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_2020/

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

M.S. Course "Artificial Intelligence Systems", academic year 2020-2021

Last update: Friday 18th December, 2020, 16:39

Outline

1. Bayesian Networks
2. Constructing Bayesian Networks
3. Exact Inference with Bayesian Networks

Outline

Bayesian Networks

- Bayesian Networks (aka Belief Networks):
  - 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
- 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 $\mathbf{P}(X_i | Parents(X_i))$ for each node X_i
- Conditional distribution represented as a conditional probability table (CPT)
  - distribution over X_i for each combination of parent values
- Topology encodes conditional independence assertions:
  - Toothache and Catch conditionally independent given Cavity
  - Tootchache, Catch depend on Cavity
  - Weather independent from others
- No arc $\Longleftrightarrow$ independence

Bayesian Networks

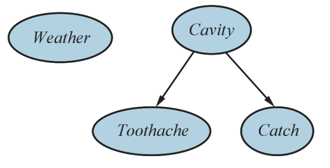
- Bayesian Networks (aka Belief Networks):
  - allow for compact specification of full joint distributions
  - represent explicit conditional dependencies among variables: an arc from X to Y means that X has a direct influence on Y
- Syntax: a directed acyclic graph (DAG):
  - each node represents a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

Bayesian Networks

- Bayesian Networks (aka Belief Networks):
  - 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
- 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 $\mathbf{P}(X_i|Parents(X_i))$ for each node X_i
- Conditional distribution represented as a conditional probability table (CPT)
  - distribution over X_i for each combination of parent values
- Topology encodes conditional independence assertions:
  - Toothache and Catch conditionally independent given Cavity
  - Tootchache, Catch depend on Cavity
  - Weather independent from others
- No arc $\Longleftrightarrow$ independence

Bayesian Networks

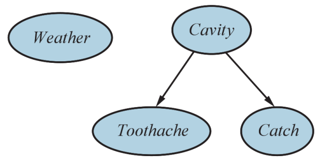
- Bayesian Networks (aka Belief Networks):
  - allow for compact specification of full joint distributions
  - represent explicit conditional dependencies among variables: an arc from X to Y means that X has a direct influence on Y
- Syntax: a directed acyclic graph (DAG):
  - each node represents a random variable (discrete or continuous)
  - directed arcs connect pairs of nodes: $X \rightarrow Y$ (X is a parent of Y)
  - a conditional distribution $\mathbf{P}(X_i|Parents(X_i))$ for each node X_i
- Conditional distribution represented as a conditional probability table (CPT)
  - distribution over X_i for each combination of parent values
- Topology encodes conditional independence assertions:
  - Toothache and Catch conditionally independent given Cavity
  - Tootchache, Catch depend on Cavity
  - Weather independent from others



- No arc $\Longleftrightarrow$ independence

Bayesian Networks

- Bayesian Networks (aka Belief Networks):
  - 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
- 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 $\mathbf{P}(X_i|Parents(X_i))$ for each node X_i
- Conditional distribution represented as a conditional probability table (CPT)
  - distribution over X_i for each combination of parent values
- Topology encodes conditional independence assertions:
  - Toothache and Catch conditionally independent given Cavity
  - Tootchache, Catch depend on Cavity
  - Weather independent from others
- No arc $\Longleftrightarrow$ independence



(© S. Russell & P. Norwig, AIMA)

Example (from Judea Pearl, UCLA)

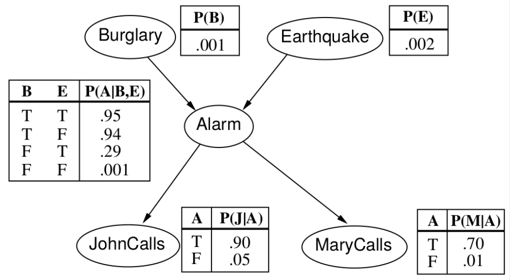
"*The burglary alarm goes off very likely on burglary and occasionally on earthquakes. John and Mary are neighbors who agreed to call when the alarm goes off. Their reliability is different ...*"

- 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 hearthq. in 29% of cases
  - false alarm setoff in 0.1% of cases

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 hearthq. in 29% of cases
  - false alarm setoff in 0.1% of cases

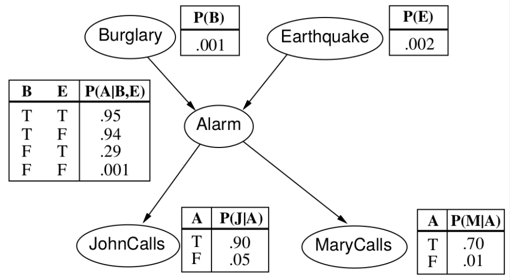


Example (from Judea Pearl, UCLA)

"*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 hearthq. in 29% of cases
  - false alarm setoff in 0.1% of cases

(© S. Russell & P. Norwig, AIMA)

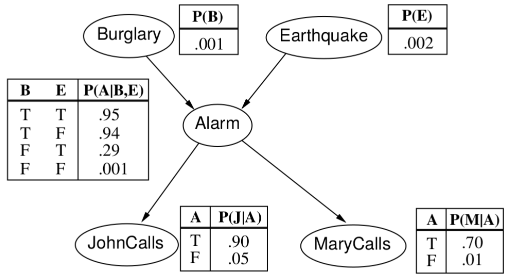


Example (from Judea Pearl, UCLA)

"*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 hearthq. in 29% of cases
  - false alarm setoff in 0.1% of cases

(© S. Russell & P. Norwig, AIMA)



Compactness of Bayesian Networks

- In most domains, it is reasonable to suppose that each random variable is directly influenced by at most k others, for some k.
- A CPT for Boolean X_i with k Boolean parents has
  - 2^k rows for the combinations of parent values
  - each row requires one number p for $P(X_i = true)$ $(P(X_i = false) = 1 - P(X_i = true))$

$\Longrightarrow$ 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 is directly influenced by at most k others, for some k.
- A CPT for Boolean X_i with k Boolean parents has
  - 2^k rows for the combinations of parent values
  - each row requires one number p for $P(X_i = true)$ $(P(X_i = false) = 1 - P(X_i = true))$

$\Longrightarrow$ 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 is directly influenced by at most k others, for some k.
- A CPT for Boolean X_i with k Boolean parents has
  - 2^k rows for the combinations of parent values
  - each row requires one number p for $P(X_i = true)$ $(P(X_i = false) = 1 - P(X_i = true))$

$\Longrightarrow$ 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 is directly influenced by at most k others, for some k.
- A CPT for Boolean X_i with k Boolean parents has
  - 2^k rows for the combinations of parent values
  - each row requires one number p for $P(X_i = true)$ $(P(X_i = false) = 1 - P(X_i = true))$

$\Longrightarrow$ 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:
  $\mathbf{P}(X_1, ..., X_N) = \prod_{i=1} \mathbf{P}(X_i | parents(X_i))$
  - if X_i has no parent, then prior probability $\mathbf{P}(X_i)$
- Intuition: order $X_1, ..., X_n$ s.t. $parents(X_i) \prec X_i$ for each i:
  $\mathbf{P}(X_1, ..., X_n) = \prod_{i=1}^{n} \mathbf{P}(X_i | X_1, ..., X_{i-1})$ // chain rule
  $= \prod_{i=1} \mathbf{P}(X_i | parents(X_i))$ // conditional independence

$\Longrightarrow$ 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:
  $\mathbf{P}(X_1, ..., X_N) = \prod_{i=1} \mathbf{P}(X_i | parents(X_i))$
  - if X_i has no parent, then prior probability $\mathbf{P}(X_i)$
- Intuition: order $X_1, ..., X_n$ s.t. $parents(X_i) \prec X_i$ for each i:
  $\mathbf{P}(X_1, ..., X_n) = \prod_{i=1}^{n} \mathbf{P}(X_i | X_1, ..., X_{i-1})$ // chain rule
  $= \prod_{i=1} \mathbf{P}(X_i | parents(X_i))$ // conditional independence

$\Longrightarrow$ 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:
$\mathbf{P}(X_1, ..., X_N) = \prod_{i=1} \mathbf{P}(X_i | parents(X_i))$
  - if X_i has no parent, then prior probability $\mathbf{P}(X_i)$
- Intuition: order $X_1, ..., X_n$ s.t. $parents(X_i) \prec X_i$ for each i:
$\mathbf{P}(X_1, ..., X_n) = \prod_{i=1}^{n} \mathbf{P}(X_i | X_1, ..., X_{i-1})$ // chain rule
$= \prod_{i=1} \mathbf{P}(X_i | parents(X_i))$ // conditional independence

$\Longrightarrow$ 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:
  $\mathbf{P}(X_1, ..., X_N) = \prod_{i=1} \mathbf{P}(X_i | parents(X_i))$
  - if X_i has no parent, then prior probability $\mathbf{P}(X_i)$
- Intuition: order $X_1, ..., X_n$ s.t. $parents(X_i) \prec X_i$ for each i:
  $\mathbf{P}(X_1, ..., X_n) = \prod_{i=1}^{n} \mathbf{P}(X_i | X_1, ..., X_{i-1}))$ // chain rule
  $= \prod_{i=1} \mathbf{P}(X_l | parents(X_i))$ // conditional independence

$\Longrightarrow$ 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:
  $\mathbf{P}(X_1, ..., X_N) = \prod_{i=1} \mathbf{P}(X_i | \mathit{parents}(X_i))$
  - if X_i has no parent, then prior probability $\mathbf{P}(X_i)$
- Intuition: order $X_1, ..., X_n$ s.t. $\mathit{parents}(X_i) \prec X_i$ for each i:
  $\mathbf{P}(X_1, ..., X_n) = \prod_{i=1}^{n} \mathbf{P}(X_i | X_1, ..., X_{i-1})$ // chain rule
  $= \prod_{i=1} \mathbf{P}(X_I | \mathit{parents}(X_i))$ // conditional independence

