BAYESIAN NETWORKS

Bayesian networks

- Network models
  - to reason under uncertainty
  - according to the laws of probability theory

Bayesian network

- A simple graphical notation
  - to represent the **dependencies among variables** and
  - for **compact specification** of any **full joint probability distribution**

Outline

- Syntax
- Semantics

Bayesian networks

- Syntax:
  - a **directed** graph
  - a **set of nodes,** one per variable
  - a set of **oriented arcs** (X→Y means X "directly influences" Y)
  - **For each node** X_i **,** a **conditional probability distribution** given parents of X_i **P** $(X_i \mid \text{Parents}(X_i))$

represented as a *conditional probability table* (**CPT**) giving the **probability distribution over** X_i **for each combination of parents values**

Example

- You have a new burglar alarm installed at home
  - fairly reliable at detecting a burglary, but
  - responds on occasion to minor earthquakes

- You also have two 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 and calls then, too
  - **Mary**, on the other hand, likes rather loud music and often misses the alarm altogether

- Given the evidence of who has or has not called, we would like to **estimate** the probability of a burglary

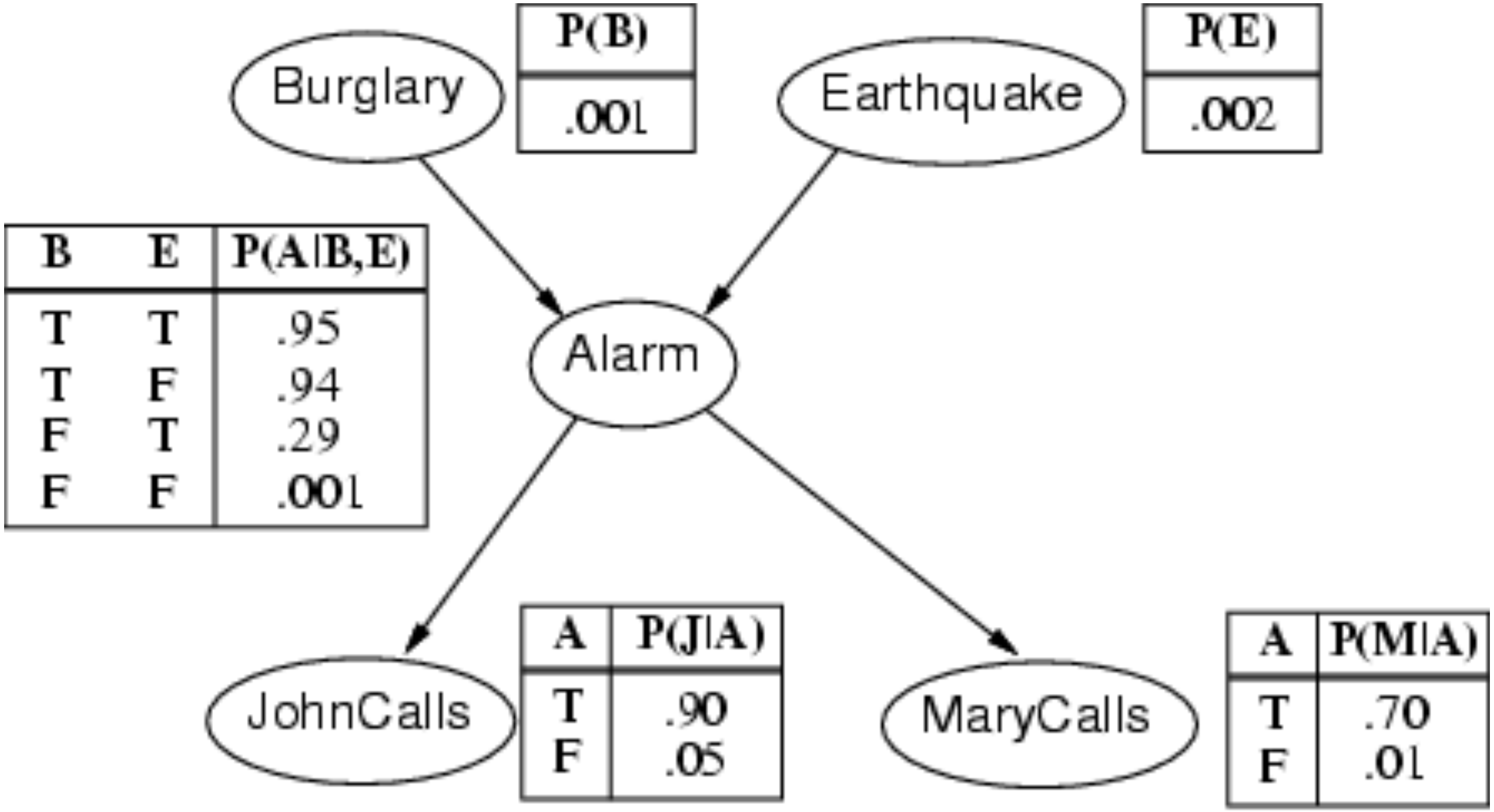
Example

- 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: *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

Example contd.



