

Fundamentals of Artificial Intelligence

Chapter 14: Probabilistic Reasoning

Roberto Sebastiani

DISI, Università di Trento, Italy – roberto.sebastiani@unitn.it
http://disi.unitn.it/rseba/DIDATTICA/fai_2022/

Teaching assistant: **Mauro Dragoni** – dragoni@fbk.eu
<http://www.maurodragoni.com/teaching/fai/>

M.S. Course “Artificial Intelligence Systems”, academic year 2022-2023

Last update: Monday 12th December, 2022, 15:08

Copyright notice: Most examples and images displayed in the slides of this course are taken from [\[Russell & Norvig, “Artificial Intelligence, a Modern Approach”, 3rd ed., Pearson\]](#), including explicitly figures from the above-mentioned book, so that their copyright is detained by the authors. A few other material (text, figures, examples) is authored by (in alphabetical order): [Pieter Abbeel](#), [Bonnie J. Dorr](#), [Anca Dragan](#), [Dan Klein](#), [Nikita Kitaev](#), [Tom Lenaerts](#), [Michela Milano](#), [Dana Nau](#), [Maria Simi](#), who detain its copyright.

These slides cannot be displayed in public without the permission of the author.

- 1 Bayesian Networks
 - Basics
 - Global Semantics
 - Local Semantics
 - Independence Property: Markov Blanket
- 2 Constructing Bayesian Networks
- 3 Exact Inference with Bayesian Networks
 - Inference by Enumeration
 - Inference by Variable Elimination
 - Complexity of Exact Inference

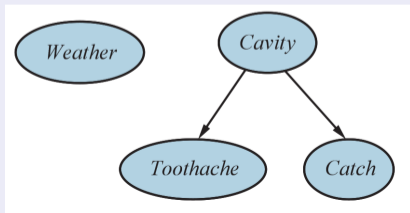
- 1 Bayesian Networks
 - Basics
 - Global Semantics
 - Local Semantics
 - Independence Property: Markov Blanket
- 2 Constructing Bayesian Networks
- 3 Exact Inference with Bayesian Networks
 - Inference by Enumeration
 - Inference by Variable Elimination
 - Complexity of Exact Inference

- 1 Bayesian Networks
 - Basics
 - Global Semantics
 - Local Semantics
 - Independence Property: Markov Blanket
- 2 Constructing Bayesian Networks
- 3 Exact Inference with Bayesian Networks
 - Inference by Enumeration
 - Inference by Variable Elimination
 - Complexity of Exact Inference

Bayesian Networks

Bayesian Networks (aka Belief Networks):

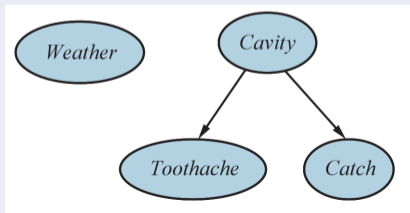
- Syntax: a directed acyclic graph (DAG):
 - each node represents a random variable (discrete or continuous)
 - directed arcs connect pairs of nodes: $X \rightarrow Y$ (X is a **parent** of Y)
 - a **conditional distribution** $\mathbf{P}(X_i | \mathbf{Parents}(X_i))$ for each node X_i
- Conditional distribution represented as a **conditional probability table (CPT)**
 - distribution over X_i for each combination of parent values
- Allow for **compact specification of full joint distributions**
- Represent explicit conditional dependencies among variables:
an arc from X to Y means that X has a direct influence on Y
- Topology encodes conditional independence assertions:
 - Toothache, Catch conditionally independent given Cavity
 - Toothache, Catch depend on Cavity
 - Weather independent from others
- **No arc \iff independence**



Bayesian Networks

Bayesian Networks (aka Belief Networks):

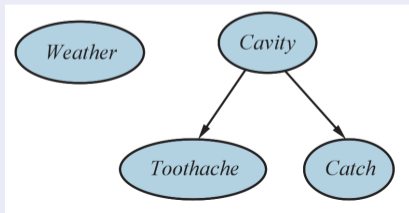
- Syntax: a directed acyclic graph (DAG):
 - each node represents a random variable (discrete or continuous)
 - directed arcs connect pairs of nodes: $X \rightarrow Y$ (X is a **parent** of Y)
 - a **conditional distribution** $\mathbf{P}(X_i | \mathbf{Parents}(X_i))$ for each node X_i
- Conditional distribution represented as a **conditional probability table (CPT)**
 - distribution over X_i for each combination of parent values
- Allow for **compact specification of full joint distributions**
- Represent explicit conditional dependencies among variables:
an arc from X to Y means that X has a direct influence on Y
- Topology encodes conditional independence assertions:
 - Toothache, Catch conditionally independent given Cavity
 - Toothache, Catch depend on Cavity
 - Weather independent from others
- **No arc \iff independence**



Bayesian Networks

Bayesian Networks (aka Belief Networks):

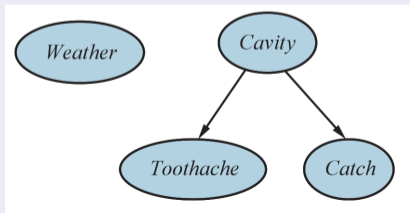
- Syntax: a directed acyclic graph (DAG):
 - each node represents a random variable (discrete or continuous)
 - directed arcs connect pairs of nodes: $X \rightarrow Y$ (X is a **parent** of Y)
 - a **conditional distribution** $\mathbf{P}(X_i | \mathbf{Parents}(X_i))$ for each node X_i
- Conditional distribution represented as a **conditional probability table (CPT)**
 - distribution over X_i for each combination of parent values
- Allow for **compact specification of full joint distributions**
- Represent explicit conditional dependencies among variables:
an arc from X to Y means that X has a direct influence on Y
- Topology encodes conditional independence assertions:
 - Toothache, Catch conditionally independent given Cavity
 - Toothache, Catch depend on Cavity
 - Weather independent from others
- No arc \iff independence



Bayesian Networks

Bayesian Networks (aka Belief Networks):

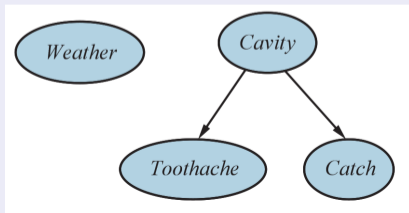
- Syntax: a directed acyclic graph (DAG):
 - each node represents a random variable (discrete or continuous)
 - directed arcs connect pairs of nodes: $X \rightarrow Y$ (X is a **parent** of Y)
 - a **conditional distribution** $\mathbf{P}(X_i | \mathbf{Parents}(X_i))$ for each node X_i
- Conditional distribution represented as a **conditional probability table (CPT)**
 - distribution over X_i for each combination of parent values
- Allow for **compact specification of full joint distributions**
- Represent explicit conditional dependencies among variables:
an arc from X to Y means that X has a direct influence on Y
- Topology encodes conditional independence assertions:
 - Toothache, Catch conditionally independent given Cavity
 - Toothache, Catch depend on Cavity
 - Weather independent from others
- No arc \iff independence



Bayesian Networks

Bayesian Networks (aka Belief Networks):

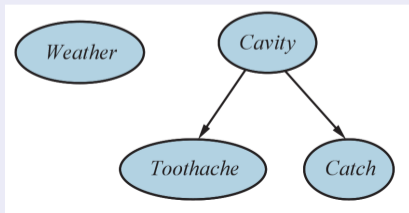
- Syntax: a directed acyclic graph (DAG):
 - each node represents a random variable (discrete or continuous)
 - directed arcs connect pairs of nodes: $X \rightarrow Y$ (X is a **parent** of Y)
 - a **conditional distribution** $\mathbf{P}(X_i | \mathbf{Parents}(X_i))$ for each node X_i
- Conditional distribution represented as a **conditional probability table (CPT)**
 - distribution over X_i for each combination of parent values
- Allow for **compact specification of full joint distributions**
- Represent explicit conditional dependencies among variables:
an arc from X to Y means that X has a direct influence on Y
- Topology encodes conditional independence assertions:
 - Toothache, Catch conditionally independent given Cavity
 - Toothache, Catch depend on Cavity
 - Weather independent from others
- No arc \iff independence



Bayesian Networks

Bayesian Networks (aka Belief Networks):

- Syntax: a directed acyclic graph (DAG):
 - each node represents a random variable (discrete or continuous)
 - directed arcs connect pairs of nodes: $X \rightarrow Y$ (X is a **parent** of Y)
 - a **conditional distribution** $\mathbf{P}(X_i | \mathbf{Parents}(X_i))$ for each node X_i
- Conditional distribution represented as a **conditional probability table (CPT)**
 - distribution over X_i for each combination of parent values
- Allow for **compact specification of full joint distributions**
- Represent explicit conditional dependencies among variables:
an arc from X to Y means that X has a direct influence on Y
- Topology encodes conditional independence assertions:
 - Toothache, Catch conditionally independent given Cavity
 - Toothache, Catch depend on Cavity
 - Weather independent from others
- **No arc \iff independence**



Example (from Judea Pearl, UCLA)

“The burglary alarm goes off very likely on burglary and occasionally on earthquakes. John and Mary are neighbors who agreed to call when the alarm goes off. Their reliability is different ...”

- Variables: Burglary, Earthquake, Alarm, JohnCalls, MaryCalls
- Network topology reflects “causal” knowledge:
 - A burglar can set the alarm off
 - An earthquake can set the alarm off
 - The alarm can cause Mary to call
 - The alarm can cause John to call
- CPTs:
 - alarm setoff if bunglar
in 94% of cases
 - alarm setoff if hearthquake
in 29% of cases
 - false alarm setoff
in 0.1% of cases
- Notice: in CPTs like $\mathbf{P}(A|B)$, only $P(a|B)$ are reported, because $P(\neg a|B) = 1 - P(a|B)$

Example (from Judea Pearl, UCLA)

“The burglary alarm goes off very likely on burglary and occasionally on earthquakes. John and Mary are neighbors who agreed to call when the alarm goes off. Their reliability is different ...”

- Variables: Burglary, Earthquake, Alarm, JohnCalls, MaryCalls

- Network topology reflects “causal” knowledge:

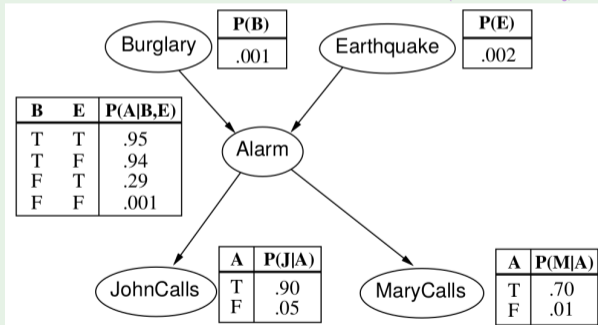
- A burglar can set the alarm off
- An earthquake can set the alarm off
- The alarm can cause Mary to call
- The alarm can cause John to call

- CPTs:

- alarm setoff if bunglar in 94% of cases
- alarm setoff if hearthquake in 29% of cases
- false alarm setoff in 0.1% of cases

- Notice: in CPTs like $P(A|B)$, only $P(a|B)$ are reported, because $P(\neg a|B) = 1 - P(a|B)$

(© S. Russell & P. Norwig, AIMA)



Example (from Judea Pearl, UCLA)

“The burglary alarm goes off very likely on burglary and occasionally on earthquakes. John and Mary are neighbors who agreed to call when the alarm goes off. Their reliability is different ...”

- Variables: Burglary, Earthquake, Alarm, JohnCalls, MaryCalls
- Network topology reflects “causal” knowledge:

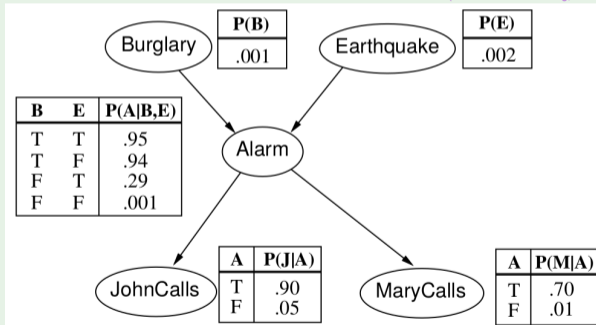
- A burglar can set the alarm off
- An earthquake can set the alarm off
- The alarm can cause Mary to call
- The alarm can cause John to call

- CPTs:

- alarm set off if burglar in 94% of cases
- alarm set off if earthquake in 29% of cases
- false alarm set off in 0.1% of cases

- Notice: in CPTs like $P(A|B)$, only $P(a|B)$ are reported, because $P(\neg a|B) = 1 - P(a|B)$

(© S. Russell & P. Norwig, AIMA)



Example (from Judea Pearl, UCLA)

“The burglary alarm goes off very likely on burglary and occasionally on earthquakes. John and Mary are neighbors who agreed to call when the alarm goes off. Their reliability is different ...”

- Variables: Burglary, Earthquake, Alarm, JohnCalls, MaryCalls
- Network topology reflects “causal” knowledge:

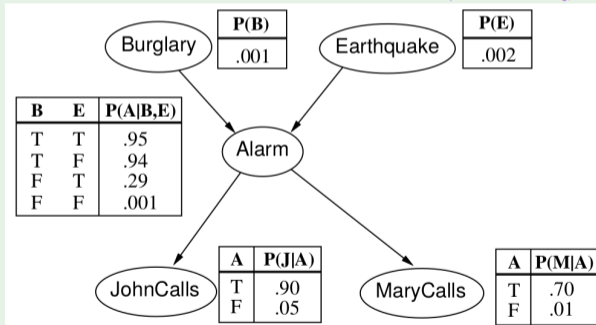
- A burglar can set the alarm off
- An earthquake can set the alarm off
- The alarm can cause Mary to call
- The alarm can cause John to call

- CPTs:

- alarm setoff if bunglar in 94% of cases
- alarm setoff if hearthquake in 29% of cases
- false alarm setoff in 0.1% of cases

- Notice: in CPTs like $P(A|B)$, only $P(a|B)$ are reported, because $P(\neg a|B) = 1 - P(a|B)$

(© S. Russell & P. Norwig, AIMA)



- 1 Bayesian Networks
 - Basics
 - **Global Semantics**
 - Local Semantics
 - Independence Property: Markov Blanket
- 2 Constructing Bayesian Networks
- 3 Exact Inference with Bayesian Networks
 - Inference by Enumeration
 - Inference by Variable Elimination
 - Complexity of Exact Inference

Compactness of Bayesian Networks

- In most domains, it is reasonable to suppose that **each random variable X_i is directly influenced by only a small number k_i of other variables**, called **parents** of X_i (**parents(X_i)**)
 - A CPT for Boolean X_i with k_i Boolean parents has
 - 2^{k_i} rows for the combinations of parent values
 - each row requires one number p for $P(X_i = \text{true})$
($P(X_i = \text{false}) = 1 - P(X_i = \text{true})$)
- ⇒ If each variable has no more than k parents, **the complete network requires $O(n \cdot 2^k)$ numbers**
- a full joint distribution requires $2^n - 1$ numbers
 - **linear vs. exponential!**
 - Ex: for burglary example:
 - $1 + 1 + 4 + 2 + 2 = 10$ numbers vs. $2^5 - 1 = 31$

Compactness of Bayesian Networks

- In most domains, it is reasonable to suppose that **each random variable X_i is directly influenced by only a small number k_i of other variables**, called **parents** of X_i (**parents(X_i)**)
- A CPT for Boolean X_i with k_i Boolean parents has
 - 2^{k_i} rows for the combinations of parent values
 - each row requires one number p for $P(X_i = \text{true})$
($P(X_i = \text{false}) = 1 - P(X_i = \text{true})$)

⇒ If each variable has no more than k parents, **the complete network requires $O(n \cdot 2^k)$ numbers**

- a full joint distribution requires $2^n - 1$ numbers
- **linear vs. exponential!**
- Ex: for burglary example:
 - $1 + 1 + 4 + 2 + 2 = 10$ numbers vs. $2^5 - 1 = 31$

Compactness of Bayesian Networks

- In most domains, it is reasonable to suppose that **each random variable X_i is directly influenced by only a small number k_i of other variables**, called **parents** of X_i (**parents(X_i)**)
- A CPT for Boolean X_i with k_i Boolean parents has
 - 2^{k_i} rows for the combinations of parent values
 - each row requires one number p for $P(X_i = \text{true})$
($P(X_i = \text{false}) = 1 - P(X_i = \text{true})$)

