

BAYESIAN NETWORKS

CHAPTER 14, SECTIONS 1–2

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”)

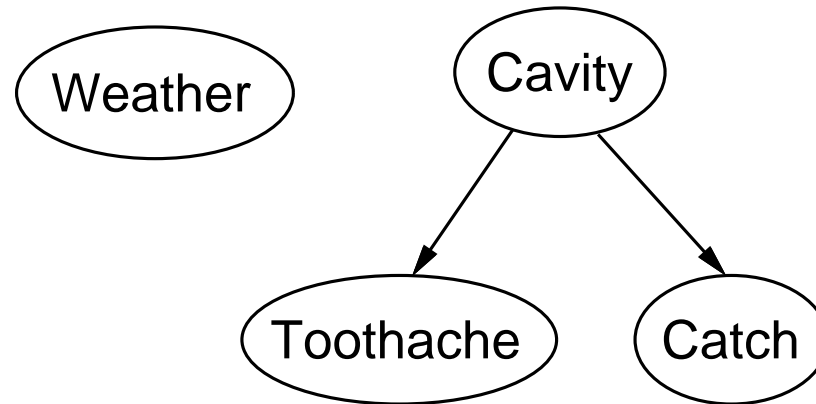
- a conditional distribution for each node given its parents:

$$P(X_i | Parents(X_i))$$

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

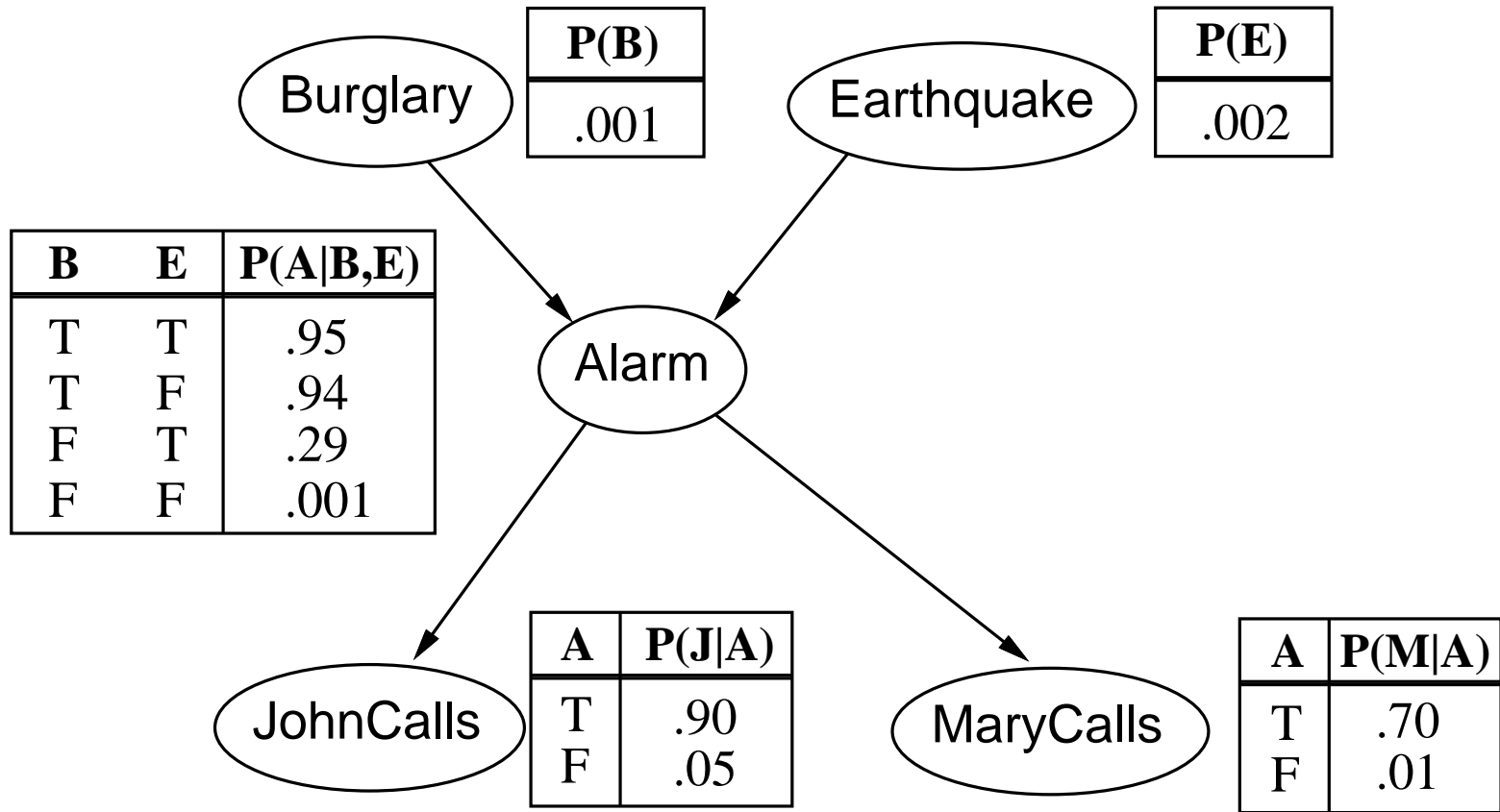
I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar?

