Bayesian networks

Chapter 14.1–3

Outline

◇ Syntax

◇ Semantics

◇ Parameterized distributions

Bayesian networks

A simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions

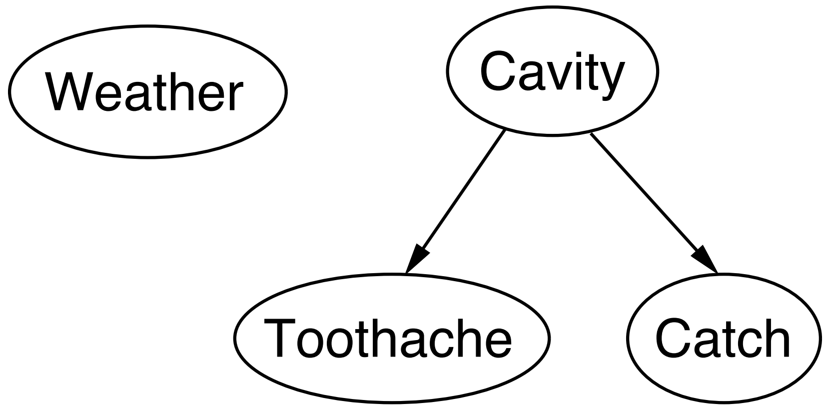
Syntax:

a set of nodes, one per variable
a directed, acyclic graph (link $\approx$ "directly influences")
a conditional distribution for each node given its parents:
$\mathbf{P}(X_i|Parents(X_i))$

In the simplest case, conditional distribution represented as a conditional probability table (CPT) giving the distribution over X_i for each combination of parent values

Example

Topology of network encodes conditional independence assertions:



