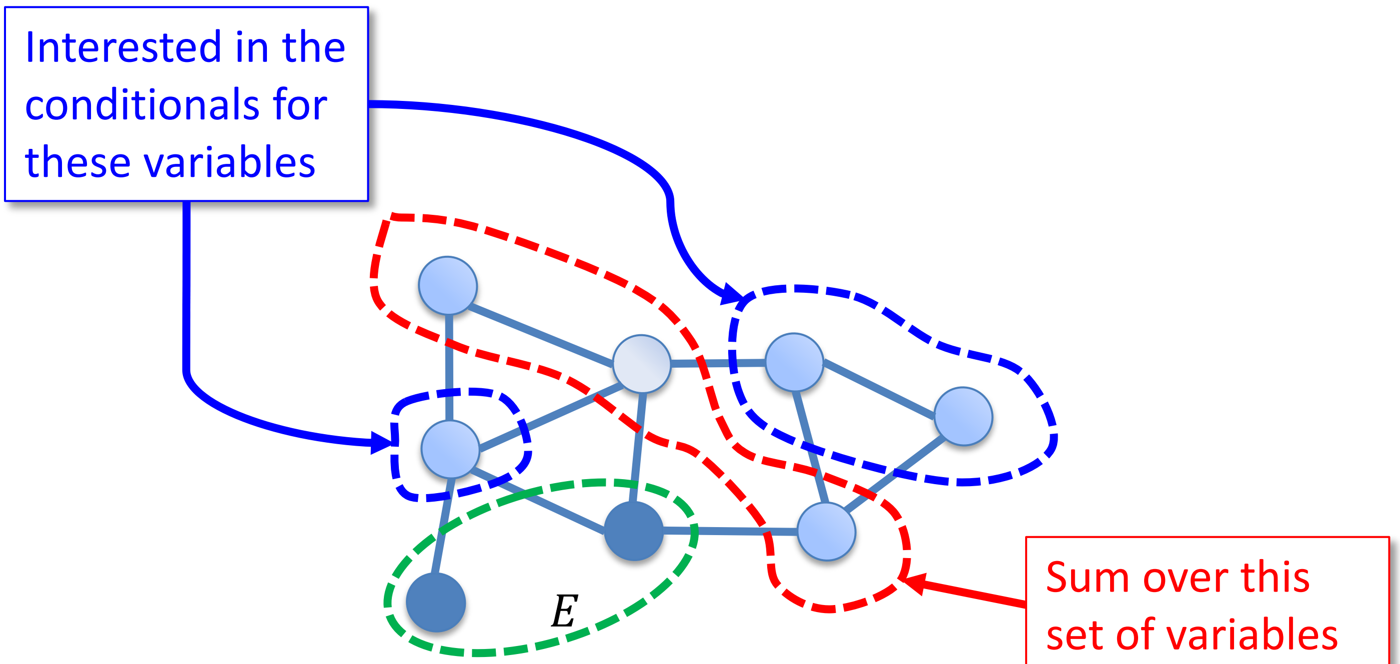
2. Conditional Probability Query:

- Evidence **E** and an assignment **e**
- Query: a subset of variables **Y**
- Task: compute P(**Y** | **E**=**e**) (*a posteriori* belief in **Y**, given evidence **e**)

$$P(\boldsymbol{Y}=\boldsymbol{y} \mid \boldsymbol{E}=\boldsymbol{e}) = \frac{P(\boldsymbol{Y}=\boldsymbol{y}, \boldsymbol{E}=\boldsymbol{e})}{P(\boldsymbol{E}=\boldsymbol{e})} = \frac{\sum_{\boldsymbol{w}\in \boldsymbol{U}-\boldsymbol{Y}-\boldsymbol{E}} P(\boldsymbol{W}=\boldsymbol{w}, \boldsymbol{Y}=\boldsymbol{y}, \boldsymbol{E}=\boldsymbol{e})}{\sum_{z\in \boldsymbol{U}-\boldsymbol{E}} P(\boldsymbol{Z}=\boldsymbol{z}, \boldsymbol{E}=\boldsymbol{e})}$$

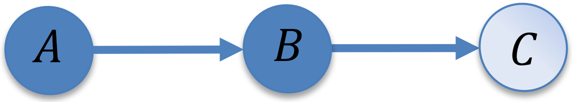
Illustrative Example of *a posteriori* belief



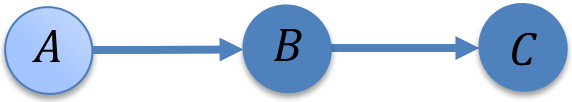


Applications of *a posteriori* belief

- 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 is an ancestor of the evidence