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.



Compactness

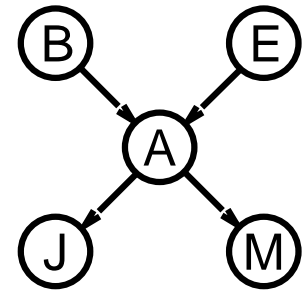
A CPT for Boolean X_i with k Boolean parents has 2^k rows for the combinations of parent values

Each row requires one number p for $X_i = \text{true}$ (the number for $X_i = \text{false}$ 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. $2^5 - 1 = 31$)



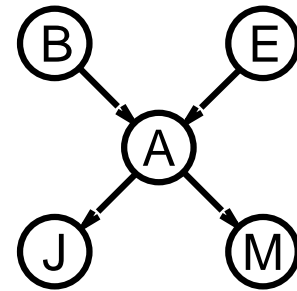
Global semantics

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

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

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:

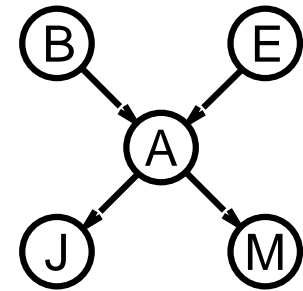
$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{parents}(X_i))$$

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

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

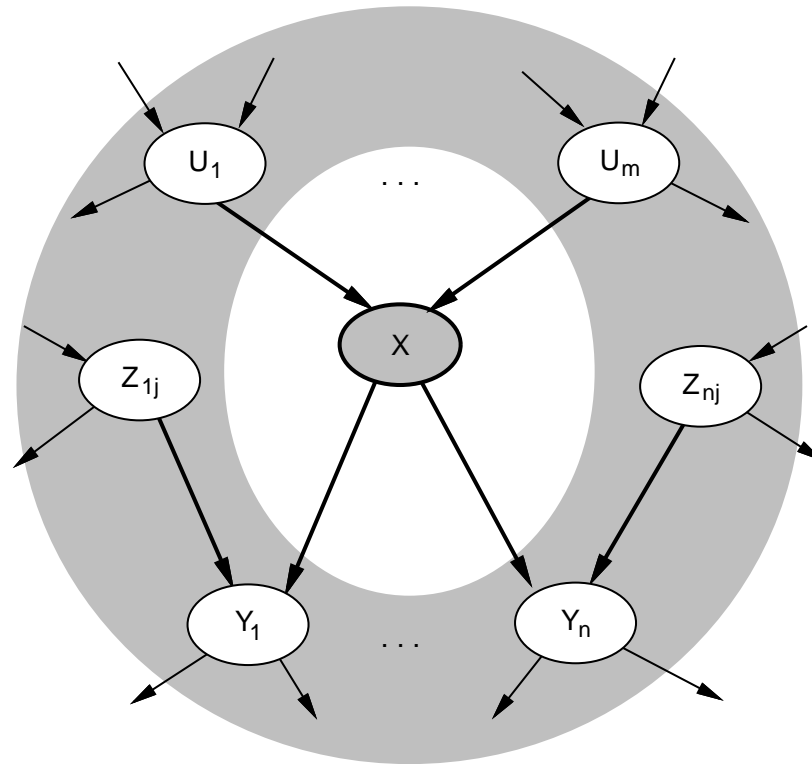
$$= 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998$$

$$\approx 0.00063$$



Markov blanket

Theorem: 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, \dots, X_n
2. For $i = 1$ to n
 - add X_i to the network
 - select parents from X_1, \dots, X_{i-1} such that
$$\mathbf{P}(X_i | \text{Parents}(X_i)) = \mathbf{P}(X_i | X_1, \dots, X_{i-1})$$

