Lecture 12: Bayesian Networks

Artificial Intelligence

CS-6364

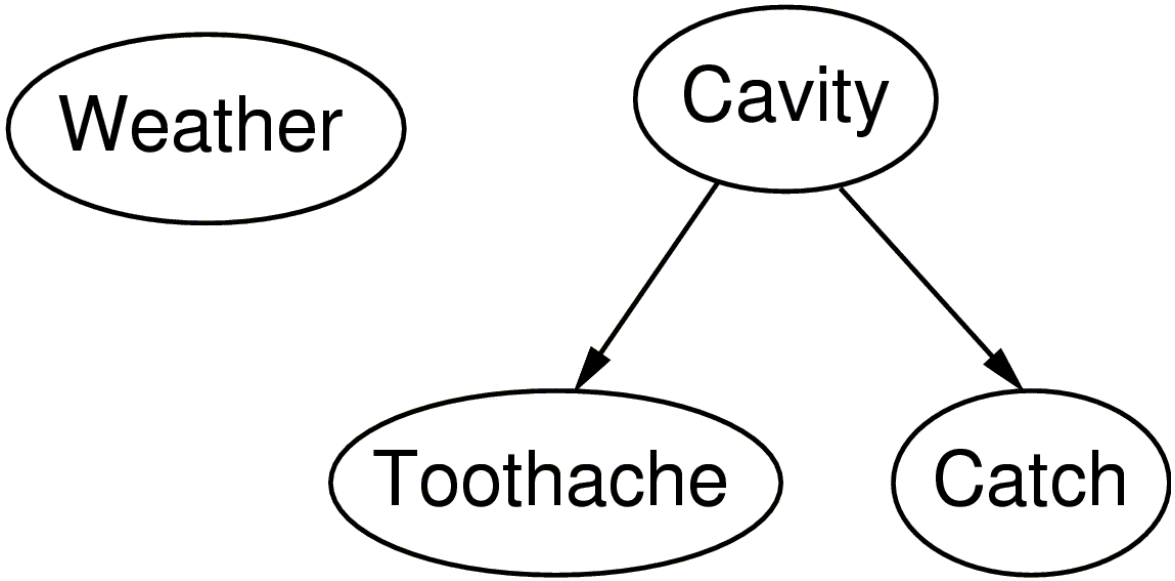
Bayesian Networks

Directed graphs in which each node is annotated with quantitative probability information. The full specification is:

1. *A set of random variables makes up the nodes of the network. Random variables may be discreet or continuous.*
2. *A set of directed links or arrows connects pairs of nodes. If there is an arrow from node X to node Y, X is said to be a parent to Y; The graph has no directed cycles (it is a directed acyclic graph, or DAG)*
3. *Each node X_i has a conditional probability distribution: $P(X_i \mid Parents(X_i))$ that quantify the effects of the parents on the node/*

Example 1

- Dental world: *Variables*: Toothache, Cavity, Catch, Weather

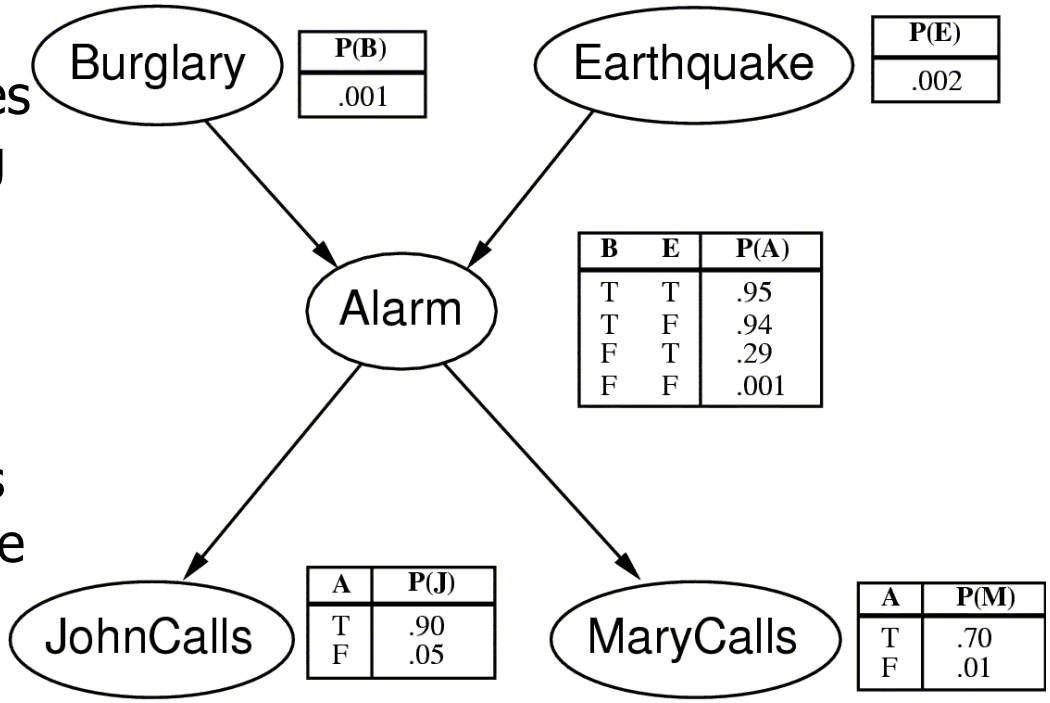


Intuitive idea: an arrow between *X* and *Y* means that *X* has direct influence on *Y*.
In this example, the Cavity has an influence on the Toothache and on the Catch! But there is no direct influence between Toothache and Catch.
The random variable Weather is independent of the other random variables.

Example 2

New burglar alarm is installed at home. It is fairly reliable at detecting a burglary, but it also responds on occasion to minor earthquakes. There are 2 neighbors: John and Mary, who have promised to call you at work when they hear the alarm.

John always calls when he hears the alarm, but sometimes confuses the telephone ringing with the alarm calls and calls then, too. Mary likes rather loud music and sometimes misses the alarm together. Given the evidence of who has or has not called, we would like to estimate the probability of burglary.



Conditional Distributions

Data Structures:

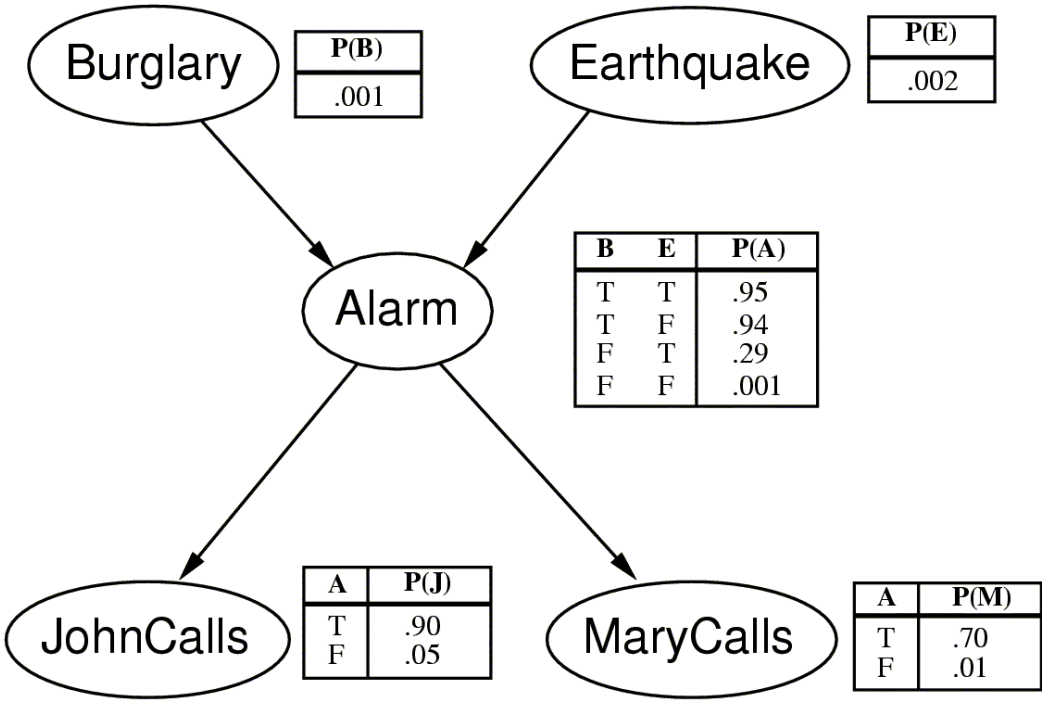
Conditional Probability Table (CPT)

Each row in a CPT contains the conditional probability of each node value for a conditioning case.

A **conditioning case** is a combination of values for the present nodes

(*A miniature atomic event*)

Property: *each row must sum to 1*. *That is particularly important for discrete, non-binary variables*. A node without parent has a CPT with only one row: the prior probability of each possible value of the variable.



Semantics of Bayesian Network

Two ways to understanding the semantics of a Bayesian network:

1. See the network as a representation of the joint probability distribution
2. View the Bayesian network as an encoding of a collection of conditional independence statements

They are equivalent. The first view helps in understanding how to construct networks. The second view helps in designing inference procedures.

Representing the full joint distribution

- A Bayesian Network provides a description of a domain. Every entry in the full joint probability distribution ("joint") can be calculated from the information in the Bayesian network.
- *A generic entry in the joint is the probability of a conjunction of particular assignments to each variable, such as:*

$$P(X_1 = x_1 \wedge \ldots \wedge X_n = x_n)$$ abbreviated as $$P(x_1,\ldots,x_n)$$

- The value for the entry is given by

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

where parents(X_i denotes the specific values of the variables in Parents(X_i)

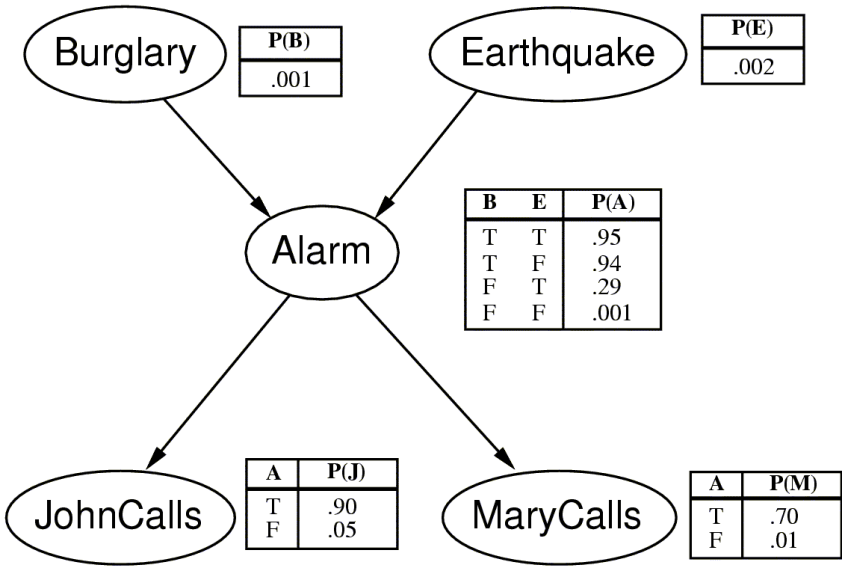
Each configuration $$(X_1 = x_1 \wedge \ldots \wedge X_n = x_n)$$ *represents a possible world*

Representing the Joint

- Each entry in the joint is represented by the product of the appropriate elements of the CPT in the Bayesian Network → the CPTs provide a decomposed representation of the joint

Example: *Compute the probability that the alarm has sounded, but neither a burglary nor an earthquake has occurred, and both John and Mary call*

