

CS 2750 Machine Learning
Lecture 16

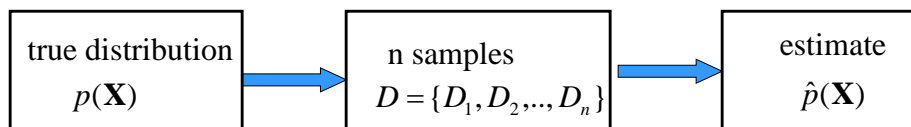
Bayesian belief networks II

Milos Hauskrecht
milos@pitt.edu
5329 Sennott Square

Density estimation

Data: $D = \{D_1, D_2, \dots, D_n\}$
 $D_i = \mathbf{x}_i$ a vector of attribute values

Objective: try to estimate the underlying true probability distribution over variables \mathbf{X} , $p(\mathbf{X})$, using examples in D



Standard (iid) assumptions: Samples

- are **independent** of each other
- come from the same **(i)dentical (d)istribution** (fixed $p(\mathbf{X})$)

Modeling complex distributions

Question: How to model and learn complex multivariate distributions $\hat{p}(\mathbf{X})$ with a large number of variables?

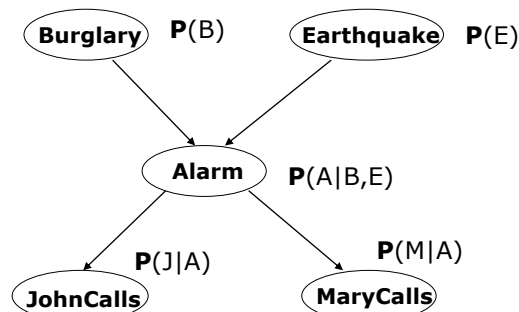
- Represent the full joint distribution over the variables more compactly with a **smaller number of parameters**.
- Take advantage of **conditional and marginal independences** among random variables

Bayesian belief network

1. Directed acyclic graph

- **Nodes** = random variables
Burglary, Earthquake, Alarm, Mary calls and John calls
- **Links** = direct (causal) dependencies between variables.

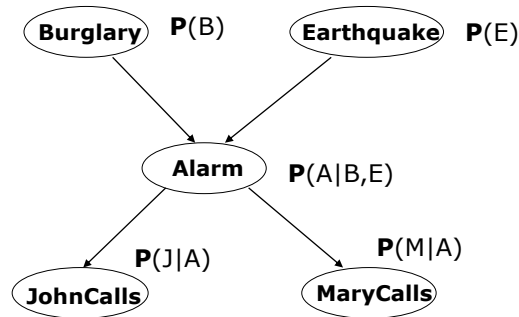
The chance of Alarm being is influenced by Earthquake,
The chance of John calling is affected by the Alarm



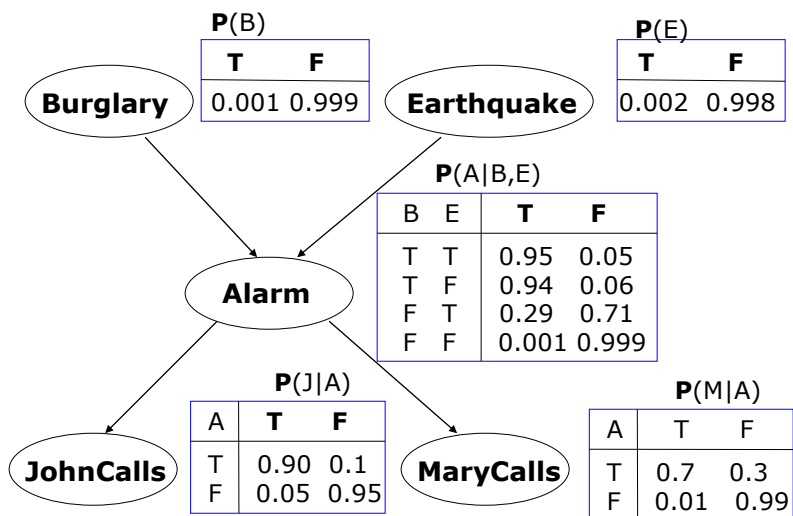
Bayesian belief network

2. Local conditional distributions

- relating variables and their parents



Bayesian belief network



Full joint distribution in BBNs

Full joint distribution is defined in terms of local conditional distributions (obtained via the chain rule):

$$\mathbf{P}(X_1, X_2, \dots, X_n) = \prod_{i=1, \dots, n} \mathbf{P}(X_i \mid pa(X_i))$$

Example:

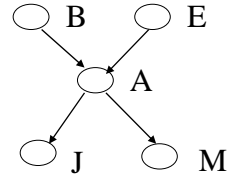
Assume the following assignment of values to random variables

$$B = T, E = T, A = T, J = T, M = F$$

Then its probability is:

$$P(B = T, E = T, A = T, J = T, M = F) =$$

$$P(B = T)P(E = T)P(A = T \mid B = T, E = T)P(J = T \mid A = T)P(M = F \mid A = T)$$



Parameter complexity problem

- In the BBN the **full joint distribution** is defined as:

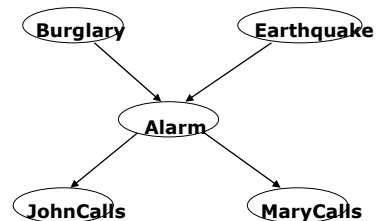
$$\mathbf{P}(X_1, X_2, \dots, X_n) = \prod_{i=1, \dots, n} \mathbf{P}(X_i \mid pa(X_i))$$

- What did we save?**

Alarm example: binary (True, False) variables

of parameters of the full joint:

?