This choice of parents guarantees the global semantics:

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

Example

Suppose we choose the ordering M, J, A, B, E

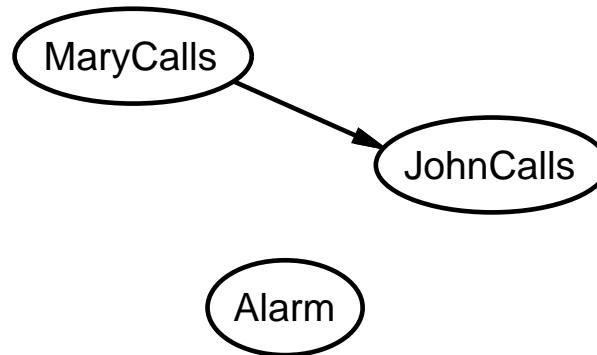
MaryCalls

JohnCalls

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

Example

Suppose we choose the ordering M, J, A, B, E

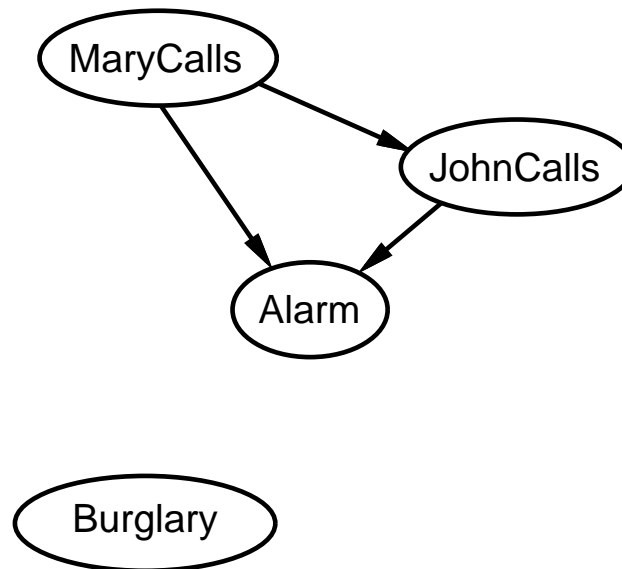


$P(J|M) = P(J)$? No

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

Example

Suppose we choose the ordering M, J, A, B, E



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

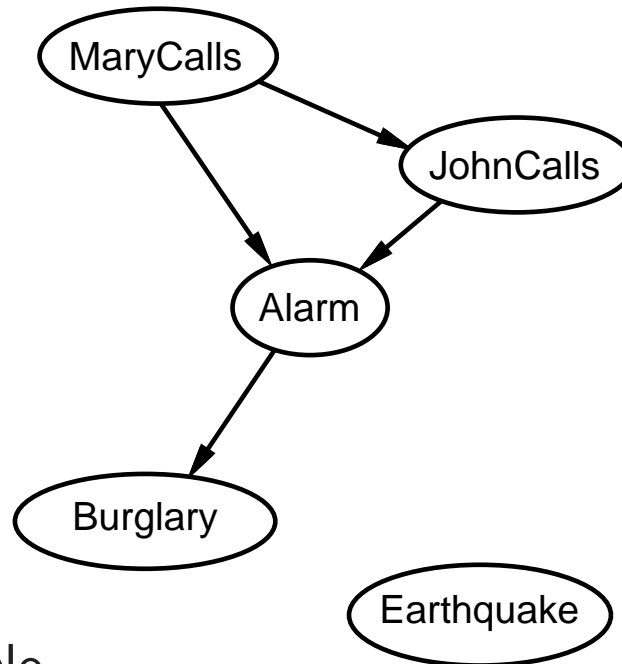
$P(A|J, M) = P(A|J)$? $P(A|J, M) = P(A)$? No