P(B)
.001

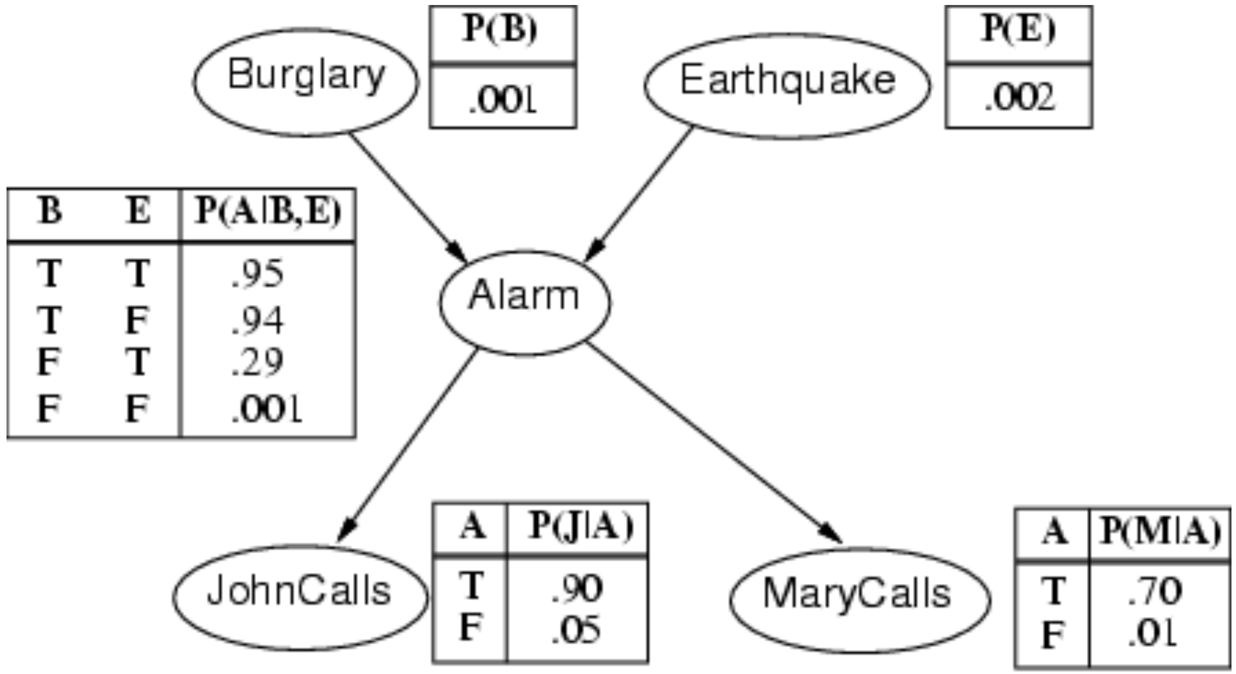
P(E)
.002

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

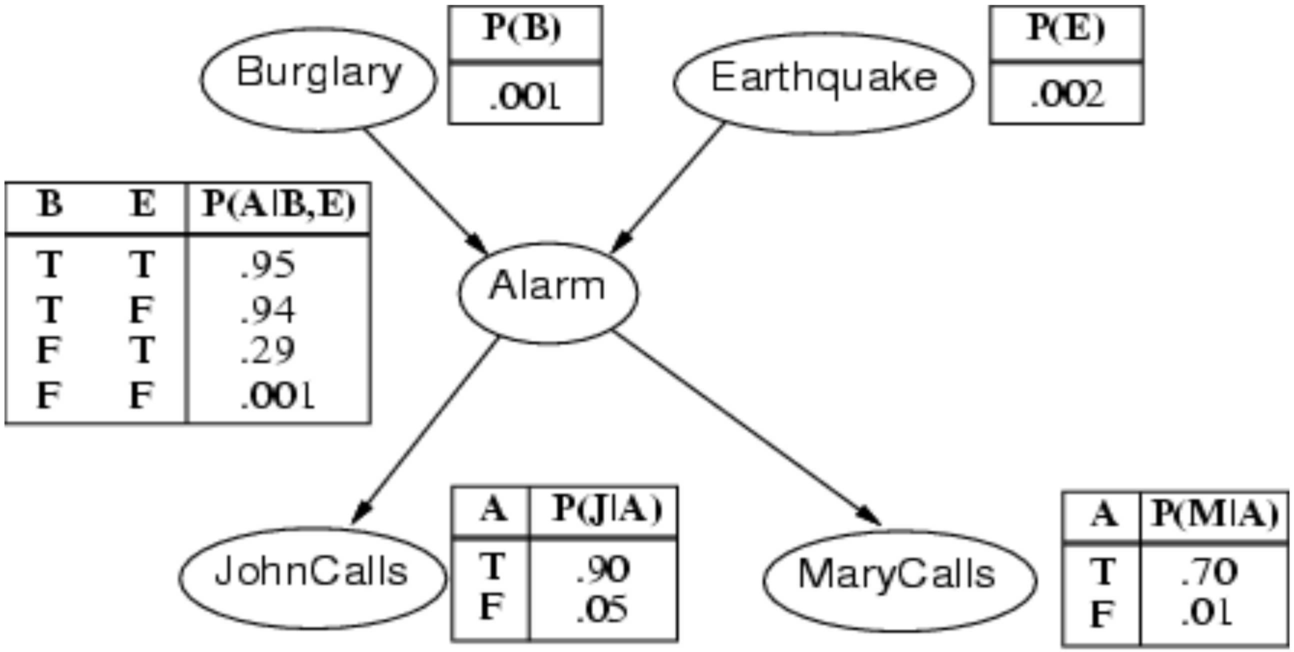
Example contd.