Parameter complexity problem

- In the BBN the **full joint distribution** is defined as:

$$\mathbf{P}(X_1, X_2, \dots, X_n) = \prod_{i=1, \dots, n} \mathbf{P}(X_i \mid pa(X_i))$$

- What did we save?

Alarm example: binary (True, False) variables

of parameters of the full joint:

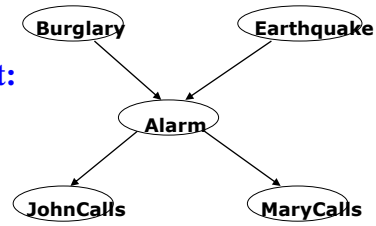
$$2^5 = 32$$

One parameter depends on the rest:

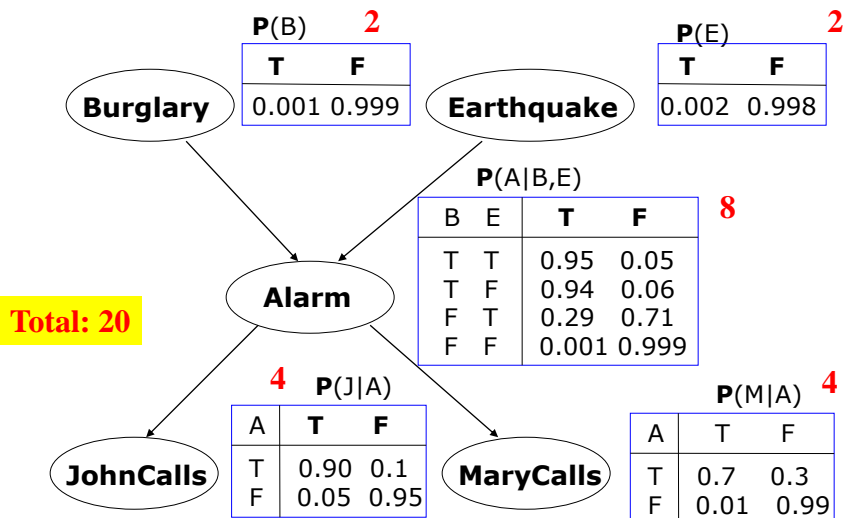
$$2^5 - 1 = 31$$

of parameters of the BBN:

?



Bayesian belief network: parameters count



Parameter complexity problem

- In the BBN the **full joint distribution** is defined as:

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1, \dots, n} P(X_i | pa(X_i))$$

- What did we save?**

Alarm example: 5 binary (True, False) variables

of parameters of the full joint:

$$2^5 = 32$$

One parameter depends on the rest:

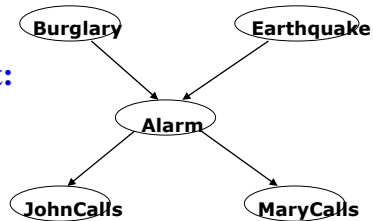
$$2^5 - 1 = 31$$

of parameters of the BBN:

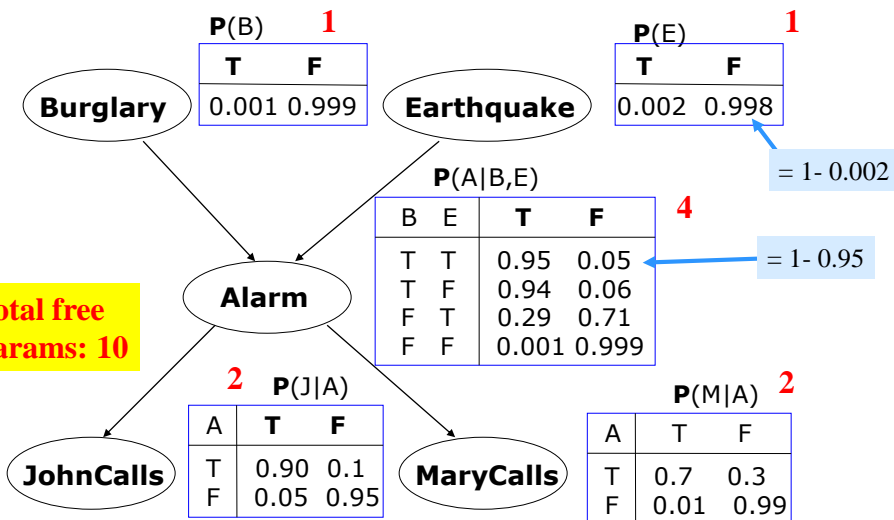
$$2^3 + 2(2^2) + 2(2) = 20$$

One parameter in every conditional depends on the rest:

?