⇒ If each variable has no more than k parents, **the complete network requires $O(n \cdot 2^k)$ numbers**

- a full joint distribution requires $2^n - 1$ numbers
- **linear vs. exponential!**
- Ex: for burglary example:
 - $1 + 1 + 4 + 2 + 2 = 10$ numbers vs. $2^5 - 1 = 31$

Compactness of Bayesian Networks

- In most domains, it is reasonable to suppose that each random variable X_i is directly influenced by only a small number k_i of other variables, called parents of X_i ($\text{parents}(X_i)$)
- A CPT for Boolean X_i with k_i Boolean parents has
 - 2^{k_i} rows for the combinations of parent values
 - each row requires one number p for $P(X_i = \text{true})$
($P(X_i = \text{false}) = 1 - P(X_i = \text{true})$)

⇒ If each variable has no more than k parents, the complete network requires $O(n \cdot 2^k)$ numbers

- a full joint distribution requires $2^n - 1$ numbers
- linear vs. exponential!
- Ex: for burglary example:
 - $1 + 1 + 4 + 2 + 2 = 10$ numbers vs. $2^5 - 1 = 31$

Global Semantics of Bayesian Networks

- **Global semantics** defines the full joint distribution as the product of the local conditional distributions:

$$\mathbf{P}(X_1, \dots, X_N) = \prod_{i=1} \mathbf{P}(X_i | \text{parents}(X_i))$$

- if X_i has no parent, then conditional distributions reduce to prior probability $\mathbf{P}(X_i)$
- Intuition: order X_1, \dots, X_n s.t. $\text{parents}(X_i) \prec X_i$ for each i :

$$\mathbf{P}(X_1, \dots, X_n)$$

$$= \prod_{i=1}^n \mathbf{P}(X_i | X_1, \dots, X_{i-1}) \quad // \text{ chain rule}$$

$$= \prod_{i=1} \mathbf{P}(X_i | \text{parents}(X_i)) \quad // \text{ conditional independence}$$

⇒ A Bayesian network is a distributed representation of the full joint distribution

Global Semantics of Bayesian Networks

- **Global semantics** defines the full joint distribution as the product of the local conditional distributions:

$$\mathbf{P}(X_1, \dots, X_N) = \prod_{i=1} \mathbf{P}(X_i | \text{parents}(X_i))$$

- if X_i has no parent, then conditional distributions reduce to prior probability $\mathbf{P}(X_i)$
- Intuition: order X_1, \dots, X_n s.t. $\text{parents}(X_i) \prec X_i$ for each i :

$$\mathbf{P}(X_1, \dots, X_n)$$

$$= \prod_{i=1}^n \mathbf{P}(X_i | X_1, \dots, X_{i-1}) \text{ // chain rule}$$

$$= \prod_{i=1} \mathbf{P}(X_i | \text{parents}(X_i)) \text{ // conditional independence}$$

⇒ A Bayesian network is a distributed representation of the full joint distribution

Global Semantics of Bayesian Networks

- **Global semantics** defines the full joint distribution as the product of the local conditional distributions:

$$\mathbf{P}(X_1, \dots, X_N) = \prod_{i=1} \mathbf{P}(X_i | \text{parents}(X_i))$$

- if X_i has no parent, then conditional distributions reduce to prior probability $\mathbf{P}(X_i)$
- Intuition: order X_1, \dots, X_n s.t. $\text{parents}(X_i) \prec X_i$ for each i :

$$\mathbf{P}(X_1, \dots, X_n)$$

$$= \prod_{i=1}^n \mathbf{P}(X_i | X_1, \dots, X_{i-1}) \text{ // chain rule}$$

$$= \prod_{i=1} \mathbf{P}(X_i | \text{parents}(X_i)) \text{ // conditional independence}$$

⇒ A Bayesian network is a distributed representation of the full joint distribution

Global Semantics of Bayesian Networks

- **Global semantics** defines the full joint distribution as the product of the local conditional distributions:

$$\mathbf{P}(X_1, \dots, X_N) = \prod_{i=1} \mathbf{P}(X_i | \text{parents}(X_i))$$

- if X_i has no parent, then conditional distributions reduce to prior probability $\mathbf{P}(X_i)$
- Intuition: order X_1, \dots, X_n s.t. $\text{parents}(X_i) \prec X_i$ for each i :

$$\mathbf{P}(X_1, \dots, X_n)$$

$$= \prod_{i=1}^n \mathbf{P}(X_i | X_1, \dots, X_{i-1}) \text{ // chain rule}$$

$$= \prod_{i=1} \mathbf{P}(X_i | \text{parents}(X_i)) \text{ // conditional independence}$$

⇒ A Bayesian network is a distributed representation of the full joint distribution

Global Semantics of Bayesian Networks

- **Global semantics** defines the full joint distribution as the product of the local conditional distributions:

$$\mathbf{P}(X_1, \dots, X_N) = \prod_{i=1} \mathbf{P}(X_i | \text{parents}(X_i))$$

- if X_i has no parent, then conditional distributions reduce to prior probability $\mathbf{P}(X_i)$
- Intuition: order X_1, \dots, X_n s.t. $\text{parents}(X_i) \prec X_i$ for each i :

$$\mathbf{P}(X_1, \dots, X_n)$$

$$= \prod_{i=1}^n \mathbf{P}(X_i | X_1, \dots, X_{i-1}) \text{ // chain rule}$$

$$= \prod_{i=1} \mathbf{P}(X_i | \text{parents}(X_i)) \text{ // conditional independence}$$

⇒ A Bayesian network is a distributed representation of the full joint distribution

Global Semantics: Example

- $\mathbf{P}(X_1, \dots, X_N) = \prod_{i=1} \mathbf{P}(X_i | \text{parents}(X_i))$
- Ex: “Prob. that both John and Mary call, the alarm sets off but no burglary nor earthquake”
 $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) =$
 $P(j|m \wedge a \wedge \neg b \wedge \neg e)P(m|a \wedge \neg b \wedge \neg e)P(a|\neg b \wedge \neg e)P(\neg b|\neg e)P(\neg e) =$
 $P(j|a) P(m|a) P(a|\neg b \wedge \neg e) P(\neg b) P(\neg e) = 0.9 \cdot 0.7 \cdot 0.001 \cdot 0.999 \cdot 0.998$
 ≈ 0.00063

Global Semantics: Example

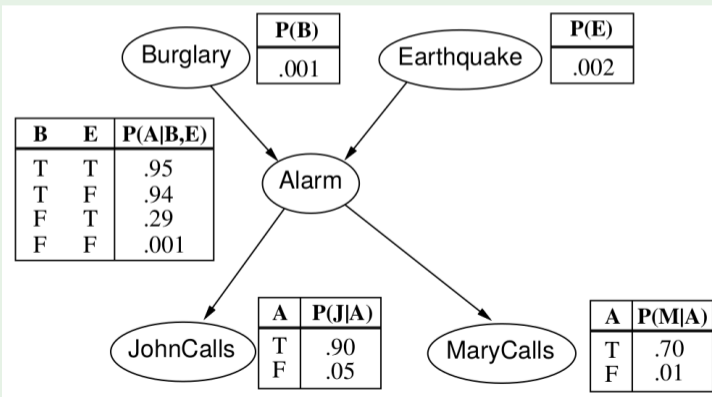
- $P(X_1, \dots, X_N) = \prod_{i=1} P(X_i | \text{parents}(X_i))$
- Ex: “Prob. that both John and Mary call, the alarm sets off but no burglary nor earthquake”

$$P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) =$$

$$P(j|m \wedge a \wedge \neg b \wedge \neg e) P(m|a \wedge \neg b \wedge \neg e) P(a|\neg b \wedge \neg e) P(\neg b|\neg e) P(\neg e) =$$

$$P(j|a) P(m|a) P(a|\neg b \wedge \neg e) P(\neg b) P(\neg e) = 0.9 \cdot 0.7 \cdot 0.001 \cdot 0.999 \cdot 0.998$$

$$\approx 0.00063$$



Global Semantics: Example

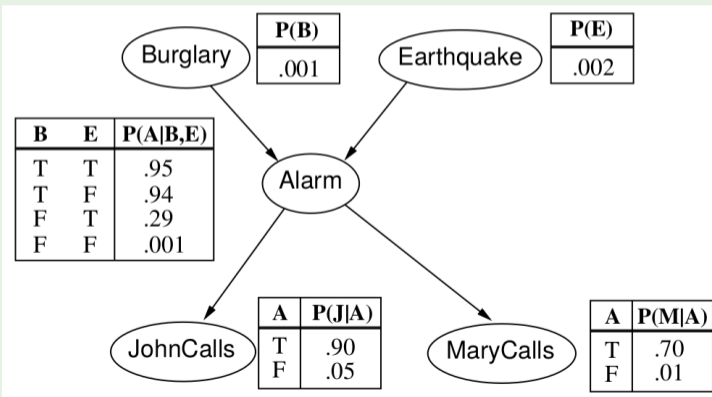
- $P(X_1, \dots, X_N) = \prod_{i=1} P(X_i | \text{parents}(X_i))$
- Ex: “Prob. that both John and Mary call, the alarm sets off but no burglary nor earthquake”

$$P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) =$$

$$P(j|m \wedge a \wedge \neg b \wedge \neg e) P(m|a \wedge \neg b \wedge \neg e) P(a|\neg b \wedge \neg e) P(\neg b|\neg e) P(\neg e) =$$

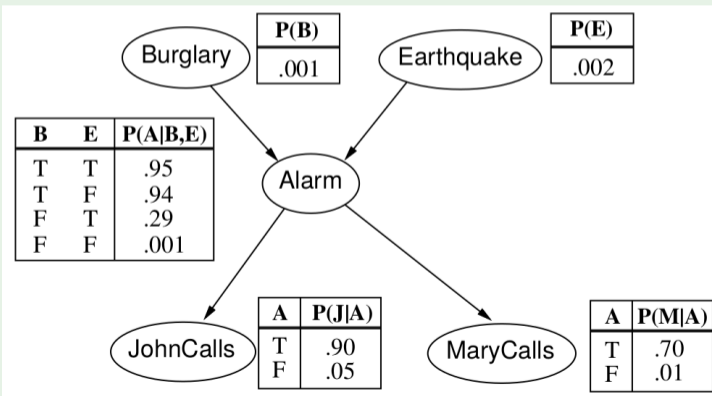
$$P(j|a) P(m|a) P(a|\neg b \wedge \neg e) P(\neg b) P(\neg e) = 0.9 \cdot 0.7 \cdot 0.001 \cdot 0.999 \cdot 0.998$$

$$\approx 0.00063$$



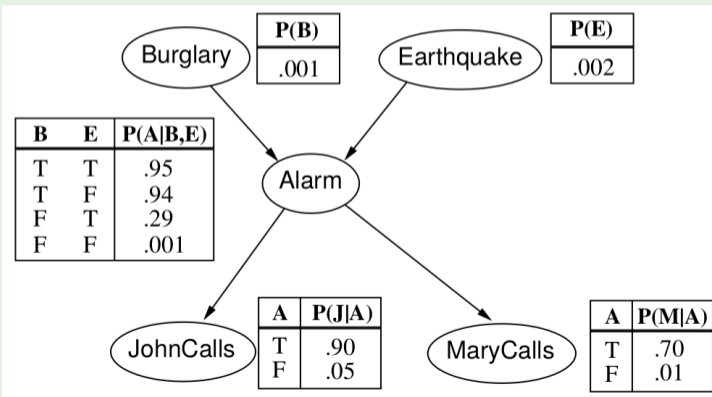
Global Semantics: Example

- $P(X_1, \dots, X_N) = \prod_{i=1} P(X_i | \text{parents}(X_i))$
- Ex: “Prob. that both John and Mary call, the alarm sets off but no burglary nor earthquake”
 $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) =$
 $P(j|m \wedge a \wedge \neg b \wedge \neg e)P(m|a \wedge \neg b \wedge \neg e)P(a|\neg b \wedge \neg e)P(\neg b|\neg e)P(\neg e) =$
 $P(j|a) P(m|a) P(a|\neg b \wedge \neg e) P(\neg b) P(\neg e) = 0.9 \cdot 0.7 \cdot 0.001 \cdot 0.999 \cdot 0.998$
 ≈ 0.00063



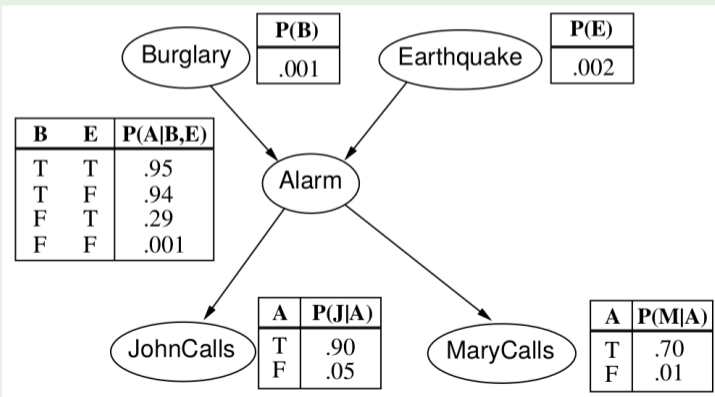
Global Semantics: Example

- $P(X_1, \dots, X_N) = \prod_{i=1} P(X_i | \text{parents}(X_i))$
- Ex: “Prob. that both John and Mary call, the alarm sets off but no burglary nor earthquake”
 $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) =$
 $P(j | m \wedge a \wedge \neg b \wedge \neg e) P(m | a \wedge \neg b \wedge \neg e) P(a | \neg b \wedge \neg e) P(\neg b | \neg e) P(\neg e) =$
 $P(j | a) P(m | a) P(a | \neg b \wedge \neg e) P(\neg b) P(\neg e) = 0.9 \cdot 0.7 \cdot 0.001 \cdot 0.999 \cdot 0.998$
 ≈ 0.00063



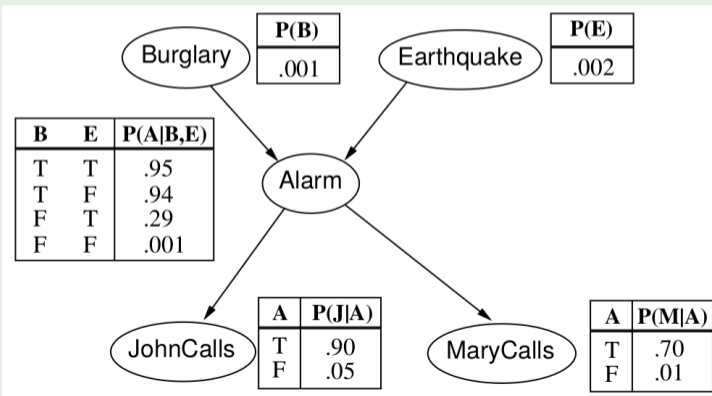
Global Semantics: Example

- $P(X_1, \dots, X_N) = \prod_{i=1} P(X_i | \text{parents}(X_i))$
- Ex: “Prob. that both John and Mary call, the alarm sets off but no burglary nor earthquake”
 $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) =$
 $P(j|m \wedge a \wedge \neg b \wedge \neg e)P(m|a \wedge \neg b \wedge \neg e)P(a|\neg b \wedge \neg e)P(\neg b|\neg e)P(\neg e) =$
 $P(j|a) P(m|a) P(a|\neg b \wedge \neg e) P(\neg b) P(\neg e) = 0.9 \cdot 0.7 \cdot 0.001 \cdot 0.999 \cdot 0.998$
 ≈ 0.00063