The **network structure** shows that

- burglary and earthquakes **directly affect** the probability of the alarm's going off
- whether John and Mary call **depends** only on the alarm.

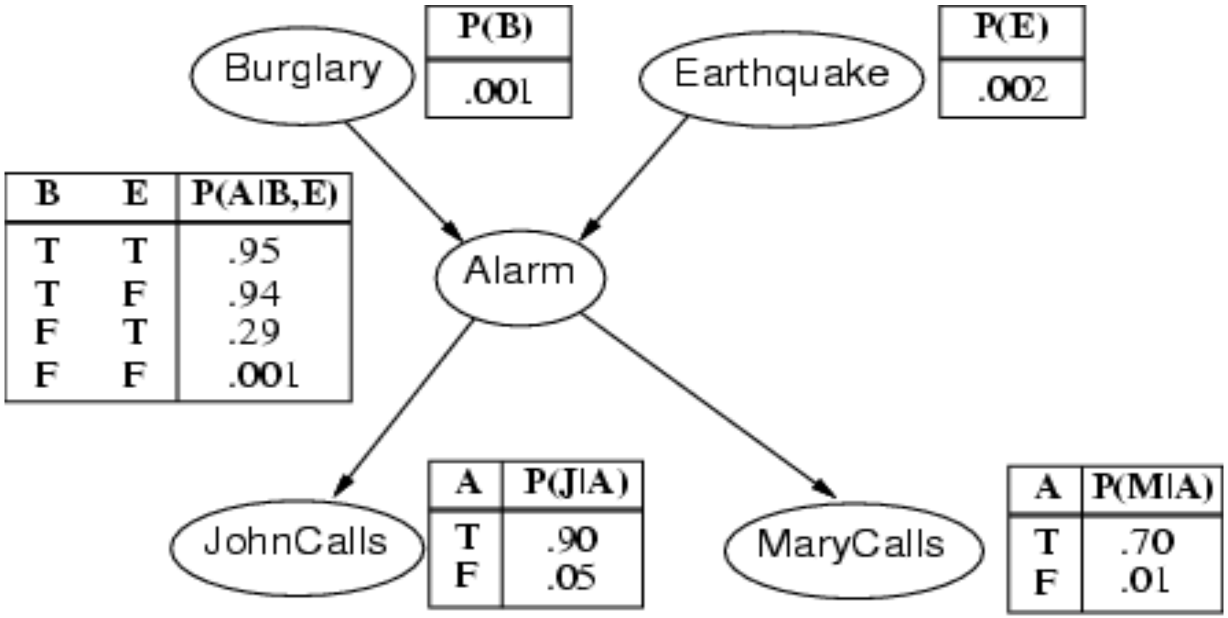
Example contd.



The network thus represents our **assumptions:**

- Mary and John **do not perceive** burglaries directly
- They **do not notice** minor earthquakes
- They **do not confer** before calling

Example contd.



The network does **not have nodes** corresponding to

- Mary's currently listening to loud music or
- the telephone ringing and confusing John