- Learning under partial observations (Fill in the unobserved)

Queries (II):

3. Maximum A Posteriori Assignment (MAP)

- Evidence: subset of variables **E** and an assignment **e**
- Query: a subset of variables **Y**

$$\text{MAP}(\mathbf{Y}|\mathbf{E}=\mathbf{e}) = \text{argmax}_y \, P(\mathbf{Y}=\mathbf{y} \mid \mathbf{E}=\boldsymbol{e})$$

- Task: compute

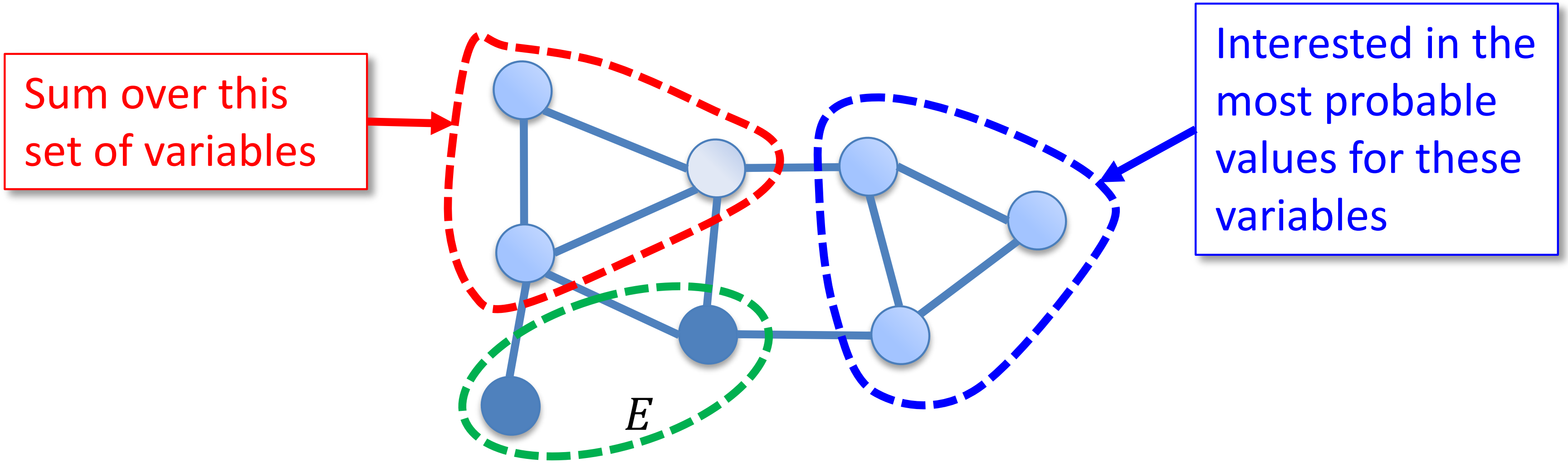
$$\text{MAP}(\mathbf{Y}=\mathbf{y} \mid \mathbf{e}) = \text{argmax}_{\mathbf{y}'} \sum_{\mathbf{w} \in \mathbf{U}-\mathbf{Y}-\mathbf{E}} P(\mathbf{W}=\mathbf{w}, \mathbf{Y}=\mathbf{y}' \mid \mathbf{E}=\mathbf{e})$$

- Note 1: there may be more than one possible solution
- Note 2: equivalent to computing

$$\text{argmax}_y \, P(\mathbf{Y}=\mathbf{y} \, , \, \mathbf{E}=\boldsymbol{e})$$

Why? $P(\mathbf{Y}=\mathbf{y} \mid \mathbf{E}=\boldsymbol{e}) = P(\mathbf{Y}=\mathbf{y} \, , \, \mathbf{E}=\boldsymbol{e}) \, / \, P(\mathbf{E}=\boldsymbol{e})$

Illustrative Example of *MAP*



Queries (III):

4. Most Probable Explanation/Assignment (MPE)

- Evidence: subset of variables **E** and an assignment **e**
- Query: **all other variables** **Y** (**Y**=**U**-**E**)

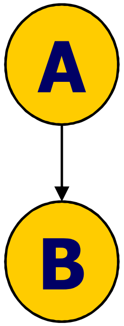
  Task: compute

$$\text{MPE}(\mathbf{Y}|\mathbf{E}=\mathbf{e}) = \text{argmax}_y \, P(\mathbf{Y}=\mathbf{y} \mid \mathbf{E}=\boldsymbol{e})$$

- Note: there may be more than one possible solution

Most Probable Assignment/Explanation (MPE)

- Note: We are searching for the most likely joint assignment to all variables
  - May be different than most likely assignment (MAP) of each variable.
  - Any example?