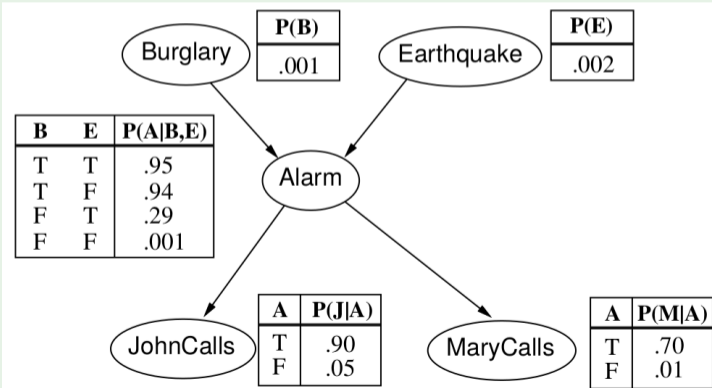
Global Semantics: Example

- $P(X_1, \dots, X_N) = \prod_{i=1} P(X_i | \text{parents}(X_i))$
- Ex: “Prob. that both John and Mary call, the alarm sets off but no burglary nor earthquake”
 $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) =$
 $P(j|m \wedge a \wedge \neg b \wedge \neg e)P(m|a \wedge \neg b \wedge \neg e)P(a|\neg b \wedge \neg e)P(\neg b|\neg e)P(\neg e) =$
 $P(j|a) P(m|a) P(a|\neg b \wedge \neg e) P(\neg b) P(\neg e) = 0.9 \cdot 0.7 \cdot 0.001 \cdot 0.999 \cdot 0.998$
 ≈ 0.00063



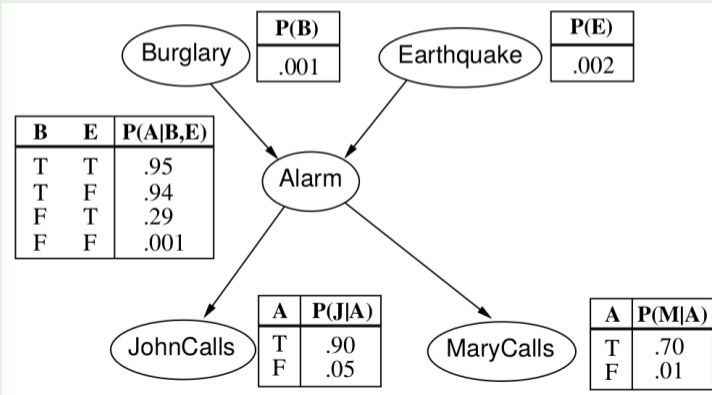
Global Semantics: Example

- $P(X_1, \dots, X_N) = \prod_{i=1} P(X_i | \text{parents}(X_i))$
- Ex: “Prob. that both John and Mary call, the alarm sets off but no burglary nor earthquake”
 $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) =$
 $P(j|m \wedge a \wedge \neg b \wedge \neg e)P(m|a \wedge \neg b \wedge \neg e)P(a|\neg b \wedge \neg e)P(\neg b|\neg e)P(\neg e) =$
 $P(j|a) P(m|a) P(a|\neg b \wedge \neg e) P(\neg b) P(\neg e) = 0.9 \cdot 0.7 \cdot 0.001 \cdot 0.999 \cdot 0.998$
 ≈ 0.00063



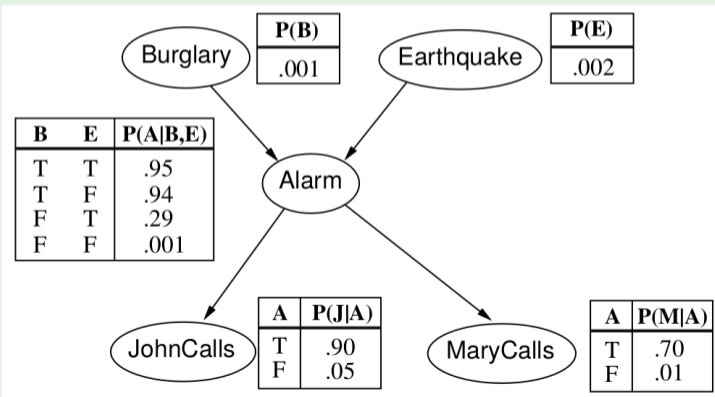
Global Semantics: Example

- $P(X_1, \dots, X_N) = \prod_{i=1} P(X_i | \text{parents}(X_i))$
- Ex: “Prob. that both John and Mary call, the alarm sets off but no burglary nor earthquake”
 $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) =$
 $P(j | m \wedge a \wedge \neg b \wedge \neg e) P(m | a \wedge \neg b \wedge \neg e) P(a | \neg b \wedge \neg e) P(\neg b | \neg e) P(\neg e) =$
 $P(j | a) P(m | a) P(a | \neg b \wedge \neg e) P(\neg b) P(\neg e) = 0.9 \cdot 0.7 \cdot 0.001 \cdot 0.999 \cdot 0.998$
 ≈ 0.00063



Global Semantics: Example

- $P(X_1, \dots, X_N) = \prod_{i=1} P(X_i | \text{parents}(X_i))$
- Ex: “Prob. that both John and Mary call, the alarm sets off but no burglary nor earthquake”
 $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) =$
 $P(j | m \wedge a \wedge \neg b \wedge \neg e) P(m | a \wedge \neg b \wedge \neg e) P(a | \neg b \wedge \neg e) P(\neg b | \neg e) P(\neg e) =$
 $P(j | a) P(m | a) P(a | \neg b \wedge \neg e) P(\neg b) P(\neg e) = 0.9 \cdot 0.7 \cdot 0.001 \cdot 0.999 \cdot 0.998$
 ≈ 0.00063

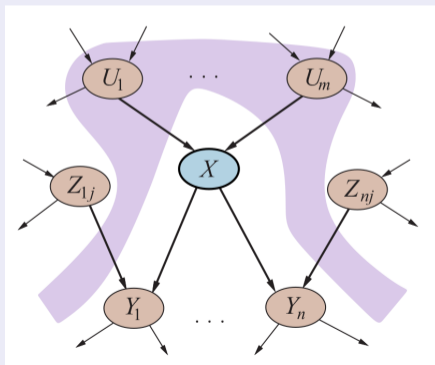


- Compute:
 - The probability that John calls and Mary does not, the alarm is not set off with a burglar entering during an earthquake
 - The probability that John calls and Mary does not, given a burglar entering the house
 - The probability of an earthquake given the fact that John has called
 - ...

- 1 Bayesian Networks
 - Basics
 - Global Semantics
 - **Local Semantics**
 - Independence Property: Markov Blanket
- 2 Constructing Bayesian Networks
- 3 Exact Inference with Bayesian Networks
 - Inference by Enumeration
 - Inference by Variable Elimination
 - Complexity of Exact Inference

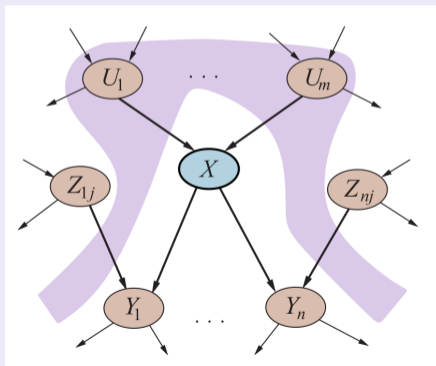
Local Semantics

- **Local Semantics:** each node is conditionally independent of its nondescendants given its parents: $\mathbf{P}(X|U_1, \dots, U_m, Z_{1j}, \dots, Z_{nj}) = \mathbf{P}(X|U_1, \dots, U_m)$, for each X
 - “nondescendants” include ancestors
- Theorem: Local semantics holds iff global semantics holds:
 $\mathbf{P}(X_1, \dots, X_N) = \prod_{i=1} \mathbf{P}(X_i | \text{parents}(X_i))$



Local Semantics

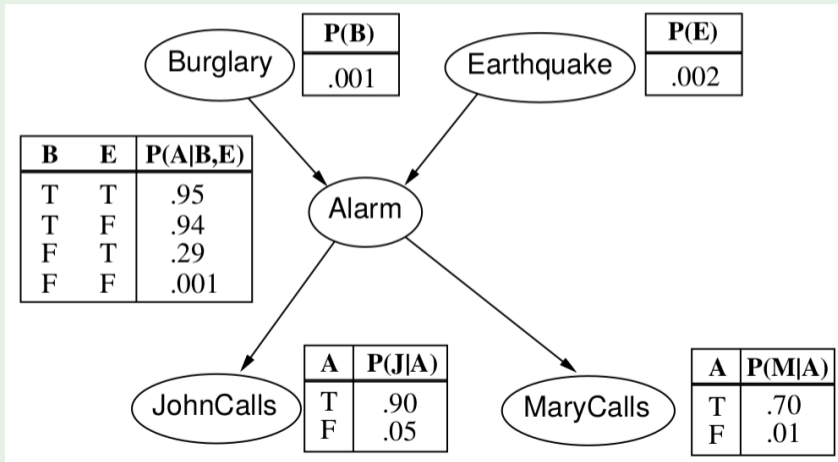
- **Local Semantics:** each node is conditionally independent of its nondescendants given its parents: $\mathbf{P}(X|U_1, \dots, U_m, Z_{1j}, \dots, Z_{nj}) = \mathbf{P}(X|U_1, \dots, U_m)$, for each X
 - “nondescendants” include ancestors
- **Theorem:** Local semantics holds iff global semantics holds:
 $\mathbf{P}(X_1, \dots, X_N) = \prod_{i=1} \mathbf{P}(X_i | \text{parents}(X_i))$



Local Semantics: Example

Ex: JohnCalls is independent of Burglary, Earthquake, and MaryCalls given the value of Alarm

$$P(\text{JohnCalls} | \text{Alarm}, \text{Burglary}, \text{Earthquake}, \text{MaryCalls}) = P(\text{JohnCalls} | \text{Alarm})$$

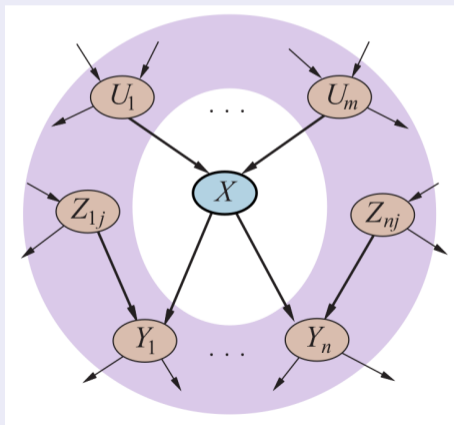


- 1 Bayesian Networks
 - Basics
 - Global Semantics
 - Local Semantics
 - Independence Property: Markov Blanket
- 2 Constructing Bayesian Networks
- 3 Exact Inference with Bayesian Networks
 - Inference by Enumeration
 - Inference by Variable Elimination
 - Complexity of Exact Inference

Independence Property: Markov Blanket

In an Bayesian Network, each node is conditionally independent of all others given its **Markov blanket**: parents + children + children's parents:

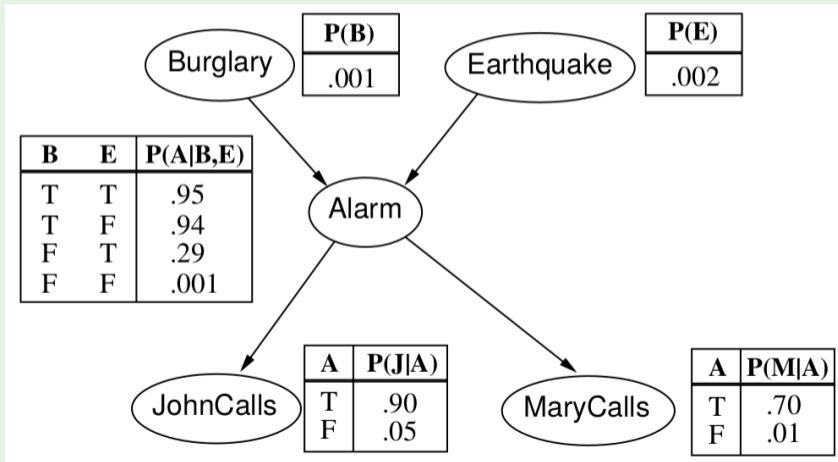
$P(X|U_1, \dots, U_m, Y_1, \dots, Y_n, Z_{1j}, \dots, Z_{nj}, W_1, \dots, W_k) = P(X|U_1, \dots, U_m, Y_1, \dots, Y_n, Z_{1j}, \dots, Z_{nj})$, for each X



Markov Blanket: Example

Ex: Burglary is independent of JohnCalls and MaryCalls, given Alarm and Earthquake

$$P(\text{Burglary} | \text{Alarm}, \text{Earthquake}, \text{JohnCalls}, \text{MaryCalls}) = P(\text{Burglary} | \text{Alarm}, \text{Earthquake})$$



Exercise

Verify numerically the two previous examples:

- Local Semantics
- Markov Blanket

- 1 Bayesian Networks
 - Basics
 - Global Semantics
 - Local Semantics
 - Independence Property: Markov Blanket
- 2 **Constructing Bayesian Networks**
- 3 Exact Inference with Bayesian Networks
 - Inference by Enumeration
 - Inference by Variable Elimination
 - Complexity of Exact Inference

Constructing Bayesian Networks

Building the graph

Given a set of random variables

1. Choose an ordering $\{X_1, \dots, X_n\}$

- in principle, any ordering will work (but some may cause blowups)
- general rule: **follow causality**, $X \prec Y$ if $X \in \text{causes}(Y)$

2. For $i=1$ to n do

1. add X_i to the network

2. as $\text{Parents}(X_i)$, choose a subset of $\{X_1, \dots, X_{i-1}\}$ s.t. $P(X_i | \text{Parents}(X_i)) = P(X_i | X_1, \dots, X_{i-1})$

⇒ Guarantees the global semantics by construction

$$\mathbf{P}(X_1, \dots, X_N) = \prod_{i=1} \mathbf{P}(X_i | \text{parents}(X_i))$$

Constructing Bayesian Networks

Building the graph

Given a set of random variables

1. Choose an ordering $\{X_1, \dots, X_n\}$
 - in principle, any ordering will work (but some may cause blowups)
 - general rule: **follow causality**, $X \prec Y$ if $X \in \text{causes}(Y)$
2. For $i=1$ to n do
 1. add X_i to the network
 2. as $\text{Parents}(X_i)$, choose a subset of $\{X_1, \dots, X_{i-1}\}$ s.t. $P(X_i | \text{Parents}(X_i)) = P(X_i | X_1, \dots, X_{i-1})$

⇒ Guarantees the global semantics by construction

$$\mathbf{P}(X_1, \dots, X_N) = \prod_{i=1} \mathbf{P}(X_i | \text{parents}(X_i))$$

Constructing Bayesian Networks

Building the graph

Given a set of random variables

1. Choose an ordering $\{X_1, \dots, X_n\}$
 - in principle, any ordering will work (but some may cause blowups)
 - general rule: **follow causality**, $X \prec Y$ if $X \in \text{causes}(Y)$
2. For $i=1$ to n do
 1. add X_i to the network
 2. as $\text{Parents}(X_i)$, choose a subset of $\{X_1, \dots, X_{i-1}\}$ s.t. $P(X_i | \text{Parents}(X_i)) = P(X_i | X_1, \dots, X_{i-1})$

⇒ Guarantees the global semantics by construction

$$\mathbf{P}(X_1, \dots, X_N) = \prod_{i=1} \mathbf{P}(X_i | \text{parents}(X_i))$$

Constructing Bayesian Networks: Example

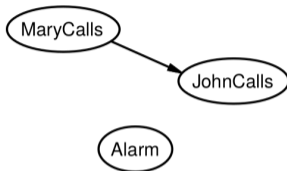
Suppose we choose the ordering $\{M, J, A, B, E\}$ (non-causal ordering):



$$P(J|M) = P(J)?$$

Constructing Bayesian Networks: Example

Suppose we choose the ordering $\{M, J, A, B, E\}$ (non-causal ordering):

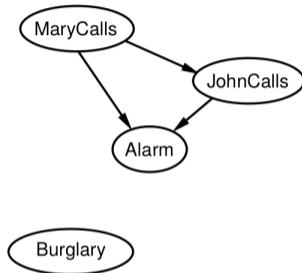


$P(J|M) = P(J)$? No

$P(A|J, M) = P(A|J)$? $P(A|J, M) = P(A)$?

Constructing Bayesian Networks: Example

Suppose we choose the ordering $\{M, J, A, B, E\}$ (non-causal ordering):



$P(J|M) = P(J)$? No

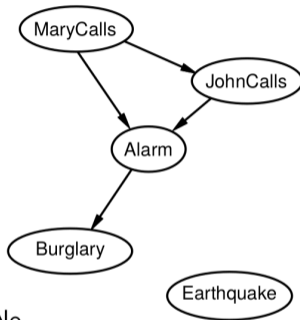
$P(A|J, M) = P(A|J)$? $P(A|J, M) = P(A)$? No

$P(B|A, J, M) = P(B|A)$?