$Weather$ is independent of the other variables

$Toothache$ and $Catch$ are conditionally independent given $Cavity$

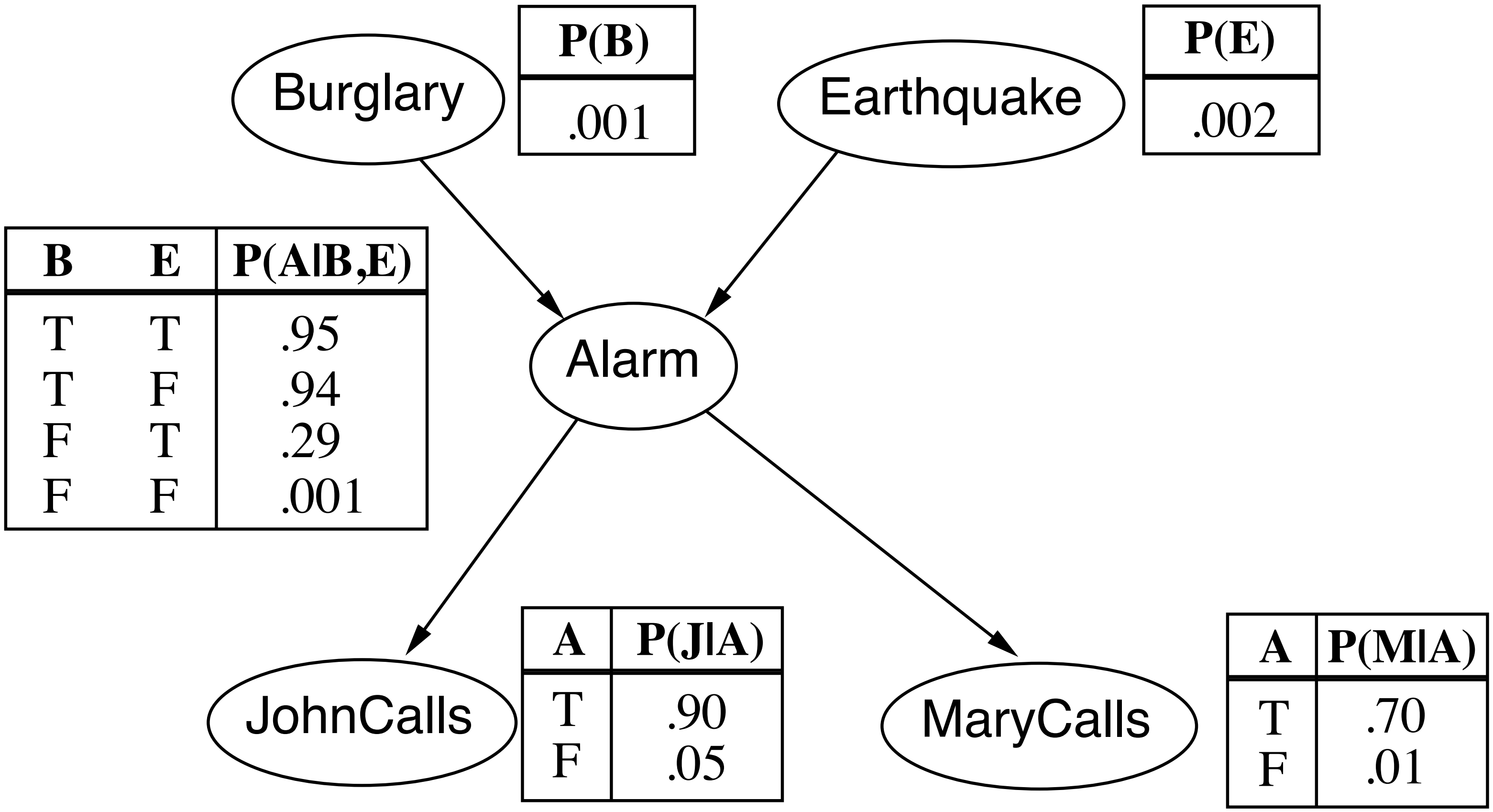
Example

I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar?

Variables: $Burglar$, $Earthquake$, $Alarm$, $JohnCalls$, $MaryCalls$
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

Example contd.



P(B)
.001

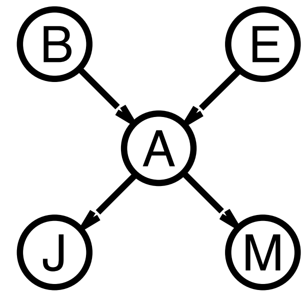
P(E)
.002

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

Compactness



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 $X_i = true$
(the number for $X_i = false$ is just $1 - p$)

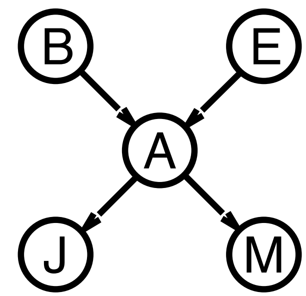
If each variable has no more than k parents,
the complete network requires $O(n \cdot 2^k)$ numbers

I.e., grows linearly with n, vs. $O(2^n)$ for the full joint distribution

For burglary net, $1 + 1 + 4 + 2 + 2 = 10$ numbers (vs. $2^5 - 1 = 31$)

Global semantics

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



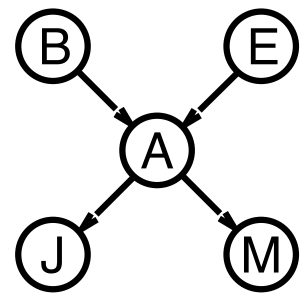
$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i))$$