- Given $\mathbf{E} = \phi$
- $P(a^1) > P(a^0) \rightarrow MAP(A) = a^1$
- $MPE(A,B) = \{a^0, b^1\}$
  - $P(a^0, b^0) = 0.04$
  - $P(a^0, b^1) = 0.36$
  - $P(a^1, b^0) = 0.3$
  - $P(a^1, b^1) = 0.3$

P(A)

I	
a^0	a^1
0.4	0.6

P(B|A)

	B	
A	B^0	B^1
a^0	0.1	0.9
a^1	0.5	0.5

-Lee

Exact Inference

To answer any of those four queries:

- Naïve approach
  - Generate joint distribution
  - Depending on query, compute sum/max
    - Exponential blowup

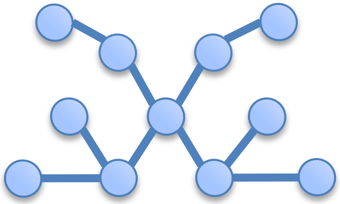
- Graphical Models can be used to answer any queries:
  - Exploit independencies for efficient inference

Complexity of Inference

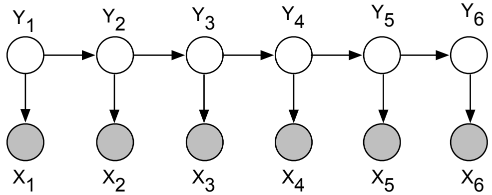
- Computing the a posteriori belief $P(X|e)$ in a GM is NP-hard in general (unless P=NP, there does *not* exist a more efficient algorithm)

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

eg. trees

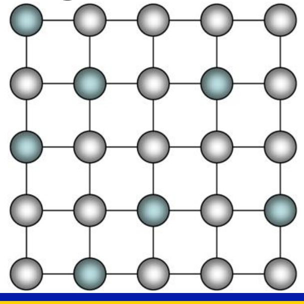


HMM



  - For some families of GMs, we need to design efficient approximate inference algorithms

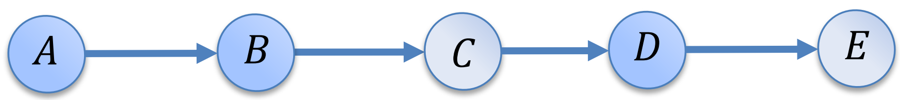
eg. grids



Exact Inference Methods

- Variable elimination algorithm (VE)
- Message-passing algorithm (sum-product, belief propagation algorithm) (MP/BP)
- The junction tree algorithm (JT)

Example:



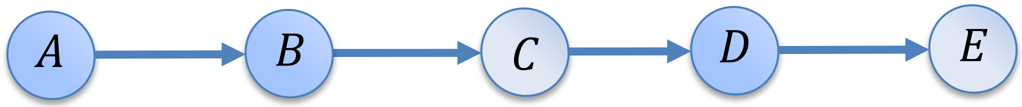
Using graphical models, we get

- $P(E) = \underbrace{\sum_d \sum_c \sum_b \sum_a}_{} P(a)P(b|a)P(c|b)P(d|c)P(E|d)$

Naïve summation needs to enumerate over an exponential number of terms

Exact Inference via Variable Elimination

- Inference in a directed chain



Rearranging terms and the summations

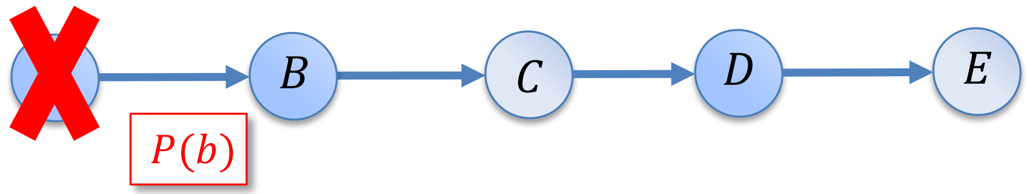
$$P(E) = \sum_d \sum_c \sum_b \sum_a P(a)P(b|a)P(c|b)P(d|c)P(E|d)$$

$$= \sum_d \sum_c \sum_b P(c|b)P(d|c)P(E|d)\left(\sum_a P(a)P(b|a)\right)$$

$$= \sum_d \sum_c \sum_b P(c|b)P(d|c)P(E|d)P(b)$$

The innermost summation eliminates one variable from our summation argument at a local cost.