Bayesian belief network: free parameters



Parameter complexity problem

- In the BBN the **full joint distribution** is defined as:

$$\mathbf{P}(X_1, X_2, \dots, X_n) = \prod_{i=1, \dots, n} \mathbf{P}(X_i \mid pa(X_i))$$

- What did we save?**

Alarm example: 5 binary (True, False) variables

of parameters of the full joint:

$$2^5 = 32$$

One parameter depends on the rest:

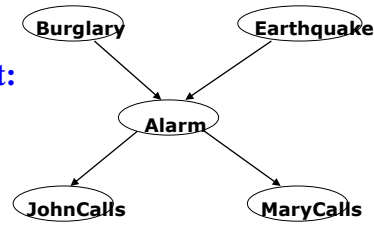
$$2^5 - 1 = 31$$

of parameters of the BBN:

$$2^3 + 2(2^2) + 2(2) = 20$$

One parameter in every conditional depends on the rest:

$$2^2 + 2(2) + 2(1) = 10$$



Naïve Bayes model

A special (simple) Bayesian belief network

- Defines a generative classifier model
- Model of $\mathbf{P}(\mathbf{x}, y) = \mathbf{P}(\mathbf{x} \mid y) \mathbf{P}(y)$

– Class variable y

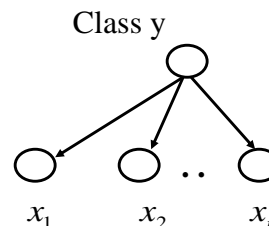
$$p(y)$$

– Attributes are independent given y

$$p(\mathbf{x} \mid y = i) = \prod_{j=1}^d p(x_j \mid y = i)$$

Learning:

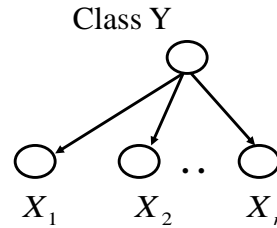
- Parameterize models of $p(y)$ and all $p(x_j \mid y=i)$
- ML estimates of the parameters



Naïve Bayes model

A **special (simple) Bayesian belief network**

- **Defines a generative classifier model**
- Model of $P(\mathbf{x}, y) = P(\mathbf{x} | y) P(y)$



Classification: given \mathbf{x} select the class

- Select the class with the maximum posterior
- Calculation of a posterior is an example of BBN inference

$$p(y = i | \mathbf{x}) = \frac{p(y = i) p(\mathbf{x} | y = i)}{\sum_{u=1}^k p(y = u) p(\mathbf{x} | y = u)} = \frac{p(y = i) \prod_{j=1}^d p(x_j | y = i)}{\sum_{u=1}^k p(y = u) \prod_{j=1}^d p(x_j | y = u)}$$

Remember: we can calculate the probabilities from the full joint

Learning of BBN

Learning.

- **Learning of parameters of conditional probabilities**
- **Learning of the network structure**

Variables:

- **Observable** – values present in every data sample
- **Hidden** – they values are never observed in data
- **Missing values** – values sometimes present, sometimes not

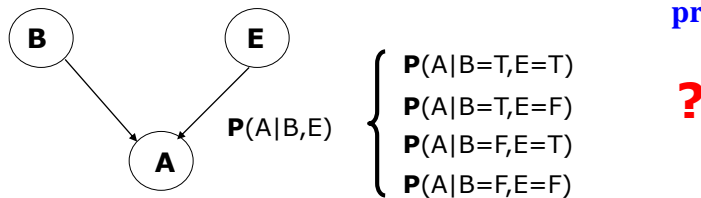
Next:

- **Learning of the parameters of BBN**
- **Values for all variables are observable**

Estimation of parameters of BBN

- **Idea:** decompose the estimation problem for the full joint over a large number of variables to a set of smaller estimation problems corresponding to local parent-variable conditionals.
- **Example:** Assume A,E,B are binary with *True, False* values

Learning of $P(A|B,E) = 4$ estimation problems



- **Assumption that enables the decomposition:** parameters of conditional distributions are independent

Estimates of parameters of BBN

- Two assumptions that permit the decomposition:
 - **Sample independence**

$$P(D | \Theta, \xi) = \prod_{u=1}^N P(D_u | \Theta, \xi)$$

- **Parameter independence**

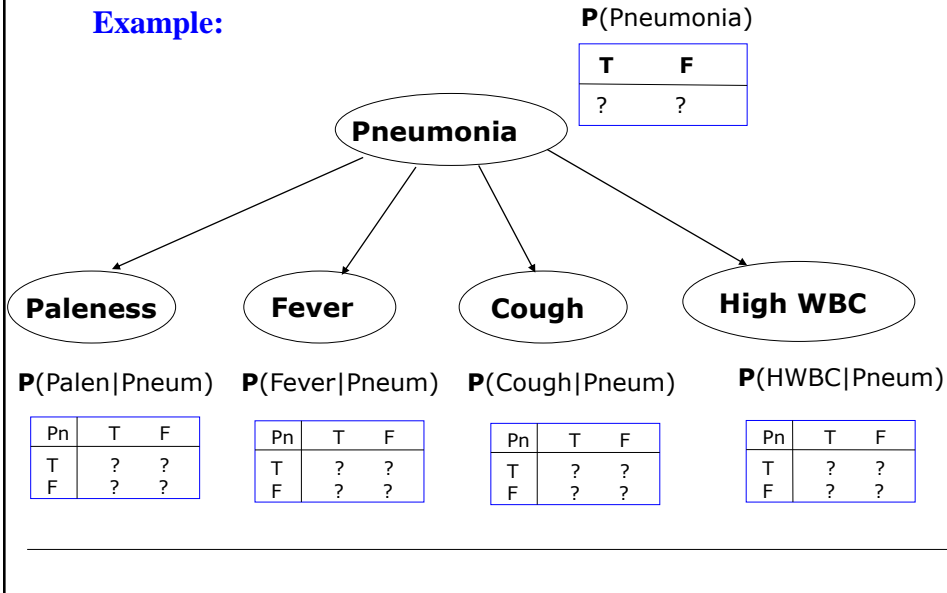
$$p(\Theta | D, \xi) = \prod_{i=1}^n \prod_{j=1}^{q_i} p(\theta_{ij} | D, \xi)$$