e.g., $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$

$=$

Global semantics

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



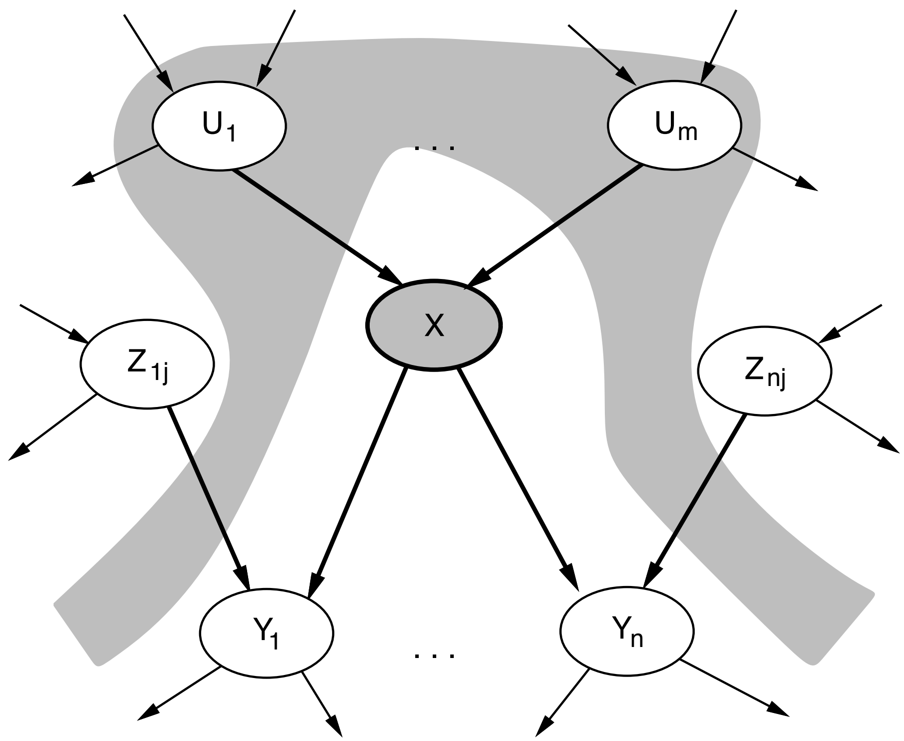
$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i))$$

e.g., $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$

$$\begin{aligned} &= P(j|a)P(m|a)P(a|\neg b, \neg e)P(\neg b)P(\neg e) \\ &= 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998 \\ &\approx 0.00063 \end{aligned}$$

Local semantics

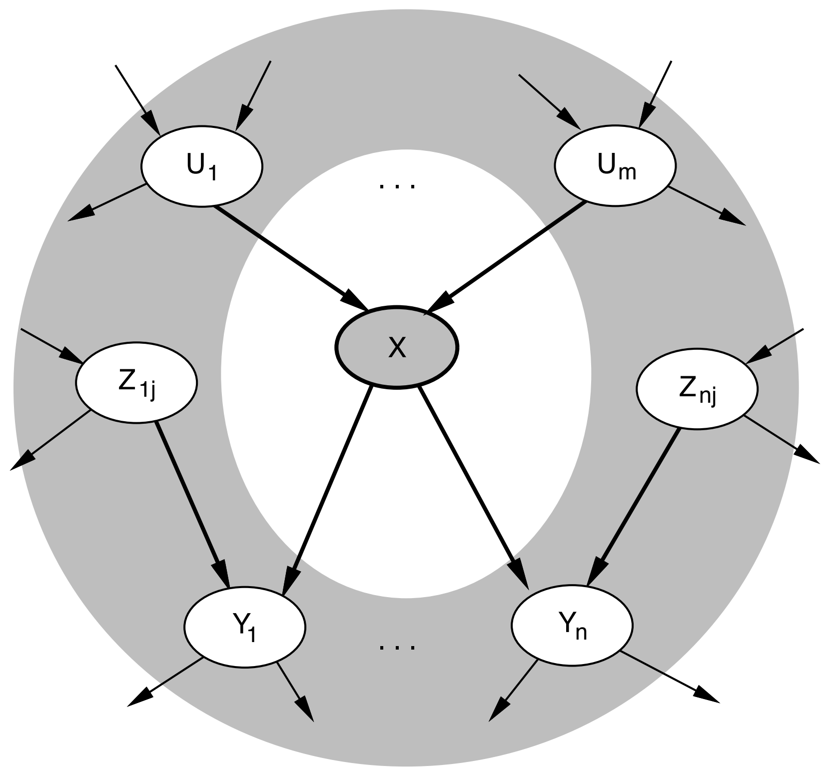
Local semantics: each node is conditionally independent of its nondescendants given its parents



Theorem: Local semantics $\Leftrightarrow$ global semantics

Markov blanket

Each node is conditionally independent of all others given its Markov blanket: parents + children + children's parents



Constructing Bayesian networks

Need a method such that a series of locally testable assertions of conditional independence guarantees the required global semantics

1. Choose an ordering of variables $X_1, \ldots, X_n$
2. For $i = 1$ to n
   add X_i to the network
   select parents from $X_1, \ldots, X_{i-1}$ such that
   $\mathbf{P}(X_i | Parents(X_i)) = \mathbf{P}(X_i | X_1, \ldots, X_{i-1})$