VE in a Directed Chain



$$P(E) = \sum_d \sum_c \sum_b P(c|b)P(d|c)P(e|d)P(b)$$
$$= \sum_d \sum_c P(d|c)P(E|d)\left(\sum_b P(c|b)P(b)\right)$$
$$= \sum_d \sum_c P(d|c)P(E|d)P(c)$$

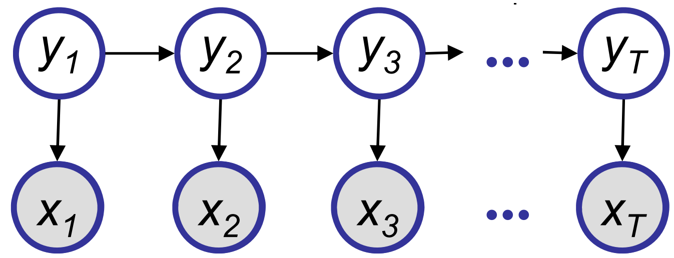
Eliminate nodes one by one all the way to the end $P(E) = \sum_d P(E|d)P(d)$

Computational Complexity for a chain of length *k*, and |***vol(X_i)***|=***n***

- Each step costs O(|Val(X_i)| * |Val(X_{i+1})|) operations: O(kn^2)
- Compare to naïve summation: O(n^k)

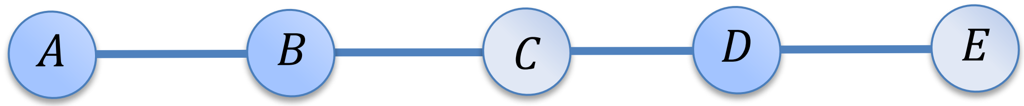
VE Example: HMM

Conditional probability:



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

VE in a Undirected Chain



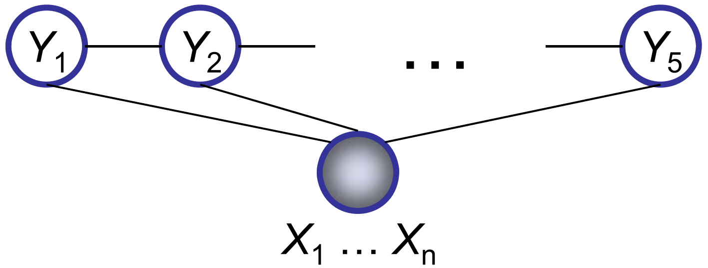
$$P(E) = \sum_d \sum_c \sum_b \sum_a \frac{1}{Z} \Psi(b,a)\Psi(c,b)\Psi(d,c)\Psi(E,d)$$

$$= \frac{1}{Z} \sum_d \sum_c \sum_b \Psi(c,b)\Psi(d,c)\Psi(E,d) \left(\sum_a \Psi(b,a) \right)$$

$$= \frac{1}{Z} \sum_d \sum_c \sum_b \Psi(c,b)\Psi(d,c)\Psi(E,d)\Psi(b)$$

Pushing summations = Dynamic programming

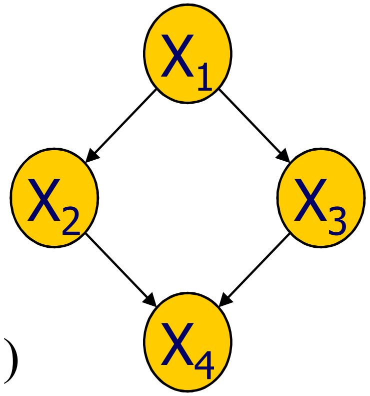
VE Example: Conditional Random Fields



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

VE in a Directed Loop

- Computing $P(X_4)$



$$
\begin{aligned}
P(X_4) &= \sum_{X_1}\sum_{X_2}\sum_{X_3} P(X_1, X_2, X_3, X_4) \\
&= \sum_{X_1}\sum_{X_2}\sum_{X_3} P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1)P(X_4 \mid X_2, X_3) \\
&= \sum_{X_2}\sum_{X_3} P(X_4 \mid X_2, X_3)\sum_{X_1} P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1) \\
&= \sum_{X_2}\sum_{X_3} P(X_4 \mid X_2, X_3)\phi(X_{2,3}) \\
&= \sum_{X_2} \phi(X_2, X_4) \\
&= \phi(X_4)
\end{aligned}
$$

- Differences
  - Summations are not "pushed in" as far as before.
  - The scope of ϕ includes two variables, not one.
- Depends on network structure