$\Longrightarrow$ A Bayesian network is a distributed representation of the full joint distribution

Global Semantics: Example

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

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

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

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

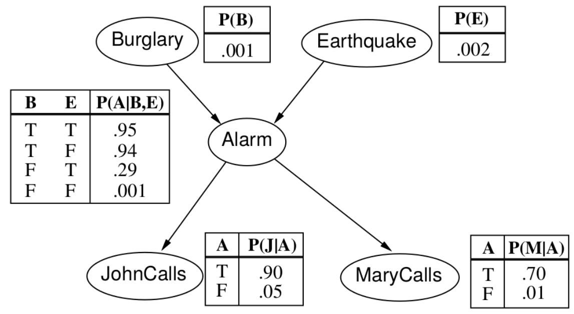
  $0.9 \cdot 0.7 \cdot 0.001 \cdot 0.999 \cdot 0.998$

  ≈ 0.00063

Global Semantics: Example

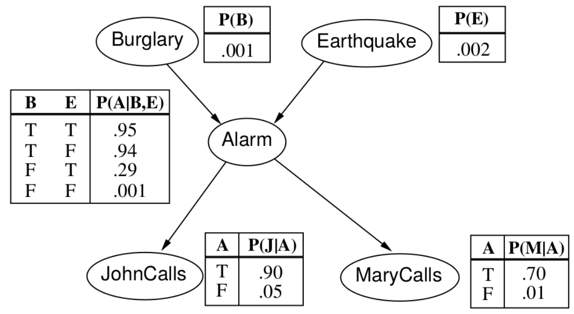
- $\mathbf{P}(X_1, ..., X_N) = \prod_{i=1} \mathbf{P}(X_i | parents(X_i))$
- Ex: "Prob. that both John and Mary call, the alarm sets off but no burglary nor earthquake"

$P(j \land m \land a \land \neg b \land \neg e) =$
$P(j|m \land a \land \neg b \land \neg e) P(m|a \land \neg b \land \neg e) P(a|\neg b \land \neg e) P(\neg b|\neg e) P(\neg e) =$
$P(j|a)\ P(m|a)\ P(a|\neg b \land \neg e)\ P(\neg b)\ P(\neg e) =$
$0.9 \cdot 0.7 \cdot 0.001 \cdot 0.999 \cdot 0.998$
≈ 0.00063



Global Semantics: Example

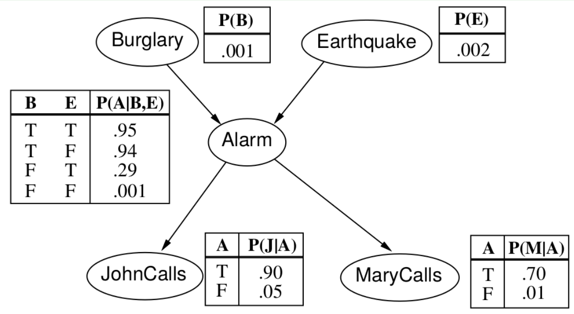
- $\mathbf{P}(X_1, ..., X_N) = \prod_{i=1} \mathbf{P}(X_i | parents(X_i))$
- Ex: "Prob. that both John and Mary call, the alarm sets off but no burglary nor earthquake"
  $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) =$
  $P(j|m \wedge a \wedge \neg b \wedge \neg e) P(m|a \wedge \neg b \wedge \neg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



(© S. Russell & P. Norwig, AIMA)

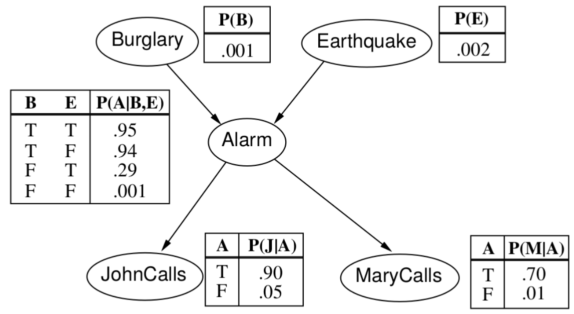
Global Semantics: Example

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



Global Semantics: Example

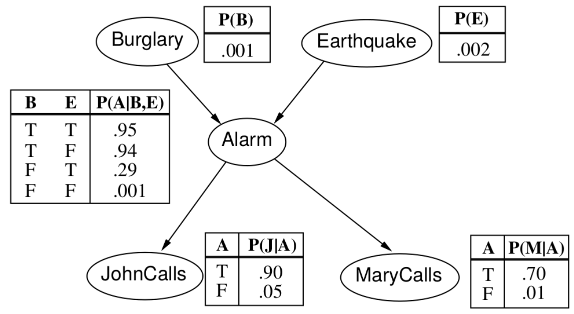
- $\mathbf{P}(X_1, ..., X_N) = \prod_{i=1} \mathbf{P}(X_i | parents(X_i))$
- Ex: "Prob. that both John and Mary call, the alarm sets off but no burglary nor earthquake"
  $P(j \land m \land a \land \neg b \land \neg e) =$
  $P(j|m \land a \land \neg b \land \neg e) P(m|a \land \neg b \land \neg e) P(a|\neg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



(© S. Russell & P. Norwig, AIMA)

Global Semantics: Example

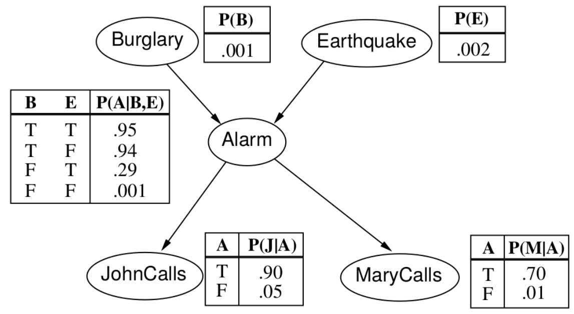
- $\mathbf{P}(X_1, ..., X_N) = \prod_{i=1} \mathbf{P}(X_i | parents(X_i))$
- Ex: "Prob. that both John and Mary call, the alarm sets off but no burglary nor earthquake"
  $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) =$
  $P(j|m \wedge a \wedge \neg b \wedge \neg e) P(m|a \wedge \neg b \wedge \neg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



(© S. Russell & P. Norwig, AIMA)

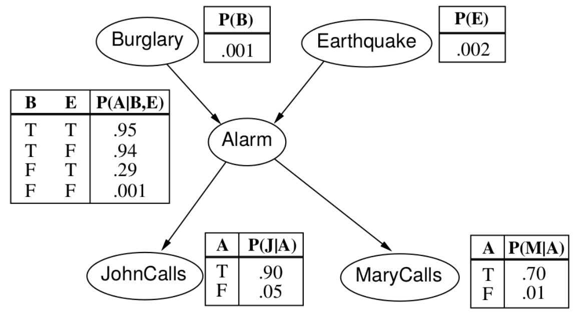
Global Semantics: Example

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



(© S. Russell & P. Norwig, AIMA)

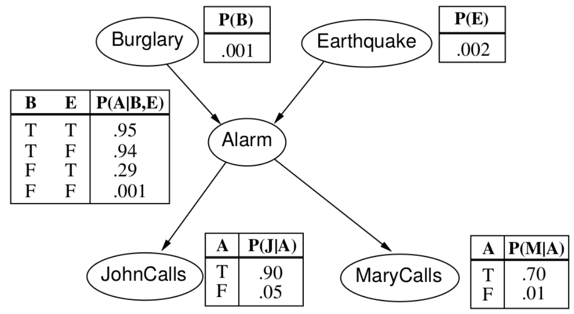
- $\mathbf{P}(X_1, ..., X_N) = \prod_{i=1} \mathbf{P}(X_i | parents(X_i))$
- Ex: "Prob. that both John and Mary call, the alarm sets off but no burglary nor earthquake"
  $P(j \land m \land a \land \neg b \land \neg e) =$
  $P(j|m \land a \land \neg b \land \neg e) P(m|a \land \neg b \land \neg e) P(a|\neg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



(© S. Russell & P. Norwig, AIMA)

Global Semantics: Example

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



P(B)
.001

P(E)
.002

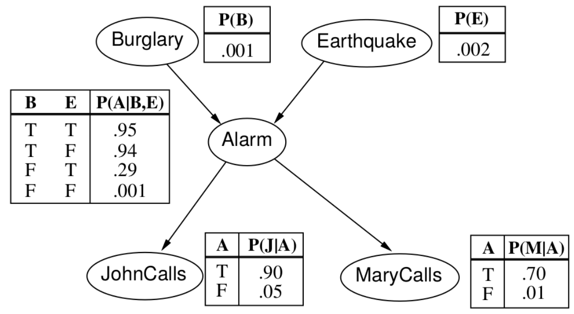
B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

A	P(M|A)
T	.70
F	.01

Global Semantics: Example

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



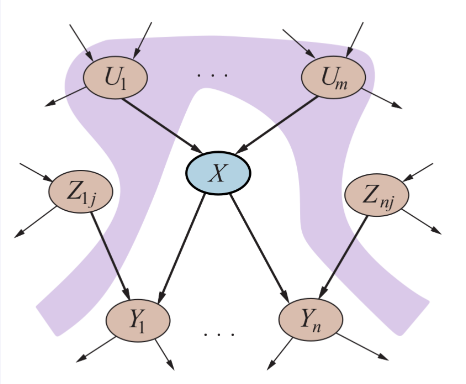
(© S. Russell & P. Norwig, AIMA)

Exercises

- 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
  - ...