This choice of parents guarantees the global semantics:

$$\begin{aligned} \mathbf{P}(X_1, \ldots, X_n) &= \prod_{i=1}^{n} \mathbf{P}(X_i | X_1, \ldots, X_{i-1}) \quad \text{(chain rule)} \\ &= \prod_{i=1}^{n} \mathbf{P}(X_i | Parents(X_i)) \quad \text{(by construction)} \end{aligned}$$

Inference tasks

Simple queries: compute posterior marginal $\mathbf{P}(X_i|\mathbf{E}=\mathbf{e})$
e.g., $P(NoGas|Gauge=empty, Lights=on, Starts=false)$

Conjunctive queries: $\mathbf{P}(X_i, X_j|\mathbf{E}=\mathbf{e}) = \mathbf{P}(X_i|\mathbf{E}=\mathbf{e})\mathbf{P}(X_j|X_i, \mathbf{E}=\mathbf{e})$

Optimal decisions: decision networks include utility information;
probabilistic inference required for $P(outcome|action, evidence)$

Value of information: which evidence to seek next?

Sensitivity analysis: which probability values are most critical?

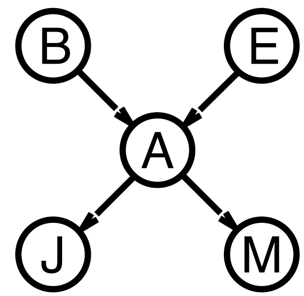
Explanation: why do I need a new starter motor?

Inference by enumeration

Slightly intelligent way to sum out variables from the joint without actually constructing its explicit representation

Simple query on the burglary network:

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



Rewrite full joint entries using product of CPT entries:

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

Recursive depth-first enumeration: $O(n)$ space, $O(d^n)$ time

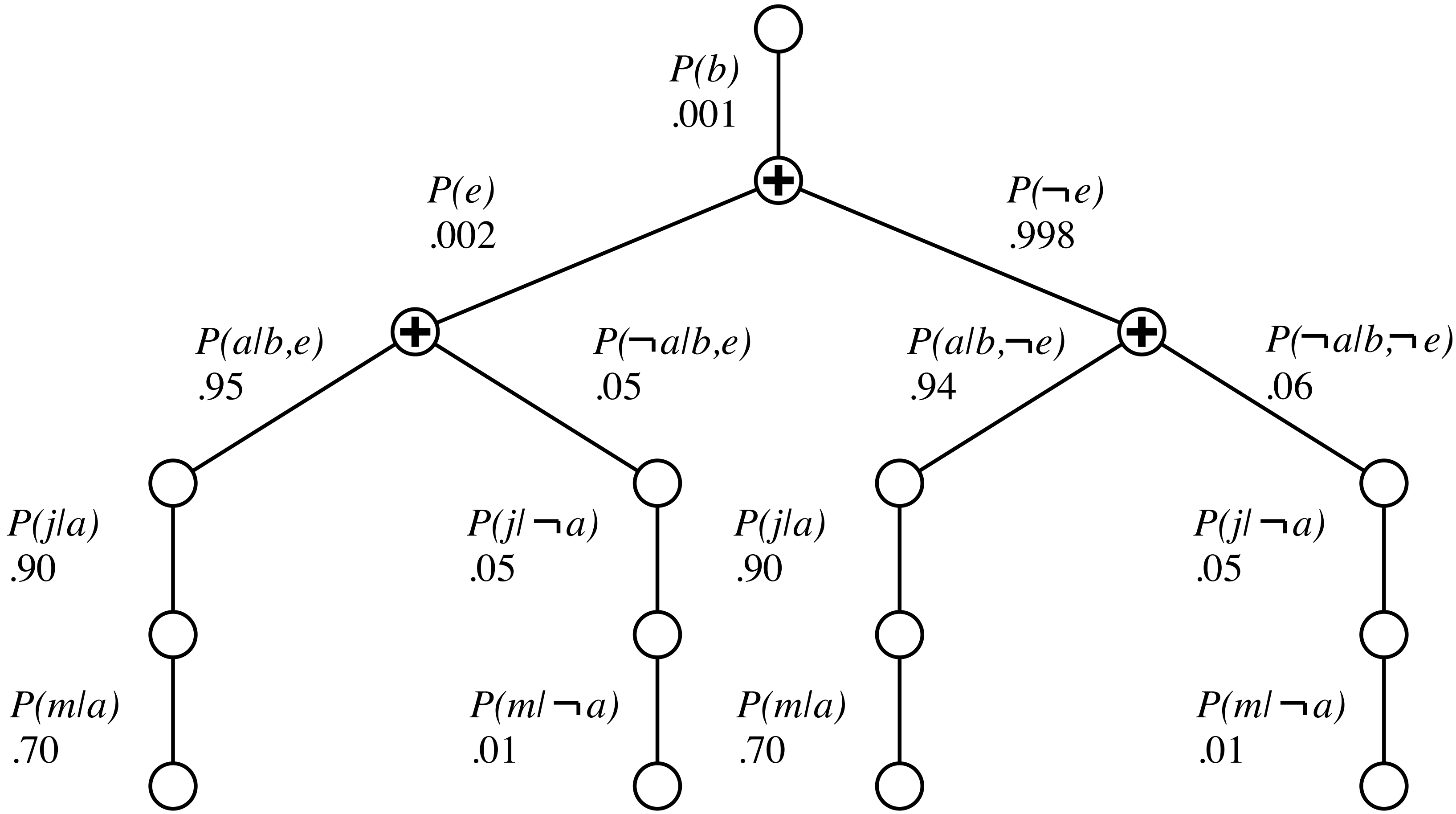
Enumeration algorithm

```
function Enumeration-Ask(X, e, bn) returns a distribution over X
   inputs: X, the query variable
           e, observed values for variables E
           bn, a Bayesian network with variables {X} ∪ E ∪ Y

   Q(X) ← a distribution over X, initially empty
   for each value x_i of X do
       extend e with value x_i for X
       Q(x_i) ← Enumerate-All(Vars[bn], e)
   return Normalize(Q(X))
```

```
function Enumerate-All(vars, e) returns a real number
   if Empty?(vars) then return 1.0
   Y ← First(vars)
   if Y has value y in e
       then return P(y | Pa(Y)) × Enumerate-All(Rest(vars), e)
       else return Σ_y P(y | Pa(Y)) × Enumerate-All(Rest(vars), e_y)
            where e_y is e extended with Y = y
```