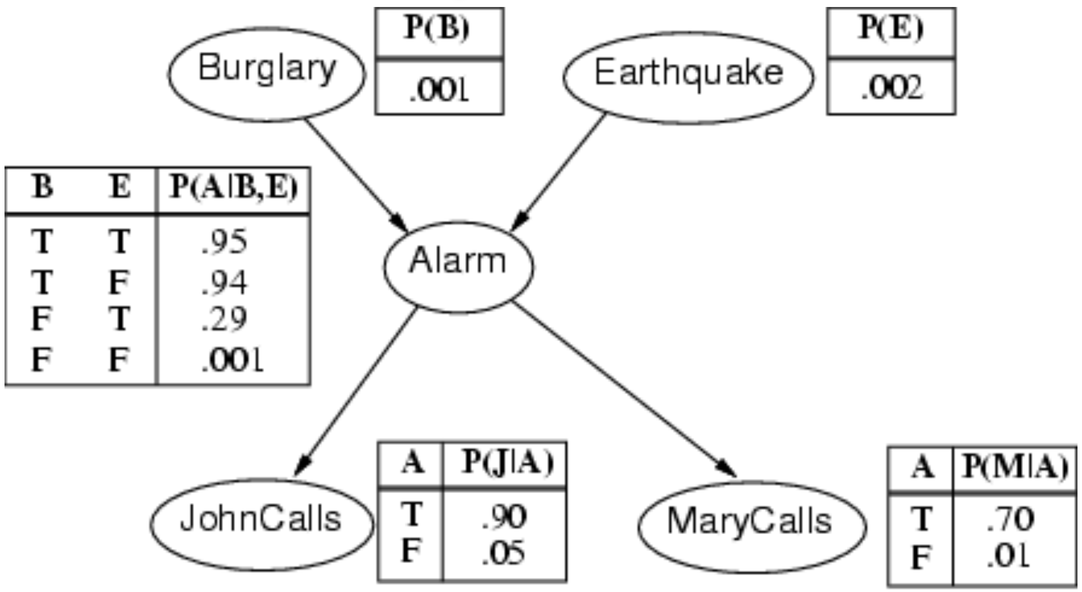
These factors **are summarized** in the **uncertainty associated with the links** from Alarm to JohnCalls and MaryCalls.

Example contd.

Notice that in the network

P(B) = **P(B=true)**

- **P(B = true)** = 0.001
- **P(B = false) =** 1- P(B=true) = 0.999



The **conditional distributions** are shown as a **conditional probability table** (CPT)

- Each row in a CPT contains the conditional probability of each node value for a **conditioning case,** that is, for each possible combination of values for the parent nodes
- For **Boolean variables**, once you know that the probability of a true value is p, the probability of false must be 1 – p, so we often omit the second number

BAYESIAN NETWORKS - PART II

Compactness

Bayesian network: **compact** representation than the **full joint distribution**

- A CPT for **a Boolean variable** X_i with k **Boolean parents** has

  2^k **rows** for the combinations of parents values

  Each row requires **one number** p for X_i = *true*

  (since the number for X_i = *false* is $1-p$)

- Assume there are **n Boolean variables**
  - If each variable has **no more than** k **parents,**

    the **Bayesian network** can be **specified** by at most $n \cdot 2^k$ numbers

  - The **full joint distribution** contains 2^n *numbers*

- For **burglary net,** 1 + 1 + 4 + 2 + 2 = **10** numbers

  (vs. 2^5-1 = **31** numbers in full joint distribution)

P(b, e, a, j, m) =
P(B=true, E= true, A=true, J=true, M=true)

Full joint distribution

- $P(b, e, a, j, m) = \ldots?$
- $P(b, e, a, j, \neg m) = \ldots?$
- $P(b, e, a, \neg j, m) = \ldots?$
- $P(b, e, a, \neg j, \neg m) = \ldots?$
- …
- … all the possible combinations! 32 numbers!

- With 5 boolean variables: $2^5 = 32$ numbers
- We need to recall only 31 numbers

Compactness

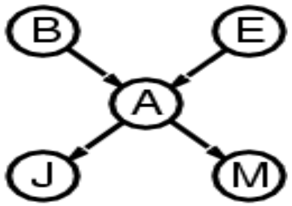
Assume there are **n Boolean variables**
If each variable has **no more than *k* parents**,
- **Bayesian network** can be **specified** by at most $n \cdot 2^k$ numbers
- **Full joint distribution** contains 2^n *numbers*

- Example:
  - Assume Boolean variables
  - Suppose we have 30 nodes ($n = 30$)
  - Suppose each node has 5 parents ($k = 5$)
  - **Bayesian network** requires $30*2^5 = 960$ numbers
  - **Full joint distribution** requires **over a billion** of numbers

Semantics

The full joint distribution is defined as the **product** of the **local** conditional distributions:

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



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

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

Example

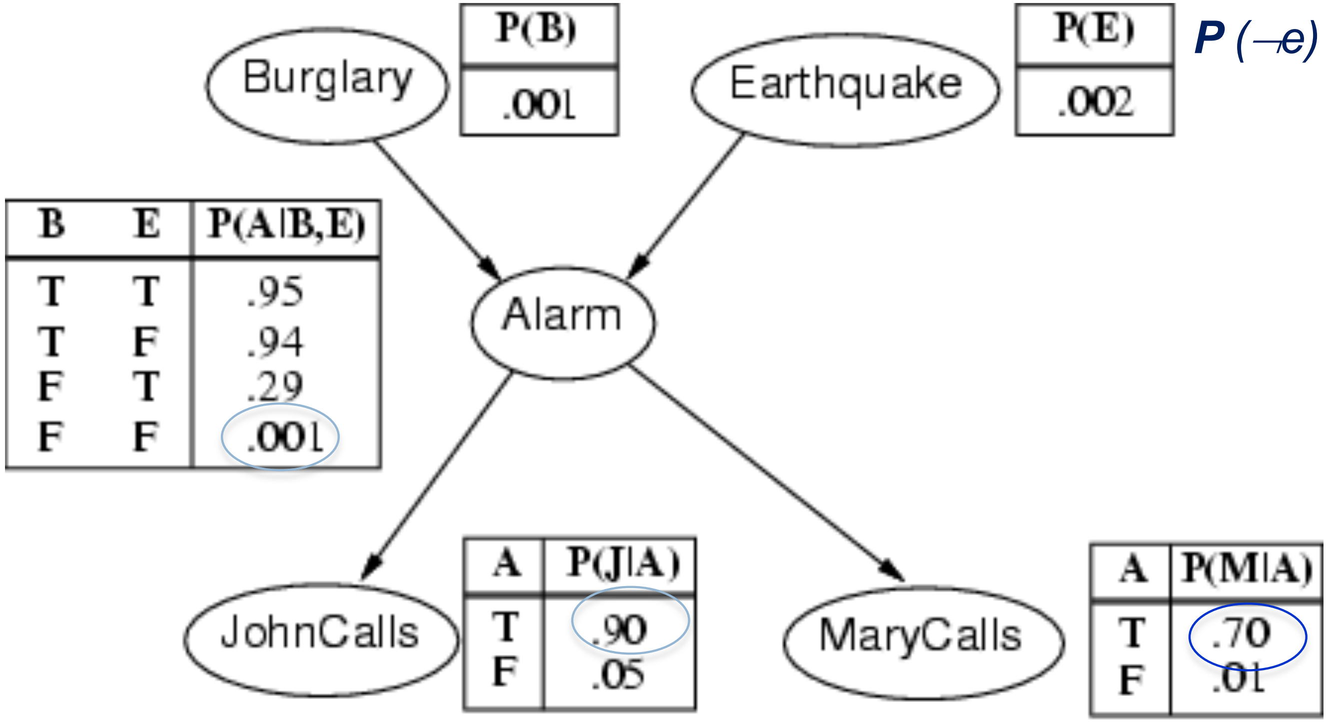
$P(x_1, \ldots, x_n) = \Pi_{i=1}^{n} P(x_i \mid parents(X_i))$

In the network P(B) = P(B=true)

- P(b) = P(B=true) = 0.001
- P(¬b) = P(B=false)= 1- P(B=true) = 0.999

P (¬b) = P(B =false) = 0.999

P (¬e) = P(E=false) = 0.998



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

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

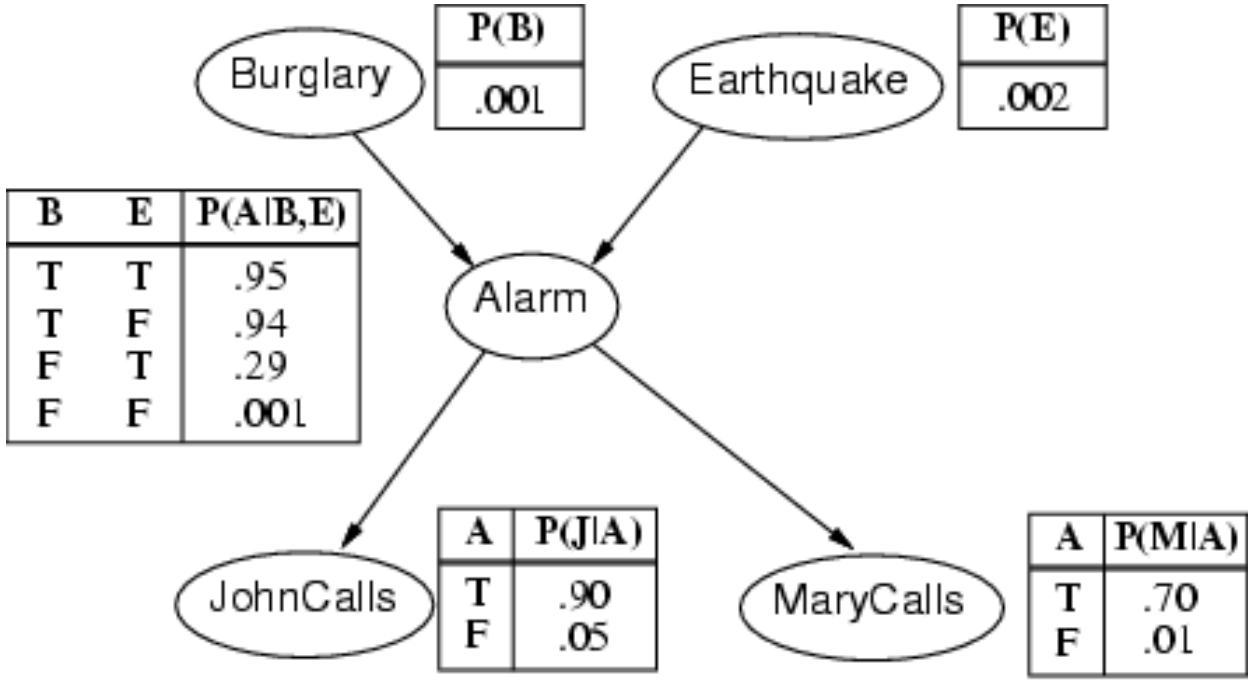
$= 0.90 \times\ 0.70 \times 0.001 \times\ 0.999 \times 0.998 \quad =$