$P(B|A, J, M) = P(B)$?

Constructing Bayesian Networks: Example

Suppose we choose the ordering $\{M, J, A, B, E\}$ (non-causal ordering):



$P(J|M) = P(J)$? No

$P(A|J, M) = P(A|J)$? $P(A|J, M) = P(A)$? No

$P(B|A, J, M) = P(B|A)$? Yes

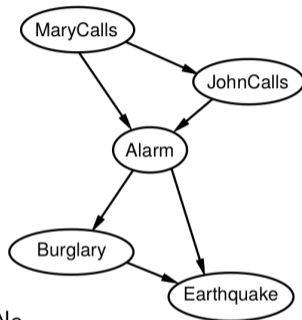
$P(B|A, J, M) = P(B)$? No

$P(E|B, A, J, M) = P(E|A)$?

$P(E|B, A, J, M) = P(E|A, B)$?

Constructing Bayesian Networks: Example

Suppose we choose the ordering $\{M, J, A, B, E\}$ (non-causal ordering):



$P(J|M) = P(J)$? No

$P(A|J, M) = P(A|J)$? $P(A|J, M) = P(A)$? No

$P(B|A, J, M) = P(B|A)$? Yes

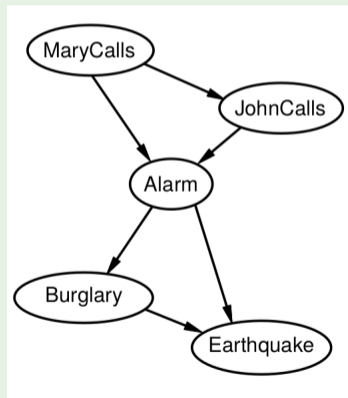
$P(B|A, J, M) = P(B)$? No

$P(E|B, A, J, M) = P(E|A)$? No

$P(E|B, A, J, M) = P(E|A, B)$? Yes

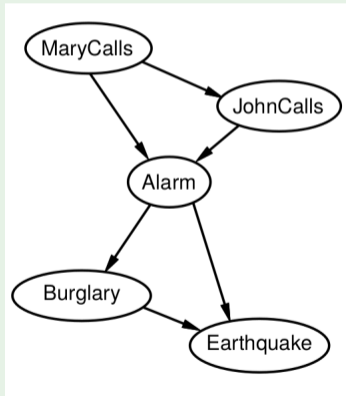
Constructing Bayesian Networks: Example [cont.]

- In non-causal directions
 - deciding conditional independence is hard
 - assessing conditional probabilities is hard
 - typically networks less compact
- Ex: $1+2+4+2+4=13$ numbers needed (rather than 10)
- Can be much worse
 - ex: try $\{M, J, E, B, A\}$ (see AIMA)
 - ex: try $\{J, M, E, B, A\}$
 - ex: try $\{J, M, E, A, B\}$
- Much better with causal orderings
 - ex: try either
 - $\{B, E, A, J, M\}$
 - $\{E, B, A, J, M\}$
 - $\{B, E, A, M, J\}$
 - $\{E, B, A, M, J\}$
 - i.e. $\{B, E\} \prec A \prec \{J, M\}$
(both B and E cause A, A causes both M and J)



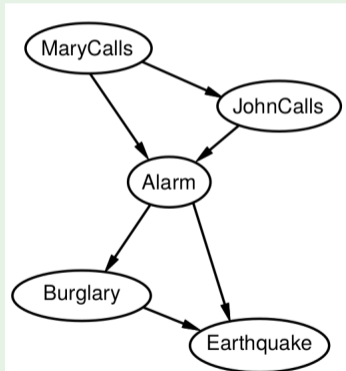
Constructing Bayesian Networks: Example [cont.]

- In non-causal directions
 - deciding conditional independence is hard
 - assessing conditional probabilities is hard
 - typically networks less compact
- Ex: $1+2+4+2+4=13$ numbers needed (rather than 10)
- Can be much worse
 - ex: try $\{M, J, E, B, A\}$ (see AIMA)
 - ex: try $\{J, M, E, B, A\}$
 - ex: try $\{J, M, E, A, B\}$
- Much better with causal orderings
 - ex: try either
 - $\{B, E, A, J, M\}$
 - $\{E, B, A, J, M\}$
 - $\{B, E, A, M, J\}$
 - $\{E, B, A, M, J\}$
 - i.e. $\{B, E\} \prec A \prec \{J, M\}$
(both B and E cause A, A causes both M and J)



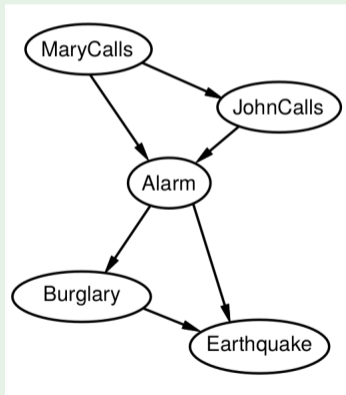
Constructing Bayesian Networks: Example [cont.]

- In non-causal directions
 - deciding conditional independence is hard
 - assessing conditional probabilities is hard
 - typically networks less compact
- Ex: $1+2+4+2+4=13$ numbers needed (rather than 10)
- Can be much worse
 - ex: try $\{M, J, E, B, A\}$ (see AIMA)
 - ex: try $\{J, M, E, B, A\}$
 - ex: try $\{J, M, E, A, B\}$
- Much better with causal orderings
 - ex: try either
 - $\{B, E, A, J, M\}$
 - $\{E, B, A, J, M\}$
 - $\{B, E, A, M, J\}$
 - $\{E, B, A, M, J\}$
 - i.e. $\{B, E\} \prec A \prec \{J, M\}$
(both B and E cause A, A causes both M and J)



Constructing Bayesian Networks: Example [cont.]

- In non-causal directions
 - deciding conditional independence is hard
 - assessing conditional probabilities is hard
 - typically networks less compact
- Ex: $1+2+4+2+4=13$ numbers needed (rather than 10)
- Can be much worse
 - ex: try $\{M, J, E, B, A\}$ (see AIMA)
 - ex: try $\{J, M, E, B, A\}$
 - ex: try $\{J, M, E, A, B\}$
- Much better with causal orderings
 - ex: try either
 - $\{B, E, A, J, M\}$
 - $\{E, B, A, J, M\}$
 - $\{B, E, A, M, J\}$
 - $\{E, B, A, M, J\}$
 - i.e. $\{B, E\} \prec A \prec \{J, M\}$
(both B and E cause A, A causes both M and J)



Building Conditional Probability Tables, CPTs

- **Problem: CPT grow exponentially with number of parents**
- **If the causes don't interact:** use a **Noisy-OR distribution** (generalization of logical or)
 - assume parents U_1, \dots, U_k include all causes (can add leak node representing "other causes"):
 $P(\neg X | \neg U_1 \dots \neg U_k) = 1.0$
 - assume independent failure probability $q_i \stackrel{\text{def}}{=} P(\neg X | U_i \wedge \bigwedge_{j \neq i} \neg U_j)$ for each cause U_i :
 $P(\neg X | U_1 \dots U_j, \neg U_{j+1} \dots \neg U_k) = \prod_{i=1}^j q_i$
 \implies number of parameters linear in number of parents!
- Ex: $q_{Cold} = 0.6$, $q_{Flu} = 0.2$, $q_{Malaria} = 0.1$:

Building Conditional Probability Tables, CPTs

- Problem: CPT grow exponentially with number of parents
- If the causes don't interact: use a Noisy-OR distribution (generalization of logical or)
 - assume parents U_1, \dots, U_k include all causes (can add leak node representing "other causes"):
 $P(\neg X | \neg U_1 \dots \neg U_k) = 1.0$
 - assume independent failure probability $q_i \stackrel{\text{def}}{=} P(\neg X | U_i \wedge \bigwedge_{j \neq i} \neg U_j)$ for each cause U_i :
 $P(\neg X | U_1 \dots U_j, \neg U_{j+1} \dots \neg U_k) = \prod_{i=1}^j q_i$

⇒ number of parameters linear in number of parents!
- Ex: $q_{Cold} = 0.6$, $q_{Flu} = 0.2$, $q_{Malaria} = 0.1$:

Building Conditional Probability Tables, CPTs

- Problem: CPT grow exponentially with number of parents
 - If the causes don't interact: use a Noisy-OR distribution (generalization of logical or)
 - assume parents U_1, \dots, U_k include all causes (can add leak node representing "other causes"):
 $P(\neg X | \neg U_1 \dots \neg U_k) = 1.0$
 - assume independent failure probability $q_i \stackrel{\text{def}}{=} P(\neg X | U_i \wedge \bigwedge_{j \neq i} \neg U_j)$ for each cause U_i :
 $P(\neg X | U_1 \dots U_j, \neg U_{j+1} \dots \neg U_k) = \prod_{i=1}^j q_i$
- ⇒ number of parameters linear in number of parents!
- Ex: $q_{\text{Cold}} = 0.6$, $q_{\text{Flu}} = 0.2$, $q_{\text{Malaria}} = 0.1$:

<i>Cold</i>	<i>Flu</i>	<i>Malaria</i>	$P(\text{Fever})$	$P(\neg \text{Fever})$
F	F	F	0.0	1.0
F	F	T	0.9	0.1
F	T	F	0.8	0.2
F	T	T	0.98	$0.02 = 0.2 \times 0.1$
T	F	F	0.4	0.6
T	F	T	0.94	$0.06 = 0.6 \times 0.1$
T	T	F	0.88	$0.12 = 0.6 \times 0.2$
T	T	T	0.988	$0.012 = 0.6 \times 0.2 \times 0.1$

Building Conditional Probability Tables, CPTs

- Problem: CPT grow exponentially with number of parents
- If the causes don't interact: use a Noisy-OR distribution (generalization of logical or)
 - assume parents U_1, \dots, U_k include all causes (can add leak node representing "other causes"):
 $P(\neg X | \neg U_1 \dots \neg U_k) = 1.0$
 - assume independent failure probability $q_i \stackrel{\text{def}}{=} P(\neg X | U_i \wedge \bigwedge_{j \neq i} \neg U_j)$ for each cause U_i :
 $P(\neg X | U_1 \dots U_j, \neg U_{j+1} \dots \neg U_k) = \prod_{i=1}^j q_i$
- ⇒ number of parameters linear in number of parents!
- Ex: $q_{\text{Cold}} = 0.6$, $q_{\text{Flu}} = 0.2$, $q_{\text{Malaria}} = 0.1$:

<i>Cold</i>	<i>Flu</i>	<i>Malaria</i>	$P(\text{Fever})$	$P(\neg \text{Fever})$
F	F	F	0.0	1.0
F	F	T	0.9	0.1
F	T	F	0.8	0.2
F	T	T	0.98	$0.02 = 0.2 \times 0.1$
T	F	F	0.4	0.6
T	F	T	0.94	$0.06 = 0.6 \times 0.1$
T	T	F	0.88	$0.12 = 0.6 \times 0.2$
T	T	T	0.988	$0.012 = 0.6 \times 0.2 \times 0.1$

Building Conditional Probability Tables, CPTs

- Problem: CPT grow exponentially with number of parents
- If the causes don't interact: use a Noisy-OR distribution (generalization of logical or)
 - assume parents U_1, \dots, U_k include all causes (can add leak node representing "other causes"):
 $P(\neg X | \neg U_1 \dots \neg U_k) = 1.0$
 - assume independent failure probability $q_i \stackrel{\text{def}}{=} P(\neg X | U_i \wedge \bigwedge_{j \neq i} \neg U_j)$ for each cause U_i :
 $P(\neg X | U_1 \dots U_j, \neg U_{j+1} \dots \neg U_k) = \prod_{i=1}^j q_i$ \Rightarrow number of parameters linear in number of parents!
- Ex: $q_{\text{Cold}} = 0.6$, $q_{\text{Flu}} = 0.2$, $q_{\text{Malaria}} = 0.1$:

<i>Cold</i>	<i>Flu</i>	<i>Malaria</i>	$P(\text{Fever})$	$P(\neg \text{Fever})$
F	F	F	0.0	1.0
F	F	T	0.9	0.1
F	T	F	0.8	0.2
F	T	T	0.98	$0.02 = 0.2 \times 0.1$
T	F	F	0.4	0.6
T	F	T	0.94	$0.06 = 0.6 \times 0.1$
T	T	F	0.88	$0.12 = 0.6 \times 0.2$
T	T	T	0.988	$0.012 = 0.6 \times 0.2 \times 0.1$

1. Consider the probabilistic Wumpus World of previous chapter
 - (a) Describe it as a Bayesian network

- 1 Bayesian Networks
 - Basics
 - Global Semantics
 - Local Semantics
 - Independence Property: Markov Blanket
- 2 Constructing Bayesian Networks
- 3 Exact Inference with Bayesian Networks**
 - Inference by Enumeration
 - Inference by Variable Elimination
 - Complexity of Exact Inference

Exact inference in Bayesian Networks

- Given:

- X : the **query variable** (we assume one for simplicity)
- E/e : the set of **evidence variables** $\{E_1, \dots, E_m\}$ and of **evidence values** $\{e_1, \dots, e_m\}$
- Y/y : the set of **unknown variables** (aka **hidden variables**) $\{Y_1, \dots, Y_l\}$ and **unknown values** $\{y_1, \dots, y_l\}$

$$\Rightarrow \mathbf{X} = X \cup \mathbf{E} \cup \mathbf{Y}$$

A typical query asks for the posterior probability distribution:

$P(X | E=e)$ (also written $P(X | e)$)

- Ex: $P(\text{Burglar} | \text{JohnCalls} = \text{true}, \text{MaryCalls} = \text{true})$
 - query: *Burglar*
 - evidence variables: $\mathbf{E} = \{\text{JohnCalls}, \text{MaryCalls}\}$
 - hidden variables: $\mathbf{Y} = \{\text{Earthquake}, \text{Alarm}\}$

Exact inference in Bayesian Networks

- Given:

- X : the **query variable** (we assume one for simplicity)
- E/e : the set of **evidence variables** $\{E_1, \dots, E_m\}$ and of **evidence values** $\{e_1, \dots, e_m\}$
- Y/y : the set of **unknown variables** (aka **hidden variables**) $\{Y_1, \dots, Y_l\}$ and **unknown values** $\{y_1, \dots, y_l\}$

$$\Rightarrow \mathbf{X} = X \cup \mathbf{E} \cup \mathbf{Y}$$

A typical query asks for the posterior probability distribution:

$P(X | E=e)$ (also written $P(X | e)$)

- Ex: $P(\text{Burglar} | \text{JohnCalls} = \text{true}, \text{MaryCalls} = \text{true})$

- query: *Burglar*
- evidence variables: $\mathbf{E} = \{\text{JohnCalls}, \text{MaryCalls}\}$
- hidden variables: $\mathbf{Y} = \{\text{Earthquake}, \text{Alarm}\}$

- 1 Bayesian Networks
 - Basics
 - Global Semantics
 - Local Semantics
 - Independence Property: Markov Blanket
- 2 Constructing Bayesian Networks
- 3 Exact Inference with Bayesian Networks
 - Inference by Enumeration
 - Inference by Variable Elimination
 - Complexity of Exact Inference

Inference by Enumeration

- We defined a procedure for the task as: $\mathbf{P}(X|\mathbf{e}) = \alpha\mathbf{P}(X, \mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})$

⇒ $\mathbf{P}(X, \mathbf{e}, \mathbf{y})$ can be rewritten as product of prior and conditional probabilities according to the Bayesian Network

- then apply factorization and simplify algebraically when possible

- Ex:

$$\begin{aligned} \mathbf{P}(B|j, m) &= \\ \alpha \sum_e \sum_a \mathbf{P}(B, e, a, j, m) &= \\ \alpha \sum_e \sum_a \mathbf{P}(B)P(e)\mathbf{P}(a|B, e)P(j|a)P(m|a) &= \\ \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e)P(j|a)P(m|a) & \end{aligned}$$

⇒ $P(b|j, m) =$
 $\alpha P(b) \sum_e P(e) \sum_a P(a|b, e)P(j|a)P(m|a)$

- Recursive depth-first enumeration:
 $O(n)$ space, $O(2^n)$ time with n propositional variables
- Enumeration can be inefficient: repeated computation

Inference by Enumeration

- We defined a procedure for the task as: $\mathbf{P}(X|\mathbf{e}) = \alpha\mathbf{P}(X, \mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})$