Denote: b – burglary; e – earthquake; a – alarm has sounded,
j – John; m – Mary

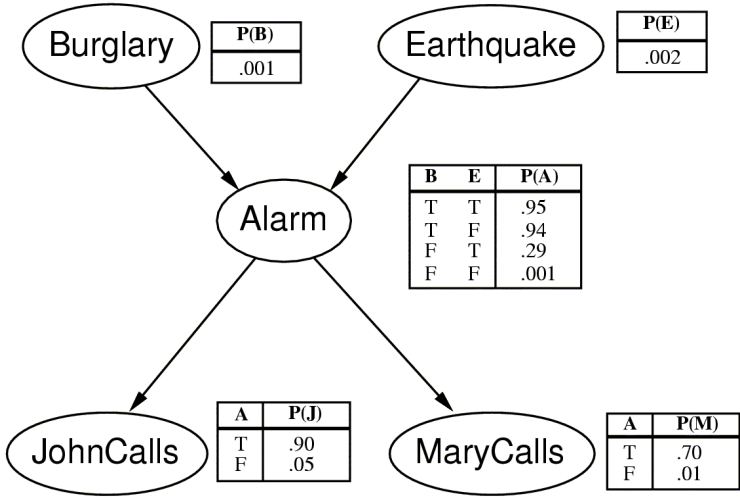


$$P(j \wedge m \wedge a \wedge \neg b \wedge \neg e) =$$
$$= P(j|a)P(m|a)P(a|\neg b \wedge \neg e)P(\neg b)P(\neg e)$$
$$= 0.90 \times 0.70 \times 0.001 \times 0.89 \times 0.998$$
$$= 0.00062$$

The Joint for Example 2 (Part 1)

- The Joint Table for variables *b, e, a, j, m*

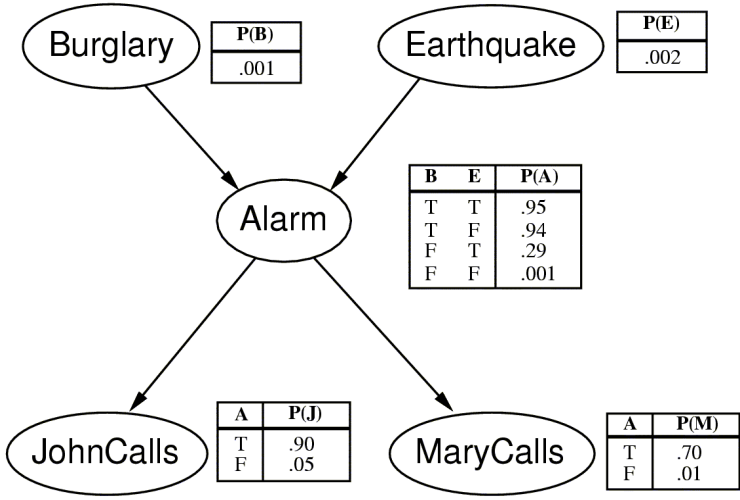
b	e	a	j	m	Probability
0	0	0	0	0	(1-0.001)×(1-0.002) ×(1-0.001) ×(1-0.05) ×(1-0.01)
0	0	0	0	1	(1-0.001) ×(1-0.002) ×(1-0.001) ×(1-0.05) × 0.01
0	0	0	1	0	(1-0.001)×(1-0.002) ×(1-0.001) ×0.05 ×(1-0.01)
0	0	0	1	1	(1-0.001)×(1-0.002) ×(1-0.001) ×0.05 ×0.01
0	0	1	0	0	(1-0.001)×(1-0.002) ×0.001 ×(1-0.90) ×(1-0.70)
0	0	1	0	1	(1-0.001)×(1-0.002) ×0.001×(1-0.90) ×0.70
0	0	1	1	0	(1-0.001)×(1-0.002) ×0.001×0.90 × (1-0.70)
0	0	1	1	1	(1-0.001)×(1-0.002) ×0.001×0.90×0.70
0	1	0	0	0	(1-0.001)×0.002×(1-0.29) ×(1-0.05) ×(1-0.01)
0	1	0	0	1	(1-0.001)×0.002×(1-0.29) ×(1-0.05) ×0.01
0	1	0	1	0	(1-0.001)×0.002×(1-0.29) ×0.05×(1-0.01)
0	1	0	1	1	(1-0.001)×0.002×(1-0.29) ×0.05×0.01
0	1	1	0	0	(1-0.001)×0.002×0.29×(1-0.90) ×(1-0.70)
0	1	1	0	1	(1-0.001)×0.002 ×0.29×(1-0.90) ×0.70
0	1	1	1	0	(1-0.001)×0.002×0.29×0.90×(1-0.70)
0	1	1	1	1	(1-0.001)×0.002×0.29×0.90×0.70



The Joint for Example 2 (Part 2)

- The Joint Table for variables *b, e, a, j, m*

b	e	a	j	m	Prob
1	0	0	0	0	0.001×(1-0.002) ×(1-0.94) ×(1-0.05) ×(1-0.01)
1	0	0	0	1	0.001 ×(1-0.002) ×(1-0.94) ×(1-0.05) × 0.01
1	0	0	1	0	0.001×(1-0.002) ×(1-0.94) ×0.05 ×(1-0.01)
1	0	0	1	1	0.001×(1-0.002) ×(1-0.94) ×0.05 ×0.01
1	0	1	0	0	0.001×(1-0.002) ×0.94 ×(1-0.90) ×(1-0.70)
1	0	1	0	1	0.001×(1-0.002) ×0.94×(1-0.90) ×0.70
1	0	1	1	0	0.001×(1-0.002) ×0.94×0.90 ×(1-0.70)
1	0	1	1	1	0.001×(1-0.002) ×0.94×0.90×0.70
1	1	0	0	0	0.001×0.002×(1-0.95) ×(1-0.05) ×(1-0.01)
1	1	0	0	1	0.001×0.002×(1-0.95) ×(1-0.05) ×0.01
1	1	0	1	0	0.001×0.002×(1-0.95) ×0.05×(1-0.01)
1	1	0	1	1	0.001×0.002×(1-0.95) ×0.05×0.01
1	1	1	0	0	0.001×0.002×0.95×(1-0.90) ×(1-0.70)
1	1	1	0	1	0.001×0.002 ×0.95×(1-0.90) ×0.70
1	1	1	1	0	0.001×0.002×0.95×0.90×(1-0.70)
1	1	1	1	1	0.001×0.002×0.95×0.90×0.70



The Joint for Example 2 (Part 3)

- We can compute any probability
  - E.g. $P(\neg b,e, \neg a,j, \neg m)=Prob(0,1,0,1,0)=$(1-0.001)×0.002×(1-0.29) ×0.05×(1-0.01)
  - $P(a|b, \neg e) = P(a,b, \neg e) / P(b, \neg e)=$(0.001×(1-0.002) ×0.94 ×(1-0.05) ×(1-0.01) + 0.001×(1-0.002) ×0.94×(1-0.05) ×0.01 + 0.001×(1-0.002) ×0.94×(1-0.05) ×0.01+ 0.001×(1-0.002) ×0.94×0.05×0.01)/ ….

b	e	a	j	m	Prob
0	0	0	0	0	(1-0.001)×(1-0.002) ×(1-0.001) ×(1-0.05) ×(1-0.01)
0	0	0	0	1	(1-0.001) ×(1-0.002) ×(1-0.001) ×(1-0.05) × 0.01
0	0	0	1	0	(1-0.001)×(1-0.002) ×(1-0.001) ×0.05 ×(1-0.01)
0	0	0	1	1	(1-0.001)×(1-0.002) ×(1-0.001) ×0.05 ×0.01
0	0	1	0	0	(1-0.001)×(1-0.002) ×0.001 ×(1-0.90) ×(1-0.70)
0	0	1	0	1	(1-0.001)×(1-0.002) ×0.001×(1-0.90) ×0.70
0	0	1	1	0	(1-0.001)×(1-0.002) ×0.001×0.90 × (1-0.70)
0	0	1	1	1	(1-0.001)×(1-0.002) ×0.001×0.90×0.70
0	1	0	0	0	(1-0.001)×0.002×(1-0.29) ×(1-0.05) ×(1-0.01)
0	1	0	0	1	(1-0.001)×0.002×(1-0.29) ×(1-0.05) ×0.01
0	1	0	1	0	(1-0.001)×0.002×(1-0.29) ×0.05×(1-0.01)
0	1	0	1	1	(1-0.001)×0.002×(1-0.29) ×0.05×0.01
0	1	1	0	0	(1-0.001)×0.002×0.29×(1-0.90) ×(1-0.70)
0	1	1	0	1	(1-0.001)×0.002 ×0.29×(1-0.90) ×0.70
0	1	1	1	0	(1-0.001)×0.002×0.29×0.90×(1-0.70)
0	1	1	1	1	(1-0.001)×0.002×0.29×0.90×0.70

b	e	a	j	m	Prob
1	0	0	0	0	0.001×(1-0.002) ×(1-0.94) ×(1-0.05) ×(1-0.01)
1	0	0	0	1	0.001 ×(1-0.002) ×(1-0.94) ×(1-0.05) × 0.01
1	0	0	1	0	0.001×(1-0.002) ×(1-0.94) ×0.05 ×(1-0.01)
1	0	0	1	1	0.001×(1-0.002) ×(1-0.94) ×0.05 ×0.01
1	0	1	0	0	0.001×(1-0.002) ×0.94 ×(1-0.90) ×(1-0.70)
1	0	1	0	1	0.001×(1-0.002) ×0.94×(1-0.90) ×0.70
1	0	1	1	0	0.001×(1-0.002) ×0.94×0.90 × (1-0.70)
1	0	1	1	1	0.001×(1-0.002) ×0.94×0.90×0.70
1	1	0	0	0	0.001×0.002×(1-0.95) ×(1-0.05) ×(1-0.01)
1	1	0	0	1	0.001×0.002×(1-0.95) ×(1-0.05) ×0.01
1	1	0	1	0	0.001×0.002×(1-0.95) ×0.05×(1-0.01)
1	1	0	1	1	0.001×0.002×(1-0.95) ×0.05×0.01
1	1	1	0	0	0.001×0.002×0.95×(1-0.90) ×(1-0.70)
1	1	1	0	1	0.001×0.002 ×0.95×(1-0.90) ×0.70
1	1	1	1	0	0.001×0.002×0.95×0.90×(1-0.70)
1	1	1	1	1	0.001×0.002×0.95×(1-0.90)×(1-0.70)

A Method for Constructing Bayesian Networks

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

defines what a given Bayesian Network means

- *It does not explain <u>how to build</u> a Bayesian Network such that the resulting joint distribution is a good representation of a given domain*

However, it implies certain additional independence relationships that can be used to guide the knowledge engineer in constructing the topology of the network. *How???*

Steps

1. Rewrite the joint distribution in terms of a conditional probability using the product rule:

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

becomes

$$P(x_1,\ldots,x_n) = P(x_n \mid x_{n-1},\ldots,x_1)P(x_{n-1},\ldots,x_1)$$

2. Repeat the process, reducing each conjunctive probability to a conditional probability and a smaller conjunction →

$$P(x_1,\ldots,x_n) = P(x_n \mid x_{n-1},\ldots,x_1)P(x_{n-1} \mid x_{n-2},\ldots,x_1)..P(x_2 \mid x_1)P(x_1)$$

$$= \prod_{i=1}^{n} P(x_i \mid x_{i-1},\ldots,x_1)$$

(*the chain rule*)

Semantics

Because

$$P(x_1,\ldots,x_n) = P(x_n \mid x_{n-1},\ldots,x_1)P(x_{n-1} \mid x_{n-2},\ldots,x_1)\ldots P(x_2 \mid x_1)P(x_1)$$

$$= \prod_{i=1}^{n} P(x_i \mid x_{i-1},\ldots,x_1)$$

We have:

$$P(x_i \mid x_{i-1},\ldots,x_1) = P(x_i \mid Parents(x_i))$$

$$if \;\; Parents(x_i) \subseteq \{x_{i-1},\ldots,x_1\}$$

This condition is satisfied by numbering the nodes in a way that is consistent with the partial order implicit in the graph structure

Construction Rules

$$P(x_i \mid x_{i-1},\ldots,x_1) = P(x_i \mid Parents(x_i))$$

The Bayesian Network is a correct representation of a domain ONLY if each node is conditionally Independent of its predecessors in the Node ordering, given its parents!