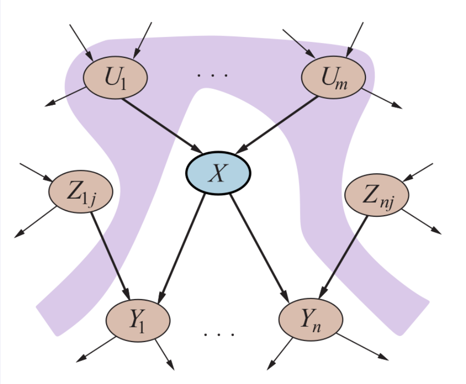
Local Semantics

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



Local Semantics

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

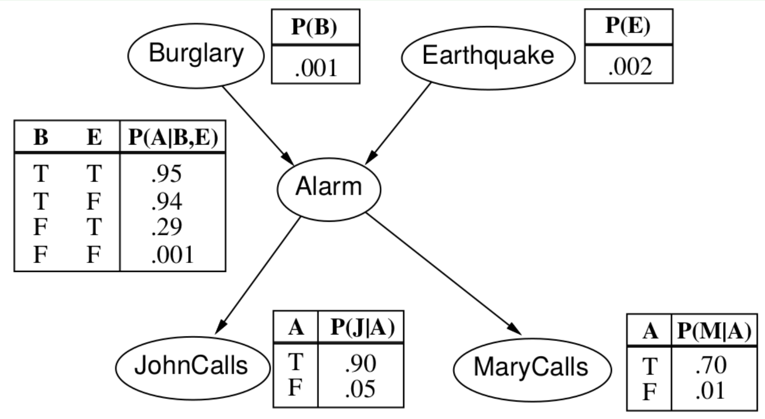


(© S. Russell & P. Norwig, AIMA)

Local Semantics: Example

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

$$\mathbf{P}(JohnCalls \mid Alarm, Burglary, Earthquake, MaryCalls) = \mathbf{P}(JohnCalls \mid Alarm)$$



P(B)
.001

P(E)
.002

B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

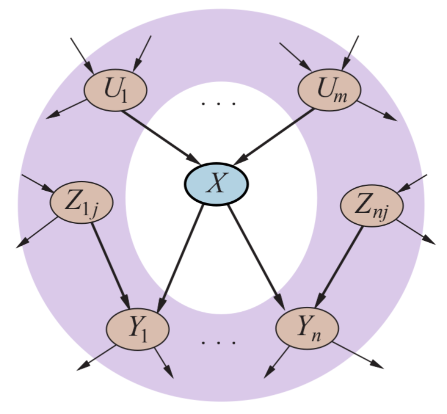
A	P(M|A)
T	.70
F	.01

(© S. Russell & P. Norwig, AIMA)

Independence Property: Markov Blanket

In an B.N., each node is conditionally independent of all others given its Markov blanket: parents + children + children's parents:

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

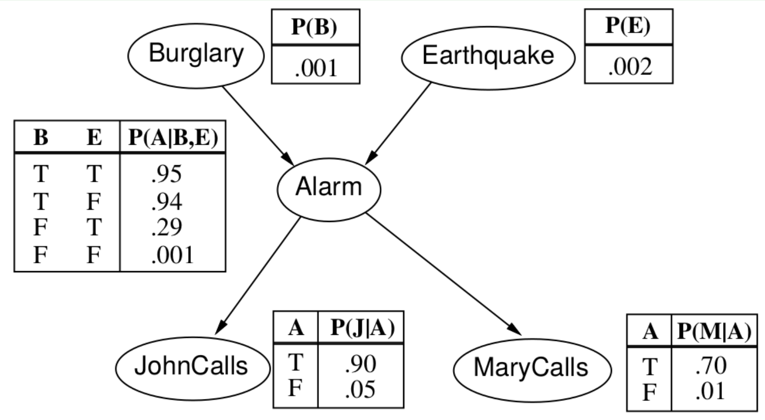


(© S. Russell & P. Norwig, AIMA)

Markov Blanket: Example

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

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



(© S. Russell & P. Norwig, AIMA)

Exercise

Verify numerically the two previous examples:

- Local Semantics
- Markov Blanket

Outline

Constructing Bayesian Networks

Building the graph

Given a set of random variables

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

$\Longrightarrow$ Guarantees the global semantics by construction

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

Constructing Bayesian Networks

Building the graph

Given a set of random variables

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

$\Longrightarrow$ Guarantees the global semantics by construction

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

Constructing Bayesian Networks

Building the graph

Given a set of random variables

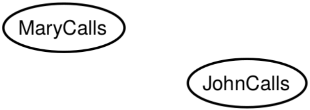
1. Choose an ordering $\{X_1, ..., X_n\}$
   - in principle, any ordering will work (but some may cause blowups)
   - general rule: follow causality, $X \prec Y$ if $X \in causes(Y)$
2. For i=1 to n do
   1. add X_i to the network
   2. as $Parents(X_i)$, choose a subset of $\{X_1, ..., X_{i-1}\}$ s.t.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

$\Longrightarrow$ Guarantees the global semantics by construction

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

Constructing Bayesian Networks: Example

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

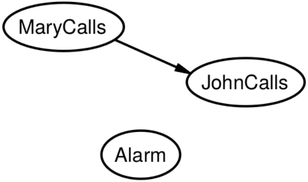


$P(J|M) = P(J)$?

(© S. Russell & P. Norwig, AIMA)

Constructing Bayesian Networks: Example

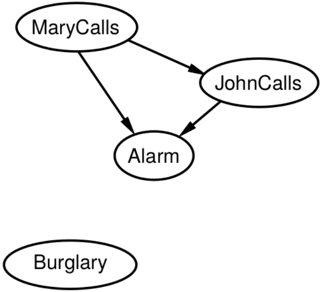
Suppose we choose the ordering $\{M, J, A, B, E\}$ (non-causal ordering):



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

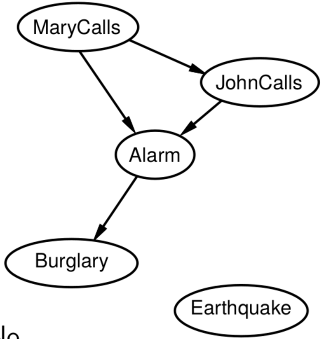


$P(J|M) = P(J)$? No
$P(A|J, M) = P(A|J)$? $P(A|J, M) = P(A)$? No
$P(B|A, J, M) = P(B|A)$?
$P(B|A, J, M) = P(B)$?

(© S. Russell & P. Norwig, AIMA)

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

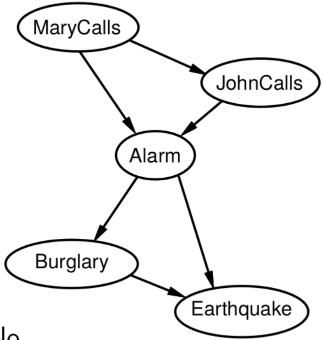
$P(B|A,J,M) = P(B)$? No

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

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

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

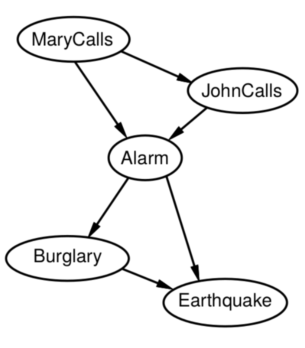
$P(B|A,J,M) = P(B)$? No

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

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

Constructing Bayesian Networks: Example [cont.]

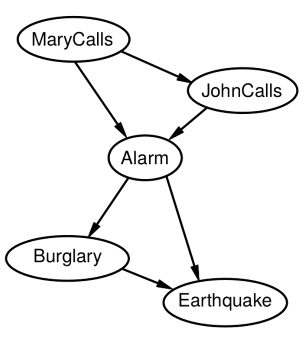
- In non-causal directions
  - deciding conditional independence is hard
  - assessing conditional probabilities is hard
  - typically networks less compact
- Ex: $1+2+4+2+4=13$ numbers needed (rather than 10)
- Can be much worse
  - ex: try $\{M, J, E, B, A\}$ (see AIMA)
- Much better with causal orderings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



(© S. Russell & P. Norwig, AIMA)

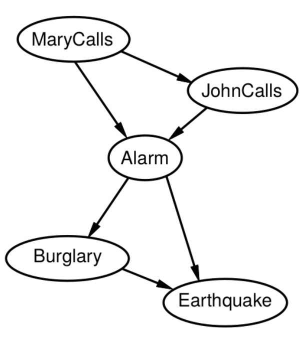
Constructing Bayesian Networks: Example [cont.]

- 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)
- 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)



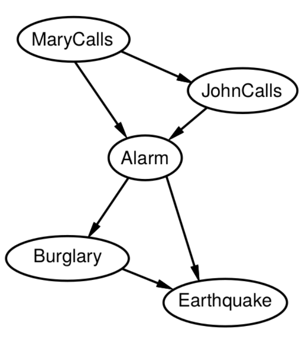
Constructing Bayesian Networks: Example [cont.]