-Lee

Sum-Product Inference

- During inference, we try to compute an expression

  Sum-product form: $\sum_Z \prod_{\Psi \in \boldsymbol{\mathcal{F}}} \Psi$

  - $\boldsymbol{\mathcal{X}} = \{X_1, \dots, X_n\}$ the set of variables
  - $\boldsymbol{\mathcal{F}}$ a set of factors such that for each $\Psi \in \boldsymbol{\mathcal{F}}$, $Scope[\Psi] \in \boldsymbol{\mathcal{X}}$
  - $\boldsymbol{\mathcal{Y}} \subset \boldsymbol{\mathcal{X}}$ a set of query variables
  - $\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{Y}}$ the variables to eliminate
- The result of eliminating the variables in $\boldsymbol{\mathcal{Z}}$ is a factor

$$\tau(\boldsymbol{\mathcal{Y}}) = \sum_{Z} \prod_{\Psi \in \boldsymbol{\mathcal{F}}} \Psi$$

  - This factor does not necessarily correspond to any probability or conditional probability in the network.
  - $P(\boldsymbol{\mathcal{Y}}) = \frac{\tau(\boldsymbol{\mathcal{Y}})}{\sum \tau(\boldsymbol{\mathcal{Y}})}$

Sum-Product Variable Elimination in MN

- Algorithm
  - Given an ordering of variables $Z_1,\ldots,Z_n$,
  - Sum out the variables one at a time
  - When summing out each variable Z,
    - Multiply all the factors ϕ's that mention the variable, generating a product factor Ψ
    - Sum out the variable from the combined factor Ψ, generating a new factor *f* without the variable Z

Sum out

- Let $\mathbf{X}$ be a set of RVs, $Y \notin \mathbf{X}$ a RV, and $\phi(\mathbf{X},Y)$ a factor
- We define the factor marginalization of Y in $\mathbf{X}$ to be a factor ψ:Val($\mathbf{X}$) $\rightarrow \Re$ as $\psi(\mathbf{X}) = \Sigma_Y \phi(\mathbf{X},Y)$
- Also called summing out

Sum-Product Variable Elimination in BN

General Idea

- Write query in the form

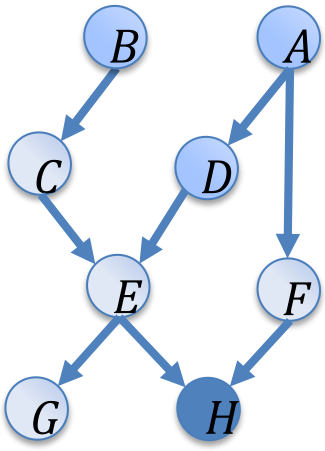
$$P(X_1, e) = \sum_{x_n} \dots \sum_{x_3} \sum_{x_2} \prod_i P\left(x_i \middle| Pa_{X_i}\right)$$

  - The sum is ordered to suggest an elimination order
  - We can pick any order we want, but some orderings introduce factors with much larger scope
- Then iteratively
  - Move all irrelevant terms outside of innermost sum
  - Perform innermost sum, getting a new term
  - Insert the new term into the product
- Finally renormalize

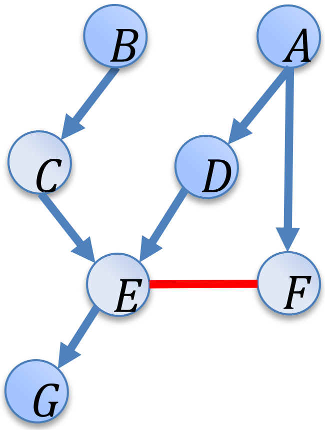
$$P(X_1|e) = \frac{\tau(X_1, e)}{\sum_{x_1} \tau(X_1, e)}$$

Example: Sum-Product Variable Elimination

- Query: $P(A|h)$, need to eliminate B, C, D, E, F, G, H

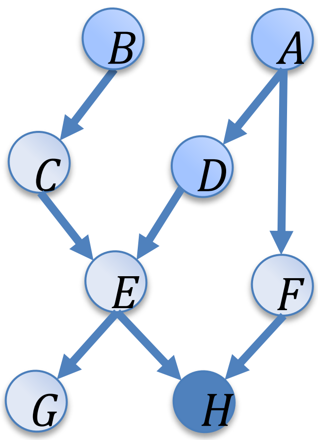


- Initial factors
  - $P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$
- Choose an elimination order: H, G, F, E, D, C, B $(<)$
- Step 1: Eliminate G
  - Conditioning (fix the evidence node on its observed value)
  - $m_h(e, f) = P(H = h|e, f)$