↖ # of nodes
 ↙ # of parents' values

Parameters of **each conditional** (one for every assignment of values to parent variables) can be learned independently

Learning of BBN parameters. Example.

Example:

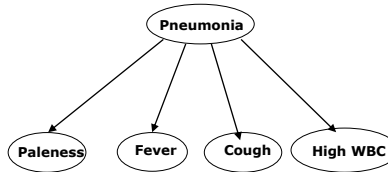


Learning of BBN parameters. Example.

Data D (different patient cases):

Pal Fev Cou HWB Pneu

T	T	T	T	F
T	F	F	F	F
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	F	T	F	F
F	F	F	F	F
T	T	F	F	F
T	T	T	T	T
F	T	F	T	T
T	F	F	T	F
F	T	F	F	F



Estimates of parameters of BBN

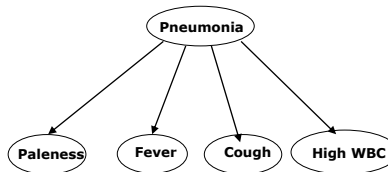
- Much like multiple **coin toss or roll of a dice** problems.
- A “smaller” learning problem corresponds to the learning of exactly one conditional distribution
- **Example:** $P(\text{Fever} \mid \text{Pneumonia} = T)$
- **Problem:** How to pick the data to learn?

Learning of BBN parameters. Example.

Learn: $P(\text{Fever} \mid \text{Pneumonia} = T)$

Step 1: Select data points with Pneumonia=T

Pal	Fev	Cou	HWB	Pneu
T	T	T	T	F
T	F	F	F	F
F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	F	T	F	F
F	F	F	F	F
T	T	F	F	F
T	T	T	T	T
F	T	F	T	T
T	F	F	T	F
F	T	F	F	F



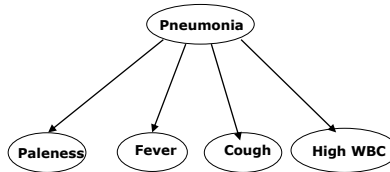
Learning of BBN parameters. Example.

Learn: $P(\text{Fever} \mid \text{Pneumonia} = T)$

Step 1: Ignore the rest

Pal Fev Cou HWB Pneu

F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	T	T	T	T
F	T	F	T	T



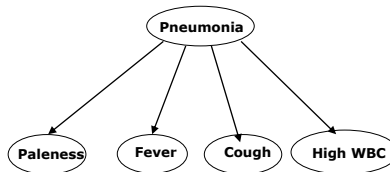
Learning of BBN parameters. Example.

Learn: $P(\text{Fever} \mid \text{Pneumonia} = T)$

Step 2: Select values of the random variable defining the distribution of Fever

Pal Fev Cou HWB Pneu

F	F	T	T	T
F	F	T	F	T
F	T	T	T	T
T	T	T	T	T
F	T	F	T	T



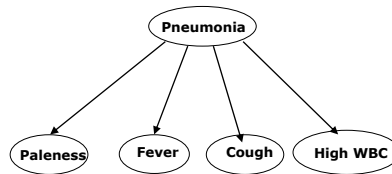
Learning of BBN parameters. Example.

Learn: $P(\text{Fever} \mid \text{Pneumonia} = T)$

Step 2: Ignore the rest

Fev

F
F
T
T
T



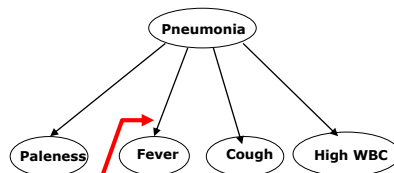
Learning of BBN parameters. Example.

Learn: $P(\text{Fever} \mid \text{Pneumonia} = T)$

Step 3a: Learning the ML estimate

Fev

F
F
T
T
T



$P(\text{Fever} \mid \text{Pneumonia} = T)$

	T	F
Pneum = T	0.6	0.4

Learning of BBN parameters. Bayesian learning.