$= 0.000628$

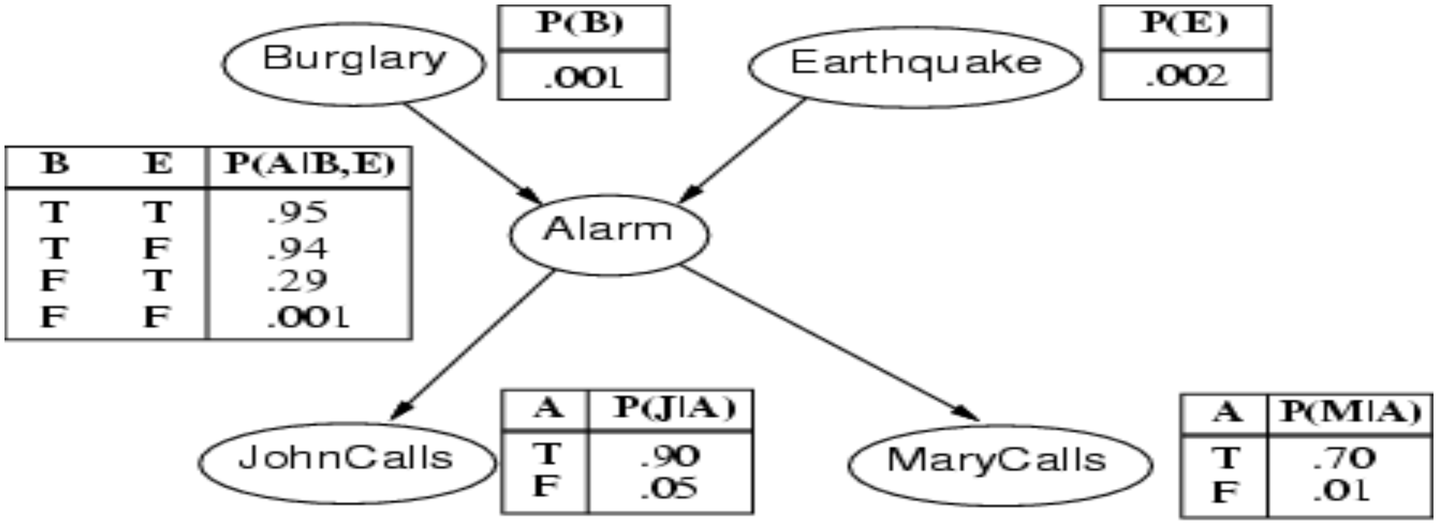
EXERCISE (BAYESIAN NETWORK)

Bayesian networks

- Given the BN below, compute the probability P(e, -b, a, j, -m)



Bayesian networks



P(e, -b, a, j, -m) =

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

$= 0.002 \times 0.999 \times 0.29 \times 0.90 \times 0.30$

$= 0.00015644$

BAYESIAN NETWORKS - PART III

Review: Bayesian network

- A simple graphical notation
  - to represent the **dependencies among variables** and
  - for **compact specification** of any **full joint probability distribution**

Review: Bayesian networks

- **Syntax:**
  - a **directed** graph
  - a **set of nodes,** one per variable
  - a set of **oriented arcs** (X→Y means X "directly influences" Y)
  - **For each node** X_i ,

    a **conditional probability distribution** given parents of X_i

    $\mathbf{P}(X_i \mid \text{Parents}(X_i))$

represented as a *conditional probability table* (**CPT**) giving the **probability distribution over** X_i **for each combination of parents values**

Review: Semantics

The full joint distribution is defined as the **product** of the **local** conditional distributions:

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

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

Constructing Bayesian networks

- 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

$$\boxed{P(X_i \mid Parents(X_i)) = P(X_i \mid X_1, \ldots, X_{i-1})}$$

Intuitively, **parents of node X_i** should contain **all those nodes** in $X_1, \ldots, X_{i-1}$ that ***directly influence*** X_i