We can satisfy this condition with this methdolology:

1. Nodes: First determine the set of variables that are required to model the domain and order them: $\{X_1,\ldots,X_n\}$ – ideal order: causes precede effects, e.g. the causes of X_i should be among $\{X_1,\ldots,X_{i-1}\}$

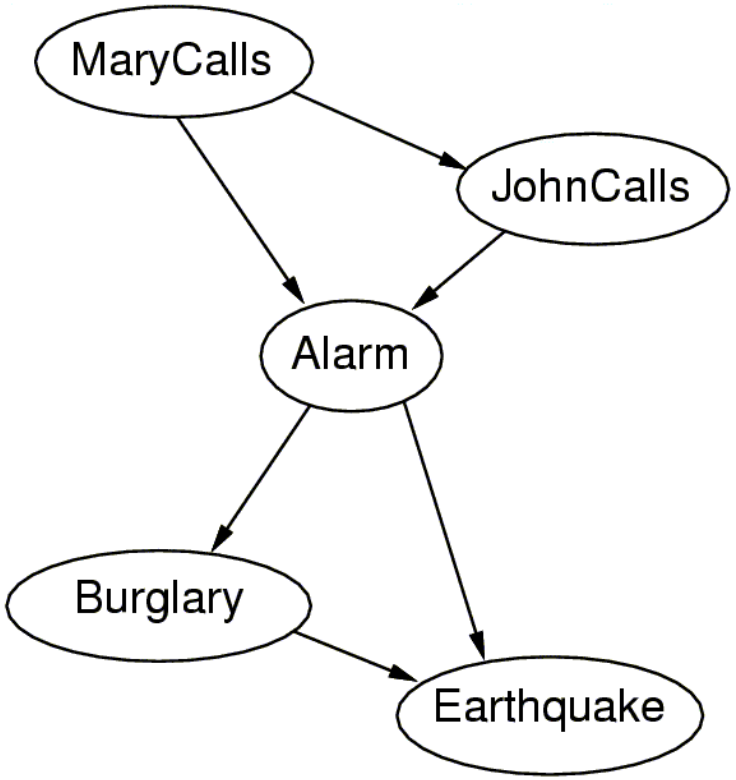
2. Links: For *i=1* to *n* do
   - Chose, from $x_1,\ldots,x_{i-1}$ a minimal set of parents for x_i such that:
     $$P(x_i \mid x_{i-1},\ldots,x_1) = P(x_i \mid Parents(x_i))$$
   - For each parent insert a link from the parent to x_i
   - Write down the conditional probability table (CPT): $P(x_i|Parents(x_i))$

Properties of Bayes Nets

- Because each node is connected only to earlier nodes, the net is guaranteed to be acyclic!
- Bayesian Networks contain no redundant probability values. If there is no redundancy, there is no change of inconsistency.⇒ there is no change for the knowledge engineer to create a Bayesian Network that violates the rules of probability!
- The compactness representation of Bayesian Networks

Compactness and Node Ordering

If we add the nodes in the order: MaryCalls, JohnCalls, Alarm, Burglary, Earthquake we obtain a more complicated network:



(a)

The process:

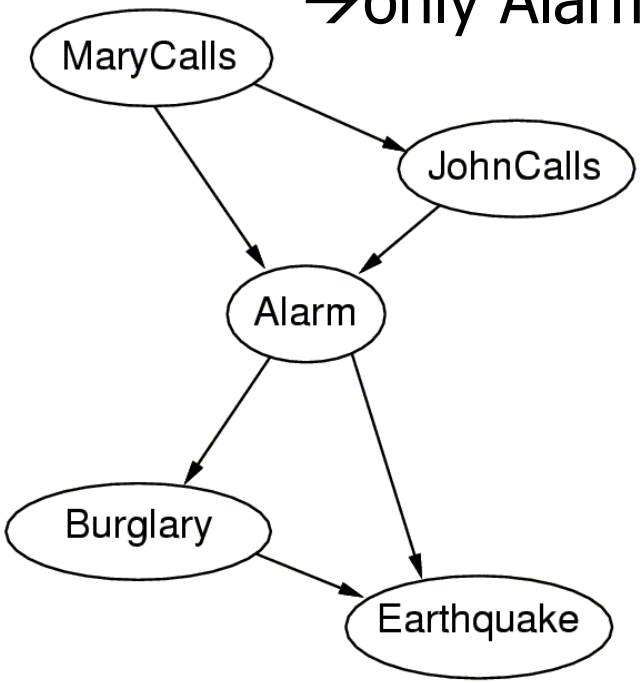
1) Add MaryCalls – no parents
2) Add JohnCalls → if MaryCalls, probably the alarm went off → it makes it more likely that JohnCalls →JohnCalls will need MaryCalls as a parent
3) Add Alarm → if both John and Mary call, it is more likely the alarm went off →both MaryCalls and JohnCalls are parents

More Problems

4) Add Burglary → if we know the alarm state, then the call from John or Mary might give us information about our phone ringing or Mary's music, but not about the burglary

$$P(Burglary \mid Alarm, JohnCalls, MaryCalls) = P(Burglary, Alarm)$$

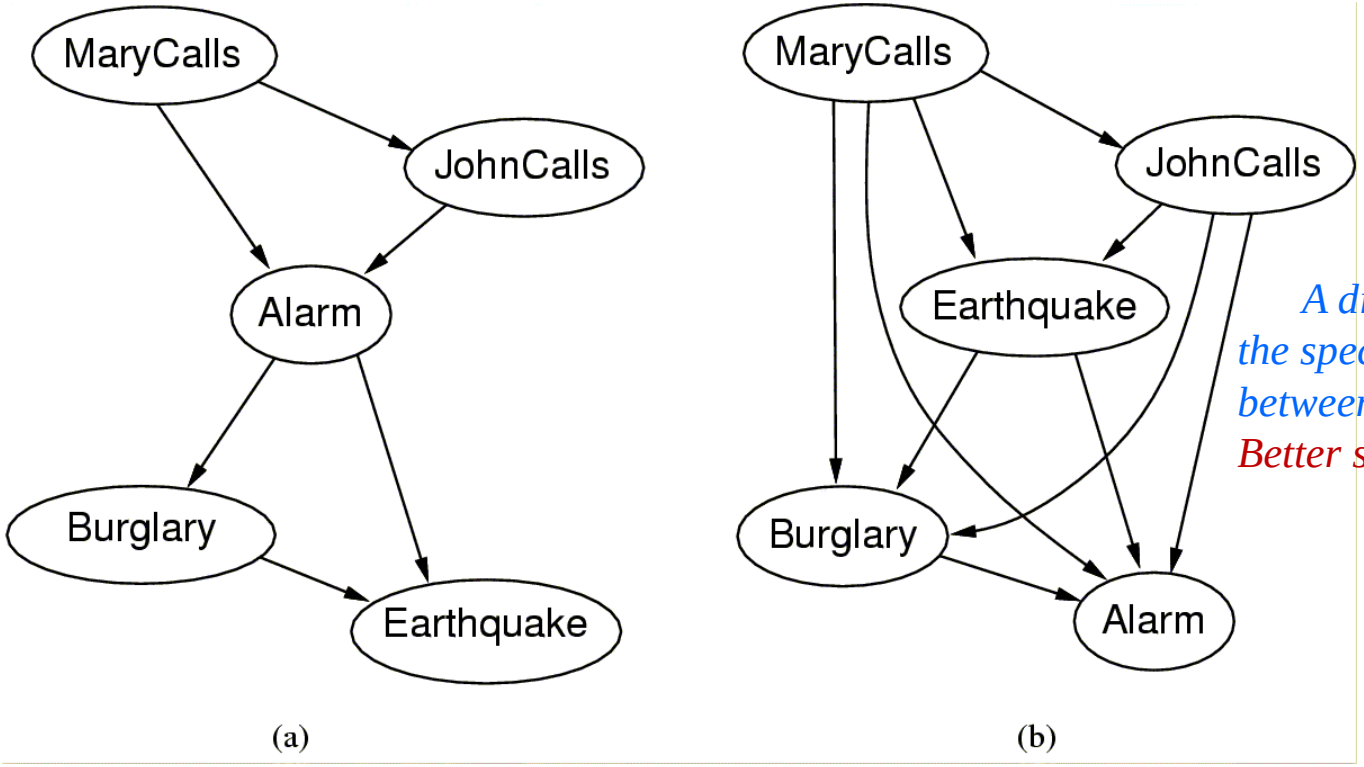
→only Alarm is a parent



(a)

5) Adding Earthquake: if the alarm is on, it is more likely that there has been an earthquake. But if we know that there has been a burglary, then that explains the alarm. Both Alarm and Burglary are parents

What is the Problem?



A diagnostic model requires the specification of dependencies between independent causes!
Better specify only a causal model!

→ We do not have only causal links!
→ There are also links from symptoms to Causes! There is no distinction between causal and diagnostic models!!!

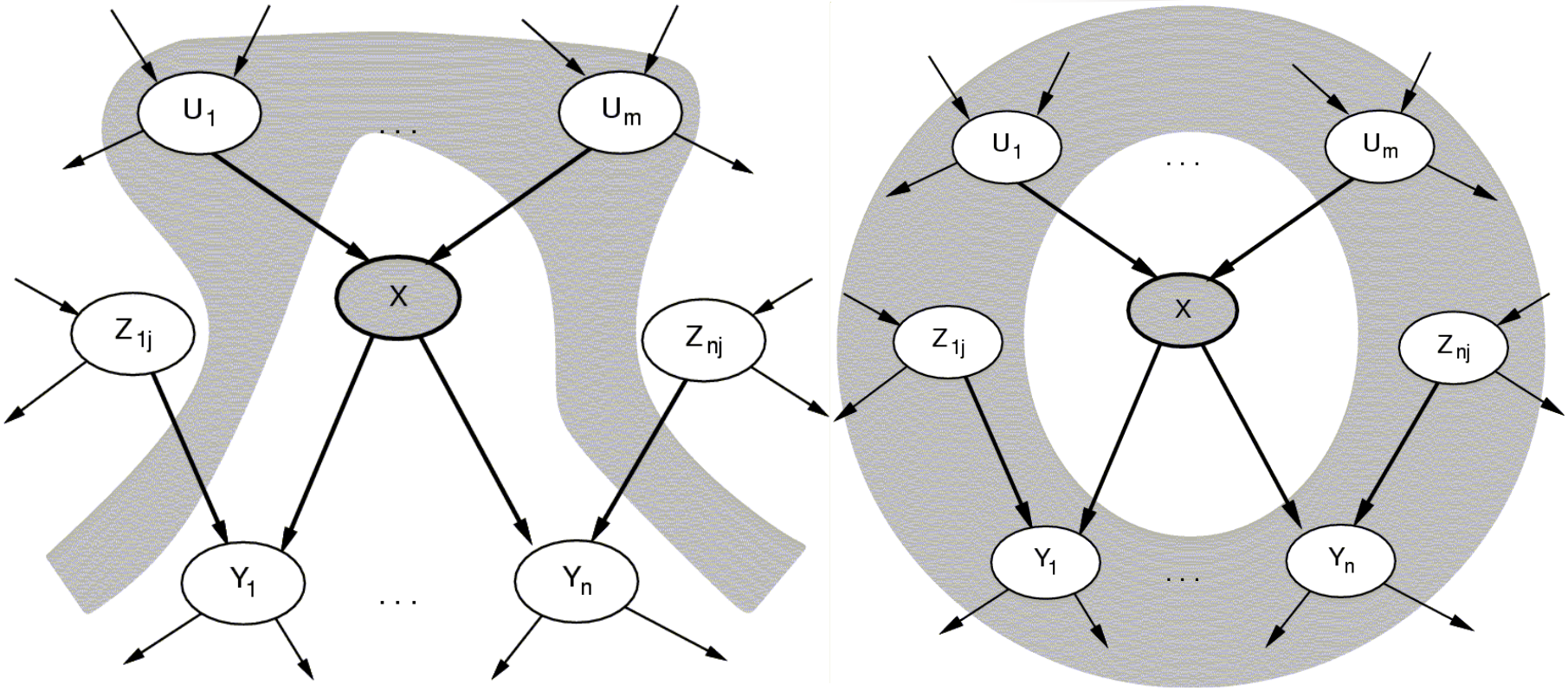
Conditional Independence in Bayesian Networks