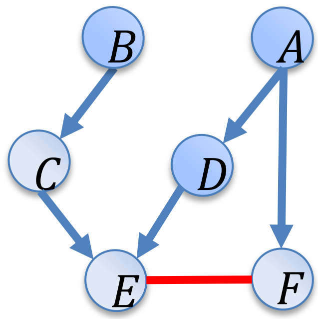
Example: Sum-Product (II)

- Query: $P(A|h)$, need to eliminate B, C, D, E, F, G



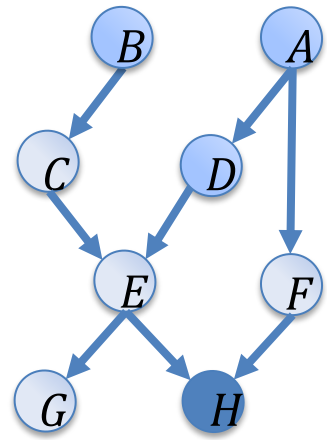
- Initial factors
  - $P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f)$

- Step 2: Eliminate G
  - Compute $m_g(e) = \sum_g P(g|e) \ = 1$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_g(e)m_h(e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_h(e,f)$



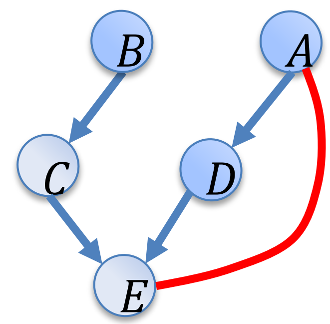
Example: Sum-Product (III)

- Query: $P(A|h)$, need to eliminate B, C, D, E, F



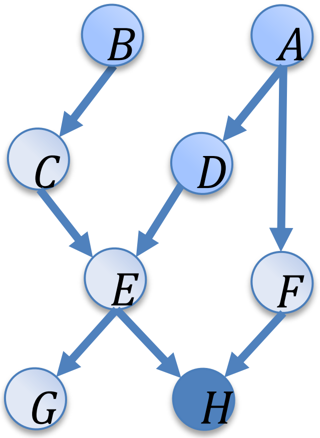
- Initial factors
  - $P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_h(e,f)$

- Step 3: Eliminate F
  - Compute $m_f(e,a) = \sum_f P(f|a)m_h(e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)m_f(e,a)$

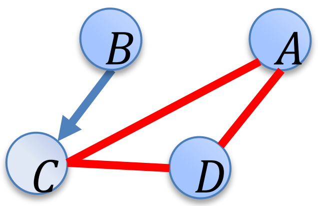


Example: Sum-Product (IV)

- Query: $P(A|h)$, need to eliminate B, C, D, E



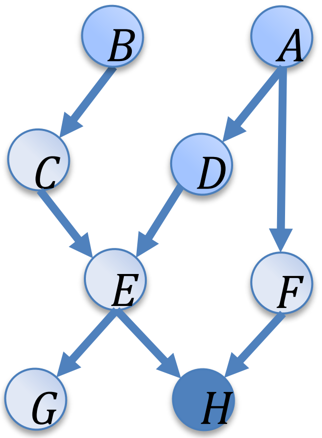
- Initial factors
  - $P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_h(e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)m_f(a,e)$



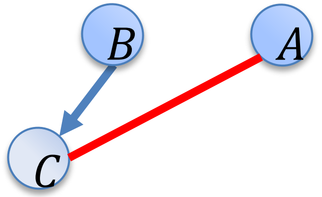
- Step 3: Eliminate E
  - Compute $m_e(a,c,d) = \sum_e P(e|c,d)m_f(a,e)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)m_e(a,c,d)$

Example: Sum-Product (V)

- Query: $P(A|h)$, need to eliminate B, C, D



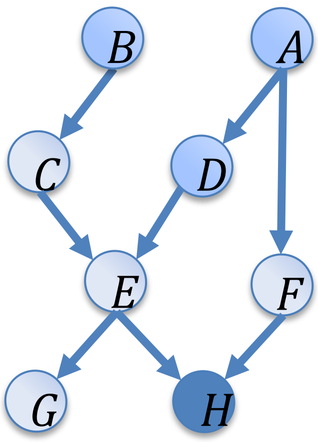
- Initial factors
  - $P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_h(e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)m_f(a,e)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)m_e(a,c,d)$