⇒ $\mathbf{P}(X, \mathbf{e}, \mathbf{y})$ can be rewritten as product of prior and conditional probabilities according to the Bayesian Network

- then apply factorization and simplify algebraically when possible

- Ex:

$$\begin{aligned} \mathbf{P}(B|j, m) &= \\ \alpha \sum_e \sum_a \mathbf{P}(B, e, a, j, m) &= \\ \alpha \sum_e \sum_a \mathbf{P}(B)P(e)\mathbf{P}(a|B, e)P(j|a)P(m|a) &= \\ \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e)P(j|a)P(m|a) & \end{aligned}$$

⇒ $P(b|j, m) =$
 $\alpha P(b) \sum_e P(e) \sum_a P(a|b, e)P(j|a)P(m|a)$

- Recursive depth-first enumeration:
 $O(n)$ space, $O(2^n)$ time with n propositional variables
- Enumeration can be inefficient: repeated computation

Inference by Enumeration

- We defined a procedure for the task as: $\mathbf{P}(X|\mathbf{e}) = \alpha\mathbf{P}(X, \mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})$

⇒ $\mathbf{P}(X, \mathbf{e}, \mathbf{y})$ can be rewritten as product of prior and conditional probabilities according to the Bayesian Network

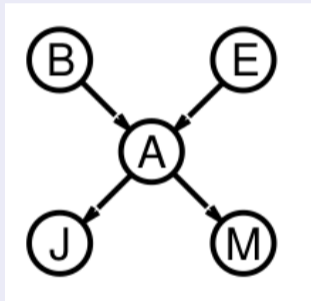
- then apply factorization and simplify algebraically when possible

- Ex:

$$\begin{aligned} \mathbf{P}(B|j, m) &= \\ \alpha \sum_e \sum_a \mathbf{P}(B, e, a, j, m) &= \\ \alpha \sum_e \sum_a \mathbf{P}(B)P(e)\mathbf{P}(a|B, e)P(j|a)P(m|a) &= \\ \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e)P(j|a)P(m|a) & \end{aligned}$$

⇒ $P(b|j, m) =$
 $\alpha P(b) \sum_e P(e) \sum_a P(a|b, e)P(j|a)P(m|a)$

- Recursive depth-first enumeration:
 $O(n)$ space, $O(2^n)$ time with n propositional variables
- Enumeration can be inefficient: repeated computation



(© S. Russell & P. Norwig, AIMA)

Inference by Enumeration

- We defined a procedure for the task as: $\mathbf{P}(X|\mathbf{e}) = \alpha\mathbf{P}(X, \mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})$

⇒ $\mathbf{P}(X, \mathbf{e}, \mathbf{y})$ can be rewritten as product of prior and conditional probabilities according to the Bayesian Network

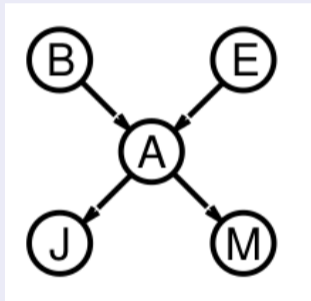
- then apply factorization and simplify algebraically when possible

- Ex:

$$\begin{aligned} \mathbf{P}(B|j, m) &= \\ \alpha \sum_e \sum_a \mathbf{P}(B, e, a, j, m) &= \\ \alpha \sum_e \sum_a \mathbf{P}(B)P(e)\mathbf{P}(a|B, e)P(j|a)P(m|a) &= \\ \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e)P(j|a)P(m|a) & \end{aligned}$$

⇒ $P(b|j, m) =$
 $\alpha P(b) \sum_e P(e) \sum_a P(a|b, e)P(j|a)P(m|a)$

- Recursive depth-first enumeration:
 $O(n)$ space, $O(2^n)$ time with n propositional variables
- Enumeration can be inefficient: repeated computation



(© S. Russell & P. Norwig, AIMA)

Inference by Enumeration

- We defined a procedure for the task as: $\mathbf{P}(X|\mathbf{e}) = \alpha\mathbf{P}(X, \mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})$

⇒ $\mathbf{P}(X, \mathbf{e}, \mathbf{y})$ can be rewritten as product of prior and conditional probabilities according to the Bayesian Network

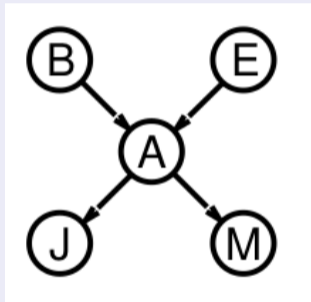
- then apply factorization and simplify algebraically when possible

- Ex:

$$\begin{aligned} \mathbf{P}(B|j, m) &= \\ \alpha \sum_e \sum_a \mathbf{P}(B, e, a, j, m) &= \\ \alpha \sum_e \sum_a \mathbf{P}(B)P(e)\mathbf{P}(a|B, e)P(j|a)P(m|a) &= \\ \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e)P(j|a)P(m|a) \end{aligned}$$

⇒ $P(b|j, m) =$
 $\alpha P(b) \sum_e P(e) \sum_a P(a|b, e)P(j|a)P(m|a)$

- Recursive depth-first enumeration:
 $O(n)$ space, $O(2^n)$ time with n propositional variables
- Enumeration can be inefficient: repeated computation



(© S. Russell & P. Norwig, AIMA)

Inference by Enumeration

- We defined a procedure for the task as: $\mathbf{P}(X|\mathbf{e}) = \alpha\mathbf{P}(X, \mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})$

⇒ $\mathbf{P}(X, \mathbf{e}, \mathbf{y})$ can be rewritten as product of prior and conditional probabilities according to the Bayesian Network

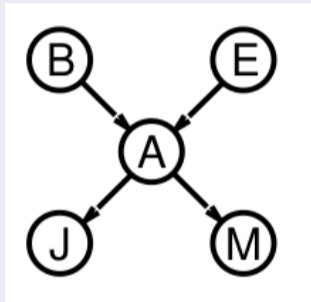
- then apply factorization and simplify algebraically when possible

- Ex:

$$\begin{aligned} \mathbf{P}(B|j, m) &= \\ \alpha \sum_e \sum_a \mathbf{P}(B, e, a, j, m) &= \\ \alpha \sum_e \sum_a \mathbf{P}(B)P(e)\mathbf{P}(a|B, e)P(j|a)P(m|a) &= \\ \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e)P(j|a)P(m|a) \end{aligned}$$

⇒ $P(b|j, m) =$
 $\alpha P(b) \sum_e P(e) \sum_a P(a|b, e)P(j|a)P(m|a)$

- Recursive depth-first enumeration:
 $O(n)$ space, $O(2^n)$ time with n propositional variables
- Enumeration can be inefficient: repeated computation



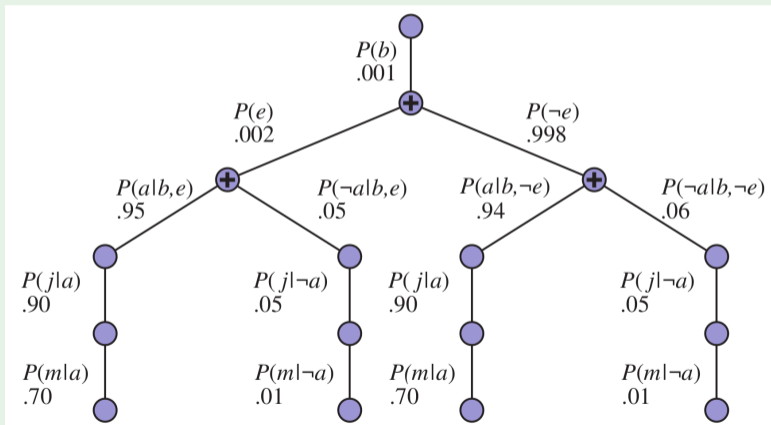
(© S. Russell & P. Norwig, AIMA)

Inference by Enumeration: Example

$$P(b|j, m) = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) P(j|a) P(m|a) = \alpha \cdot 0.00059224$$

$$P(\neg b|j, m) = \alpha P(\neg b) \sum_e P(e) \sum_a P(a|\neg b, e) P(j|a) P(m|a) = \alpha \cdot 0.0014919$$

$$\Rightarrow \mathbf{P}(B|j, m) = \alpha \cdot \langle 0.00059224, 0.0014919 \rangle = [\text{normal.}] \approx \langle 0.284, 0.716 \rangle$$



© S. Russell & P. Norwig, AIMA

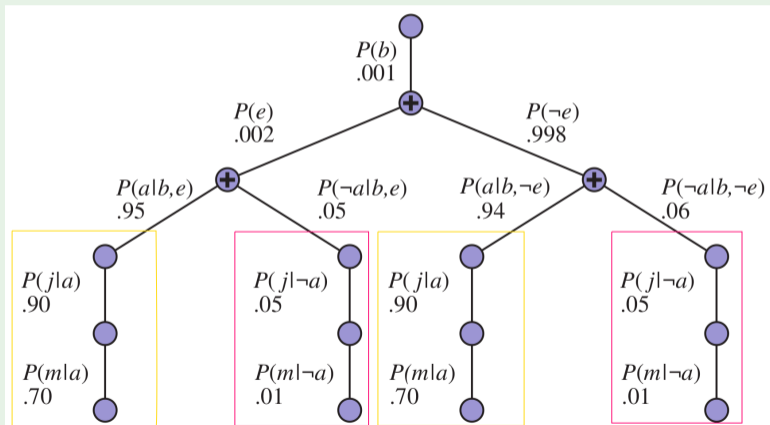
Repeated computation: $P(j|a)P(m|a)$ & $P(j|\neg a)P(m|\neg a)$ for each value of e

Inference by Enumeration: Example

$$P(b|j, m) = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) P(j|a) P(m|a) = \alpha \cdot 0.00059224$$

$$P(\neg b|j, m) = \alpha P(\neg b) \sum_e P(e) \sum_a P(a|\neg b, e) P(j|a) P(m|a) = \alpha \cdot 0.0014919$$

$$\Rightarrow \mathbf{P}(B|j, m) = \alpha \cdot \langle 0.00059224, 0.0014919 \rangle = [\text{normal.}] \approx \langle 0.284, 0.716 \rangle$$



© S. Russell & P. Norwig, AIMA

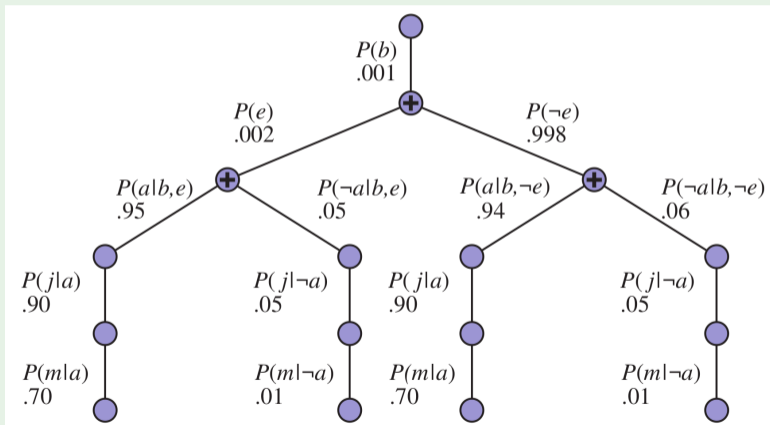
Repeated computation: $P(j|a)P(m|a)$ & $P(j|\neg a)P(m|\neg a)$ for each value of e

Inference by Enumeration: Example

$$P(b|j, m) = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) P(j|a) P(m|a) = \alpha \cdot 0.00059224$$

$$P(\neg b|j, m) = \alpha P(\neg b) \sum_e P(e) \sum_a P(a|\neg b, e) P(j|a) P(m|a) = \alpha \cdot 0.0014919$$

$$\Rightarrow P(B|j, m) = \alpha \cdot \langle 0.00059224, 0.0014919 \rangle = [\text{normal.}] \approx \langle 0.284, 0.716 \rangle$$



© S. Russell & P. Norwig, AIMA

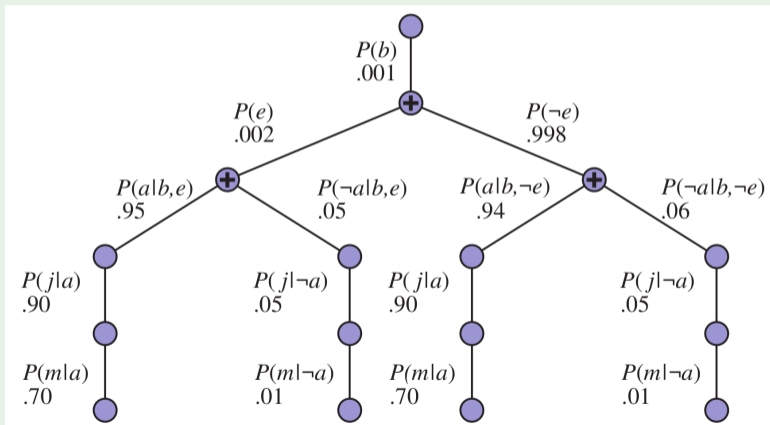
Repeated computation: $P(j|a)P(m|a)$ & $P(j|\neg a)P(m|\neg a)$ for each value of e

Inference by Enumeration: Example

$$P(b|j, m) = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) P(j|a) P(m|a) = \alpha \cdot 0.00059224$$

$$P(\neg b|j, m) = \alpha P(\neg b) \sum_e P(e) \sum_a P(a|\neg b, e) P(j|a) P(m|a) = \alpha \cdot 0.0014919$$

$$\Rightarrow \mathbf{P}(B|j, m) = \alpha \cdot \langle 0.00059224, 0.0014919 \rangle = [\text{normal.}] \approx \langle 0.284, 0.716 \rangle$$



© S. Russell & P. Norwig, AIMA

Repeated computation: $P(j|a)P(m|a)$ & $P(j|\neg a)P(m|\neg a)$ for each value of e

Enumeration Algorithm

function ENUMERATION-ASK(X, \mathbf{e}, bn) **returns** a distribution over X **computes** $\mathbf{P}(X \mid \mathbf{e})$

inputs: X , the query variable

\mathbf{e} , observed values for variables \mathbf{E}

bn , a Bayes net with variables $\{X\} \cup \mathbf{E} \cup \mathbf{Y}$ /* $\mathbf{Y} = \text{hidden variables}$ */

$\mathbf{Q}(X) \leftarrow$ a distribution over X , initially empty

for each value x_i of X **do**

$\mathbf{Q}(x_i) \leftarrow$ ENUMERATE-ALL($bn.VARS, \mathbf{e}_{x_i}$) **computes** $\mathbf{P}(x_i, \mathbf{Y}, \mathbf{e})$ (single probability value)

where \mathbf{e}_{x_i} is \mathbf{e} extended with $X = x_i$

return NORMALIZE($\mathbf{Q}(X)$)

function ENUMERATE-ALL($vars, \mathbf{e}$) **returns** a real number

if EMPTY?($vars$) **then return** 1.0

$Y \leftarrow$ FIRST($vars$)

if Y has value y in \mathbf{e}

then return $P(y \mid \text{parents}(Y)) \times$ ENUMERATE-ALL(REST($vars$), \mathbf{e}) **X or evidence var**

else return $\sum_y P(y \mid \text{parents}(Y)) \times$ ENUMERATE-ALL(REST($vars$), \mathbf{e}_y) **hidden var**

where \mathbf{e}_y is \mathbf{e} extended with $Y = y$

1. Consider the probabilistic Wumpus World of previous chapter
 - (a) Describe it as a Bayesian network
 - (b) Compute the query $P(P_{1,3}|b^*, p^*)$ via enumeration
 - (c) Compare the result with that of the example in Ch. 13

- 1 Bayesian Networks
 - Basics
 - Global Semantics
 - Local Semantics
 - Independence Property: Markov Blanket
- 2 Constructing Bayesian Networks
- 3 Exact Inference with Bayesian Networks
 - Inference by Enumeration
 - Inference by Variable Elimination
 - Complexity of Exact Inference

Inference by Variable Elimination

- Variable elimination:
 - carry out summations right-to-left (i.e., bottom-up in the tree)
 - store intermediate results (factors) to avoid recomputation