- A node is conditionally independent of its predecessors, given its parents because:

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

- Conditional independence is also dictated by topological semantics:
  1. A node is conditionally independent of its non-descendants, given its parents. Example: JohnCalls is independent of Burglary and Earthquake, given the value of Alarm
  2. A node is conditionally independent of all other nodes in the network, given its parents, children and children's parents →called Markov blanket

Examples



(a) A node X is conditionally independent of its non-descendants (e.g., the Z_{ij}s) given its parents (the U_is shown in the gray area).

(b) A node X is conditionally independent of all other nodes in the network given its Markov blanket (the gray area)

D-separation

- The most general conditional independence question: are X nodes *conditionally independent* of another set Y, given a set Z?

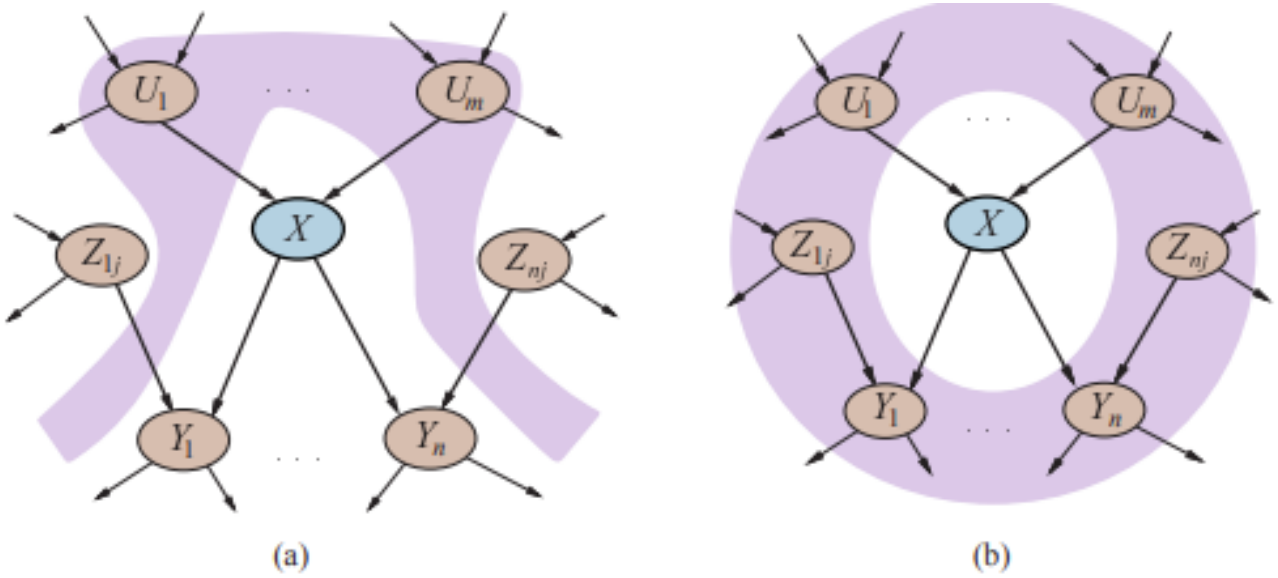


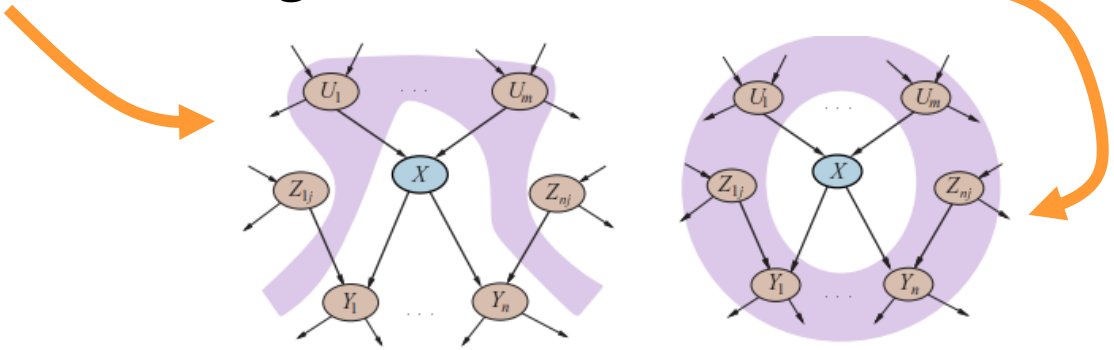
Figure 13.4 (a) A node X is conditionally independent of its non-descendants (e.g., the Z_{ij}s) given its parents (the U_is shown in the gray area). (b) A node X is conditionally independent of all other nodes in the network given its Markov blanket (the gray area).

- This can be achieved by examining the Bayes Net to see whether Z D-separates X and Y.

Ancestral and Moral Graphs

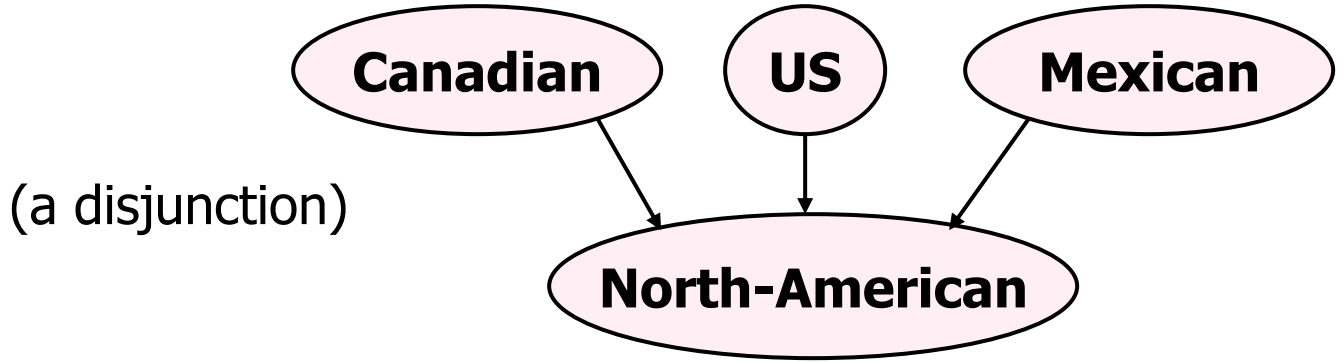
- *The process of examining d-separation is:*

1. *Consider just the **ancestral sub-graph**: X, Y, Z and their ancestors;*
2. *Add links between any unlinked pairs of nodes that share a common child: this is the **moral graph***
3. *Relace all directed links by undirected links*
4. *If Z blocks all paths between X and Y in the resulting graph, then Z d-separates X and Y. In that case,* X is *conditionally independent* of Y, given a set Z. Otherwise:



Efficient Representation of Conditional Distribution

- Even when a node has only K parents → the CPT needs $O(2^k)$ members → this is the worst-case scenario. Usually, relations between parents and child are described by *a canonical distribution* that fits some standard pattern → the CPT can be specified by naming the pattern and few parameters
- Simplest Example: **deterministic nodes.** A deterministic node has its value specified exactly by the value of its parents, with no uncertainty
  - *Example*: The relationship may be a logical one!



Context-specific Independence

Another important pattern that occurs in practice is context-specific independence. When??

- A conditional distribution exhibits CSI if a variable is conditionally independent of its parents, given certain values or others.

Example: *Damage (your car)* during a specific period of time depends on the *Ruggedness* of your car and whether or not an *Accident* occurred in that period. If *Accident* is FALSE, then *Damage (your car)* doesn't depend on the *Ruggedness* (there might be vandalism).

$\Rightarrow$We say that *Damage* is context-specific independent of *Ruggedness* given *Accident*=FALSE

How to implement CSI in Bayes Nets

- Bayes Nets often implement context-specific independence using *if-then-else* syntax for specifying conditional distributions:

$$P(Damage|Ruggedness, Accident) = if\,(Accident = FALSE)\ then\ \ d_1 \\ else\ \ d_2(Ruggedness)$$

-where d_1 and d_2 are arbitrary distributions.

Noisy-OR

- Uncertain relationship can be characterized often by noisy logical relationship! A standard example: The NOISY-OR relation, which is a generalization of the logical OR.
- In Prepositional Logic, we may say that *Fever* is True if *Cold*, *Flu* and *Malaria* are true. The Noisy-OR model allows for uncertainty about the ability of each parent to cause the child to be True.
- The causal relation between parent and child may be inhibited: e.g. a patient may have a *Cold,* but not exhibit *Fever!*

The model makes 2 assumptions:

1. *All possible causes are listed (if some are missing, we can add a leak-node that covers "miscellaneous causes")*
2. *It assumes that the inhibition of a parent is independent of the inhibition of other parents.*

Noisy-OR example

- Given the assumption that *Fever* is False if and only if all its True parents are inhibited, and the probability of this is the product of the ***inhibition probabilities*** of each parent, we may have:

inhibition probs

$$\begin{cases} q_{Cold} = P(\neg Fever | Cold, \neg Flu, \neg Malaria) = 0.6 \\ q_{Flu} = P(\neg Fever | \neg Cold, Flu, \neg Malaria) = 0.2 \\ q_{Malaria} = P(\neg Fever | \neg Cold, \neg Flu, Malaria) = 0.1 \end{cases}$$

From this information and the Noisy-OR assumption, the CPT can be built.