This choice of parents guarantees:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid X_1, \ldots, X_{i-1})$$

(chain rule)

$$= \prod_{i=1}^{n} P(X_i \mid Parents(X_i))$$

(by construction)

Inference in Bayesian Networks

- Exact inference by enumeration
- Exact inference by variable elimination

Inference in Bayesian Networks

- **Basic task** for any probabilistic inference system:

  Computing the posterior **probability distribution**

  for a **set of query variables**

  given some **observed event**

  - **observed event =** an assignment of values to a set of ***evidence variables***

- We assume **one query variable**
  - Algorithms can be **easily extended** to queries with multiple variables

Inference in Bayesian Networks

- X denotes the query variable
- **E** denotes the set of **evidence variables** $E1, \ldots, Em$

  e is a particular observed event
- **Y** denotes **hidden variables** $Y1, \ldots, Yl$

  (that are the nonevidence, nonquery variables)
- Complete set of variables: $\mathbf{X} = \{X\} \cup \mathbf{E} \cup \mathbf{Y}$
- **Typical query: posterior probability distribution**

  $\mathbf{P}(X \mid \mathbf{e})$?

Inference in Bayesian Networks

X :query variable
E: evidence variables
Y: hidden variables

- Complete set of variables: $\mathbf{X} = \{X\} \cup \mathbf{E} \cup \mathbf{Y}$

- Typical query: **posterior probability distribution**

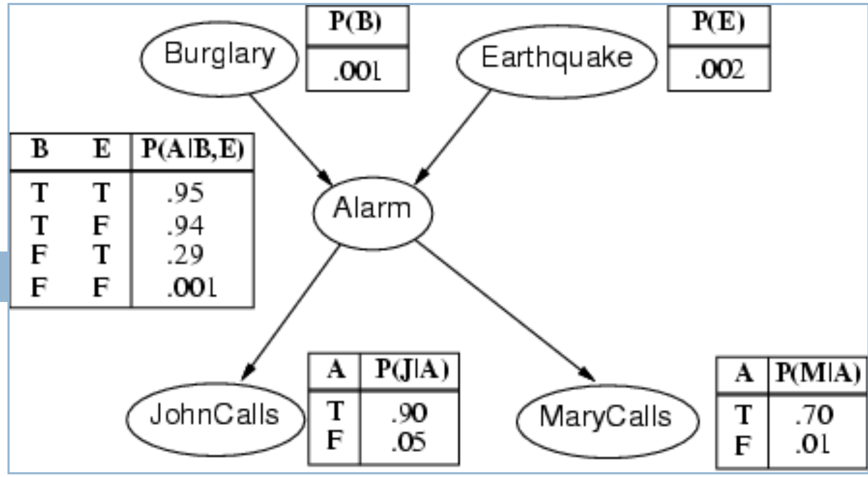
$$\mathbf{P}(X \mid \mathbf{e})?$$

bayes regola marginale

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

$$\alpha = \frac{1}{P(e)} = \frac{1}{\sum_x P(x, e)}$$

Inference



- Query on the burglary network
- **P** (B | j, m) = ?

Inference in Bayesian Networks

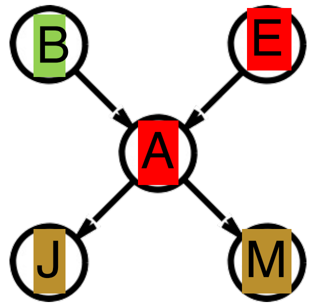
- Exact inference by enumeration
- Exact inference by variable elimination

Inference by enumeration

Review:
X: query variable
E: evidence variables
Y: hidden variables
$\mathbf{P}(X|\mathbf{e}) = \alpha \Sigma_y \mathbf{P}(X, \mathbf{e}, y)$

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

- Query on the burglary network



- $\boxed{\mathbf{P}(B \mid j, m) = ?}$

$= \alpha\, \mathbf{P}(B, j, m)$

$= \alpha\, \Sigma_e\ \Sigma_a\, P(B, e, a, j, m)$

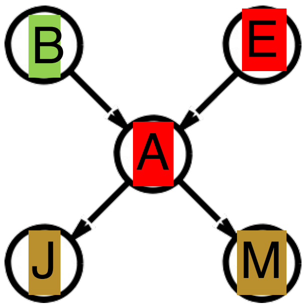
$\mathbf{P}(X|\mathbf{e}) = \alpha\, \Sigma_y\, \mathbf{P}(X, \mathbf{e}, \mathbf{y})$

Bayesian network: a representation of the **full joint distribution**

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

Inference by enumeration

- $\mathbf{P}(B \mid j, m) = \alpha\, \Sigma_e\ \Sigma_a\, \boxed{\mathbf{P}(B, e, a, j, m)}$