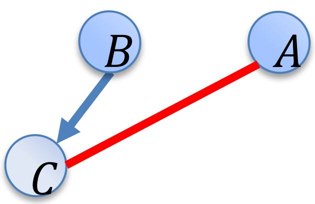
- Step 3: Eliminate D
  - Compute $m_d(a,c) = \sum_d P(d|a)m_e(a,c,d)$
  - $\Rightarrow P(a)P(b)P(c|b)m_d(a,c)$

Example: Sum-Product (VI)

- Query: $P(A|h)$, need to eliminate B, C



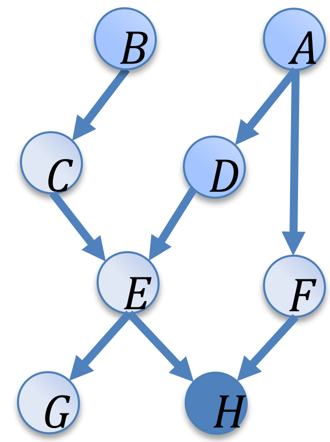
- Initial factors
  - $P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_h(e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)m_f(a,e)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)m_e(a,c,d)$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



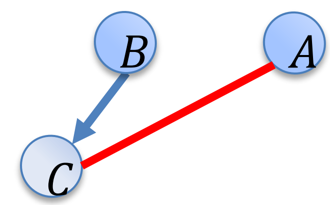
- Step 3: Eliminate C
  - Compute $m_c(a,b) = \sum_c P(c|b)m_d(a,c)$
  - $\Rightarrow P(a)P(b)m_c(a,b)$

Example: Sum-Product (VII)

- Query: $P(A|h)$, need to eliminate B



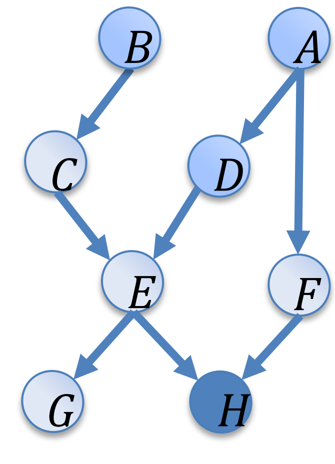
- Initial factors
  - $P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)P(h|e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)P(g|e)m_h(e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)P(f|a)m_h(e,f)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)P(e|c,d)m_f(a,e)$
  - $\Rightarrow P(a)P(b)P(c|b)P(d|a)m_e(a,c,d)$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



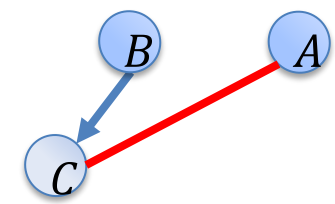
- Step 3: Eliminate B
  - Compute $m_b(a) = \sum_b P(b)m_c(a,b)$
  - $\Rightarrow P(a)m_b(a)$

Example: Sum-Product (VIII)

- Query: $P(A|h)$, need to renormalize over A



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



- Step 3: renormalize
  - $P(a,h) = P(a)m_b(a)$, compute $P(h) = \sum_a P(a)m_b(a)$
  - $\Rightarrow P(a|h) = \frac{P(a)m_b(a)}{\sum_a P(a)m_b(A)}$

Factor Marginalization

$\varphi(A,B,C)$

a^1	b^1	c^1	0.25
a^1	b^1	c^2	0.35
a^1	b^2	c^1	0.08
a^1	b^2	c^2	0.16
a^2	b^1	c^1	0.05
a^2	b^1	c^2	0.07
a^2	b^2	c^1	0
a^2	b^2	c^2	0
a^3	b^1	c^1	0.15
a^3	b^1	c^2	0.21
a^3	b^2	c^1	0.09
a^3	b^2	c^2	0.18

$$\varphi_1(A,C) = \sum_B \varphi(A,B,C)$$

a^1	c^1	0.33
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

Factor Product

$\varphi_1(A,B)$

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

$\varphi_2(B,C)$

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

$\varphi_3(A,B,C)$

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	$0.9 \cdot 0.2 = 0.18$

Factor Reduction

$\varphi(A,B,C)$

a^1	b^1	c^1	0.25
a^1	b^1	c^2	0.35
a^1	b^2	c^1	0.08
a^1	b^2	c^2	0.16
a^2	b^1	c^1	0.05
a^2	b^1	c^2	0.07
a^2	b^2	c^1	0
a^2	b^2	c^2	0
a^3	b^1	c^1	0.15
a^3	b^1	c^2	0.21
a^3	b^2	c^1	0.09
a^3	b^2	c^2	0.18