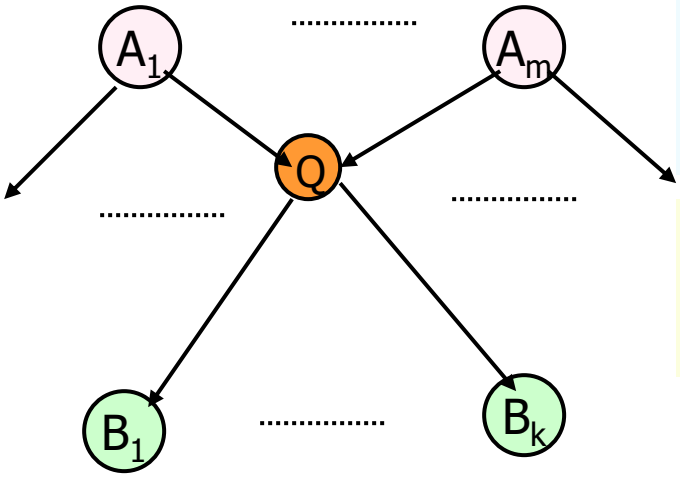
The general Rule is that: $P(x_i | Parents(X_i)) = 1 - \prod_{\{j:X_j=True\}} q_j$

Cold	*Flu*	*Malaria*	*P(Fever)*	*P(¬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=0.2×0.1
T	F	F	0.4	**0.6**
T	F	T	0.94	0.06=0.6×0.1
T	T	F	0.88	0.12=0.6×0.2
T	T	T	0.988	0.012=0.6×0.2×0.1

Patterns of inference in Bayesian Nets

- *We are interested in inference that can be computed directly from the Bayesian network, without the explicit computation of an exponentially long table.*

Consider a special network topology: **the polytree**
A polytree is a directed acyclic graph for which there is just one path along the undirected graph arcs between any two nodes.
Any node connected to a node **Q** will not be connected to any other node except through node Q



The evidence might include variables associated with components above Q (E^+) or below Q (E^-). Then:
$Prob(A_1,\ldots,A_m|E^+) = Prob(A_1|E^+) \times \ldots Prob(A_m|E^+)$
$Prob(B_1,\ldots,B_k|E^-,Q) = Prob(B_1|E^-,Q) \times \ldots Prob(B_k|E^-,Q)$

Inference is based on Bayes rule for all *n* possible worlds of variable X:

$Prob(X) = Prob(X|w_1) \times Prob(w_1) + \ldots + Prob(X|w_n) \times Prob(w_n)$

There are three cases:

1/ No evidence

2/ Evidence above the query

3/ Evidence below the query

Case 1: No evidence

- A query variable Q is given, and there is no evidence. Goal: compute P(Q)

Method:

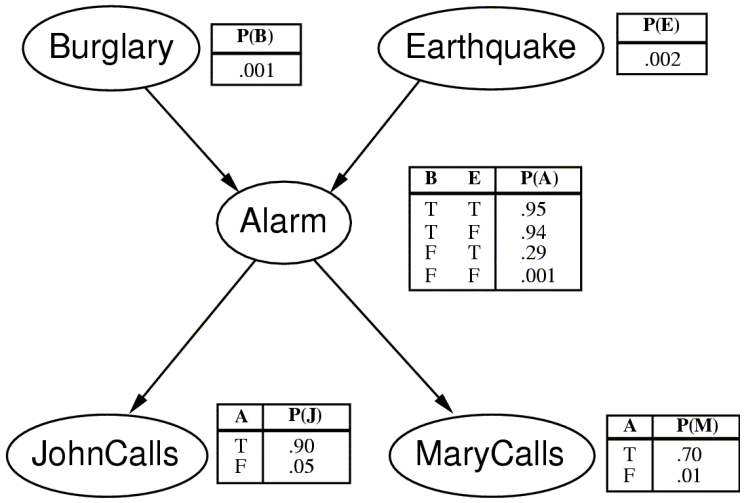
1/ Compute the probability of variables A_1, …, A_m above Q

2/ Compute the probability of all possible worlds w_i based on A_1, …, A_m using:

$$Prob(A_1,\ldots,A_m|E^+)= Prob(A_1|\ E^+) \times \ldots\ Prob(A_m|\ E^+)$$

3/ Compute the probability of Q using:

$$Prob(Q)= Prob(Q|w_1) \times Prob(w_1) + \ldots + Prob(Q|w_n) \times Prob(w_n)$$



Example: Q=B then P(B)=0.001 Q=E then P(E) = 0.002

At node A:

B	E	P(A)	P(w)
T	T	0.95	0.001× 0.002=0.000002
T	F	0.94	0.001 × 0.998 = 0.000998
F	T	0.29	0.999 × 0.002 = 0.001998
F	F	0.001	0.999 × 0.998 = 0.997002

P(A) = 0.000002 × 0.95 + 0.000998 × 0.94 +
+ 0.001998 × 0.29 + 0.997002 × 0.001=
0.0000019+0.00093812+0.00057942+0.000997002+
= 0.002509522 then P(¬A) = 0.997489048

Case 1: No evidence (cont)_

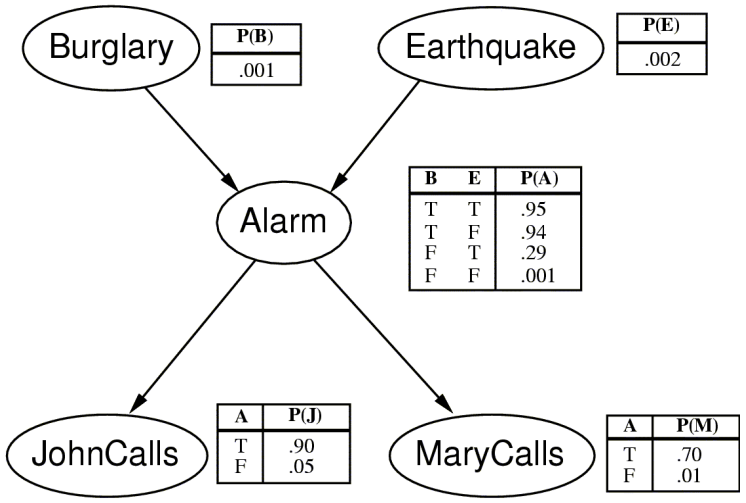
Continue:

Example: P(B)=0.001 P(E) = 0.002

At node A:

B	E	P(A)	P(w)
T	T	0.95	0.001× 0.002=0.000002
T	F	0.94	0.001 × 0.998 = 0.000998
F	T	0.29	0.999 × 0.002 = 0.001998
F	F	0.001	0.999 × 0.998 = 0.997002

P(A) = 0.002509522 then P(¬A) = 0.99749048



At node J:

A	P(J)	P(w)
T	0.9	0.002509522
F	0.05	0.99749048

P(J) = 0.9 × 0.002509522 + 0.05 × 0.99749048
= 0.052133093
P(¬J) = 0.94786091

At node M:

A	P(M)	P(w)
T	0.7	0.002509522
F	0.01	0.99749048

P(M) = 0.7 × 0.002509522 + 0.01 × 0.99749048
= 0.01173157
P(¬M) = 0.98826843

Case 2: Evidence above the query

A query variable Q is given, and

1. Q is part of evidence $P(Q|E^+) = 0$ or 1 (according to evidence)
2. Q is at the top, then P(Q) is given by the CPT

Method:

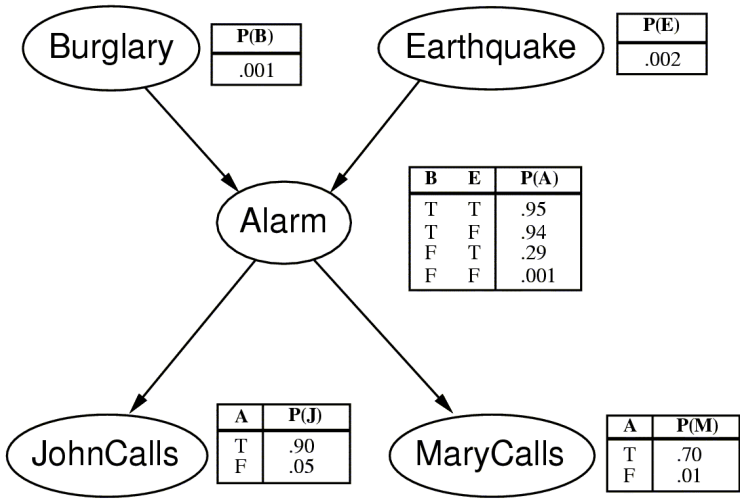
1/ Compute the probability of variables $A_1, \ldots, A_m$ above Q

2/ Compute the probability of all possible worlds w_i based on $A_1, \ldots, A_m$ using:

$$Prob(A_1,\ldots,A_m|E^+) = Prob(A_1|E^+) \times \ldots \ldots Prob(A_m|E^+)$$

3/ Compute the probability of Q using:

$$Prob(Q) = Prob(Q|w_1) \times Prob(w_1) + \ldots + Prob(Q|w_n) \times Prob(w_n)$$



Example1: P(A|B)

At node A:

B	E	P(A)	P(w)
T	T	0.95	1× 0.002=0.002
T	F	0.94	1 × 0.998 = 0.998
F	T	0.29	0 × 0.002 = 0
F	F	0.001	0× 0.998 = 0

$P(A) = 0.002 \times 0.95 + 0.0998 \times 0.94 = 0.00019+0.093812$

$= 0.94002$ then $P(\neg A) = 0.05998$

Example 2

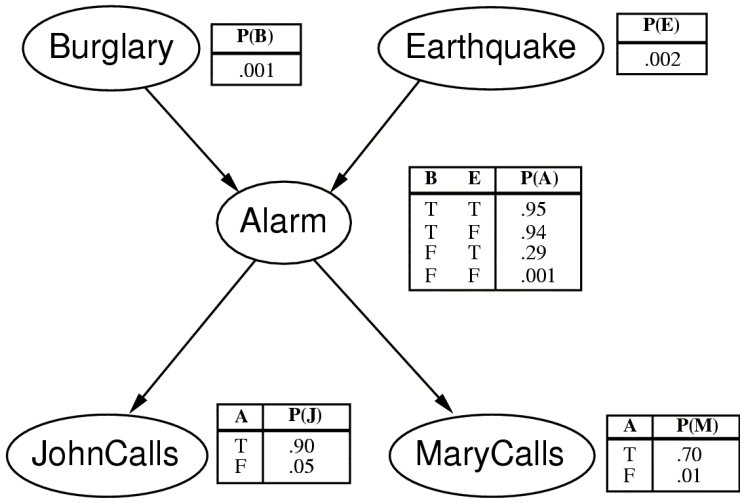
Example 2: P(M|B)

At node A:

B	E	P(A)	P(w)
T	T	0.95	1× 0.002=0.002
T	F	0.94	1 × 0.998 = 0.998
F	T	0.29	0 × 0.002 = 0
F	F	0.001	0× 0.998 = 0

P(A) = 0.002 × 0.95 + 0.998 × 0.94 =

0.0019+0.93812 = 0.94002 then P(¬A) = 0.05998



At node M:

A	P(M)	P(w)
T	0.7	0.94002
F	0.01	0.05998

P(M) = 0.7 × 0.94002 + 0.01 × 0.05998

= 0.6586

P(¬M) = 0.3414

Example 3: P(M|A,B)=0.7

Case 3: Evidence below the query

- A query variable Q is given, and the evidence E^-.