Evaluation tree



Enumeration is inefficient: repeated computation
e.g., computes $P(j|a)P(m|a)$ for each value of e

Inference by variable elimination

Variable elimination: carry out summations right-to-left,
storing intermediate results (factors) to avoid recomputation

$$\mathbf{P}(B|j,m)$$
$$= \alpha \underbrace{\mathbf{P}(B)}_{B} \textstyle\sum_e \underbrace{P(e)}_{E} \textstyle\sum_a \underbrace{\mathbf{P}(a|B,e)}_{A} \underbrace{P(j|a)}_{J} \underbrace{P(m|a)}_{M}$$
$$= \alpha \mathbf{P}(B) \textstyle\sum_e P(e) \textstyle\sum_a \mathbf{P}(a|B,e) P(j|a) f_M(a)$$
$$= \alpha \mathbf{P}(B) \textstyle\sum_e P(e) \textstyle\sum_a \mathbf{P}(a|B,e) f_J(a) f_M(a)$$
$$= \alpha \mathbf{P}(B) \textstyle\sum_e P(e) \textstyle\sum_a f_A(a,b,e) f_J(a) f_M(a)$$
$$= \alpha \mathbf{P}(B) \textstyle\sum_e P(e) f_{\bar{A}JM}(b,e) \text{ (sum out } A\text{)}$$
$$= \alpha \mathbf{P}(B) f_{\bar{E}\bar{A}JM}(b) \text{ (sum out } E\text{)}$$
$$= \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b)$$