• Ex: $P(B|j, m)$

$$\begin{aligned} & \overbrace{\alpha P(B)}^{f_1(B)} \sum_e \overbrace{P(e)}^{f_2(E)} \sum_a \overbrace{P(a|B, e)}^{f_3(A, B, E)} \overbrace{P(j|a)}^{f_4(A)} \overbrace{P(m|a)}^{f_5(A)} \\ &= \alpha f_1(B) \times \sum_e f_2(E) \times \sum_a f_3(A, B, E) \times f_4(A) \times f_5(A) \\ &= \alpha f_1(B) \times \sum_e f_2(E) \times f_6(B, E) \quad (\text{sum out } A) \\ &= \alpha f_1(B) \times f_7(B) \quad (\text{sum out } E) \\ &= \alpha f_8(B) \end{aligned}$$

-
- “ \times ” is the pointwise product (see later)
-
-

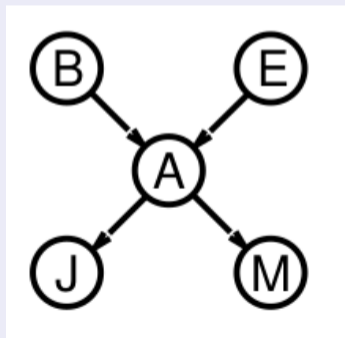
Inference by Variable Elimination

- Variable elimination:
 - carry out summations right-to-left (i.e., bottom-up in the tree)
 - store intermediate results (factors) to avoid recomputation

- Ex: $P(B|j, m)$

$$\begin{aligned} & \overbrace{\alpha P(B)}^{f_1(B)} \sum_e \overbrace{P(e)}^{f_2(E)} \sum_a \overbrace{P(a|B, e)}^{f_3(A, B, E)} \overbrace{P(j|a)}^{f_4(A)} \overbrace{P(m|a)}^{f_5(A)} \\ &= \alpha f_1(B) \times \sum_e f_2(E) \times \sum_a f_3(A, B, E) \times f_4(A) \times f_5(A) \\ &= \alpha f_1(B) \times \sum_e f_2(E) \times f_6(B, E) \quad (\text{sum out } A) \\ &= \alpha f_1(B) \times f_7(B) \quad (\text{sum out } E) \\ &= \alpha f_8(B) \end{aligned}$$

-
- “ \times ” is the pointwise product (see later)
-
-



Inference by Variable Elimination

- Variable elimination:

- carry out summations right-to-left (i.e., bottom-up in the tree)
- store intermediate results (factors) to avoid recomputation

- Ex: $P(B|j, m)$

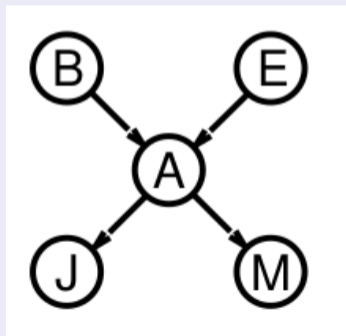
$$\begin{aligned}
 & \alpha \overbrace{P(B)}^{f_1(B)} \sum_e \overbrace{P(e)}^{f_2(E)} \sum_a \overbrace{P(a|B, e)}^{f_3(A, B, E)} \overbrace{P(j|a)}^{f_4(A)} \overbrace{P(m|a)}^{f_5(A)} \\
 & = \alpha f_1(B) \times \sum_e f_2(E) \times \sum_a f_3(A, B, E) \times f_4(A) \times f_5(A) \\
 & = \alpha f_1(B) \times \sum_e f_2(E) \times f_6(B, E) \quad (\text{sum out } A) \\
 & = \alpha f_1(B) \times f_7(B) \quad (\text{sum out } E) \\
 & = \alpha f_8(B)
 \end{aligned}$$

- $f_5(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(m|a) \\ P(m|\neg a) \end{bmatrix}$, $f_4(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(j|a) \\ P(j|\neg a) \end{bmatrix}$, ...

- " \times " is the pointwise product (see later)

- $f_6(B, E) = \sum_a f_3(A, B, E) \times f_4(A) \times f_5(A) =$
 $(f_3(a, B, E) \times f_4(a) \times f_5(a)) + (f_3(\neg a, B, E) \times f_4(\neg a) \times f_5(\neg a))$

- $f_7(B) = \sum_e f_2(E) \times f_6(B, E) =$
 $(f_2(e) \times f_6(B, e)) + (f_2(\neg e) \times f_6(B, \neg e))$



Inference by Variable Elimination

- Variable elimination:

- carry out summations right-to-left (i.e., bottom-up in the tree)
- store intermediate results (factors) to avoid recomputation

- Ex: $P(B|j, m)$

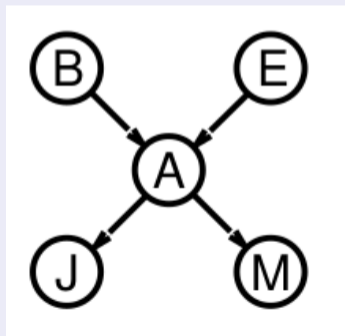
$$\begin{aligned}
 & \alpha \overbrace{P(B)}^{f_1(B)} \sum_e \overbrace{P(e)}^{f_2(E)} \sum_a \overbrace{P(a|B, e)}^{f_3(A, B, E)} \overbrace{P(j|a)}^{f_4(A)} \overbrace{P(m|a)}^{f_5(A)} \\
 & = \alpha f_1(B) \times \sum_e f_2(E) \times \sum_a f_3(A, B, E) \times f_4(A) \times f_5(A) \\
 & = \alpha f_1(B) \times \sum_e f_2(E) \times f_6(B, E) \quad (\text{sum out } A) \\
 & = \alpha f_1(B) \times f_7(B) \quad (\text{sum out } E) \\
 & = \alpha f_8(B)
 \end{aligned}$$

- $f_5(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(m|a) \\ P(m|\neg a) \end{bmatrix}$, $f_4(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(j|a) \\ P(j|\neg a) \end{bmatrix}$, ...

- " \times " is the pointwise product (see later)

- $f_6(B, E) = \sum_a f_3(A, B, E) \times f_4(A) \times f_5(A) =$
 $(f_3(a, B, E) \times f_4(a) \times f_5(a)) + (f_3(\neg a, B, E) \times f_4(\neg a) \times f_5(\neg a))$

- $f_7(B) = \sum_e f_2(E) \times f_6(B, E) =$
 $(f_2(e) \times f_6(B, e)) + (f_2(\neg e) \times f_6(B, \neg e))$



Inference by Variable Elimination

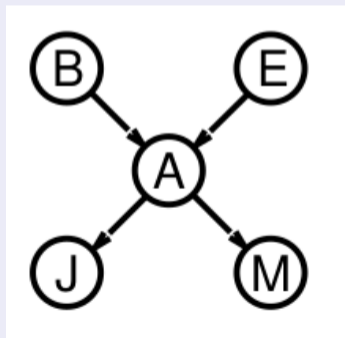
- Variable elimination:

- carry out summations right-to-left (i.e., bottom-up in the tree)
- store intermediate results (factors) to avoid recomputation

- Ex: $P(B|j, m)$

$$\begin{aligned}
 & \alpha \overbrace{P(B)}^{f_1(B)} \sum_e \overbrace{P(e)}^{f_2(E)} \sum_a \overbrace{P(a|B, e)}^{f_3(A, B, E)} \overbrace{P(j|a)}^{f_4(A)} \overbrace{P(m|a)}^{f_5(A)} \\
 &= \alpha f_1(B) \times \sum_e f_2(E) \times \sum_a f_3(A, B, E) \times f_4(A) \times f_5(A) \\
 &= \alpha f_1(B) \times \sum_e f_2(E) \times f_6(B, E) \quad (\text{sum out } A) \\
 &= \alpha f_1(B) \times f_7(B) \quad (\text{sum out } E) \\
 &= \alpha f_8(B)
 \end{aligned}$$

- $f_5(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(m|a) \\ P(m|\neg a) \end{bmatrix}$, $f_4(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(j|a) \\ P(j|\neg a) \end{bmatrix}$, ...
- " \times " is the pointwise product (see later)
- $f_6(B, E) = \sum_a f_3(A, B, E) \times f_4(A) \times f_5(A) =$
 $(f_3(a, B, E) \times f_4(a) \times f_5(a)) + (f_3(\neg a, B, E) \times f_4(\neg a) \times f_5(\neg a))$
- $f_7(B) = \sum_e f_2(E) \times f_6(B, E) =$
 $(f_2(e) \times f_6(B, e)) + (f_2(\neg e) \times f_6(B, \neg e))$



Inference by Variable Elimination

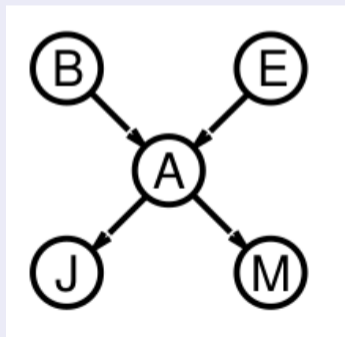
- Variable elimination:

- carry out summations right-to-left (i.e., bottom-up in the tree)
- store intermediate results (factors) to avoid recomputation

- Ex: $P(B|j, m)$

$$\begin{aligned}
 & \alpha \overbrace{P(B)}^{f_1(B)} \sum_e \overbrace{P(e)}^{f_2(E)} \sum_a \overbrace{P(a|B, e)}^{f_3(A, B, E)} \overbrace{P(j|a)}^{f_4(A)} \overbrace{P(m|a)}^{f_5(A)} \\
 &= \alpha f_1(B) \times \sum_e f_2(E) \times \sum_a f_3(A, B, E) \times f_4(A) \times f_5(A) \\
 &= \alpha f_1(B) \times \sum_e f_2(E) \times f_6(B, E) \quad (\text{sum out } A) \\
 &= \alpha f_1(B) \times f_7(B) \quad (\text{sum out } E) \\
 &= \alpha f_8(B)
 \end{aligned}$$

- $f_5(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(m|a) \\ P(m|\neg a) \end{bmatrix}$, $f_4(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(j|a) \\ P(j|\neg a) \end{bmatrix}$, ...
- " \times " is the **pointwise product** (see later)
- $f_6(B, E) = \sum_a f_3(A, B, E) \times f_4(A) \times f_5(A) = (f_3(a, B, E) \times f_4(a) \times f_5(a)) + (f_3(\neg a, B, E) \times f_4(\neg a) \times f_5(\neg a))$
- $f_7(B) = \sum_e f_2(E) \times f_6(B, E) = (f_2(e) \times f_6(B, e)) + (f_2(\neg e) \times f_6(B, \neg e))$



Inference by Variable Elimination

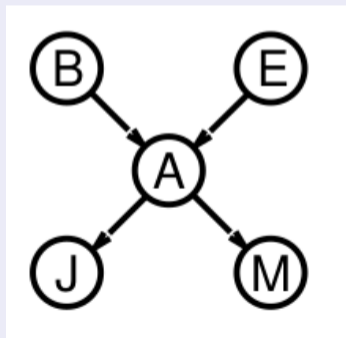
- Variable elimination:

- carry out summations right-to-left (i.e., bottom-up in the tree)
- store intermediate results (factors) to avoid recomputation

- Ex: $P(B|j, m)$

$$\begin{aligned} & \quad \quad \quad \mathbf{f}_1(B) \quad \quad \quad \mathbf{f}_2(E) \quad \quad \quad \mathbf{f}_3(A,B,E) \quad \mathbf{f}_4(A) \quad \mathbf{f}_5(A) \\ &= \alpha \overbrace{P(B)}^{\mathbf{f}_1(B)} \sum_e \overbrace{P(e)}^{\mathbf{f}_2(E)} \sum_a \overbrace{P(a|B, e)}^{\mathbf{f}_3(A,B,E)} \overbrace{P(j|a)}^{\mathbf{f}_4(A)} \overbrace{P(m|a)}^{\mathbf{f}_5(A)} \\ &= \alpha \mathbf{f}_1(B) \times \sum_e \mathbf{f}_2(E) \times \sum_a \mathbf{f}_3(A, B, E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A) \\ &= \alpha \mathbf{f}_1(B) \times \sum_e \mathbf{f}_2(E) \times \mathbf{f}_6(B, E) \quad (\text{sum out } A) \\ &= \alpha \mathbf{f}_1(B) \times \mathbf{f}_7(B) \quad (\text{sum out } E) \\ &= \alpha \mathbf{f}_8(B) \end{aligned}$$

- $\mathbf{f}_5(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(m|a) \\ P(m|\neg a) \end{bmatrix}$, $\mathbf{f}_4(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(j|a) \\ P(j|\neg a) \end{bmatrix}$, ...
- “ \times ” is the **pointwise product** (see later)
- $\mathbf{f}_6(B, E) = \sum_a \mathbf{f}_3(A, B, E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A) =$
 $(\mathbf{f}_3(a, B, E) \times \mathbf{f}_4(a) \times \mathbf{f}_5(a)) + (\mathbf{f}_3(\neg a, B, E) \times \mathbf{f}_4(\neg a) \times \mathbf{f}_5(\neg a))$
- $\mathbf{f}_7(B) = \sum_e \mathbf{f}_2(E) \times \mathbf{f}_6(B, E) =$
 $(\mathbf{f}_2(e) \times \mathbf{f}_6(B, e)) + (\mathbf{f}_2(\neg e) \times \mathbf{f}_6(B, \neg e))$



Variable Elimination: Basic Operations

- **Factor summation:** $\mathbf{f}_3(X_1, \dots, X_j) = \mathbf{f}_1(X_1, \dots, X_j) + \mathbf{f}_2(X_1, \dots, X_j)$

- standard matrix summation:

$$\begin{bmatrix} a_{11} & a_{21} & \dots \\ \dots & \dots & \dots \\ a_{n1} & a_{n1} & \dots \end{bmatrix} + \begin{bmatrix} b_{11} & b_{21} & \dots \\ \dots & \dots & \dots \\ b_{n1} & b_{n1} & \dots \end{bmatrix} = \begin{bmatrix} a_{11} + b_{11} & a_{21} + b_{21} & \dots \\ \dots & \dots & \dots \\ a_{n1} + b_{n1} & a_{n1} + b_{n1} & \dots \end{bmatrix}$$

- must have the same argument variables
- **Pointwise product:** Multiply the array elements for the same variable values

- Ex: $\mathbf{f}_4(A) \times \mathbf{f}_5(A) = \begin{bmatrix} P(j|a) \\ P(j|\neg a) \end{bmatrix} \times \begin{bmatrix} P(m|a) \\ P(m|\neg a) \end{bmatrix} = \begin{bmatrix} P(j|a)P(m|a) \\ P(j|\neg a)P(m|\neg a) \end{bmatrix}$

- General case:

$$\mathbf{f}_3(X_1, \dots, X_j, Y_1, \dots, Y_k, Z_1, \dots, Z_l) = \mathbf{f}_1(X_1, \dots, X_j, Y_1, \dots, Y_k) \times \mathbf{f}_2(Y_1, \dots, Y_k, Z_1, \dots, Z_l)$$

- union of arguments
- values: $\mathbf{f}_3(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{f}_1(\mathbf{x}, \mathbf{y}) \cdot \mathbf{f}_2(\mathbf{y}, \mathbf{z})$
- matrix size: $\mathbf{f}_1 : 2^{j+k}$, $\mathbf{f}_2 : 2^{k+l}$, $\mathbf{f}_3 : 2^{j+k+l}$

Variable Elimination: Basic Operations

- **Factor summation:** $\mathbf{f}_3(X_1, \dots, X_j) = \mathbf{f}_1(X_1, \dots, X_j) + \mathbf{f}_2(X_1, \dots, X_j)$

- standard matrix summation:

$$\begin{bmatrix} a_{11} & a_{21} & \dots \\ \dots & \dots & \dots \\ a_{n1} & a_{n1} & \dots \end{bmatrix} + \begin{bmatrix} b_{11} & b_{21} & \dots \\ \dots & \dots & \dots \\ b_{n1} & b_{n1} & \dots \end{bmatrix} = \begin{bmatrix} a_{11} + b_{11} & a_{21} + b_{21} & \dots \\ \dots & \dots & \dots \\ a_{n1} + b_{n1} & a_{n1} + b_{n1} & \dots \end{bmatrix}$$

- must have the same argument variables

- **Pointwise product:** Multiply the array elements for the same variable values