The goal: compute $P(Q|E^-)$

Method:

Compute

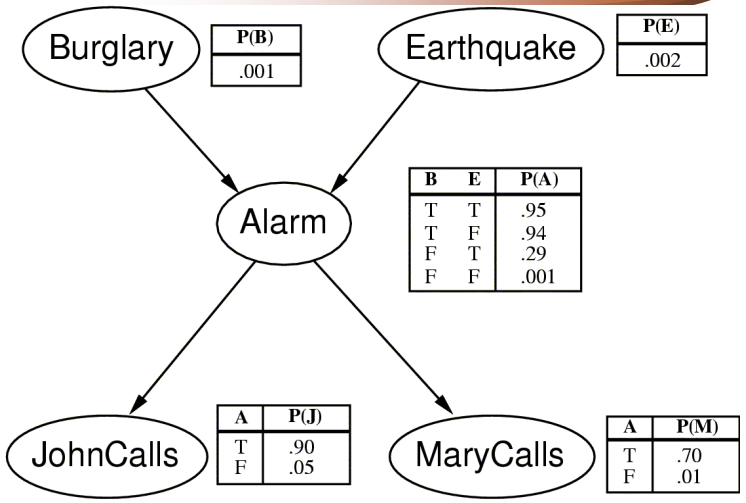
$$P(Q \mid E^-) = \frac{P(E^-|Q) \times P(Q)}{P(E^-)}$$

Thus we need to compute:

1/ $P(E^-|Q)$ – case 2

2/ $P(Q)$ – case 1

3/ a normalization constant



Example 1: P(B|JM)

$P(B|JM) = \alpha \times P(J|B) \times P(M|B) \times P(B)$

At node A:

B	E	P(A)	P(w)
T	T	0.95	1× 0.002=0.002
T	F	0.94	1 × 0.998 = 0.998
F	T	0.29	0 × 0.002 = 0
F	F	0.001	0× 0.998 = 0

$P(A|B) = 0.002 \times 0.95 + 0.998 \times 0.94 = 0.0019+0.93812$
$= 0.94002$ then $P(\neg A|B) = 0.05998$

Case 3 (Cont)

P(A|B) = 0.94002 P(¬A|B) = 0.05998

At node J:

A	P(J)	P(w)
T	0.9	0.94002
F	0.05	0.05998

P(J|B) = 0.9 × 0.94002 + 0.05 × 0.05998

≈ 0.849

P(¬J|B) = 0.151 Also P(B)= 0.001

Now P(B|JM) = α × P(J|B) × P(M|B) × P(B)=

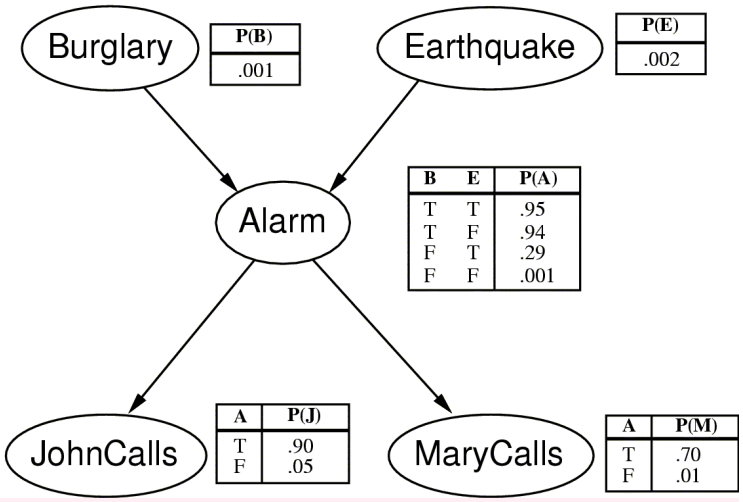
= α × 0.849 × 0.6586 × 0.001= α × 0.00059224

To compute α we need to compute also:

P(¬ B|JM) = α × P(J| ¬ B) × P(M| ¬ B) × P(¬ B)
= α × 0.0014919

α(0.00059224 + 0.0014919) = 1; α = 479.8

P(B|JM) = <0.284, 0.716>



At node M:

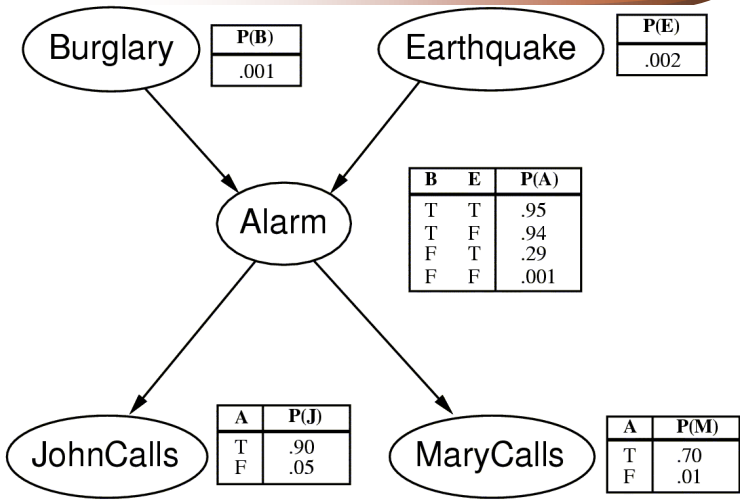
A	P(M)	P(w)
T	0.7	0.94002
F	0.01	0.05998

P(M|B) = 0.7 × 0.94002 + 0.01 × 0.05998

= 0.6586

P(¬M) = 0.3414

Case 3: (cont -2)



- How did we compute:

$P(\neg B|JM) = \alpha \times P(J|\neg B) \times$

$\times P(M|\neg B) \times P(\neg B) = \alpha \times 0.0014919$

At node A:

B	E	P(A)	P(w)
T	T	0.95	0× 0.002=0
T	F	0.94	0 × 0.998 = 0
F	T	0.29	1 × 0.002 = 0.002
F	F	0.001	1× 0.998 = 0.998

$P(A|\neg B) = 0.29 \times 0.002 + 0.998 \times 0.001 =$
$0.00058+0.000998$
$= 0.001478$ then $P(\neg A|\neg B) = 0.9998522$

Exact inference

*Basic task: compute the **posterior probability** distribution for a set of query variables, given some observed event (i.e. an assignment of values to a set of evidence variables)*

X → *query variables,*

E → *set of evidence variables* $E_1, E_2, \ldots, E_m$

and

e → *a particular observed event*

Y → *non-evidence variables* → $Y_1, Y_2, \ldots, Y_l$

(also called hidden variables)

The complete set of variables $\mathbf{X} = \{X\} \cup E \cup Y$

Typical query: P(X|e) ← the posterior distribution

Example

Burglary network: event → { JohnCalls = true MaryCalls = true}

Ask if a burglary has occurred:

P(Burglary| JohnCalls=true,MaryCalls=true)=<0.284,0.716>

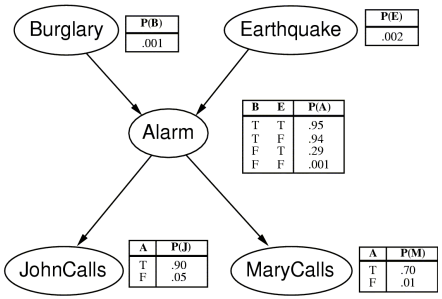
?

Inference by enumeration

$$P(X \mid e) = \alpha P(X, e) = \alpha \sum_{y} P(X, e, y)$$

For the query P(Burglary|JohnCalls=true,MaryCalls=true)

the hidden variables are Earthquake and Alarm



$$P(B \mid j, m) = \alpha P(B, j, e) = \alpha \sum_{e} \sum_{a} P(B, e, a, j, m)$$

The semantics of Bayesian networks gives us:

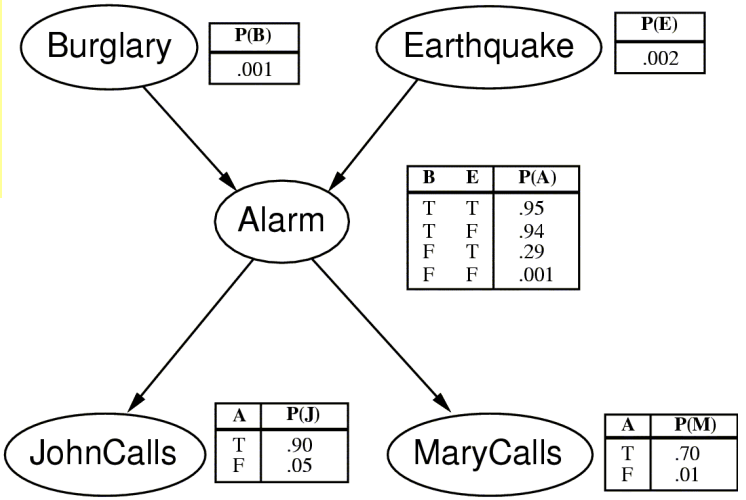
$$P(b \mid j, m) = \alpha \sum_{e} \sum_{a} P(b) P(e) P(a \mid b, e) P(j \mid a) P(m \mid a)$$

constant

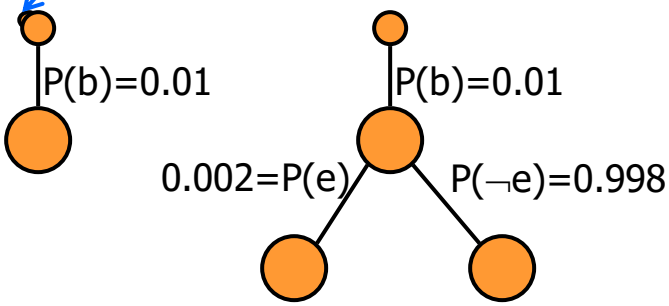
Computing the Posterior

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

How do we compute this?



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

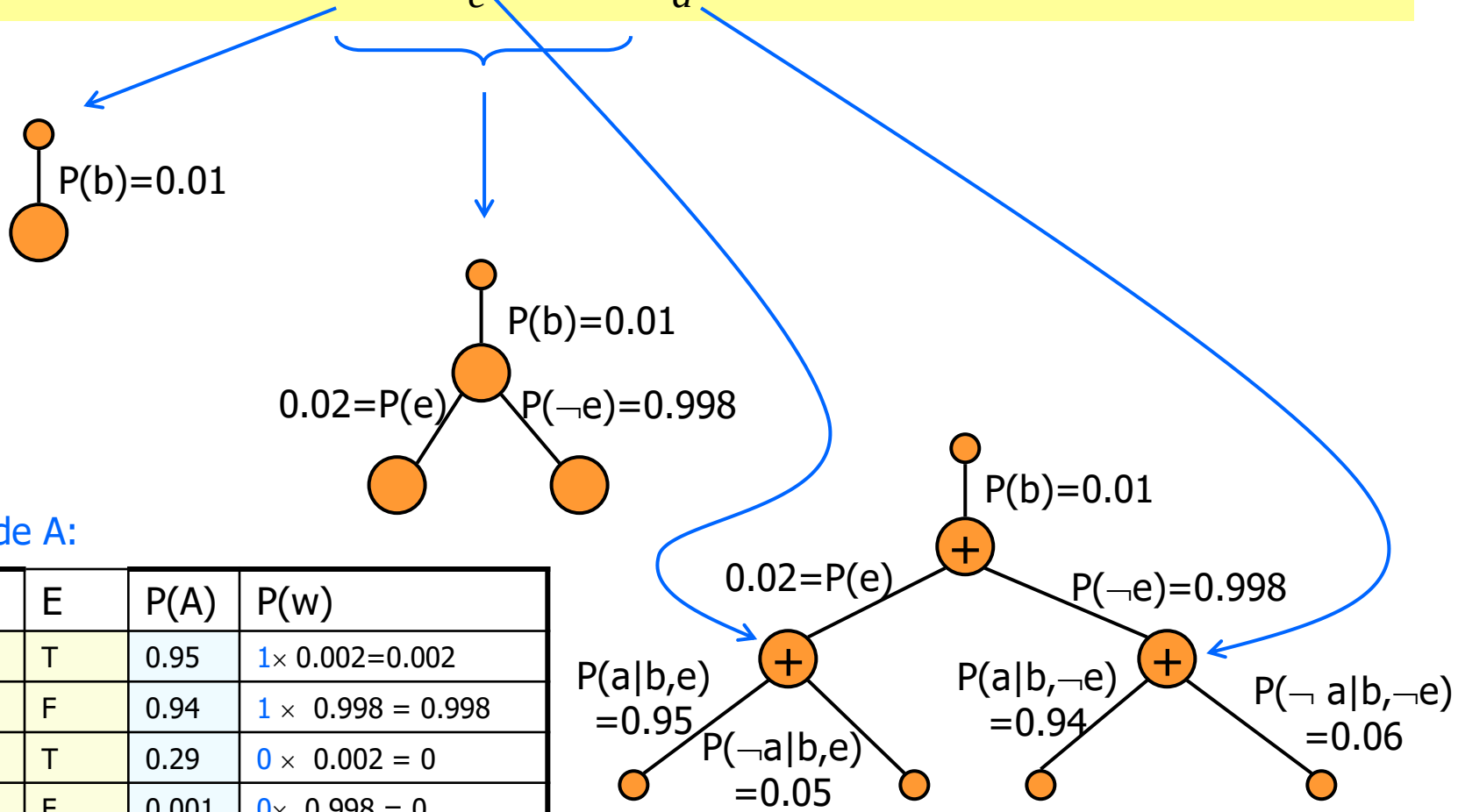


At node A:

B	E	P(A)	P(w)
T	T	0.95	1× 0.002=0.000002
T	F	0.94	1 × 0.998 = 0.000998
F	T	0.29	0 × 0.002 = 0
F	F	0.001	0× 0.998 = 0

Explanation

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



At node A:

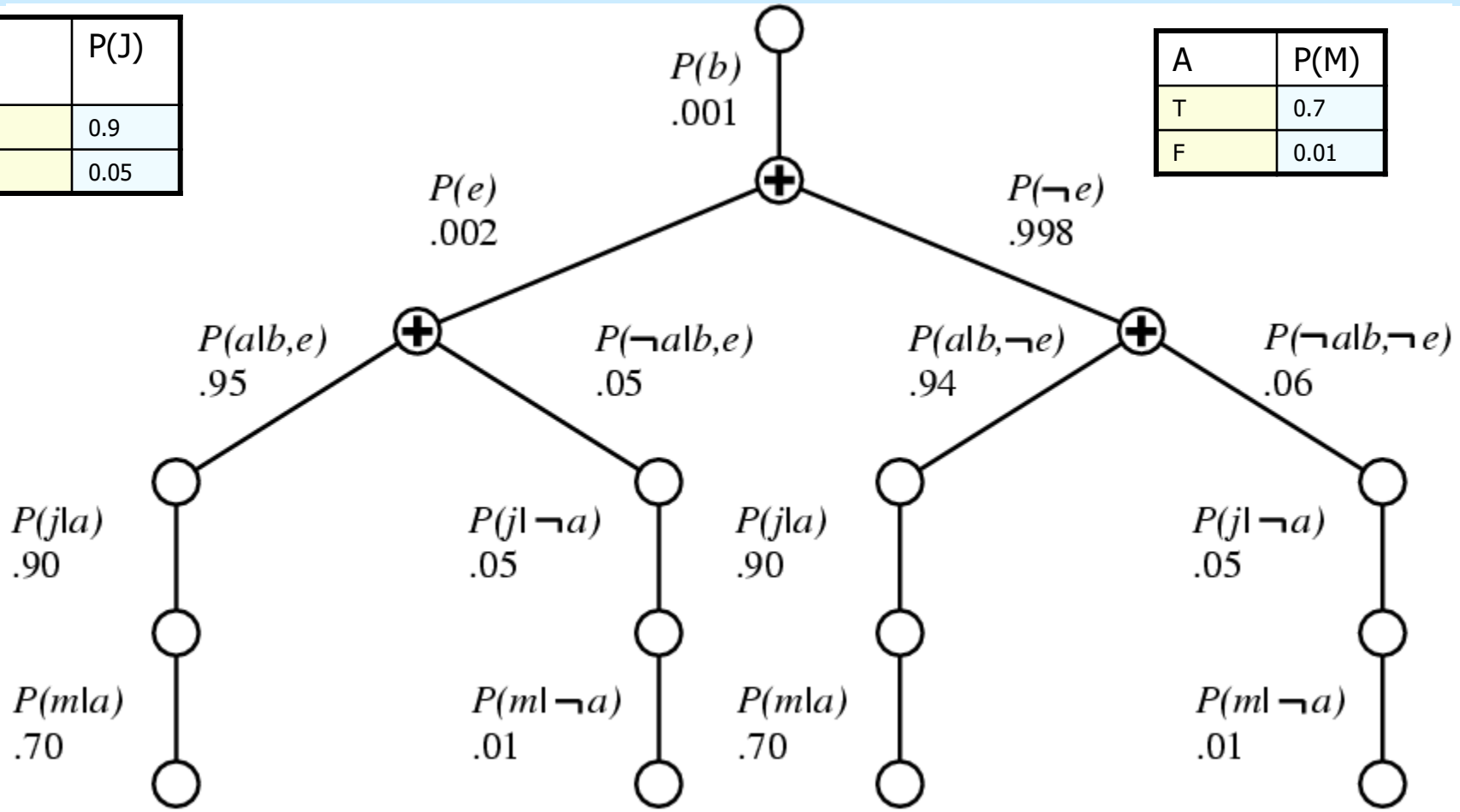
B	E	P(A)	P(w)
T	T	0.95	1× 0.002=0.002
T	F	0.94	1 × 0.998 = 0.998
F	T	0.29	0 × 0.002 = 0
F	F	0.001	0× 0.998 = 0

Final Structure

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

A	P(J)
T	0.9
F	0.05

A	P(M)
T	0.7
F	0.01



Enumerate-Ask

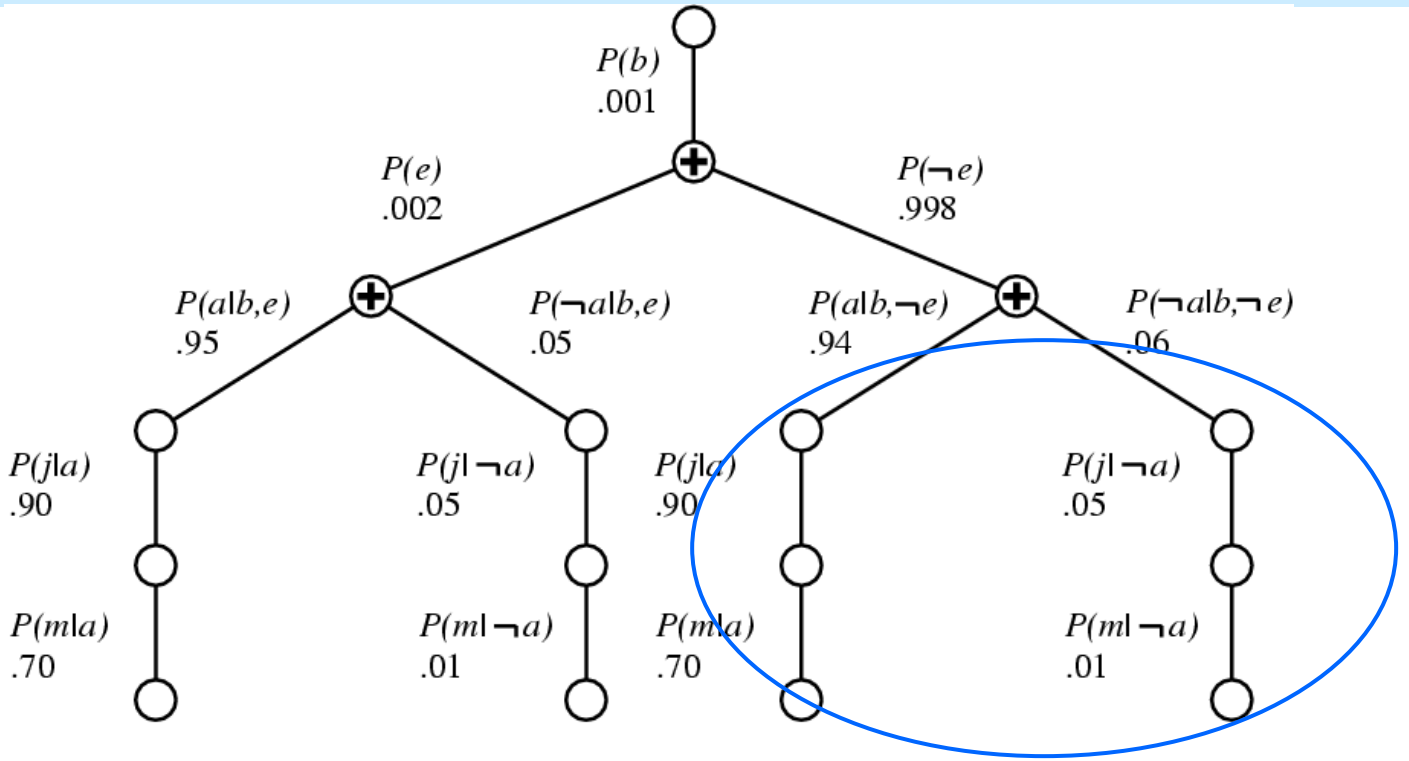
function ENUMERATION-ASK(X, **e**, bn) **returns** a distribution over X
  inputs: X, the query variable
    e, observed values for variables **E**
    bn, a Bayes net with variables $vars$

  $\mathbf{Q}(X) \leftarrow$ a distribution over X, initially empty
  for each value x_i of X **do**
    $\mathbf{Q}(x_i) \leftarrow$ ENUMERATE-ALL($vars$, $\mathbf{e}_{x_i}$)
      where $\mathbf{e}_{x_i}$ is **e** extended with $X = x_i$
  return NORMALIZE($\mathbf{Q}(X)$)

function ENUMERATE-ALL($vars$, **e**) **returns** a real number
  if EMPTY?($vars$) **then return** 1.0
  $V \leftarrow$ FIRST($vars$)
  if V is an evidence variable with value v in **e**
    then return $P(v \mid parents(V)) \times$ ENUMERATE-ALL(REST($vars$), **e**)
    else return $\sum_v P(v \mid parents(V)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}_v$)
      where $\mathbf{e}_v$ is **e** extended with $V = v$

Improvements

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



repetitions

Variable Elimination Algorithm

➢ **Idea**: *Eliminate repeated calculations.*

- How? *Do the calculations once and save the results for later use*
- *Evaluate expressions in the right-to-left order (bottom-up in the tree)*

$$P(B \mid j,m) = \alpha \underbrace{P(B)}_{F_1(B)} \sum_e \underbrace{P(e)}_{F_2(E)} \sum_a \underbrace{P(a \mid B,e) P(j \mid a) P(m \mid a)}_{F_3(A) \quad F_4(J) \quad F_5(M)}$$

factors

$$F_1(B) = \begin{pmatrix} P(B=T) \\ P(B=F) \end{pmatrix} = \begin{pmatrix} 0.001 \\ 0.999 \end{pmatrix}$$

$$F_2(E) = \begin{pmatrix} P(E=T) \\ P(E=F) \end{pmatrix} = \begin{pmatrix} 0.002 \\ 0.998 \end{pmatrix}$$

Factors

- *Each factor is a matrix indexed by the values of its argument variables*

$$P(B|j,m) = \alpha \underbrace{P(B)}_{F_1(B)} \sum_e \underbrace{P(e)}_{F_2(E)} \sum_a \underbrace{P(a|B,e)}_{F_3(A,B,E)} \underbrace{P(j|a)}_{F_4(A)} \underbrace{P(m|a)}_{F_5(A)}$$

factors

$$F_1(B) = \begin{pmatrix} P(b=T) \\ P(b=F) \end{pmatrix} = \begin{pmatrix} 0.001 \\ 0.999 \end{pmatrix}$$

$$F_2(E) = \begin{pmatrix} P(e=T) \\ P(e=F) \end{pmatrix} = \begin{pmatrix} 0.002 \\ 0.998 \end{pmatrix}$$

$F_3(A,B,E) = ????$

$$F_4(A) = \begin{pmatrix} P(j|a) \\ P(j|\neg a) \end{pmatrix} = \begin{pmatrix} 0.90 \\ 0.05 \end{pmatrix}$$

$$F_5(A) = \begin{pmatrix} P(m|a) \\ P(m|\neg a) \end{pmatrix} = \begin{pmatrix} 0.70 \\ 0.01 \end{pmatrix}$$

How does one compute the factors???