- 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)
- 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)
- 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
  - let parents $U_1, ..., U_k$ include all causes (can add leak node)
  - add Independent failure probability q_i for each cause U_i

  $\Longrightarrow P(\neg X | U_1 \ldots U_j, \neg U_{j+1} \ldots \neg U_k) = \prod_{i=1}^{j} q_i$
  - 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
  - let parents $U_1, ..., U_k$ include all causes (can add leak node)
  - add Independent failure probability q_i for each cause U_i

$\Longrightarrow$ $P(\neg X | U_1 ... U_j, \neg U_{j+1} ... \neg U_k) = \prod_{i=1}^{j} q_i$

  - 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
  - let parents $U_1, ..., U_k$ include all causes (can add leak node)
  - add Independent failure probability q_i for each cause U_i

  $\Longrightarrow$ $P(\neg X | U_1 ... U_j, \neg U_{j+1} ... \neg U_k) = \prod_{i=1}^{j} q_i$
  - number of parameters linear in number of parents!
- Ex: $q_{Cold} = 0.6,\ q_{Flu} = 0.2,\ q_{Malaria} = 0.1$:

$Cold$	Flu	$Malaria$	$P(Fever)$	$P(\neg Fever)$
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	$0.012 = 0.6 \times 0.2 \times 0.1$

Exercises

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

Outline

1. Bayesian Networks
2. Constructing Bayesian Networks
3. Exact Inference with Bayesian Networks

Exact inference in Bayesian Networks

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

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

  A typical query asks for the posterior probability distribution: **P**(X | **E**=**e**) (also written **P**(X | **e**))
- Ex: $P(Burglar|JohnCalls = true, MaryCalls = true)$
  - query: $Burglar$
  - evidence variables: $\mathbf{E} = \{JohnCalls, MaryCalls\}$
  - hidden variables: $\mathbf{Y} = \{Earthquake, Alarm\}$

Exact inference in Bayesian Networks

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

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

  A typical query asks for the posterior probability distribution: **P**(X | **E**=**e**) (also written **P**(X | **e**))
- Ex: $P(Burglar|JohnCalls = true, MaryCalls = true)$
  - query: *Burglar*
  - evidence variables: $\mathbf{E} = \{JohnCalls, MaryCalls\}$
  - hidden variables: $\mathbf{Y} = \{Earthquake, Alarm\}$

Inference by Enumeration

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

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

  - then apply factorization and simplify algebraically when possible

- Ex:
  $\mathbf{P}(B|j, m) =$
  $\alpha \sum_e \sum_a \mathbf{P}(B, e, a, j, m) =$
  $\alpha \sum_e \sum_a \mathbf{P}(B)P(e)\mathbf{P}(a|B, e)P(j|a)P(m|a) =$
  $\alpha\mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e)P(j|a)P(m|a)$

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

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

Inference by Enumeration

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

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

  - then apply factorization and simplify algebraically when possible

- Ex:
  $\mathbf{P}(B|j, m) =$
  $\alpha \sum_e \sum_a \mathbf{P}(B, e, a, j, m) =$
  $\alpha \sum_e \sum_a \mathbf{P}(B)P(e)\mathbf{P}(a|B, e)P(j|a)P(m|a) =$
  $\alpha\mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e)P(j|a)P(m|a)$

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

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

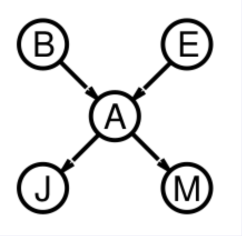
Inference by Enumeration

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

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

  - then apply factorization and simplify algebraically when possible

- Ex:
  $\mathbf{P}(B|j,m) =$
  $\alpha\sum_e\sum_a\mathbf{P}(B,e,a,j,m) =$
  $\alpha\sum_e\sum_a\mathbf{P}(B)P(e)\mathbf{P}(a|B,e)P(j|a)P(m|a) =$
  $\alpha\mathbf{P}(B)\sum_e P(e)\sum_a\mathbf{P}(a|B,e)P(j|a)P(m|a)$



(© S. Russell & P. Norwig, AIMA)

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

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

Inference by Enumeration

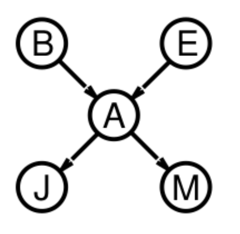
- We defined a procedure for the task as:
  $\mathbf{P}(X|\mathbf{e}) = \alpha\mathbf{P}(X, \mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})$

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

  - then apply factorization and simplify algebraically when possible

- Ex:
  $\mathbf{P}(B|j, m) =$
  $\alpha \sum_e \sum_a \mathbf{P}(B, e, a, j, m) =$
  $\alpha \sum_e \sum_a \mathbf{P}(B)P(e)\mathbf{P}(a|B, e)P(j|a)P(m|a) =$
  $\alpha\mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e)P(j|a)P(m|a)$

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



(© S. Russell & P. Norwig, AIMA)

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

Inference by Enumeration

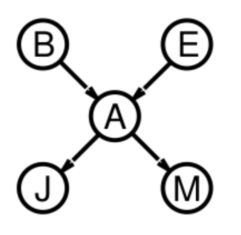
- We defined a procedure for the task as:
  $\mathbf{P}(X|\mathbf{e}) = \alpha\mathbf{P}(X, \mathbf{e}) = \alpha \sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})$

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

  - then apply factorization and simplify algebraically when possible

- Ex:
  $\mathbf{P}(B|j, m) =$
  $\alpha \sum_e \sum_a \mathbf{P}(B, e, a, j, m) =$
  $\alpha \sum_e \sum_a \mathbf{P}(B)P(e)\mathbf{P}(a|B, e)P(j|a)P(m|a) =$
  $\alpha\mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e)P(j|a)P(m|a)$

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



(© S. Russell & P. Norwig, AIMA)

- Recursive depth-first enumeration:
  $O(n)$ space, $O(2^n)$ time with n propositional variables

- Enumeration is inefficient: repeated computation

Inference by Enumeration

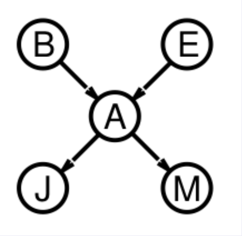
- We defined a procedure for the task as:
  $\mathbf{P}(X|\mathbf{e}) = \alpha\mathbf{P}(X, \mathbf{e}) = \alpha\sum_{\mathbf{y}} \mathbf{P}(X, \mathbf{e}, \mathbf{y})$

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

  - then apply factorization and simplify algebraically when possible

- Ex:
  $\mathbf{P}(B|j, m) =$
  $\alpha\sum_e\sum_a \mathbf{P}(B, e, a, j, m) =$
  $\alpha\sum_e\sum_a \mathbf{P}(B)P(e)\mathbf{P}(a|B, e)P(j|a)P(m|a) =$
  $\alpha\mathbf{P}(B)\sum_e P(e)\sum_a \mathbf{P}(a|B, e)P(j|a)P(m|a)$

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



(© S. Russell & P. Norwig, AIMA)

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

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 $\{X\} \cup \mathbf{E} \cup \mathbf{Y}$ /* $\mathbf{Y}$ = *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 P(xi, **Y**, **e**) (single probability value)
      where $\mathbf{e}_{x_i}$ is $\mathbf{e}$ extended with $X = x_i$
  return NORMALIZE($\mathbf{Q}(X)$)

---

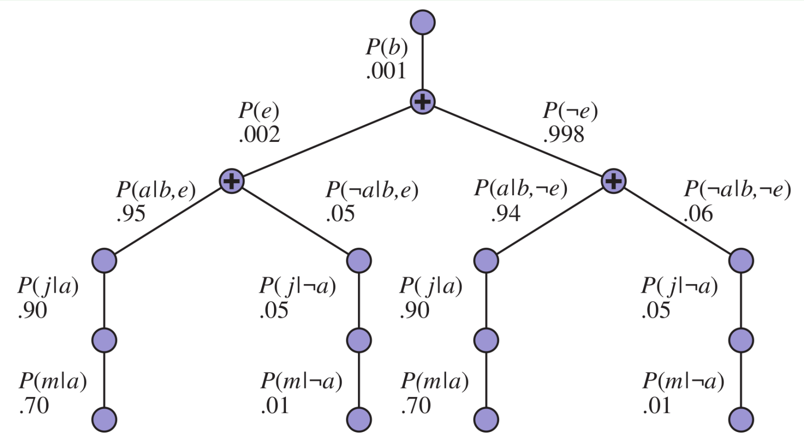
function ENUMERATE-ALL($vars$, $\mathbf{e}$) **returns** a real number
  if EMPTY?($vars$) **then return** 1.0
  $Y \leftarrow$ FIRST($vars$)
  if Y has value y in $\mathbf{e}$
    then return $P(y \mid parents(Y)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}$) X or evidence var
    else return $\sum_y P(y \mid parents(Y)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}_y$) hidden var
      where $\mathbf{e}_y$ is $\mathbf{e}$ extended with $Y = y$

Inference by Enumeration: Example

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

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

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



(© S. Russell & P. Norwig, AIMA)

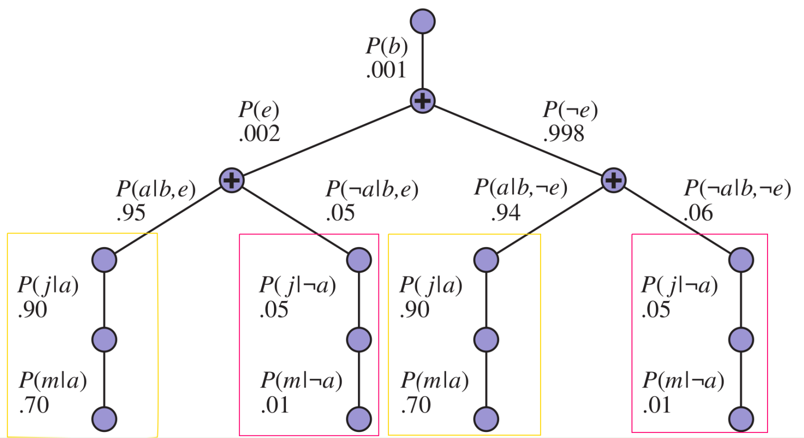
Repeated computation: $P(j|a)P(m|a)$ & $P(j|\neg a)P(m|\neg a)$ for each value of e

Inference by Enumeration: Example

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

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

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

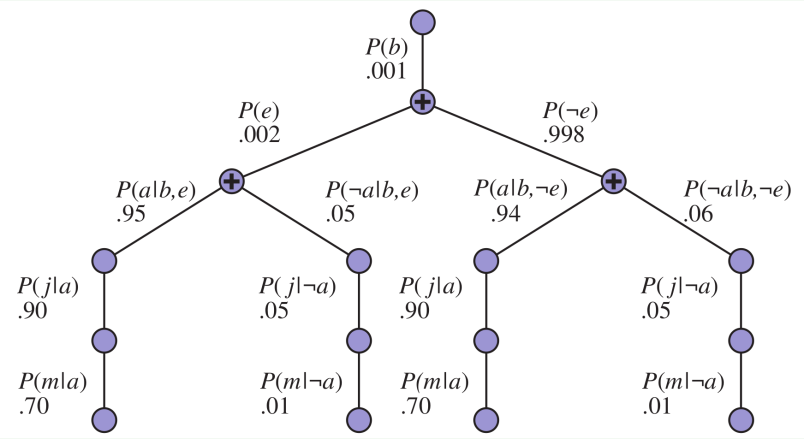


(© S. Russell & P. Norwig, AIMA)