Learn: $\mathbf{P}(Fever | Pneumonia = T)$

Step 3b: Learning the Bayesian posterior

Assume the prior

$$\theta_{Fever|Pneumonia=T} \sim Beta(3,4)$$

Fev

F

F

T

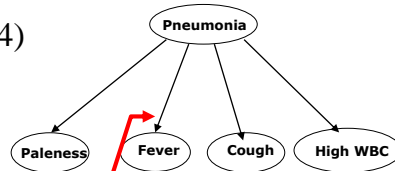
T

T

Posterior:

$$\theta_{Fever|Pneumonia=T} \sim Beta(6,6)$$

$$\theta_{Fever|Pneumonia=T}^{MAP} = \frac{6-1}{6+6-2} = 0.5$$



MAP estimates

	T	F
Pneum =T	0.5	0.5

Estimates of parameters of BBN

Much like multiple **coin toss or roll of a dice** problems.

- A “smaller” learning problem corresponds to the learning of exactly one conditional distribution

Example:

$$\mathbf{P}(Fever | Pneumonia = T)$$

Problem: How to pick the data to learn?

Answer:

1. Select data points with Pneumonia=T
(ignore the rest)
2. Focus on (select) only values of the random variable defining the distribution (Fever)
3. Learn the parameters of the local conditionals the same way as we learned the parameters of a biased coin or a die

Inference in Bayesian networks

- BBN models compactly the full joint distribution by taking advantage of existing independences between variables
- Simplifies the representation and learning of a model
- Can be used for solving various **inference tasks**:

- **Diagnostic task. (from effect to cause)**

$$\mathbf{P}(\text{Burglary} \mid \text{JohnCalls} = T)$$

- **Prediction task. (from cause to effect)**

$$\mathbf{P}(\text{JohnCalls} \mid \text{Burglary} = T)$$

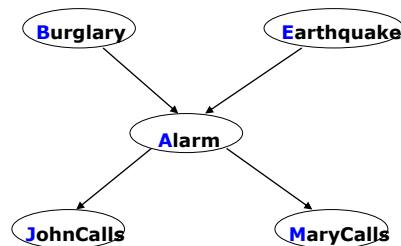
- **Other probabilistic queries** (queries on joint distributions).

$$\mathbf{P}(\text{Alarm})$$

- **Main question:** Can we take advantage of independences to construct special algorithms and speeding up the inference?
-

Inference in Bayesian network

- **Bad news:**
 - Exact inference problem in BBNs is NP-hard (Cooper)
 - Approximate inference is NP-hard (Dagum, Luby)
- **But** very often we can achieve significant improvements
- Assume our Alarm network



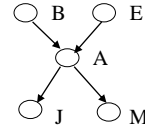
- Assume we want to compute: $P(J = T)$
-

Inference in Bayesian networks

Computing: $P(J = T)$

Approach 1. Blind approach.

- Sum out all un-instantiated variables from the full joint,
- express the joint distribution as a product of conditionals



$$\begin{aligned}
 P(J = T) &= \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(B = b, E = e, A = a, J = T, M = m) \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e)
 \end{aligned}$$

Computational cost:

Number of additions: ?

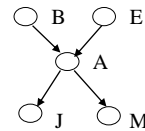
Number of products: ?

Inference in Bayesian networks

Computing: $P(J = T)$

Approach 1. Blind approach.

- Sum out all un-instantiated variables from the full joint,
- express the joint distribution as a product of conditionals



$$\begin{aligned}
 P(J = T) &= \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(B = b, E = e, A = a, J = T, M = m) \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e)
 \end{aligned}$$

Computational cost:

Number of additions: 15

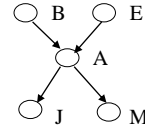
Number of products: ?

Inference in Bayesian networks

Computing: $P(J = T)$

Approach 1. Blind approach.

- Sum out all un-instantiated variables from the full joint,
- express the joint distribution as a product of conditionals



$$\begin{aligned} P(J = T) &= \\ &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(B = b, E = e, A = a, J = T, M = m) \\ &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \end{aligned}$$

Computational cost:

Number of additions: 15

Number of products: $16 * 4 = 64$

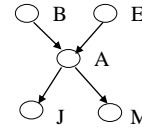
Inference in Bayesian networks

How to compute sums and products more efficiently?

$$\sum_x af(x) = a \sum_x f(x)$$

Inference in Bayesian networks

Approach 2. Interleave sums and products



- Combines sums and product in a smart way
(multiplications by constants can be taken out of the sum)

$$\begin{aligned}
 P(J = T) &= \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e)
 \end{aligned}$$

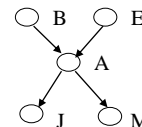
- Use variable e and its sum and

$$\sum_x a f(x) = a \sum_x f(x)$$

$$= \sum_{b \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(B = b) \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right]$$

Inference in Bayesian networks

Approach 2. Interleave sums and products



- Combines sums and product in a smart way
(multiplications by constants can be taken out of the sum)

$$\begin{aligned}
 P(J = T) &= \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \\
 &= \sum_{b \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(B = b) \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right]
 \end{aligned}$$

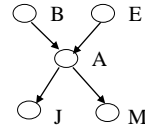
• • •

$$= \sum_{a \in T, F} P(J = T | A = a) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \left[\sum_{b \in T, F} P(B = b) \right] \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right]$$

Inference in Bayesian networks

Approach 2. Interleave sums and products

- Combines sums and product in a smart way
(multiplications by constants can be taken out of the sum)



$$= \sum_{a \in T, F} P(J = T | A = a) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \left[\sum_{b \in T, F} P(B = b) \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right] \right]$$

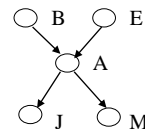
↓
 1

Computational cost:
 Number of additions: ?