A set of initial factors::

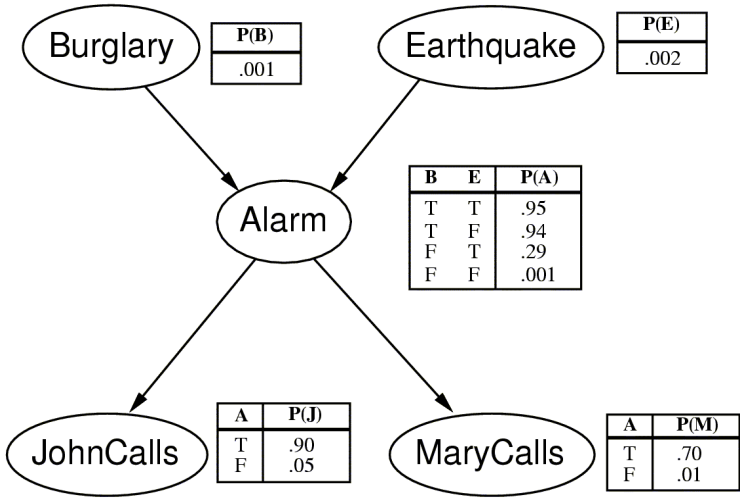
B	$F_1(B)$
T	0.001
F	0.999

E	$F_2(E)$
T	0.002
F	0.998

B	E	A	$F_3(A,B,E)$
T	T	T	0.95
T	T	F	0.05
T	F	T	0.94
T	F	F	0.06
F	T	T	0.29
F	T	F	0.71
F	F	T	0.001
F	F	F	0.999

A	J	$F^*_4(A,J)$
T	T	0.9
T	F	0.1
F	T	0.05
F	F	0.95

A	M	$F^*_5(A,M)$
T	T	0.7
T	F	0.3
F	T	0.01
F	F	0.99



Query P(B|j,m) $\Rightarrow$ j=T and m=T, thus

$$F_4(A) = \begin{pmatrix} P(j|a) \\ P(j|\neg a) \end{pmatrix} = \begin{pmatrix} 0.90 \\ 0.05 \end{pmatrix}$$

$$F_5(A) = \begin{pmatrix} P(m|a) \\ P(m|\neg a) \end{pmatrix} = \begin{pmatrix} 0.70 \\ 0.01 \end{pmatrix}$$

Variable Elimination Algorithm

- Now we can write:

$$P(B \mid j,m) = \alpha P(B) \sum_e P(e) \sum_a P(a \mid B,e) P(j \mid a) P(m \mid a)$$

as:

$$P(B|j,m) = \alpha\, F_1(B) \times \sum_e F_2(E) \times \sum_a F_3(A,B,E) \times F_4(A) \times F_5(A)$$

Where $\times$ is the **pointwise product**, NOT an ordinary matrix multiplication!!!!!!!!!!!!!!!!!

Pointwise Product

- it is not matrix multiplication!
- it is not element-by-element multiplication!

Pointwise product of two factors f_1 and f_2 yields a new factor f whose variables are the union of variables f_1 and f_2. Suppose the two factors have variables $Y_1,\ldots,Y_k$ in common.

Then we have:

$$f(X_1,\ldots,X_j,Y_1,\ldots,Y_k,Z_1,..,Z_e) =$$

$$= f_1(X_1,\ldots,X_j,Y_1,\ldots,Y_k)\, f_2(Y_1,\ldots,Y_k,Z_1,..,Z_e)$$

Computing Pointwise Product

$$f(X_1,\ldots,X_j,Y_1,\ldots,Y_k,Z_1,..,Z_l) = f_1(X_1,\ldots,X_j,Y_1,\ldots,Y_k)f_2(Y_1,\ldots,Y_k,Z_1,..,Z_l)$$

- If all the variables are binary then f_1 and f_2 have 2^{j+k} and 2^{k+l} entries respectively, and the pointwise product has 2^{j+k+l} entries.
- **Example**: Consider $f_1(A,B)$ and $f_2(B,C)$

A	B	$f_1(A,B)$	B	C	$f_2(B,C)$	A	B	C	$f_3(A,B,C)$
T	T	.3	T	T	.2	T	T	T	.3 × .2
T	F	.7	T	F	.8	T	T	F	.3 × .8
F	T	.9	F	T	.6	T	F	T	.7 × .6
F	F	.1	F	F	.4	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

Evaluating with variable elimination

- Now we evaluate*:*

$$P(B|j,m) = \alpha\, F_1(B) \times \sum_e F_2(E) \times \sum_a F_3(A,B,E) \times F_4(A) \times F_5(A)$$

By summing out variables right to left from pointwise products of factors $\Rightarrow$ to produce new factors, eventually generating the factor that constitutes the solution = *posterior distribution over the query variable*!!!!

- First, we sum out A from the pointwise product of F_3, F_4 and F_5 $\Rightarrow$ we shall obtain a new factor $F_6(B,E)$ whose indices range over the values of B and E

$F_6(E,B) = \sum_a F_3(A,B,E) \times F_4(A) \times F_5(A) = (F_3(a,B,E) \times F_4(a) \times F_5(a)) + (F_3(\neg a,B,E) \times F_4(\neg a) \times F_5(\neg a))$

Evaluating with variable elimination -2

- Now from :

$$F_6(B,E) = \sum_a F_3(A,B,E) \times F_4(A) \times F_5(A) = (F_3(a,B,E) \times F_4(a) \times F_5(a)) + (F_3(\neg a,B,E) \times F_4(\neg a) \times F_5(\neg a))$$

- We are left with evaluating:

$$P(B|j,m) = \alpha\, F_1(B) \times \sum_e F_2(E) \times F_6(B,E)$$

Next, we sum out variable E , from the product of F_2 and $F_6 \Rightarrow$

$$F_7(B) = \sum_e F_2(E) \times F_6(B,E) = (F_2(e) \times F_6(B,e) + F_2(\neg e) \times F_6(B,\neg e)$$

This leaves the expression:

$$P(B|j,m) = \alpha\, F_1(B) \times F_7(B)$$

Computing the NEW factors

- *Let us start with:*

$$F_6(E,B) = \sum_a F_3(A,B,E) \times F_4(A) \times F_5(A)$$

B	E	A	F_3
T	T	T	0.95
T	T	F	0.05
T	F	T	0.94
T	F	F	0.06
F	T	T	0.29
F	T	F	0.71
F	F	T	0.01
F	F	F	0.99

×

A	F_4
T	0.9
F	0.05

×

A	F_5
T	0.7
F	0.01

=

B	E	F_6
T	T	0.95×0.7×0.9 + 0.05×0.01×0.05
T	F	0.94×0.7×0.9 + 0.06×0.01×0.05
F	T	0.29×0.7×0.9 + 0.71×0.01×0.05
F	F	0.01×0.7×0.9 + 0.99×0.01×0.05

A-Variable Elimination!!!

Continue

$$F_7(B) = \sum_e F_2(E) \times F_6(B, E)$$

E	F_2
T	0.002
F	0.998

×

B	E	F_6
T	T	0.95×0.7×0.9 + 0.05×0.01×0.05
T	F	0.94×0.7×0.9 + 0.06×0.01×0.05
F	T	0.29×0.7×0.9 + 0.71×0.01×0.05
F	F	0.01×0.7×0.9 + 0.99×0.01×0.05

=

B	F_7
T	0.002×(0.95×0.7×0.9+0.05×0.01×0.05)+ 0.998×(0.94×0.7×0.9+0.06×0.01×0.05)
F	0.002×(0.29×0.7×0.9 +0.71×0.01×0.05)+ 0.998×(0.01×0.7×0.9 +0.99×0.01×0.05)

E-Variable Elimination!!!

Computing the answer

$$P(B|j,m) = \alpha\, F_1(B) \times F_7(B)$$

B	F_1
T	0.001
F	0.999

×

B	F_7
T	0.002×(0.95×0.7×0.9+0.05×0.01×0.05)+ 0.998×(0.94×0.7×0.9+0.06×0.01×0.05)
F	0.002×(0.29×0.7×0.9 +0.71×0.01×0.05)+ 0.998×(0.01×0.7×0.9 +0.99×0.01×0.05)

Variable Elimination!!!

Elimination-Ask

The variable elimination algorithm for **exact inference** in Bayes nets.

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 with variables $vars$

  $factors \leftarrow [\,]$
  for each V **in** ORDER($vars$) **do**
    $factors \leftarrow [\text{MAKE-FACTOR}(V, \mathbf{e})] + factors$
    if V is a hidden variable **then** $factors \leftarrow \text{SUM-OUT}(V, factors)$
  return NORMALIZE(POINTWISE-PRODUCT($factors$))

Variable Ordering and Variable Relevance

This algorithm has an unspecified ORDERING function to operate on the variables.

Variable ordering and relevance

- *Variable Ordering and Variable Relevance*

The Variable Elimination (VE) algorithm has an unspecified ORDERING function to operate on the variables. $\Rightarrow$ Every choice yields a valid algorithm! But different ordering cause different intermediary factors to be calculated!

Different ordering generate different factors. It is intractable to determine the optimal ordering, but several heuristics are available.

Example: The Greedy heuristic: Eliminate whichever variable which minimizes the Size of the next Factor.

In general, every variable which is not an ancestor of a query variable is irrelevant to the query. The VE algorithm can eliminate all of them before evaluating the query.

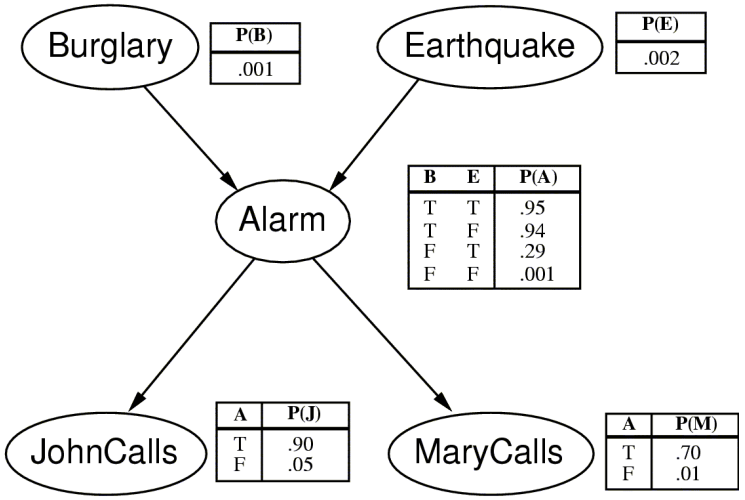
Complexity of Exact Inference

- The complexity of exact inference in Bayes nets depends **strongly** on the structure of the network.

The Burglary network is a singly connected network – a polytree.

Time and space complexity of exact inference in polytrees is linear in the size of the network!!!!

Size = number of CPT entries.



Clustering Algorithms

- If we want to compute posterior probabilities for all variables in the network – the variable elimination algorithm is not efficient.

Solution: Consider clustering algorithms also known as join tree algorithms.

The idea: Cluster individual modes of the network to form cluster nodes ⇒ the resulting network is a polytree.