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

$P(b|j,m) = \alpha P(b) \sum_e P(e) \sum_a P(a|b,e) P(j|a) P(m|a) = \alpha \cdot 0.00059224$
$P(\neg b|j,m) = \alpha P(\neg b) \sum_e P(e) \sum_a P(a|\neg b,e) P(j|a) P(m|a) = \alpha \cdot 0.0014919$
$\Longrightarrow \mathbf{P}(B|j,m) = \alpha \cdot \langle 0.00059224, 0.0014919 \rangle = [normal.] \approx \langle 0.284, 0.716 \rangle$

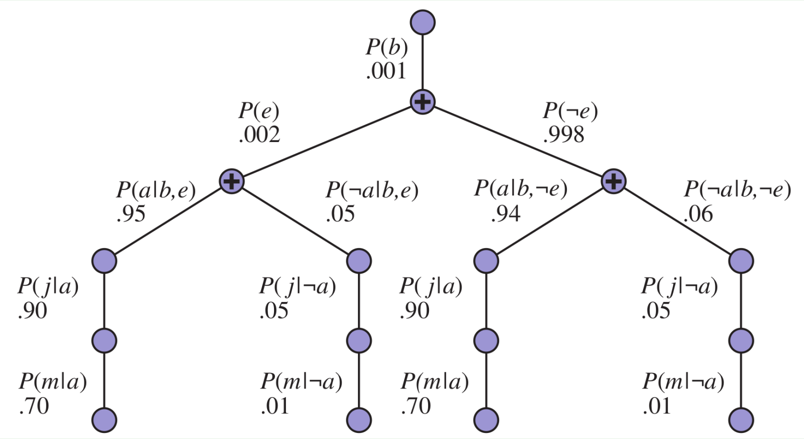


(© S. Russell & P. Norwig, AIMA)

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

$P(\ \ b|j,m) = \alpha P(\ \ b) \sum_e P(e) \sum_a P(a|\ \ b,e) P(j|a) P(m|a) = \alpha \cdot 0.00059224$
$P(\neg b|j,m) = \alpha P(\neg b) \sum_e P(e) \sum_a P(a|\neg b,e) P(j|a) P(m|a) = \alpha \cdot 0.0014919$
$\Longrightarrow \mathbf{P}(B|j,m) = \alpha \cdot \langle 0.00059224, 0.0014919 \rangle = [normal.] \approx \langle 0.284, 0.716 \rangle$



(© 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

Exercises

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

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: $P(B|j,m)$

$$= \alpha \overbrace{P(B)}^{B} \sum_e \overbrace{P(e)}^{E} \sum_a \overbrace{\mathbf{P}(a|B,e)}^{A} \overbrace{P(j|a)}^{J} \overbrace{P(m|a)}^{M}$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e) P(j|a) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{f}_A(A,B,E) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \times \mathbf{f}_{\overline{A}JM}(B,E) \quad (sum\ out\ A)$$
$$= \alpha P(B) \sum_e \mathbf{f}_E(E) \times \mathbf{f}_{\overline{A}JM}(B,E) \quad (sum\ out\ A)$$
$$= \alpha P(B) \times \mathbf{f}_{\overline{E}\overline{A}JM}(B) \quad (sum\ out\ E)$$
$$= \alpha \times \mathbf{f}_B(B) \times \mathbf{f}_{\overline{E}\overline{A}JM}(B)$$

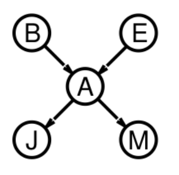
  - "$\times$" is the pointwise product (see later)
  - $\mathbf{f}_M(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(m|a) \\ P(m|\neg a) \end{bmatrix}$, $\mathbf{f}_J(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(j|a) \\ P(j|\neg a) \end{bmatrix}$, ...
  - $\mathbf{f}_{\overline{A}...}(.)$: summation over the values of A...

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: $P(B|j,m)$

$$= \alpha \overbrace{P(B)}^{B} \sum_e \overbrace{P(e)}^{E} \sum_a \overbrace{\mathbf{P}(a|B,e)}^{A} \overbrace{P(j|a)}^{J} \overbrace{P(m|a)}^{M}$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e) P(j|a) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{f}_A(A,B,E) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \times \mathbf{f}_{\bar{A}JM}(B,E) \quad (sum\ out\ A)$$
$$= \alpha P(B) \sum_e \mathbf{f}_E(E) \times \mathbf{f}_{\bar{A}JM}(B,E) \quad (sum\ out\ A)$$
$$= \alpha P(B) \times \mathbf{f}_{\overline{E}\bar{A}JM}(B) \quad (sum\ out\ E)$$
$$= \alpha \times \mathbf{f}_B(B) \times \mathbf{f}_{\overline{E}\bar{A}JM}(B)$$

- "×" is the pointwise product (see later)
- $\mathbf{f}_M(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(m|a) \\ P(m|\neg a) \end{bmatrix}$, $\mathbf{f}_J(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(j|a) \\ P(j|\neg a) \end{bmatrix}$, ...
- $\mathbf{f}_{\bar{A}...}(.)$: summation over the values of A...

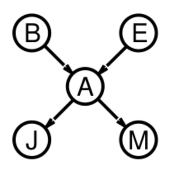


Inference by Variable Elimination

- Variable elimination:
  - carry out summations right-to-left (i.e., bottom-up in the tree)
  - store intermediate results (factors) to avoid recomputation
- Ex: $P(B|j,m)$

$$= \alpha \overbrace{P(B)}^{B} \sum_e \overbrace{P(e)}^{E} \sum_a \overbrace{\mathbf{P}(a|B,e)}^{A} \overbrace{P(j|a)}^{J} \overbrace{P(m|a)}^{M}$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e) P(j|a) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{f}_A(A,B,E) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \times \mathbf{f}_{\overline{A}JM}(B,E) \quad (sum\ out\ A)$$
$$= \alpha P(B) \sum_e \mathbf{f}_E(E) \times \mathbf{f}_{\overline{A}JM}(B,E) \quad (sum\ out\ A)$$
$$= \alpha P(B) \times \mathbf{f}_{\overline{EA}JM}(B) \quad (sum\ out\ E)$$
$$= \alpha \times \mathbf{f}_B(B) \times \mathbf{f}_{\overline{EA}JM}(B)$$

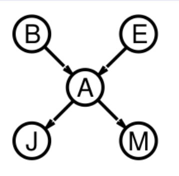
  - "$\times$" is the pointwise product (see later)
  - $\mathbf{f}_M(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(m|a) \\ P(m|\neg a) \end{bmatrix}$, $\mathbf{f}_J(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(j|a) \\ P(j|\neg a) \end{bmatrix}$, ...
  - $\mathbf{f}_{\overline{A}...}(.)$: summation over the values of A...



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: $P(B|j,m)$

$$= \alpha \overbrace{P(B)}^{B} \textstyle\sum_e \overbrace{P(e)}^{E} \sum_a \overbrace{\mathbf{P}(a|B,e)}^{A} \overbrace{P(j|a)}^{J} \overbrace{P(m|a)}^{M}$$
$$= \alpha P(B) \textstyle\sum_e P(e) \sum_a \mathbf{P}(a|B,e) P(j|a) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \textstyle\sum_e P(e) \sum_a \mathbf{P}(a|B,e) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \textstyle\sum_e P(e) \sum_a \mathbf{f}_A(A,B,E) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \textstyle\sum_e P(e) \times \mathbf{f}_{\overline{A}JM}(B,E) \quad (sum\ out\ A)$$
$$= \alpha P(B) \textstyle\sum_e \mathbf{f}_E(E) \times \mathbf{f}_{\overline{A}JM}(B,E) \quad (sum\ out\ A)$$
$$= \alpha P(B) \times \mathbf{f}_{\overline{E}\overline{A}JM}(B) \quad (sum\ out\ E)$$
$$= \alpha \times \mathbf{f}_B(B) \times \mathbf{f}_{\overline{E}\overline{A}JM}(B)$$



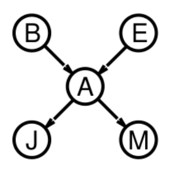
  - "$\times$" is the pointwise product (see later)
  - $\mathbf{f}_M(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(m|\ a) \\ P(m|\neg a) \end{bmatrix}$, $\mathbf{f}_J(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(j|\ a) \\ P(j|\neg a) \end{bmatrix}$, ...
  - $\mathbf{f}_{\overline{A}...}(.)$: summation over the values of A...

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: $P(B|j,m)$

$$
\begin{aligned}
&= \alpha \overbrace{P(B)}^{B} \textstyle\sum_e \overbrace{P(e)}^{E} \textstyle\sum_a \overbrace{\mathbf{P}(a|B,e)}^{A} \overbrace{P(j|a)}^{J} \overbrace{P(m|a)}^{M} \\
&= \alpha P(B) \textstyle\sum_e P(e) \textstyle\sum_a \mathbf{P}(a|B,e) P(j|a) \times \mathbf{f}_M(A) \\
&= \alpha P(B) \textstyle\sum_e P(e) \textstyle\sum_a \mathbf{P}(a|B,e) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A) \\
&= \alpha P(B) \textstyle\sum_e P(e) \textstyle\sum_a \mathbf{f}_A(A,B,E) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A) \\
&= \alpha P(B) \textstyle\sum_e P(e) \times \mathbf{f}_{\overline{A}JM}(B,E) \quad (\textit{sum out } A) \\
&= \alpha P(B) \textstyle\sum_e \mathbf{f}_E(E) \times \mathbf{f}_{\overline{A}JM}(B,E) \quad (\textit{sum out } A) \\
&= \alpha P(B) \times \mathbf{f}_{\overline{EA}JM}(B) \quad (\textit{sum out } E) \\
&= \alpha \times \mathbf{f}_B(B) \times \mathbf{f}_{\overline{EA}JM}(B)
\end{aligned}
$$

  - "$\times$" is the pointwise product (see later)
  - $\mathbf{f}_M(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(m|\ a) \\ P(m|\neg a) \end{bmatrix}$, $\mathbf{f}_J(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(j|\ a) \\ P(j|\neg a) \end{bmatrix}$, ...
  - $\mathbf{f}_{\overline{A}...}(.)$: summation over the values of A...