Inference in Bayesian networks

Approach 2. Interleave sums and products

- Combines sums and product in a smart way
(multiplications by constants can be taken out of the sum)



$$= \sum_{a \in T, F} P(J = T | A = a) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \left[\sum_{b \in T, F} P(B = b) \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right] \right]$$

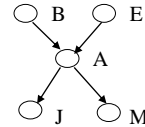
↘ ↓
 2*1

Computational cost:
 Number of additions: ?

Inference in Bayesian networks

Approach 2. Interleave sums and products

- Combines sums and product in a smart way
(multiplications by constants can be taken out of the sum)



$$= \sum_{a \in T, F} P(J = T | A = a) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \left[\sum_{b \in T, F} P(B = b) \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right] \right]$$

→ → → → $2 * 2 * 1$

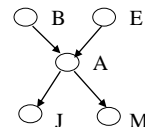
Computational cost:

Number of additions: ?

Inference in Bayesian networks

Approach 2. Interleave sums and products

- Combines sums and product in a smart way
(multiplications by constants can be taken out of the sum)



$$= \sum_{a \in T, F} P(J = T | A = a) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \left[\sum_{b \in T, F} P(B = b) \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right] \right]$$

→ → → → $2 * 1$ $2 * 2 * 1$

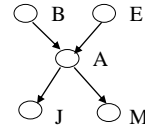
Computational cost:

Number of additions: ?

Inference in Bayesian networks

Approach 2. Interleave sums and products

- Combines sums and product in a smart way
(multiplications by constants can be taken out of the sum)



$$= \sum_{a \in T, F} P(J=T | A=a) \left[\sum_{m \in T, F} P(M=m | A=a) \right] \left[\sum_{b \in T, F} P(B=b) \left[\sum_{e \in T, F} P(A=a | B=b, E=e) P(E=e) \right] \right]$$

↓ 1
↓ 2*1
↓ 2*1
↓ 2*2*1

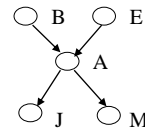
Computational cost:

Number of additions: ?

Inference in Bayesian networks

Approach 2. Interleave sums and products

- Combines sums and product in a smart way
(multiplications by constants can be taken out of the sum)



$$= \sum_{a \in T, F} P(J=T | A=a) \left[\sum_{m \in T, F} P(M=m | A=a) \right] \left[\sum_{b \in T, F} P(B=b) \left[\sum_{e \in T, F} P(A=a | B=b, E=e) P(E=e) \right] \right]$$

↓ 1
↓ 2*1
↓ 2*1
↓ 2*2*1

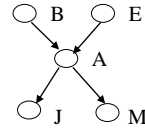
Computational cost:

Number of additions: $1+2*[1+1+2*1]=9$

Inference in Bayesian networks

Approach 2. Interleave sums and products

- Combines sums and product in a smart way
(multiplications by constants can be taken out of the sum)



$$= \sum_{a \in T, F} P(J = T | A = a) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \left[\sum_{b \in T, F} P(B = b) \right] \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right]$$

1

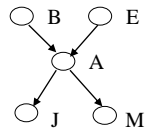
Computational cost:

Number of products: ?

Inference in Bayesian networks

Approach 2. Interleave sums and products

- Combines sums and product in a smart way
(multiplications by constants can be taken out of the sum)



$$= \sum_{a \in T, F} P(J = T | A = a) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \left[\sum_{b \in T, F} P(B = b) \right] \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right]$$

1

2*2 * 2*1

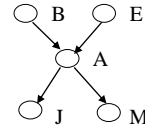
Computational cost:

Number of products: ?

Inference in Bayesian networks

Approach 2. Interleave sums and products

- Combines sums and product in a smart way
(multiplications by constants can be taken out of the sum)



$$= \sum_{a \in T, F} P(J = T | A = a) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \left[\sum_{b \in T, F} P(B = b) \right] \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right]$$

$2*2$ $2*2*1$ $2*2 * 2*1$

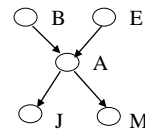
Computational cost:

Number of products: ?

Inference in Bayesian networks

Approach 2. Interleave sums and products

- Combines sums and product in a smart way
(multiplications by constants can be taken out of the sum)



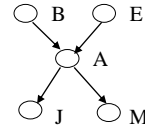
$$= \sum_{a \in T, F} P(J = T | A = a) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \left[\sum_{b \in T, F} P(B = b) \right] \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right]$$

$2*2$ $2*2*1$ $2*2 * 2*1$

Computational cost:

Number of products: $2*[2+2*(1+2*1)]=16$

Inference in Bayesian networks



Approach 2. Interleave sums and products

- Combines sums and product in a smart way
(multiplications by constants can be taken out of the sum)

$$\begin{aligned}
 P(J = T) &= \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(B = b) \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right] \\
 &= \sum_{a \in T, F} P(J = T | A = a) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \left[\sum_{b \in T, F} P(B = b) \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right] \right]
 \end{aligned}$$

Computational cost:

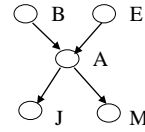
Number of additions: $1 + 2 * [1 + 1 + 2 * 1] = 9$

Number of products: $2 * [2 + 2 * (1 + 2 * 1)] = 16$

Inference in Bayesian network

- **Exact inference algorithms:**
 - Variable elimination
 - Recursive decomposition (Cooper, Darwiche)
 - Symbolic inference (D'Ambrosio)
 - Belief propagation algorithm (Pearl)
 - Clustering and joint tree approach (Lauritzen, Spiegelhalter)
 - Arc reversal (Olmsted, Schachter)
- **Approximate inference algorithms:**
 - Monte Carlo methods:
 - Forward sampling, Likelihood sampling
 - Variational methods

Variable elimination



- **Variable elimination:**

- Interleave sum and products one variable at the time during inference

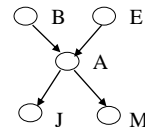
$$\sum_x af(x) = a \sum_x f(x)$$

- E.g. Query $P(J = T)$ requires to eliminate A,B,E,M and this can be done in different order

$$P(J = T) =$$

$$= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e)$$

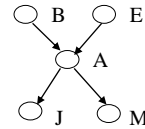
Variable elimination



Assume order: M, E, B, A to calculate $P(J = T)$

$$= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e)$$

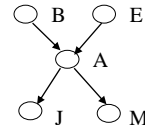
Variable elimination



Assume order: **M, E, B, A** to calculate $P(J = T)$

$$\begin{aligned}
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \left[\sum_{m \in T, F} P(M = m | A = a) \right]
 \end{aligned}$$

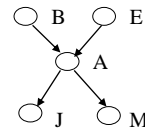
Variable elimination



Assume order: **M, E, B, A** to calculate $P(J = T)$

$$\begin{aligned}
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \mathbf{1}
 \end{aligned}$$

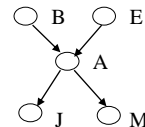
Variable elimination



Assume order: M, E, B, A to calculate $P(J = T)$

$$\begin{aligned}
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \cdot 1 \\
 &= \sum_{a \in T, F} \sum_{b \in T, F} P(J = T | A = a) P(B = b) \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right]
 \end{aligned}$$

Variable elimination



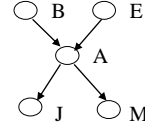
Assume order: M, E, B, A to calculate $P(J = T)$

$$\begin{aligned}
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \cdot 1 \\
 &= \sum_{a \in T, F} \sum_{b \in T, F} P(J = T | A = a) P(B = b) \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right]
 \end{aligned}$$

$\tau_1(A = a, B = b)$

$$\tau_1(A = a, B = b) = \begin{array}{l} A = T \quad A = F \\ \begin{array}{|c|c|} \hline B = T & \sum_{e \in T, F} P(A = F | B = T, E = e) P(E = e) \\ \hline B = F & \sum_{e \in T, F} P(A = F | B = F, E = e) P(E = e) \\ \hline \end{array} \end{array}$$

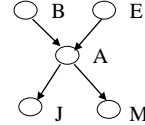
Variable elimination



Assume order: **M, E, B, A** to calculate $P(J = T)$

$$\begin{aligned}
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \quad 1 \\
 &= \sum_{a \in T, F} \sum_{b \in T, F} P(J = T | A = a) P(B = b) \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right] \\
 &= \sum_{a \in T, F} \sum_{b \in T, F} P(J = T | A = a) P(B = b) \tau_1(A = a, B = b)
 \end{aligned}$$

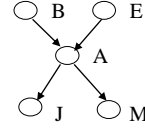
Variable elimination



Assume order: **M, E, B, A** to calculate $P(J = T)$

$$\begin{aligned}
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \quad 1 \\
 &= \sum_{a \in T, F} \sum_{b \in T, F} P(J = T | A = a) P(B = b) \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right] \\
 &= \sum_{a \in T, F} \sum_{b \in T, F} P(J = T | A = a) P(B = b) \tau_1(A = a, B = b) \\
 &= \sum_{a \in T, F} P(J = T | A = a) \left[\sum_{b \in T, F} P(B = b) \tau_1(A = a, B = b) \right]
 \end{aligned}$$

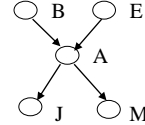
Variable elimination



Assume order: **M, E, B, A** to calculate $P(J = T)$

$$\begin{aligned}
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \quad 1 \\
 &= \sum_{a \in T, F} \sum_{b \in T, F} P(J = T | A = a) P(B = b) \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right] \\
 &= \sum_{a \in T, F} \sum_{b \in T, F} P(J = T | A = a) P(B = b) \tau_1(A = a, B = b) \\
 &= \sum_{a \in T, F} P(J = T | A = a) \left[\sum_{b \in T, F} P(B = b) \tau_1(A = a, B = b) \right] \\
 &\quad \tau_2(A = a)
 \end{aligned}$$