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

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

Example

Suppose we choose the ordering M, J, A, B, E



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

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

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

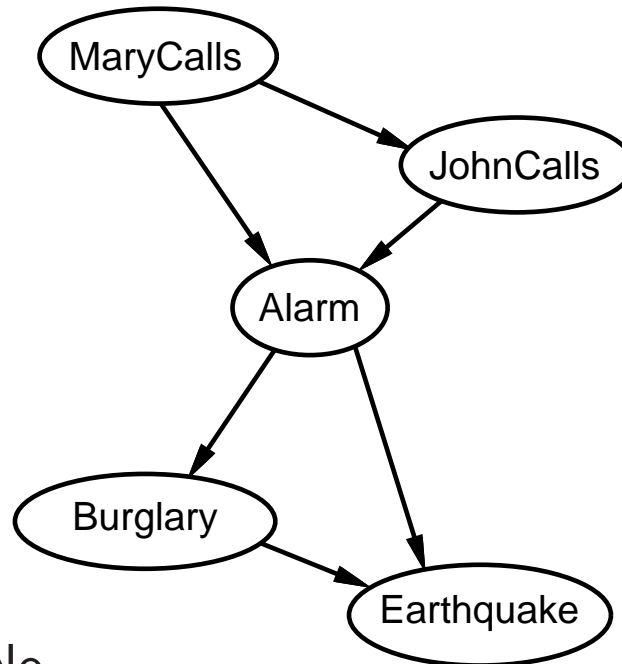
$P(B|A, J, M) = P(B)$? No

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

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

Example

Suppose we choose the ordering M, J, A, B, E



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

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

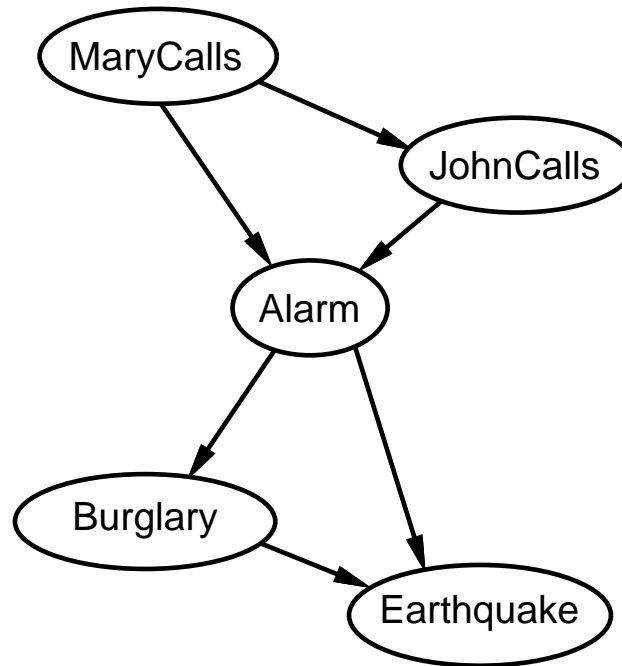
$P(B|A, J, M) = P(B|A)$? Yes

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

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

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

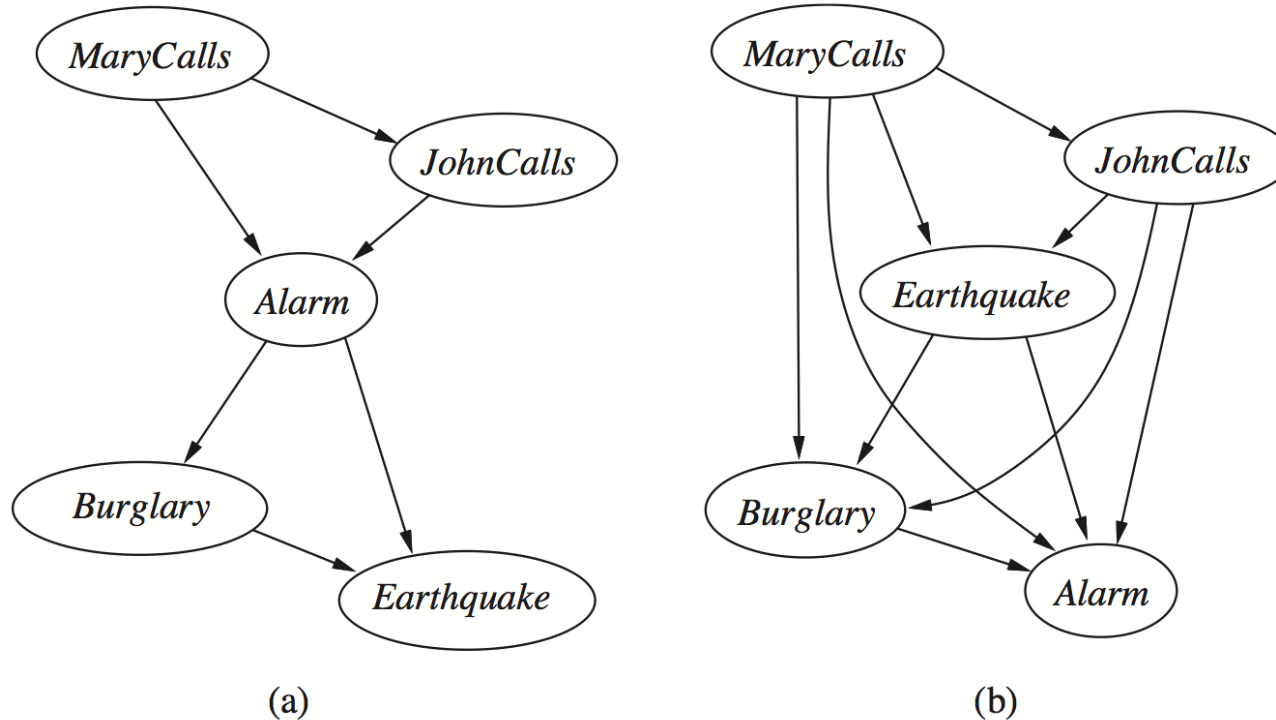
Example contd.