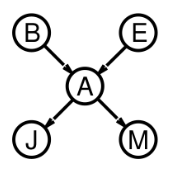


Inference by Variable Elimination

- Variable elimination:
  - carry out summations right-to-left (i.e., bottom-up in the tree)
  - store intermediate results (factors) to avoid recomputation
- Ex: $P(B|j,m)$

$$= \alpha \overbrace{P(B)}^{B} \sum_e \overbrace{P(e)}^{E} \sum_a \overbrace{\mathbf{P}(a|B,e)}^{A} \overbrace{P(j|a)}^{J} \overbrace{P(m|a)}^{M}$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e) P(j|a) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{f}_A(A,B,E) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \times \mathbf{f}_{\overline{A}JM}(B,E) \quad (\textit{sum out } A)$$
$$= \alpha P(B) \sum_e \mathbf{f}_E(E) \times \mathbf{f}_{\overline{A}JM}(B,E) \quad (\textit{sum out } A)$$
$$= \alpha P(B) \times \mathbf{f}_{\overline{E}\overline{A}JM}(B) \quad (\textit{sum out } E)$$
$$= \alpha \times \mathbf{f}_B(B) \times \mathbf{f}_{\overline{E}\overline{A}JM}(B)$$

  - "$\times$" is the pointwise product (see later)
  - $\mathbf{f}_M(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(m|\ a) \\ P(m|\neg a) \end{bmatrix}, \mathbf{f}_J(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(j|\ a) \\ P(j|\neg a) \end{bmatrix}, \ldots$
  - $\mathbf{f}_{\overline{A}\ldots}(.)$: summation over the values of A...

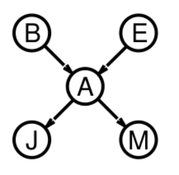


(© S. Russell & P. Norwig, AIMA)

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: $P(B|j,m)$

$$= \alpha \overbrace{P(B)}^{B} \sum_e \overbrace{P(e)}^{E} \sum_a \overbrace{\mathbf{P}(a|B,e)}^{A} \overbrace{P(j|a)}^{J} \overbrace{P(m|a)}^{M}$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e) P(j|a) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{f}_A(A,B,E) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \times \mathbf{f}_{\bar{A}JM}(B,E) \ \ (\text{sum out } A)$$
$$= \alpha P(B) \sum_e \mathbf{f}_E(E) \times \mathbf{f}_{\bar{A}JM}(B,E) \ \ (\text{sum out } A)$$
$$= \alpha P(B) \times \mathbf{f}_{\overline{E}\bar{A}JM}(B) \ \ (\text{sum out } E)$$
$$= \alpha \times \mathbf{f}_B(B) \times \mathbf{f}_{\overline{E}\bar{A}JM}(B)$$

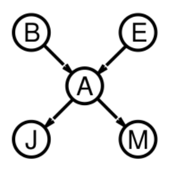


  - "$\times$" is the pointwise product (see later)
  - $\mathbf{f}_M(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(m|\ a) \\ P(m|\neg a) \end{bmatrix}$, $\mathbf{f}_J(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(j|\ a) \\ P(j|\neg a) \end{bmatrix}$, ...
  - $\mathbf{f}_{\bar{A}...}(.)$: summation over the values of A...

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: $P(B|j,m)$

$$= \alpha \overbrace{P(B)}^{B} \sum_e \overbrace{P(e)}^{E} \sum_a \overbrace{\mathbf{P}(a|B,e)}^{A} \overbrace{P(j|a)}^{J} \overbrace{P(m|a)}^{M}$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e) P(j|a) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{f}_A(A,B,E) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \times \mathbf{f}_{\bar{A}JM}(B,E) \;\; (\textit{sum out } A)$$
$$= \alpha P(B) \sum_e \mathbf{f}_E(E) \times \mathbf{f}_{\bar{A}JM}(B,E) \;\; (\textit{sum out } A)$$
$$= \alpha P(B) \times \mathbf{f}_{\overline{EA}JM}(B) \;\; (\textit{sum out } E)$$
$$= \alpha \times \mathbf{f}_B(B) \times \mathbf{f}_{\overline{EA}JM}(B)$$



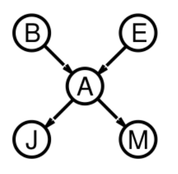
  - "$\times$" is the pointwise product (see later)
  - $\mathbf{f}_M(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(m|\;a) \\ P(m|\neg a) \end{bmatrix}$, $\mathbf{f}_J(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(j|\;a) \\ P(j|\neg a) \end{bmatrix}$, ...
  - $\mathbf{f}_{\bar{A}...}(.)$: summation over the values of A...

(© S. Russell & P. Norwig, AIMA)

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: $P(B|j,m)$

$$= \alpha \overbrace{P(B)}^{B} \sum_e \overbrace{P(e)}^{E} \sum_a \overbrace{\mathbf{P}(a|B,e)}^{A} \overbrace{P(j|a)}^{J} \overbrace{P(m|a)}^{M}$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e) P(j|a) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{f}_A(A,B,E) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \times \mathbf{f}_{\bar{A}JM}(B,E) \ \ (sum\ out\ A)$$
$$= \alpha P(B) \sum_e \mathbf{f}_E(E) \times \mathbf{f}_{\bar{A}JM}(B,E) \ \ (sum\ out\ A)$$
$$= \alpha P(B) \times \mathbf{f}_{\overline{EA}JM}(B) \ \ (sum\ out\ E)$$
$$= \alpha \times \mathbf{f}_B(B) \times \mathbf{f}_{\overline{EA}JM}(B)$$

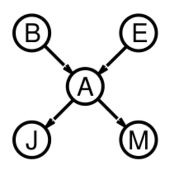


- “$\times$” is the pointwise product (see later)
- $\mathbf{f}_M(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(m|\ \ a) \\ P(m|\neg a) \end{bmatrix}$, $\mathbf{f}_J(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(j|\ \ a) \\ P(j|\neg a) \end{bmatrix}$, ...
- $\mathbf{f}_{\bar{A}...}(.)$: summation over the values of A...

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: $P(B|j,m)$

$$= \alpha \overbrace{P(B)}^{B} \sum_e \overbrace{P(e)}^{E} \sum_a \overbrace{\mathbf{P}(a|B,e)}^{A} \overbrace{P(j|a)}^{J} \overbrace{P(m|a)}^{M}$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e) P(j|a) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \sum_a \mathbf{f}_A(A,B,E) \times \mathbf{f}_J(A) \times \mathbf{f}_M(A)$$
$$= \alpha P(B) \sum_e P(e) \times \mathbf{f}_{\bar{A}JM}(B,E) \ \ (sum\ out\ A)$$
$$= \alpha P(B) \sum_e \mathbf{f}_E(E) \times \mathbf{f}_{\bar{A}JM}(B,E) \ \ (sum\ out\ A)$$
$$= \alpha P(B) \times \mathbf{f}_{\overline{E}\bar{A}JM}(B) \ \ (sum\ out\ E)$$
$$= \alpha \times \mathbf{f}_B(B) \times \mathbf{f}_{\overline{E}\bar{A}JM}(B)$$



  - "$\times$" is the pointwise product (see later)
  - $\mathbf{f}_M(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(m|\ \ a) \\ P(m|\neg a) \end{bmatrix}$, $\mathbf{f}_J(A) \stackrel{\text{def}}{=} \begin{bmatrix} P(j|\ \ a) \\ P(j|\neg a) \end{bmatrix}$, ...
  - $\mathbf{f}_{\bar{A}...}(.)$: summation over the values of A...

Variable Elimination: Basic Operations

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

$$\mathbf{f}_J(A) \times \mathbf{f}_M(A) = \begin{bmatrix} P(j|\ \ a) \\ P(j|\neg a) \end{bmatrix} \times \begin{bmatrix} P(m|\ \ a) \\ P(m|\neg a) \end{bmatrix} = \begin{bmatrix} P(j|\ \ a)P(m|\ \ a) \\ P(j|\neg a)P(m|\neg a) \end{bmatrix}$$

- General case:
  $\mathbf{f}_3(X_1, ..., X_j, Y_1, ..., Y_k, Z_1, ..., Z_l) =$
  $\mathbf{f}_1(X_1, ..., X_j, Y_1, ..., Y_k) \times \mathbf{f}_2(Y_1, ..., Y_k, Z_1, ..., Z_l)$
  - 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: $\mathbf{f}_1 : 2^{j+k}$, $\mathbf{f}_1 : 2^{k+l}$, $\mathbf{f}_3 : 2^{j+k+l}$

Variable Elimination: Basic Operations

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

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

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

$$\mathbf{f}_J(A) \times \mathbf{f}_M(A) = \begin{bmatrix} P(j| \ \ a) \\ P(j|\neg a) \end{bmatrix} \times \begin{bmatrix} P(m| \ \ a) \\ P(m|\neg a) \end{bmatrix} = \begin{bmatrix} P(j| \ \ a)P(m| \ \ a) \\ P(j|\neg a)P(m|\neg a) \end{bmatrix}$$

- General case:
  $\mathbf{f}_3(X_1, ..., X_j, Y_1, ..., Y_k, Z_1, ..., Z_l) =$
  $\mathbf{f}_1(X_1, ..., X_j, Y_1, ..., Y_k) \times \mathbf{f}_2(Y_1, ..., Y_k, Z_1, ..., Z_l)$
  - 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: $\mathbf{f}_1 : 2^{j+k}$, $\mathbf{f}_1 : 2^{k+l}$, $\mathbf{f}_3 : 2^{j+k+l}$