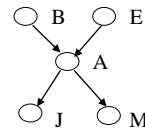
Variable elimination



Assume order: **M, E, B, A** to calculate $P(J = T)$

$$\begin{aligned}
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \quad 1 \\
 &= \sum_{a \in T, F} \sum_{b \in T, F} P(J = T | A = a) P(B = b) \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right] \\
 &= \sum_{a \in T, F} \sum_{b \in T, F} P(J = T | A = a) P(B = b) \tau_1(A = a, B = b) \\
 &= \sum_{a \in T, F} P(J = T | A = a) \left[\sum_{b \in T, F} P(B = b) \tau_1(A = a, B = b) \right] \\
 &= \sum_{a \in T, F} P(J = T | A = a) \tau_2(A = a)
 \end{aligned}$$

Variable elimination



Assume order: **M, E, B, A** to calculate $P(J = T)$

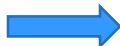
$$\begin{aligned}
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \left[\sum_{m \in T, F} P(M = m | A = a) \right] \\
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} P(J = T | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \quad 1 \\
 &= \sum_{a \in T, F} \sum_{b \in T, F} P(J = T | A = a) P(B = b) \left[\sum_{e \in T, F} P(A = a | B = b, E = e) P(E = e) \right] \\
 &= \sum_{a \in T, F} \sum_{b \in T, F} P(J = T | A = a) P(B = b) \tau_1(A = a, B = b) \\
 &= \sum_{a \in T, F} P(J = T | A = a) \left[\sum_{b \in T, F} P(B = b) \tau_1(A = a, B = b) \right] \\
 &= \sum_{a \in T, F} P(J = T | A = a) \tau_2(A = a) = \boxed{P(J = T)}
 \end{aligned}$$

Variable elimination

Assume order: **M, E, B, A** to calculate $P(J = T)$

$$\begin{aligned}
 &= \sum_{b \in T, F} \sum_{e \in T, F} \sum_{a \in T, F} \sum_{m \in T, F} P(J = T | A = a) P(M = m | A = a) P(A = a | B = b, E = e) P(B = b) P(E = e) \\
 &= \sum_{B \in T, F} \sum_{E \in T, F} \sum_{A \in T, F} \sum_{M \in T, F} f_1(A) f_2(M, A) f_3(A, B, E) f_4(B) f_4(E)
 \end{aligned}$$

Conditional probabilities defining the joint = **functions over variables** (or **factors**)



Variable elimination inference can be cast in terms of operations defined over factors

Factors

- **Factor:** is a function that maps value assignments for a subset of random variables to \mathfrak{R} (reals)
- **The scope of the factor:**
 - a set of variables defining the factor
- **Example:**
 - Assume discrete random variables x (with values a_1, a_2, a_3) and y (with values b_1 and b_2)

– Factor:

$$\phi(x, y) \longrightarrow$$

a1	b1	0.5
a1	b2	0.2
a2	b1	0.1
a2	b2	0.3
a3	b1	0.2
a3	b2	0.4

– Scope of the factor:

$$\{x, y\}$$

CS 3750 Advanced Machine Learning

Factor Product

Variables: A,B,C

$$\phi(A, B, C) = \phi(B, C) \circ \phi(A, B)$$

$\phi(B, C)$

b1	c1	0.1
b1	c2	0.6
b2	c1	0.3
b2	c2	0.4

$\phi(A, B)$

a1	b1	0.5
a1	b2	0.2
a2	b1	0.1
a2	b2	0.3
a3	b1	0.2
a3	b2	0.4

$\phi(A, B, C)$

a1	b1	c1	0.5*0.1
a1	b1	c2	0.5*0.6
a1	b2	c1	0.2*0.3
a1	b2	c2	0.2*0.4
a2	b1	c1	0.1*0.1
a2	b1	c2	0.1*0.6
a2	b2	c1	0.3*0.3
a2	b2	c2	0.3*0.4
a3	b1	c1	0.2*0.1
a3	b1	c2	0.2*0.6
a3	b2	c1	0.4*0.3
a3	b2	c2	0.4*0.4

CS 3750 Advanced Machine Learning

Factor Marginalization

Variables: A,B,C

$$\phi(A, C) = \sum_B \phi(A, B, C)$$

a1	b1	c1	0.2
a1	b1	c2	0.35
a1	b2	c1	0.4
a1	b2	c2	0.15
a2	b1	c1	0.5
a2	b1	c2	0.1
a2	b2	c1	0.3
a2	b2	c2	0.2
a3	b1	c1	0.25
a3	b1	c2	0.45
a3	b2	c1	0.15
a3	b2	c2	0.25

a1	c1	0.2+0.4=0.6
a1	c2	0.35+0.15=0.5
a2	c1	0.8
a2	c2	0.3
a3	c1	0.4
a3	c2	0.7

CS 3750 Advanced Machine Learning

Factor division

A=1	B=1	0.5
A=1	B=2	0.4
A=2	B=1	0.8
A=2	B=2	0.2
A=3	B=1	0.6
A=3	B=2	0.5

A=1	0.4
A=2	0.4
A=3	0.5

A=1	B=1	0.5/0.4=1.25
A=1	B=2	0.4/0.4=1.0
A=2	B=1	0.8/0.4=2.0
A=2	B=2	0.2/0.4=2.0
A=3	B=1	0.6/0.5=1.2
A=3	B=2	0.5/0.5=1.0

Inverse of a factor product

CS 3750 Advanced Machine Learning