$C = c^1$ ⇒

$\varphi_1(A,B,c^1)$

a^1	b^1	c^1	0.25
a^1	b^2	c^1	0.08
a^2	b^1	c^1	0.05
a^2	b^2	c^1	0
a^3	b^1	c^1	0.15
a^3	b^2	c^1	0.09

Complexity of variable elimination

- Suppose in one elimination step we compute
  - $m_x(y_1, \dots, y_k) = \sum_x m'_x(x, y_1, \dots, y_k)$
  - $m'_x(x, y_1, \dots, y_k) = \prod_{i=1}^{k} m_i(x, y_{c_i})$

- This requires
  - $k * |Val(X)| * \prod_i |Val(Y_{c_i})|$ multiplications
    - For each value of $x, y_1, \dots, y_k$, we do k multiplications
  - $|Val(X)| * \prod_i |Val(Y_{c_i})|$ additions
    - For each value of $y_1, \dots, y_k$, we do $|Val(X)|$ additions

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

Inference with Evidence (I)

- Computing P(**Y**|**E**=**e**)
- Let **Y** be the query RVs
- Let **E** be the evidence RVs and **e** their assignment
- Let **Z** be all other RVs (U-**Y**-**E**)
- The general inference task is

$$\frac{\phi(\boldsymbol{Y},\boldsymbol{e})}{\phi(\boldsymbol{e})}=\frac{\sum_{\boldsymbol{Z}}\prod_{X\in U}\phi_{X|\boldsymbol{E}=\boldsymbol{e}}}{\sum_{\boldsymbol{Y},\boldsymbol{Z}}\prod_{X\in U}\phi_{X|\boldsymbol{E}=\boldsymbol{e}}}$$

Variable 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 $\mathscr{F}$ with the full the set of factors

Procedure Evidence $(\mathbf{E})$

1. **for** each $i \in I_E$,

   $\mathscr{F} = \mathscr{F} \cup \delta(E_i, e_i)$

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

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

1. **for** $i = 1, \ldots, k$

   $\mathscr{F} \leftarrow$ Sum-Product-Eliminate-Var$(\mathscr{F}, Z_i)$
2. $\phi^* \leftarrow \prod_{\phi \in \mathscr{F}} \phi$
3. **return** ϕ^*
4. Normalization (ϕ^*)

Procedure Normalization (ϕ^*)

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

Procedure Sum-Product-Eliminate-Var (

$\mathscr{F}$, // Set of factors

Z // Variable to be eliminated

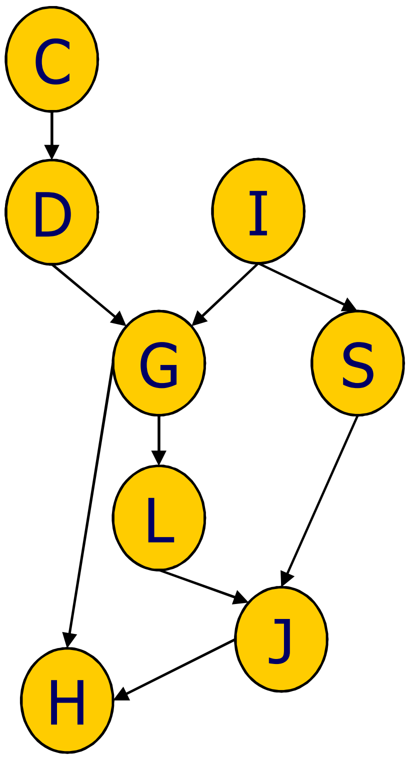
)

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

-Xing

Example: Inference with Multiple Evidence

- Goal: P(J|H=h,I=i)
- Eliminate: C,D,G,S,L
- Below, compute f(J,H=h,I=i)



$$
\begin{aligned}
P(J,h,i) &= \sum_{L,S,G,D,C} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,i)\phi_G(G,i,D)\phi_H(h,G,J)\phi_I(i)\phi_D(C,D)\phi_C(C) \\
&= \sum_{L,S,G,D} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,i)\phi_G(G,i,D)\phi_H(h,G,J)\phi_I(i)f_1(D) \\
&= \sum_{L,S,G} \phi_J(J,L,S)\phi_L(L,G)\phi_S(S,i)\phi_H(h,G,J)\phi_I(i)f_2(G,i) \\
&= \sum_{L,S} \phi_J(J,L,S)\phi_S(S,i)f_3(L,J) \\
&= \sum_{L} f_4(L,J) \\
&= f_5(J)
\end{aligned}
$$

- Differences
  - Less number of variables to be eliminated (H and I are excluded)
  - Scope of factors tend to be smaller.

-Lee