Variable Elimination: Basic Operations

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

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

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

$$\mathbf{f}_J(A) \times \mathbf{f}_M(A) = \begin{bmatrix} P(j| \ \ a) \\ P(j|\neg a) \end{bmatrix} \times \begin{bmatrix} P(m| \ \ a) \\ P(m|\neg a) \end{bmatrix} = \begin{bmatrix} P(j| \ \ a)P(m| \ \ a) \\ P(j|\neg a)P(m|\neg a) \end{bmatrix}$$

- General case:

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

  - 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: $\mathbf{f}_1 : 2^{j+k}$, $\mathbf{f}_1 : 2^{k+l}$, $\mathbf{f}_3 : 2^{j+k+l}$

Variable Elimination: Basic Operations

- $\mathbf{f}_3(A,B,C) = \mathbf{f}_1(A,B) \times \mathbf{f}_2(B,C)$
- 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	$\mathbf{f}_1(A,B)$	B	C	$\mathbf{f}_2(B,C)$	A	B	C	$\mathbf{f}_3(A,B,C)$
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	$.1 \times .4 = .04$

Variable Elimination: Basic Operations

- $\mathbf{f}_3(A, B, C) = \mathbf{f}_1(A, B) \times \mathbf{f}_2(B, C)$
- 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	$\mathbf{f}_1(A, B)$	B	C	$\mathbf{f}_2(B, C)$	A	B	C	$\mathbf{f}_3(A, B, C)$
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	$.1 \times .4 = .04$

Variable Elimination Algorithm

```
function ELIMINATION-ASK(X, e, bn) returns a distribution over X
   inputs: X, the query variable
           e, observed values for variables E
           bn, a Bayesian network specifying joint distribution P(X_1, ..., X_n)

   factors ← [ ]
   for each var in ORDER(bn.VARS) do
       factors ← [MAKE-FACTOR(var, e)|factors]
       if var is a hidden variable then factors ← SUM-OUT(var, factors)
   return NORMALIZE(POINTWISE-PRODUCT(factors))
```


- 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, **e**, bn) **returns** a distribution over X
  inputs: X, the query variable
    e, observed values for variables **E**
    bn, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

  $factors \leftarrow [\,]$
  for each var **in** ORDER(bn.VARS) **do**
    $factors \leftarrow$ [MAKE-FACTOR(var, **e**)|$factors$]
    if var is a hidden variable **then** $factors \leftarrow$ SUM-OUT(var, $factors$)
  return NORMALIZE(POINTWISE-PRODUCT($factors$))

- 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, **e**, bn) **returns** a distribution over X
  inputs: X, the query variable
    e, observed values for variables **E**
    bn, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

  $factors \leftarrow [\,]$
  for each var **in** ORDER(bn.VARS) **do**
    $factors \leftarrow$ [MAKE-FACTOR(var, **e**)|$factors$]
    if var is a hidden variable **then** $factors \leftarrow$ SUM-OUT(var, $factors$)
  return NORMALIZE(POINTWISE-PRODUCT($factors$))

- 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

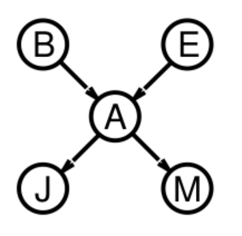
- If $f_1, ..., f_i$ do not depend on X, then move them out of a $\sum_x (...)$:
  $\sum_x f_1 \times \cdots \times f_k =$
  $f_1 \times \cdots \times f_i \sum_x (f_{i+1} \times \cdots \times f_k) =$
  $f_1 \times \cdots \times f_i \times f_X$
- Ex: $\sum_a \mathbf{f}_1(A, B) \times \mathbf{f}_2(B, C)$
  $= \mathbf{f}_2(B, C) \times \sum_a \mathbf{F}_1(A, B)$
- Ex: $\mathbf{P}(JohnCalls|Burglary = true)$:
  $\mathbf{P}(J|b) = \alpha \sum_e \sum_a \sum_m \mathbf{P}(J, m, b, e, a) =$
  $\alpha \sum_e \sum_a \sum_m P(b)P(e)P(a|b, e)\mathbf{P}(J|a)P(m|a) =$
  $\alpha P(b) \sum_e P(e) \sum_a P(a|b, e)\mathbf{P}(J|a) \sum_m P(m|a)$

Factor Out Constant Factors

- If $f_1, ..., f_i$ do not depend on X, then move them out of a $\sum_x(...)$:
  $\sum_x f_1 \times \cdots \times f_k =$
  $f_1 \times \cdots \times f_i \sum_x (f_{i+1} \times \cdots \times f_k) =$
  $f_1 \times \cdots \times f_i \times f_X$
- 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: $\mathbf{P}(JohnCalls | Burglary = true)$:
  $\mathbf{P}(J|b) = \alpha \sum_e \sum_a \sum_m \mathbf{P}(J, m, b, e, a) =$
  $\alpha \sum_e \sum_a \sum_m P(b)P(e)P(a|b, e)\mathbf{P}(J|a)P(m|a) =$
  $\alpha P(b) \sum_e P(e) \sum_a P(a|b, e)\mathbf{P}(J|a) \sum_m P(m|a)$

Factor Out Constant Factors

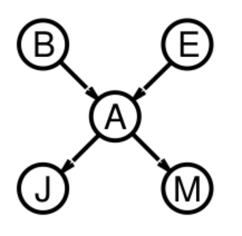
- If $f_1, ..., f_i$ do not depend on X, then move them out of a $\sum_x(...)$:
  $\sum_x f_1 \times \cdots \times f_k =$
  $f_1 \times \cdots \times f_i \sum_x(f_{i+1} \times \cdots \times f_k) =$
  $f_1 \times \cdots \times f_i \times f_X$
- Ex: $\sum_a \mathbf{f}_1(A, B) \times \mathbf{f}_2(B, C)$
  $= \mathbf{f}_2(B, C) \times \sum_a \mathbf{F}_1(A, B)$
- Ex: $\mathbf{P}(JohnCalls|Burglary = true)$:
  $\mathbf{P}(J|b) = \alpha \sum_e \sum_a \sum_m \mathbf{P}(J, m, b, e, a) =$
  $\alpha \sum_e \sum_a \sum_m P(b)P(e)P(a|b, e)\mathbf{P}(J|a)P(m|a) =$
  $\alpha P(b) \sum_e P(e) \sum_a P(a|b, e)\mathbf{P}(J|a) \sum_m P(m|a)$



(© S. Russell & P. Norwig, AIMA)

Factor Out Constant Factors

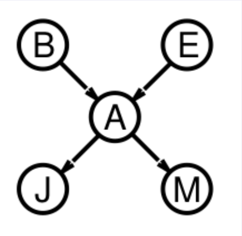
- If $f_1, ..., f_i$ do not depend on X, then move them out of a $\sum_x(...)$:
  $\sum_x f_1 \times \cdots \times f_k =$
  $f_1 \times \cdots \times f_i \sum_x (f_{i+1} \times \cdots \times f_k) =$
  $f_1 \times \cdots \times f_i \times f_X$
- Ex: $\sum_a \mathbf{f}_1(A, B) \times \mathbf{f}_2(B, C)$
  $= \mathbf{f}_2(B, C) \times \sum_a \mathbf{F}_1(A, B)$
- Ex: $\mathbf{P}(JohnCalls | Burglary = true)$:
  $\mathbf{P}(J|b) = \alpha \sum_e \sum_a \sum_m \mathbf{P}(J, m, b, e, a) =$
  $\alpha \sum_e \sum_a \sum_m P(b)P(e)P(a|b,e)\mathbf{P}(J|a)P(m|a) =$
  $\alpha P(b) \sum_e P(e) \sum_a P(a|b,e)\mathbf{P}(J|a) \sum_m P(m|a)$



(© S. Russell & P. Norwig, AIMA)

Factor Out Constant Factors

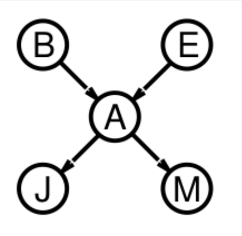
- If $f_1, ..., f_i$ do not depend on X, then move them out of a $\sum_x(...)$:
  $\sum_x f_1 \times \cdots \times f_k =$
  $f_1 \times \cdots \times f_i \sum_x (f_{i+1} \times \cdots \times f_k) =$
  $f_1 \times \cdots \times f_i \times f_X$
- Ex: $\sum_a \mathbf{f}_1(A, B) \times \mathbf{f}_2(B, C)$
  $= \mathbf{f}_2(B, C) \times \sum_a \mathbf{F}_1(A, B)$
- Ex: $\mathbf{P}(JohnCalls|Burglary = true)$:
  $\mathbf{P}(J|b) = \alpha \sum_e \sum_a \sum_m \mathbf{P}(J, m, b, e, a) =$
  $\alpha \sum_e \sum_a \sum_m P(b)P(e)P(a|b, e)\mathbf{P}(J|a)P(m|a) =$
  $\alpha P(b) \sum_e P(e) \sum_a P(a|b, e)\mathbf{P}(J|a) \sum_m P(m|a)$



(© S. Russell & P. Norwig, AIMA)

Factor Out Constant Factors

- If $f_1, ..., f_i$ do not depend on X, then move them out of a $\sum_x(...)$:
  $\sum_x f_1 \times \cdots \times f_k =$
  $f_1 \times \cdots \times f_i \sum_x (f_{i+1} \times \cdots \times f_k) =$
  $f_1 \times \cdots \times f_i \times f_X$
- 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: $\mathbf{P}(JohnCalls|Burglary = true)$:
  $\mathbf{P}(J|b) = \alpha \sum_e \sum_a \sum_m \mathbf{P}(J, m, b, e, a) =$
  $\alpha \sum_e \sum_a \sum_m P(b)P(e)P(a|b, e)\mathbf{P}(J|a)P(m|a) =$
  $\alpha P(b) \sum_e P(e) \sum_a P(a|b, e)\mathbf{P}(J|a) \sum_m P(m|a)$