Rewrite full joint entries **using <u>product</u> of CPT <u>entries</u>:**

For simplicity, we do this just for Burglary = true:

- $P(b \mid j, m) = \alpha\, \Sigma_e\ \Sigma_a\, P(b, e, a, j, m)$
- $= \alpha\, \Sigma_e\ \Sigma_a\, \boxed{P(b)\, P(e)\, P(a \mid b, e)\, P(j \mid a)\, P(m \mid a)}$
- $= \alpha\, P(b)\, \Sigma_e\, P(e)\ \Sigma_a\, P(a \mid b, e)\, P(j \mid a)\, P(m \mid a)$

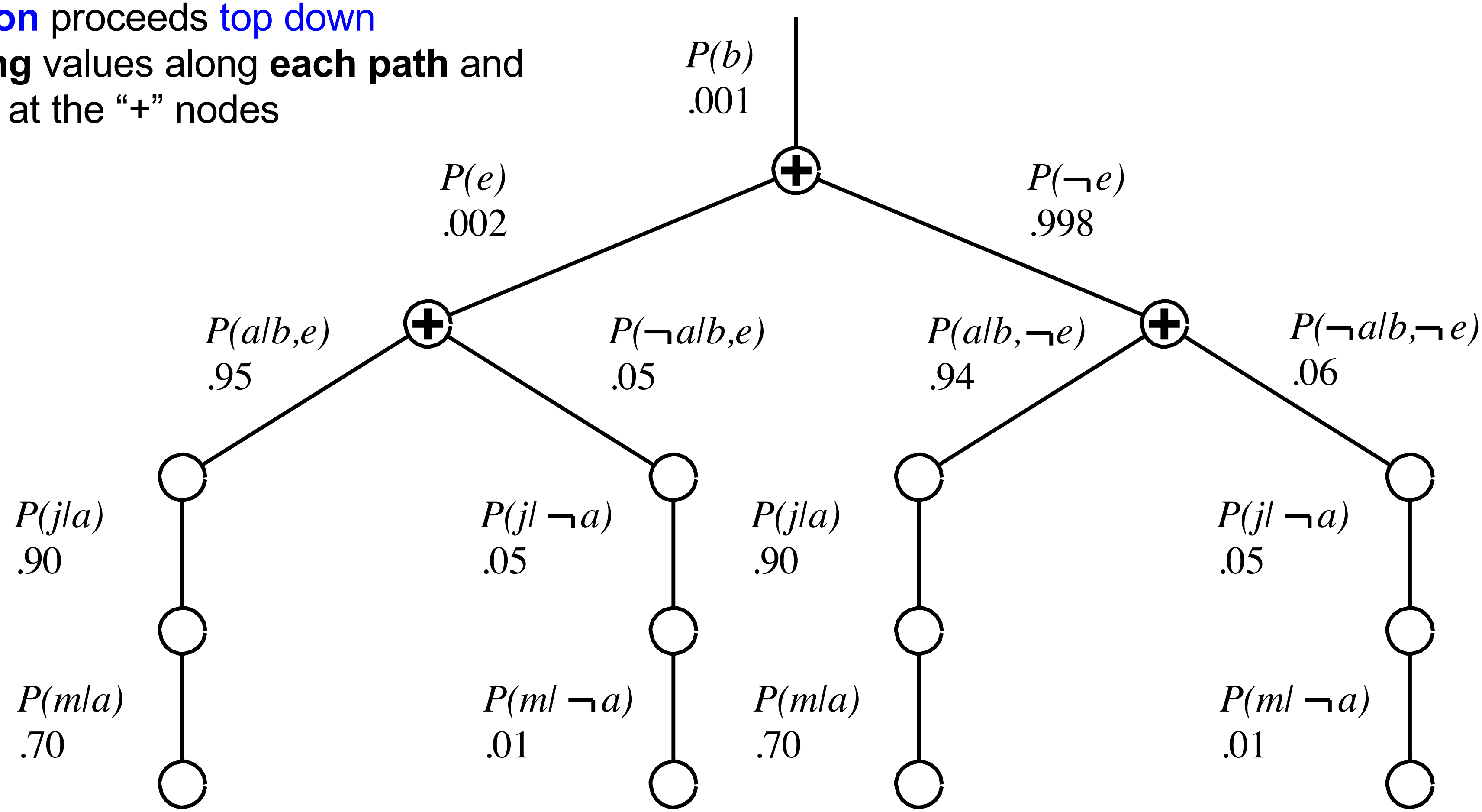
O(2^n) time complexity for n boolean variables

Evaluation tree

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

The **evaluation** proceeds top down

- **multiplying** values along **each path** and
- **summing** at the "+" nodes



Enumeration is inefficient: repeated computation

e.g., computes the product $P(j|a)P(m|a)$ for each value of e