- Ex: $\mathbf{f}_4(A) \times \mathbf{f}_5(A) = \begin{bmatrix} P(j|a) \\ P(j|\neg a) \end{bmatrix} \times \begin{bmatrix} P(m|a) \\ P(m|\neg a) \end{bmatrix} = \begin{bmatrix} P(j|a)P(m|a) \\ P(j|\neg a)P(m|\neg a) \end{bmatrix}$

- General case:

$$\mathbf{f}_3(X_1, \dots, X_j, Y_1, \dots, Y_k, Z_1, \dots, Z_l) = \mathbf{f}_1(X_1, \dots, X_j, Y_1, \dots, Y_k) \times \mathbf{f}_2(Y_1, \dots, Y_k, Z_1, \dots, Z_l)$$

- union of arguments
- values: $\mathbf{f}_3(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{f}_1(\mathbf{x}, \mathbf{y}) \cdot \mathbf{f}_2(\mathbf{y}, \mathbf{z})$
- matrix size: $\mathbf{f}_1 : 2^{j+k}$, $\mathbf{f}_2 : 2^{k+l}$, $\mathbf{f}_3 : 2^{j+k+l}$

Variable Elimination: Basic Operations

- **Factor summation:** $\mathbf{f}_3(X_1, \dots, X_j) = \mathbf{f}_1(X_1, \dots, X_j) + \mathbf{f}_2(X_1, \dots, X_j)$

- standard matrix summation:

$$\begin{bmatrix} a_{11} & a_{21} & \dots \\ \dots & \dots & \dots \\ a_{n1} & a_{n1} & \dots \end{bmatrix} + \begin{bmatrix} b_{11} & b_{21} & \dots \\ \dots & \dots & \dots \\ b_{n1} & b_{n1} & \dots \end{bmatrix} = \begin{bmatrix} a_{11} + b_{11} & a_{21} + b_{21} & \dots \\ \dots & \dots & \dots \\ a_{n1} + b_{n1} & a_{n1} + b_{n1} & \dots \end{bmatrix}$$

- must have the same argument variables

- **Pointwise product:** Multiply the array elements for the same variable values

- Ex: $\mathbf{f}_4(A) \times \mathbf{f}_5(A) = \begin{bmatrix} P(j|a) \\ P(j|\neg a) \end{bmatrix} \times \begin{bmatrix} P(m|a) \\ P(m|\neg a) \end{bmatrix} = \begin{bmatrix} P(j|a)P(m|a) \\ P(j|\neg a)P(m|\neg a) \end{bmatrix}$

- General case:

$$\mathbf{f}_3(X_1, \dots, X_j, Y_1, \dots, Y_k, Z_1, \dots, Z_l) = \mathbf{f}_1(X_1, \dots, X_j, Y_1, \dots, Y_k) \times \mathbf{f}_2(Y_1, \dots, Y_k, Z_1, \dots, Z_l)$$

- union of arguments
- values: $\mathbf{f}_3(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathbf{f}_1(\mathbf{x}, \mathbf{y}) \cdot \mathbf{f}_2(\mathbf{y}, \mathbf{z})$
- matrix size: $\mathbf{f}_1 : 2^{j+k}$, $\mathbf{f}_2 : 2^{k+l}$, $\mathbf{f}_3 : 2^{j+k+l}$

Variable Elimination: Basic Operations

- $f_3(A, B, C) = f_1(A, B) \times f_2(B, C)$

- Summing out one variable: $f(B, C) = \sum_a f_3(A, B, C) = f_3(a, B, C) + f_3(\neg a, B, C) =$
 $\begin{bmatrix} 0.06 & 0.24 \\ 0.42 & 0.28 \end{bmatrix} + \begin{bmatrix} 0.18 & 0.72 \\ 0.06 & 0.04 \end{bmatrix} = \begin{bmatrix} 0.24 & 0.96 \\ 0.48 & 0.32 \end{bmatrix}$

A	B	$f_1(A, B)$	B	C	$f_2(B, C)$	A	B	C	$f_3(A, B, C)$
T	T	.3	T	T	.2	T	T	T	$.3 \times .2 = .06$
T	F	.7	T	F	.8	T	T	F	$.3 \times .8 = .24$
F	T	.9	F	T	.6	T	F	T	$.7 \times .6 = .42$
F	F	.1	F	F	.4	T	F	F	$.7 \times .4 = .28$
						F	T	T	$.9 \times .2 = .18$
						F	T	F	$.9 \times .8 = .72$
						F	F	T	$.1 \times .6 = .06$
						F	F	F	$.1 \times .4 = .04$

(© S. Russell & P. Norwig, AIMA)

Variable Elimination: Basic Operations

- $f_3(A, B, C) = f_1(A, B) \times f_2(B, C)$
- Summing out one variable: $f(B, C) = \sum_a f_3(A, B, C) = f_3(a, B, C) + f_3(\neg a, B, C) =$
 $\begin{bmatrix} 0.06 & 0.24 \\ 0.42 & 0.28 \end{bmatrix} + \begin{bmatrix} 0.18 & 0.72 \\ 0.06 & 0.04 \end{bmatrix} = \begin{bmatrix} 0.24 & 0.96 \\ 0.48 & 0.32 \end{bmatrix}$

A	B	$f_1(A, B)$	B	C	$f_2(B, C)$	A	B	C	$f_3(A, B, C)$
T	T	.3	T	T	.2	T	T	T	$.3 \times .2 = .06$
T	F	.7	T	F	.8	T	T	F	$.3 \times .8 = .24$
F	T	.9	F	T	.6	T	F	T	$.7 \times .6 = .42$
F	F	.1	F	F	.4	T	F	F	$.7 \times .4 = .28$
						F	T	T	$.9 \times .2 = .18$
						F	T	F	$.9 \times .8 = .72$
						F	F	T	$.1 \times .6 = .06$
						F	F	F	$.1 \times .4 = .04$

(© S. Russell & P. Norwig, AIMA)

Variable Elimination Algorithm

```
function ELIMINATION-ASK( $X, \mathbf{e}, bn$ ) returns a distribution over  $X$   
inputs:  $X$ , the query variable  
           $\mathbf{e}$ , observed values for variables  $\mathbf{E}$   
           $bn$ , a Bayesian network specifying joint distribution  $\mathbf{P}(X_1, \dots, X_n)$   
  
 $factors \leftarrow []$   
for each  $var$  in ORDER( $bn.VARS$ ) do  
     $factors \leftarrow [MAKE-FACTOR(var, \mathbf{e}) | factors]$   
    if  $var$  is a hidden variable then  $factors \leftarrow SUM-OUT(var, factors)$   
return NORMALIZE(POINTWISE-PRODUCT( $factors$ ))
```

(© S. Russell & P. Norwig, AIMA)

- Efficiency depends on variable ordering ORDER(...)
- Efficiency improvements:
 - factor out of summations factors not depending on sum variable
 - remove irrelevant variables

Variable Elimination Algorithm

```
function ELIMINATION-ASK( $X$ ,  $\mathbf{e}$ ,  $bn$ ) returns a distribution over  $X$   
inputs:  $X$ , the query variable  
           $\mathbf{e}$ , observed values for variables  $\mathbf{E}$   
           $bn$ , a Bayesian network specifying joint distribution  $\mathbf{P}(X_1, \dots, X_n)$   
  
 $factors \leftarrow []$   
for each  $var$  in ORDER( $bn.VARS$ ) do  
     $factors \leftarrow [MAKE-FACTOR(var, \mathbf{e}) | factors]$   
    if  $var$  is a hidden variable then  $factors \leftarrow SUM-OUT(var, factors)$   
return NORMALIZE(POINTWISE-PRODUCT( $factors$ ))
```

(© S. Russell & P. Norwig, AIMA)

- Efficiency depends on variable ordering ORDER(...)
- Efficiency improvements:
 - factor out of summations factors not depending on sum variable
 - remove irrelevant variables

Variable Elimination Algorithm

```
function ELIMINATION-ASK( $X, \mathbf{e}, bn$ ) returns a distribution over  $X$   
inputs:  $X$ , the query variable  
           $\mathbf{e}$ , observed values for variables  $\mathbf{E}$   
           $bn$ , a Bayesian network specifying joint distribution  $\mathbf{P}(X_1, \dots, X_n)$   
  
 $factors \leftarrow []$   
for each  $var$  in ORDER( $bn.VARS$ ) do  
     $factors \leftarrow [MAKE-FACTOR(var, \mathbf{e}) | factors]$   
    if  $var$  is a hidden variable then  $factors \leftarrow SUM-OUT(var, factors)$   
return NORMALIZE(POINTWISE-PRODUCT( $factors$ ))
```

(© S. Russell & P. Norwig, AIMA)

- Efficiency depends on variable ordering ORDER(...)
- Efficiency improvements:
 - factor out of summations factors not depending on sum variable
 - remove irrelevant variables

Factor Out Constant Factors

- If f_1, \dots, f_i do not depend on X , then move them out of a $\sum_x(\dots)$:

$$\begin{aligned}\sum_x f_1 \times \dots \times f_k &= \\ f_1 \times \dots \times f_i \sum_x (f_{i+1} \times \dots \times f_k) &= \\ f_1 \times \dots \times f_i \times f_X\end{aligned}$$

- Ex: $\sum_a \mathbf{f}_1(A, B) \times \mathbf{f}_2(B, C)$
 $= \mathbf{f}_2(B, C) \times \sum_a \mathbf{F}_1(A, B)$

- Ex: $\mathbf{P}(\text{JohnCalls} | \text{Burglary} = \text{true})$:

$$\begin{aligned}\mathbf{P}(J|b) &= \alpha \sum_e \sum_a \sum_m \mathbf{P}(J, m, b, e, a) = \\ \alpha \sum_e \sum_a \sum_m \mathbf{P}(b) \mathbf{P}(e) \mathbf{P}(a|b, e) \mathbf{P}(J|a) \mathbf{P}(m|a) &= \\ \alpha \mathbf{P}(b) \sum_e \mathbf{P}(e) \sum_a \mathbf{P}(a|b, e) \mathbf{P}(J|a) \sum_m \mathbf{P}(m|a)\end{aligned}$$

Factor Out Constant Factors

- If f_1, \dots, f_i do not depend on X , then move them out of a $\sum_x(\dots)$:

$$\begin{aligned}\sum_x f_1 \times \dots \times f_k &= \\ f_1 \times \dots \times f_i \sum_x (f_{i+1} \times \dots \times f_k) &= \\ f_1 \times \dots \times f_i \times f_X &\end{aligned}$$

- Ex: $\sum_a \mathbf{f}_1(A, B) \times \mathbf{f}_2(B, C)$
 $= \mathbf{f}_2(B, C) \times \sum_a \mathbf{F}_1(A, B)$

- Ex: $\mathbf{P}(\text{JohnCalls} | \text{Burglary} = \text{true})$:

$$\begin{aligned}\mathbf{P}(J|b) &= \alpha \sum_e \sum_a \sum_m \mathbf{P}(J, m, b, e, a) = \\ \alpha \sum_e \sum_a \sum_m \mathbf{P}(b) \mathbf{P}(e) \mathbf{P}(a|b, e) \mathbf{P}(J|a) \mathbf{P}(m|a) &= \\ \alpha \mathbf{P}(b) \sum_e \mathbf{P}(e) \sum_a \mathbf{P}(a|b, e) \mathbf{P}(J|a) \sum_m \mathbf{P}(m|a) &\end{aligned}$$

Factor Out Constant Factors

- If f_1, \dots, f_i do not depend on X , then move them out of a $\sum_x(\dots)$:

$$\sum_x f_1 \times \dots \times f_k =$$

$$f_1 \times \dots \times f_i \sum_x (f_{i+1} \times \dots \times f_k) =$$

$$f_1 \times \dots \times f_i \times f_x$$

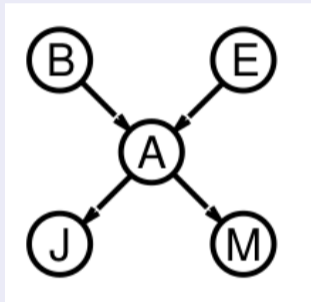
- Ex: $\sum_a \mathbf{f}_1(A, B) \times \mathbf{f}_2(B, C)$
 $= \mathbf{f}_2(B, C) \times \sum_a \mathbf{F}_1(A, B)$

- Ex: $\mathbf{P}(\text{JohnCalls} | \text{Burglary} = \text{true})$:

$$\mathbf{P}(J|b) = \alpha \sum_e \sum_a \sum_m \mathbf{P}(J, m, b, e, a) =$$

$$\alpha \sum_e \sum_a \sum_m P(b)P(e)P(a|b, e)\mathbf{P}(J|a)P(m|a) =$$

$$\alpha P(b) \sum_e P(e) \sum_a P(a|b, e)\mathbf{P}(J|a) \sum_m P(m|a)$$



(© S. Russell & P. Norvig, AIMA)

Factor Out Constant Factors

- If f_1, \dots, f_i do not depend on X , then move them out of a $\sum_x(\dots)$:

$$\sum_x f_1 \times \dots \times f_k =$$

$$f_1 \times \dots \times f_i \sum_x (f_{i+1} \times \dots \times f_k) =$$

$$f_1 \times \dots \times f_i \times f_X$$

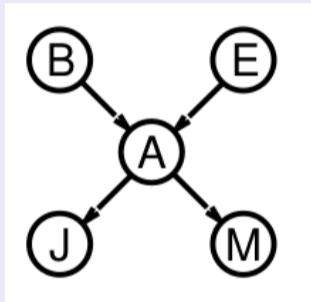
- Ex: $\sum_a \mathbf{f}_1(A, B) \times \mathbf{f}_2(B, C)$
 $= \mathbf{f}_2(B, C) \times \sum_a \mathbf{F}_1(A, B)$

- Ex: $\mathbf{P}(\text{JohnCalls} | \text{Burglary} = \text{true})$:

$$\mathbf{P}(J|b) = \alpha \sum_e \sum_a \sum_m \mathbf{P}(J, m, b, e, a) =$$

$$\alpha \sum_e \sum_a \sum_m P(b)P(e)P(a|b, e)\mathbf{P}(J|a)P(m|a) =$$

$$\alpha P(b) \sum_e P(e) \sum_a P(a|b, e)\mathbf{P}(J|a) \sum_m P(m|a)$$



(© S. Russell & P. Norvig, AIMA)

Factor Out Constant Factors

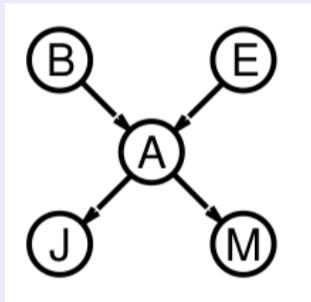
- If f_1, \dots, f_i do not depend on X , then move them out of a $\sum_x(\dots)$:

$$\begin{aligned}\sum_x f_1 \times \dots \times f_k &= \\ f_1 \times \dots \times f_i \sum_x (f_{i+1} \times \dots \times f_k) &= \\ f_1 \times \dots \times f_i \times f_x &\end{aligned}$$

- Ex: $\sum_a \mathbf{f}_1(A, B) \times \mathbf{f}_2(B, C)$
 $= \mathbf{f}_2(B, C) \times \sum_a \mathbf{F}_1(A, B)$

- Ex: $\mathbf{P}(\text{JohnCalls} | \text{Burglary} = \text{true})$:

$$\begin{aligned}\mathbf{P}(J|b) &= \alpha \sum_e \sum_a \sum_m \mathbf{P}(J, m, b, e, a) = \\ \alpha \sum_e \sum_a \sum_m P(b)P(e)P(a|b, e)\mathbf{P}(J|a)P(m|a) &= \\ \alpha P(b) \sum_e P(e) \sum_a P(a|b, e)\mathbf{P}(J|a) \sum_m P(m|a) &\end{aligned}$$



(© S. Russell & P. Norvig, AIMA)

Factor Out Constant Factors

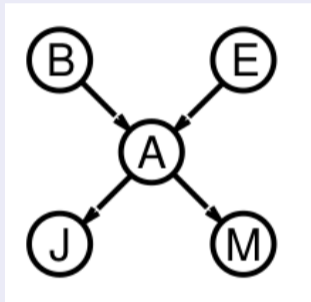
- If f_1, \dots, f_i do not depend on X , then move them out of a $\sum_x(\dots)$:

$$\begin{aligned}\sum_x f_1 \times \dots \times f_k &= \\ f_1 \times \dots \times f_i \sum_x (f_{i+1} \times \dots \times f_k) &= \\ f_1 \times \dots \times f_i \times f_x\end{aligned}$$