(© S. Russell & P. Norwig, AIMA)

Remove Irrelevant Variables

- Sometimes we fave summations like $\sum_y P(y|...)$
  - $\sum_y P(y|...) = 1 \implies$ can be dropped
- Ex: $\mathbf{P}(JohnCalls|Burglary = true)$:
  $\mathbf{P}(J|b) = ... =$
  $\alpha P(b) \sum_e P(e) \sum_a P(a|b,e) \mathbf{P}(J|a) \overbrace{\sum_m P(m|a)}^{1}$
  $\alpha P(b) \sum_e P(e) \sum_a P(a|b,e) \mathbf{P}(J|a)$
- Theorem: For query X and evidence **E**,
  Y is irrelevant unless $Y \in Ancestors(X \cup \mathbf{E})$
- Ex: $X = JohnCalls$, $\mathbf{E} = \{Burglary\}$, and
  $Ancestors(\{X\} \cup \mathbf{E}) = \{Alarm, Earthquake\}$

$\implies$ *MaryCalls* is irrelevant

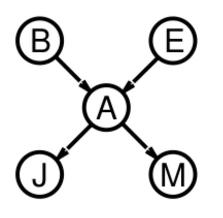
- Related to backward-chaining with Horn clauses

Remove Irrelevant Variables

- Sometimes we fave summations like $\sum_y P(y|\ldots)$
  - $\sum_y P(y|\ldots) = 1 \Longrightarrow$ can be dropped
- Ex: $\mathbf{P}(JohnCalls|Burglary = true)$:
  $\mathbf{P}(J|b) = \ldots =$
  $\alpha P(b) \sum_e P(e) \sum_a P(a|b,e) \mathbf{P}(J|a) \overbrace{\sum_m P(m|a)}^{1}$
  $\alpha P(b) \sum_e P(e) \sum_a P(a|b,e) \mathbf{P}(J|a)$
- Theorem: For query X and evidence **E**,
  Y is irrelevant unless $Y \in Ancestors(X \cup \mathbf{E})$
- Ex: $X = JohnCalls$, $\mathbf{E} = \{Burglary\}$, and
  $Ancestors(\{X\} \cup \mathbf{E}) = \{Alarm, Earthquake\}$

$\Longrightarrow$ *MaryCalls* is irrelevant

- Related to backward-chaining with Horn clauses



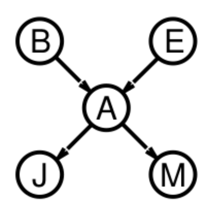
(© S. Russell & P. Norwig, AIMA)

Remove Irrelevant Variables

- Sometimes we fave summations like $\sum_y P(y|\ldots)$
  - $\sum_y P(y|\ldots) = 1 \implies$ can be dropped
- Ex: $\mathbf{P}(JohnCalls|Burglary = true)$:
  $\mathbf{P}(J|b) = \ldots =$
  $\alpha P(b)\sum_e P(e)\sum_a P(a|b,e)\mathbf{P}(J|a)\overbrace{\sum_m P(m|a)}^{1}$
  $\alpha P(b)\sum_e P(e)\sum_a P(a|b,e)\mathbf{P}(J|a)$
- Theorem: For query X and evidence **E**, Y is irrelevant unless $Y \in Ancestors(X \cup \mathbf{E})$
- Ex: $X = JohnCalls$, $\mathbf{E} = \{Burglary\}$, and $Ancestors(\{X\} \cup \mathbf{E}) = \{Alarm, Earthquake\}$

$\implies$ *MaryCalls* is irrelevant

- Related to backward-chaining with Horn clauses



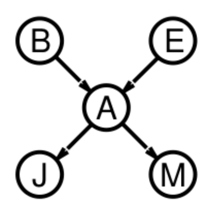
(© S. Russell & P. Norwig, AIMA)

Remove Irrelevant Variables

- Sometimes we fave summations like $\sum_y P(y|...)$
  - $\sum_y P(y|...) = 1 \Longrightarrow$ can be dropped
- Ex: $\mathbf{P}(JohnCalls|Burglary = true)$:
  $\mathbf{P}(J|b) = ... =$
  $\alpha P(b) \sum_e P(e) \sum_a P(a|b,e) \mathbf{P}(J|a) \overbrace{\sum_m P(m|a)}^{1}$
  $\alpha P(b) \sum_e P(e) \sum_a P(a|b,e) \mathbf{P}(J|a)$
- Theorem: For query X and evidence $\mathbf{E}$, Y is irrelevant unless $Y \in Ancestors(X \cup \mathbf{E})$
- Ex: $X = JohnCalls$, $\mathbf{E} = \{Burglary\}$, and $Ancestors(\{X\} \cup \mathbf{E}) = \{Alarm, Earthquake\}$

$\Longrightarrow$ *MaryCalls* is irrelevant

- Related to backward-chaining with Horn clauses

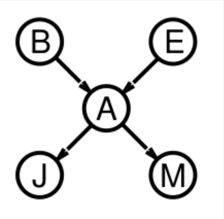


(© S. Russell & P. Norwig, AIMA)

- Sometimes we fave summations like $\sum_y P(y|...)$
  - $\sum_y P(y|...) = 1 \Longrightarrow$ can be dropped
- Ex: $\mathbf{P}(JohnCalls|Burglary = true)$:
  $\mathbf{P}(J|b) = ... =$
  $\alpha P(b)\sum_e P(e)\sum_a P(a|b,e)\mathbf{P}(J|a)\overbrace{\sum_m P(m|a)}^{1}$
  $\alpha P(b)\sum_e P(e)\sum_a P(a|b,e)\mathbf{P}(J|a)$
- Theorem: For query X and evidence **E**, Y is irrelevant unless $Y \in Ancestors(X \cup \mathbf{E})$
- Ex: $X = JohnCalls$, $\mathbf{E} = \{Burglary\}$, and $Ancestors(\{X\} \cup \mathbf{E}) = \{Alarm, Earthquake\}$

$\Longrightarrow$ $MaryCalls$ is irrelevant

- Related to backward-chaining with Horn clauses



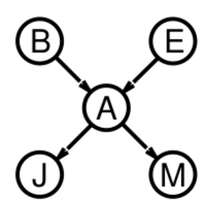
(© S. Russell & P. Norwig, AIMA)

Remove Irrelevant Variables

- Sometimes we fave summations like $\sum_y P(y|...)$
  - $\sum_y P(y|...) = 1 \Longrightarrow$ can be dropped
- Ex: $\mathbf{P}(JohnCalls|Burglary = true)$:
  $\mathbf{P}(J|b) = ... =$
  $\alpha P(b) \sum_e P(e) \sum_a P(a|b,e) \mathbf{P}(J|a) \overbrace{\sum_m P(m|a)}^{1}$
  $\alpha P(b) \sum_e P(e) \sum_a P(a|b,e) \mathbf{P}(J|a)$
- Theorem: For query X and evidence **E**, Y is irrelevant unless $Y \in Ancestors(X \cup \mathbf{E})$
- Ex: $X = JohnCalls$, $\mathbf{E} = \{Burglary\}$, and $Ancestors(\{X\} \cup \mathbf{E}) = \{Alarm, Earthquake\}$

$\Longrightarrow$ *MaryCalls* is irrelevant

- Related to backward-chaining with Horn clauses



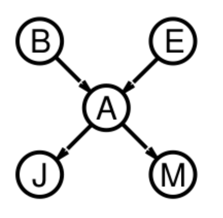
(© S. Russell & P. Norwig, AIMA)

Remove Irrelevant Variables

- Sometimes we fave summations like $\sum_y P(y|\ldots)$
  - $\sum_y P(y|\ldots) = 1 \implies$ can be dropped
- Ex: $\mathbf{P}(JohnCalls|Burglary = true)$:
  $\mathbf{P}(J|b) = \ldots =$
  $\alpha P(b)\sum_e P(e)\sum_a P(a|b,e)\mathbf{P}(J|a)\overbrace{\sum_m P(m|a)}^{1}$
  $\alpha P(b)\sum_e P(e)\sum_a P(a|b,e)\mathbf{P}(J|a)$
- Theorem: For query X and evidence **E**,
  Y is irrelevant unless $Y \in Ancestors(X \cup \mathbf{E})$
- Ex: $X = JohnCalls$, $\mathbf{E} = \{Burglary\}$, and
  $Ancestors(\{X\} \cup \mathbf{E}) = \{Alarm, Earthquake\}$

$\Longrightarrow$ *MaryCalls* is irrelevant

- Related to backward-chaining with Horn clauses



(© S. Russell & P. Norwig, AIMA)

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

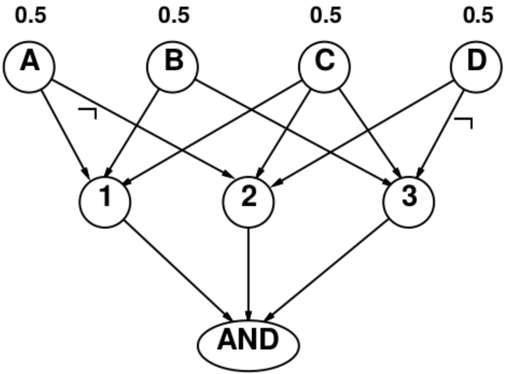
Complexity of Exact Inference

- We can reduce 3SAT to exact inference in Bayesian Networks $\Longrightarrow$ NP-Hard
- Ex:

Complexity of Exact Inference

- We can reduce 3SAT to exact inference in Bayesian Networks $\Longrightarrow$ NP-Hard
- Ex:

1. A v B v C
2. C v D v ¬A
3. B v C v ¬D



(© S. Russell & P. Norwig, AIMA)


"That's all Folks!"