Variable elimination: Basic operations

Summing out a variable from a product of factors:

move any constant factors outside the summation

add up submatrices in pointwise product of remaining factors

$$\sum_x f_1 \times \cdots \times f_k = f_1 \times \cdots \times f_i \sum_x f_{i+1} \times \cdots \times f_k = f_1 \times \cdots \times f_i \times f_{\bar{X}}$$

assuming $f_1, \ldots, f_i$ do not depend on X

Pointwise product of factors f_1 and f_2:

$$f_1(x_1, \ldots, x_j, y_1, \ldots, y_k) \times f_2(y_1, \ldots, y_k, z_1, \ldots, z_l)$$
$$= f(x_1, \ldots, x_j, y_1, \ldots, y_k, z_1, \ldots, z_l)$$

E.g., $f_1(a, b) \times f_2(b, c) = f(a, b, c)$

Pointwise Product

- Pointwise multiplication of factors when variable is summed out or at last step
- Pointwise product of factors f_1 and f_2:

$$f_1(x_1,\ldots,x_j,y_1,\ldots,y_k) \times f_2(y_1,\ldots,y_k,z_1,\ldots,z_l) = f(x_1,\ldots,x_j,y_1,\ldots,y_k,z_1,\ldots,z_l)$$

- E.g. $f_1(a,b) \times f_2(b,c) = f(a,b,c)$:

a	b	$f_1(a,b)$
T	T	.3
T	F	.7
F	T	.9
F	F	.1

b	c	$f_2(b,c)$
T	T	.2
T	F	.8
F	T	.6
F	F	.4

a	b	c	$f(a,b,c)$
T	T	T	.3 * .2
T	T	F	.3 * .8
T	F	T	.7 * .6
T	F	F	.7 * .4
F	T	T	.9 * .2
F	T	F	.9 * .8
F	F	T	.1 * .6
F	F	F	.1 * .4

Summing Out

- Summing out a variable from a product of factors
  - Move any constant factors outside the summation
  - Add up sub-matrices in pointwise product of remaining factors

$$\sum_x f_1 \times \cdots \times f_k = f_1 \times \cdots \times f_l \times \left(\sum_x f_{l+1} \times \cdots \times f_k \right) = f_1 \times \cdots \times f_l \times f_{\overline{X}}$$

- E.g. $\sum_a f(a,b,c) = f_{\overline{a}}(b,c)$:

a	b	c	$f(a,b,c)$	b	c	$f_{\overline{a}}(b,c)$
T	T	T	.3 * .2	T	T	.3 * .2 + .9 * .2
T	T	F	.3 * .8	T	F	.3 * .8 + .9 * .8
T	F	T	.7 * .6	F	T	.7 * .6 + .1 * .6
T	F	F	.7 * .4	F	F	.7 * .4 + .1 * .4
F	T	T	.9 * .2			
F	T	F	.9 * .8			
F	F	T	.1 * .6			
F	F	F	.1 * .4			

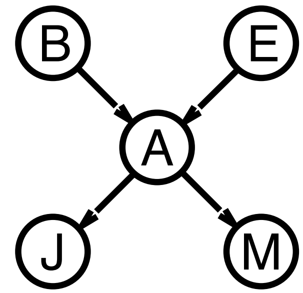
Variable elimination algorithm

```
function ELIMINATION-ASK(X, e, bn) returns a distribution over X
   inputs: X, the query variable
           e, evidence specified as an event
           bn, a belief network specifying joint distribution P(X_1, ..., X_n)

   factors ← [ ]; vars ← REVERSE(VARS[bn])
   for each var in 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))
```

Irrelevant variables

Consider the query $P(JohnCalls|Burglary = true)$



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

Sum over m is identically 1; M is **irrelevant** to the query

Thm 1: Y is irrelevant unless $Y \in Ancestors(\{X\} \cup \mathbf{E})$

Here, $X = JohnCalls$, $\mathbf{E} = \{Burglary\}$, and
$Ancestors(\{X\} \cup \mathbf{E}) = \{Alarm, Earthquake\}$
so $MaryCalls$ is irrelevant

(Compare this to backward chaining from the query in Horn clause KBs)

Complexity of exact inference

Singly connected networks (or polytrees):
- any two nodes are connected by at most one (undirected) path
- time and space cost of variable elimination are $O(d^k n)$

Multiply connected networks:
- can reduce 3SAT to exact inference $\Rightarrow$ NP-hard
- equivalent to **counting** 3SAT models $\Rightarrow$ #P-complete

1. A v B v C

2. C v D v ¬A

3. B v C v ¬D