- Ex: $\sum_a \mathbf{f}_1(A, B) \times \mathbf{f}_2(B, C)$
 $= \mathbf{f}_2(B, C) \times \sum_a \mathbf{F}_1(A, B)$

- Ex: $\mathbf{P}(\text{JohnCalls} | \text{Burglary} = \text{true})$:

$$\begin{aligned}\mathbf{P}(J|b) &= \alpha \sum_e \sum_a \sum_m \mathbf{P}(J, m, b, e, a) = \\ \alpha \sum_e \sum_a \sum_m P(b)P(e)P(a|b, e)\mathbf{P}(J|a)P(m|a) &= \\ \alpha P(b) \sum_e P(e) \sum_a P(a|b, e)\mathbf{P}(J|a) \sum_m P(m|a)\end{aligned}$$



(© S. Russell & P. Norwig, AIMA)

Remove Irrelevant Variables

- Sometimes we have summations like $\sum_y P(y|\dots)$
 - $\sum_y P(y|\dots) = 1 \implies$ can be dropped

- Ex: $P(\text{JohnCalls} | \text{Burglary} = \text{true})$:

$$P(J|b) = \dots = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) P(J|a) \overbrace{\sum_m P(m|a)}^1 = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) P(J|a)$$

- Theorem: For query X and evidence \mathbf{E} ,

Y is irrelevant unless $Y \in \text{Ancestors}(X \cup \mathbf{E})$

- Ex: $X = \text{JohnCalls}$, $\mathbf{E} = \{\text{Burglary}\}$, and $\text{Ancestors}(\{X\} \cup \mathbf{E}) = \{\text{Alarm}, \text{Earthquake}\}$

$\implies \text{MaryCalls}$ is irrelevant

- Related to backward-chaining with Horn clauses

Remove Irrelevant Variables

- Sometimes we have summations like $\sum_y P(y|\dots)$
 - $\sum_y P(y|\dots) = 1 \implies$ can be dropped

- Ex: $P(\text{JohnCalls} | \text{Burglary} = \text{true})$:

$$P(J|b) = \dots = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) P(J|a) \overbrace{\sum_m P(m|a)}^1 = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) P(J|a)$$

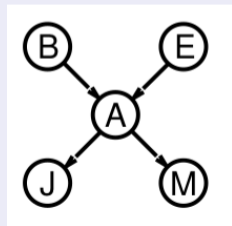
- Theorem: For query X and evidence \mathbf{E} ,

Y is irrelevant unless $Y \in \text{Ancestors}(X \cup \mathbf{E})$

- Ex: $X = \text{JohnCalls}$, $\mathbf{E} = \{\text{Burglary}\}$, and $\text{Ancestors}(\{X\} \cup \mathbf{E}) = \{\text{Alarm}, \text{Earthquake}\}$

\implies MaryCalls is irrelevant

- Related to backward-chaining with Horn clauses



Remove Irrelevant Variables

- Sometimes we have summations like $\sum_y P(y|\dots)$
 - $\sum_y P(y|\dots) = 1 \implies$ can be dropped

- Ex: $P(\text{JohnCalls} | \text{Burglary} = \text{true})$:

$$P(J|b) = \dots = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) P(J|a) \overbrace{\sum_m P(m|a)}^1 = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) P(J|a)$$

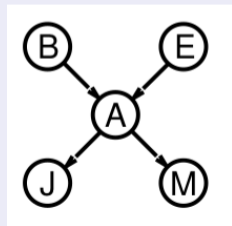
- Theorem: For query X and evidence \mathbf{E} ,

Y is irrelevant unless $Y \in \text{Ancestors}(X \cup \mathbf{E})$

- Ex: $X = \text{JohnCalls}$, $\mathbf{E} = \{\text{Burglary}\}$, and $\text{Ancestors}(\{X\} \cup \mathbf{E}) = \{\text{Alarm}, \text{Earthquake}\}$

\implies MaryCalls is irrelevant

- Related to backward-chaining with Horn clauses



Remove Irrelevant Variables

- Sometimes we have summations like $\sum_y P(y|\dots)$
 - $\sum_y P(y|\dots) = 1 \implies$ can be dropped

- Ex: $\mathbf{P}(\text{JohnCalls} | \text{Burglary} = \text{true})$:

$$\mathbf{P}(J|b) = \dots = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) \mathbf{P}(J|a) \overbrace{\sum_m P(m|a)}^1 = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) \mathbf{P}(J|a)$$

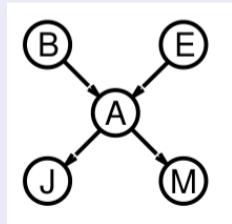
- Theorem: For query X and evidence \mathbf{E} ,

Y is irrelevant unless $Y \in \text{Ancestors}(X \cup \mathbf{E})$

- Ex: $X = \text{JohnCalls}$, $\mathbf{E} = \{\text{Burglary}\}$, and $\text{Ancestors}(\{X\} \cup \mathbf{E}) = \{\text{Alarm}, \text{Earthquake}\}$

$\implies \text{MaryCalls}$ is irrelevant

- Related to backward-chaining with Horn clauses



Remove Irrelevant Variables

- Sometimes we have summations like $\sum_y P(y|\dots)$
 - $\sum_y P(y|\dots) = 1 \implies$ can be dropped

- Ex: $\mathbf{P}(\text{JohnCalls} | \text{Burglary} = \text{true})$:

$$\mathbf{P}(J|b) = \dots = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) \mathbf{P}(J|a) \overbrace{\sum_m P(m|a)}^1 = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) \mathbf{P}(J|a)$$

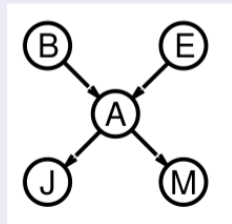
- Theorem: For query X and evidence \mathbf{E} ,

Y is irrelevant unless $Y \in \text{Ancestors}(X \cup \mathbf{E})$

- Ex: $X = \text{JohnCalls}$, $\mathbf{E} = \{\text{Burglary}\}$, and $\text{Ancestors}(\{X\} \cup \mathbf{E}) = \{\text{Alarm}, \text{Earthquake}\}$

$\implies \text{MaryCalls}$ is irrelevant

- Related to backward-chaining with Horn clauses



Remove Irrelevant Variables

- Sometimes we have summations like $\sum_y P(y|\dots)$
 - $\sum_y P(y|\dots) = 1 \implies$ can be dropped

- Ex: $\mathbf{P}(\text{JohnCalls} | \text{Burglary} = \text{true})$:

$$\mathbf{P}(J|b) = \dots = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) \mathbf{P}(J|a) \overbrace{\sum_m P(m|a)}^1 = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) \mathbf{P}(J|a)$$

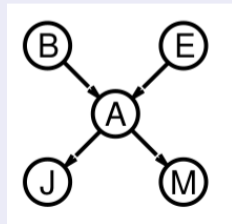
- Theorem: For query X and evidence \mathbf{E} ,

Y is irrelevant unless $Y \in \text{Ancestors}(X \cup \mathbf{E})$

- Ex: $X = \text{JohnCalls}$, $\mathbf{E} = \{\text{Burglary}\}$, and $\text{Ancestors}(\{X\} \cup \mathbf{E}) = \{\text{Alarm}, \text{Earthquake}\}$

\implies MaryCalls is irrelevant

- Related to backward-chaining with Horn clauses



Remove Irrelevant Variables

- Sometimes we have summations like $\sum_y P(y|\dots)$
 - $\sum_y P(y|\dots) = 1 \implies$ can be dropped

- Ex: $\mathbf{P}(\text{JohnCalls} | \text{Burglary} = \text{true})$:

$$\mathbf{P}(J|b) = \dots = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) \mathbf{P}(J|a) \overbrace{\sum_m P(m|a)}^1 = \alpha P(b) \sum_e P(e) \sum_a P(a|b, e) \mathbf{P}(J|a)$$

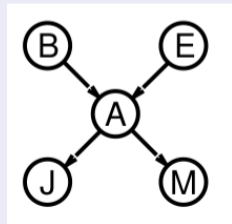
- Theorem: For query X and evidence \mathbf{E} ,

Y is irrelevant unless $Y \in \text{Ancestors}(X \cup \mathbf{E})$

- Ex: $X = \text{JohnCalls}$, $\mathbf{E} = \{\text{Burglary}\}$, and $\text{Ancestors}(\{X\} \cup \mathbf{E}) = \{\text{Alarm}, \text{Earthquake}\}$

\implies MaryCalls is irrelevant

- Related to backward-chaining with Horn clauses



Exercises

1. Try to compute queries (your choice) on the burglary problem using variable elimination
2. Consider the probabilistic Wumpus World of previous chapter
 - (a) Describe it as a Bayesian network
 - (b) Compute the query $P(P_{1,3}|b^*, p^*)$ via variable elimination
 - (c) Compare the result with that of the example in Ch. 13

Exercises

1. Try to compute queries (your choice) on the burglary problem using variable elimination
2. Consider the probabilistic Wumpus World of previous chapter
 - (a) Describe it as a Bayesian network
 - (b) Compute the query $P(P_{1,3}|b^*, p^*)$ via variable elimination
 - (c) Compare the result with that of the example in Ch. 13

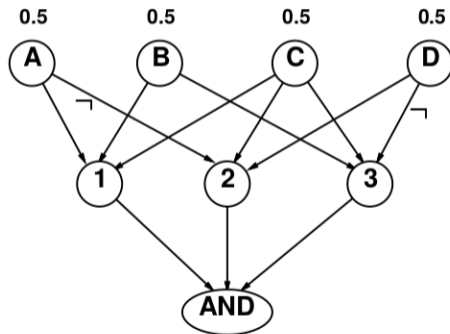
- 1 Bayesian Networks
 - Basics
 - Global Semantics
 - Local Semantics
 - Independence Property: Markov Blanket
- 2 Constructing Bayesian Networks
- 3 Exact Inference with Bayesian Networks**
 - Inference by Enumeration
 - Inference by Variable Elimination
 - Complexity of Exact Inference**

Complexity of Exact Inference

- We can reduce SAT to exact inference in Bayesian Networks

- $P(AND = \top) = \frac{|Models(\varphi)|}{2^{\#vars}}$
- φ satisfiable $\iff P(AND = \top) > 0$

1. $A \vee B \vee C$
2. $C \vee D \vee \neg A$
3. $B \vee C \vee \neg D$



(© S. Russell & P. Norwig, AIMA)

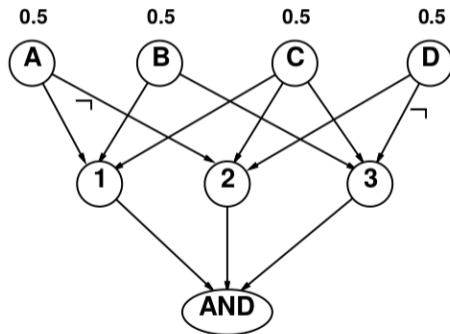
\implies Exact inference in Bayesian Networks is NP-Hard

Complexity of Exact Inference

- We can reduce SAT to exact inference in Bayesian Networks

- $P(AND = \top) = \frac{|Models(\varphi)|}{2^{\#vars}}$
- φ satisfiable $\iff P(AND = \top) > 0$

1. $A \vee B \vee C$
2. $C \vee D \vee \neg A$
3. $B \vee C \vee \neg D$

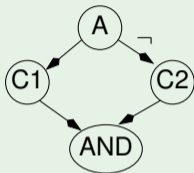


(© S. Russell & P. Norwig, AIMA)

\implies Exact inference in Bayesian Networks is NP-Hard

Example: From SAT to BN Inference

$$\varphi \stackrel{\text{def}}{=} (a \wedge \neg a)$$



$$c1 \stackrel{\text{def}}{=} a \quad \implies \quad P(c1|a) = 1, P(c1|\neg a) = 0$$

$$c2 \stackrel{\text{def}}{=} \neg a \quad \implies \quad P(c2|a) = 0, P(c2|\neg a) = 1$$

$$and \stackrel{\text{def}}{=} c1 \wedge c2 \quad \implies \quad \begin{cases} P(and|c1c2) = 1 \\ P(and|c1\neg c2) = P(and|\neg c1c2) = P(and|\neg c1\neg c2) = 0 \end{cases}$$

$$P(and) = \sum_{c1} \sum_{c2} \sum_a P(and|c1c2)P(c1|a)P(c2|a)P(a)$$

$$= \sum_{c1} \sum_{c2} P(and|c1c2) \sum_a P(c1|a)P(c2|a)P(a)$$

$$\implies = 1 \cdot \sum_a P(c1|a)P(c2|a)P(a) + 0 \cdot \dots + 0 \cdot \dots + 0 \cdot \dots$$

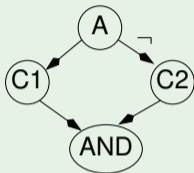
$$= 1 \cdot (P(c1|a)P(c2|a)P(a) + P(c1|\neg a)P(c2|\neg a)P(a\neg))$$

$$= 1 \cdot ((1 \cdot 0 \cdot 0.5) + (0 \cdot 1 \cdot 0.5)) = 0$$

\implies UNSAT

Example: From SAT to BN Inference

$$\varphi \stackrel{\text{def}}{=} (a \wedge \neg a)$$



$$c1 \stackrel{\text{def}}{=} a \quad \implies \quad P(c1|a) = 1, P(c1|\neg a) = 0$$

$$c2 \stackrel{\text{def}}{=} \neg a \quad \implies \quad P(c2|a) = 0, P(c2|\neg a) = 1$$

$$and \stackrel{\text{def}}{=} c1 \wedge c2 \quad \implies \quad \begin{cases} P(and|c1c2) = 1 \\ P(and|c1\neg c2) = P(and|\neg c1c2) = P(and|\neg c1\neg c2) = 0 \end{cases}$$

$$P(and) = \sum_{c1} \sum_{c2} \sum_a P(and|c1c2)P(c1|a)P(c2|a)P(a)$$

$$= \sum_{c1} \sum_{c2} P(and|c1c2) \sum_a P(c1|a)P(c2|a)P(a)$$

$$\implies = 1 \cdot \sum_a P(c1|a)P(c2|a)P(a) + 0 \cdot \dots + 0 \cdot \dots + 0 \cdot \dots$$

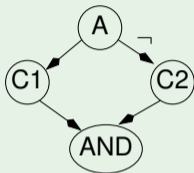
$$= 1 \cdot (P(c1|a)P(c2|a)P(a) + P(c1|\neg a)P(c2|\neg a)P(a\neg))$$

$$= 1 \cdot ((1 \cdot 0 \cdot 0.5) + (0 \cdot 1 \cdot 0.5)) = 0$$

\implies UNSAT

Example: From SAT to BN Inference

$$\varphi \stackrel{\text{def}}{=} (a \wedge \neg a)$$



$$c1 \stackrel{\text{def}}{=} a \quad \implies \quad P(c1|a) = 1, P(c1|\neg a) = 0$$

$$c2 \stackrel{\text{def}}{=} \neg a \quad \implies \quad P(c2|a) = 0, P(c2|\neg a) = 1$$

$$and \stackrel{\text{def}}{=} c1 \wedge c2 \quad \implies \quad \begin{cases} P(and|c1c2) = 1 \\ P(and|c1\neg c2) = P(and|\neg c1c2) = P(and|\neg c1\neg c2) = 0 \end{cases}$$

$$P(and) = \sum_{c1} \sum_{c2} \sum_a P(and|c1c2)P(c1|a)P(c2|a)P(a)$$

$$= \sum_{c1} \sum_{c2} P(and|c1c2) \sum_a P(c1|a)P(c2|a)P(a)$$

$$\implies = 1 \cdot \sum_a P(c1|a)P(c2|a)P(a) + 0 \cdot \dots + 0 \cdot \dots + 0 \cdot \dots$$

$$= 1 \cdot (P(c1|a)P(c2|a)P(a) + P(c1|\neg a)P(c2|\neg a)P(a\neg))$$

$$= 1 \cdot ((1 \cdot 0 \cdot 0.5) + (0 \cdot 1 \cdot 0.5)) = 0$$

\implies UNSAT

Exercise

- For each of the following formulas φ , convert φ into a Bayesian network, and determine the number of its models
 - $(\neg A \vee B) \wedge (A \vee \neg B)$
 - $A \wedge (\neg A \vee B) \wedge \neg B$