Network is less compact: $1 + 2 + 4 + 2 + 4 = 13$ numbers needed

Compare with the original burglary net: $1 + 1 + 4 + 2 + 2 = 10$ numbers

Example contd.



The chosen ordering of the variables can have a big impact on the size of the network! Network (b) has $2^5 - 1 = 31$ numbers—exactly the same as the full joint distribution

Inference tasks

Simple queries: compute posterior marginal $\mathbf{P}(X_i|\mathbf{E} = \mathbf{e})$

e.g., $P(\textit{Burglar}|\textit{JohnCalls} = \textit{true}, \textit{MaryCalls} = \textit{true})$

or shorter, $P(B|j, m)$

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 $P(\textit{outcome}|\textit{action}, \textit{evidence})$

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?

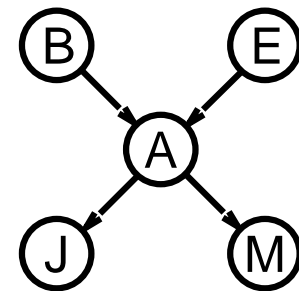
Inference by enumeration

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

Simple query on the burglary network:

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

(where e and a are the hidden variables)

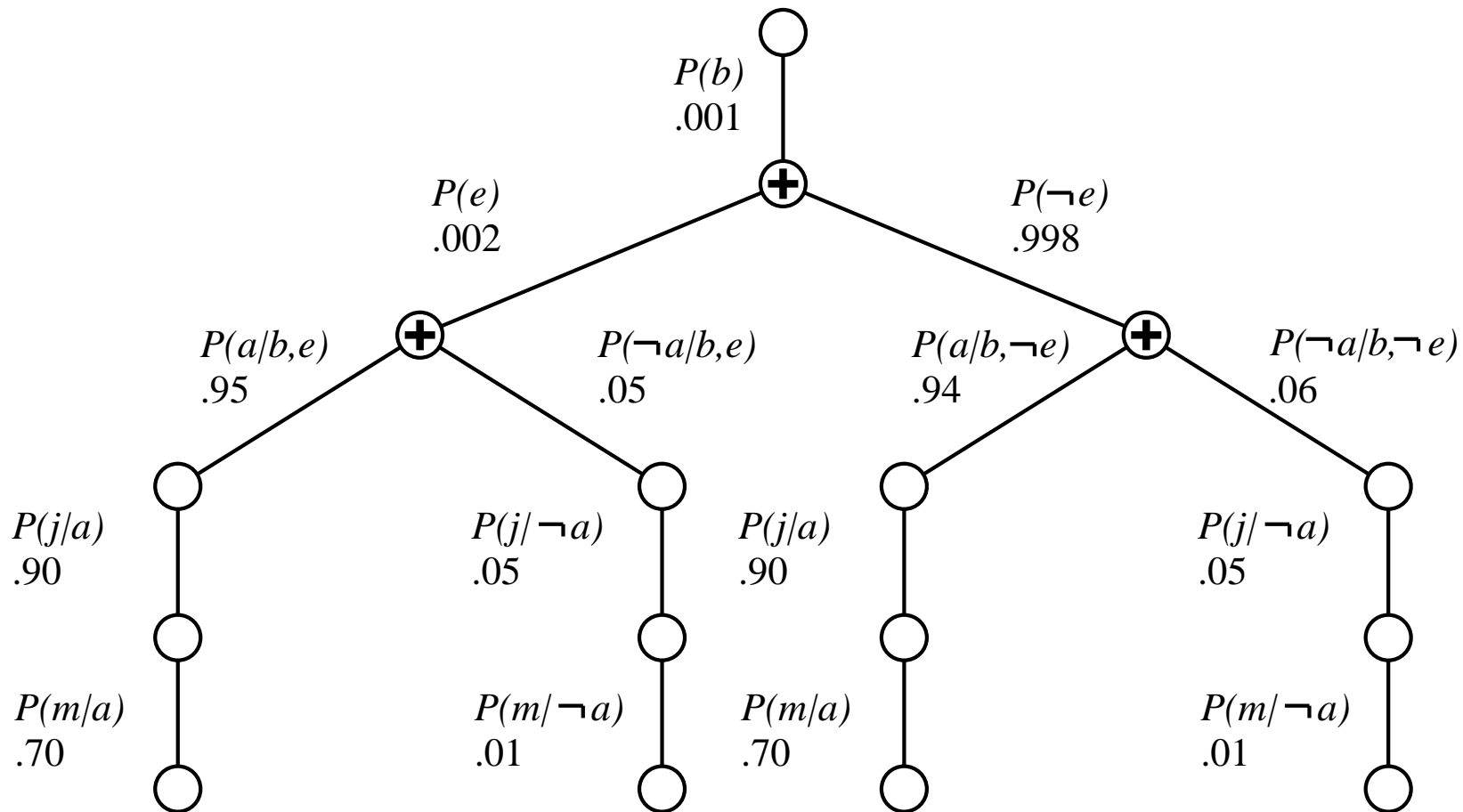


Rewrite full joint entries using product of CPT entries:

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

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

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

$$\begin{aligned}\mathbf{P}(B|j, m) &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) P(j|a) P(m|a) \\ &= \alpha \mathbf{f}_1(B) \sum_e \mathbf{f}_2(E) \sum_a \mathbf{f}_3(A, B, E) \mathbf{f}_4(A) \mathbf{f}_5(A)\end{aligned}$$

(where $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_4, \mathbf{f}_5$, are 2-element vectors, and \mathbf{f}_3 is a $2 \times 2 \times 2$ matrix)

Sum out A to get the 2×2 matrix \mathbf{f}_6 , and then E to get the 2-vector \mathbf{f}_7 :

$$\begin{aligned}\mathbf{f}_6(B, E) &= \sum_a \mathbf{f}_3(A, B, E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A) \\ &= \mathbf{f}_3(a, B, E) \times \mathbf{f}_4(a) \times \mathbf{f}_5(a) + \mathbf{f}_3(\neg a, B, E) \times \mathbf{f}_4(\neg a) \times \mathbf{f}_5(\neg a) \\ \mathbf{f}_7(B) &= \sum_e \mathbf{f}_2(E) \times \mathbf{f}_6(B, E) = \mathbf{f}_2(e) \times \mathbf{f}_6(B, e) + \mathbf{f}_2(\neg e) \times \mathbf{f}_6(B, \neg e)\end{aligned}$$

Finally, we get this:

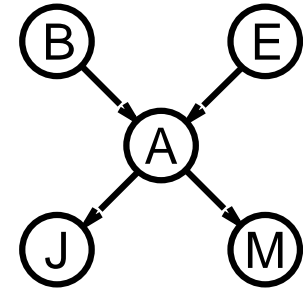
$$\mathbf{P}(B|j, m) = \alpha \mathbf{f}_1(B) \times \mathbf{f}_7(B)$$

Irrelevant variables

Consider the query $P(\text{JohnCalls} | \text{Burglary} = \text{true})$

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

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



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

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

Summary

Bayes nets provide a natural representation for (causally induced) conditional independence

Topology + CPTs = compact representation of joint distribution

Generally easy for (non)experts to construct

Inference can be computed exactly:

- variable elimination avoids recomputations
- irrelevant variables can be removed, which reduces complexity