Artificial Intelligence

Hai Thi Tuyet Nguyen

Outline

CHAPTER 1: INTRODUCTION (CHAPTER 1)

CHAPTER 2: INTELLIGENT AGENTS (CHAPTER 2)

CHAPTER 3: SOLVING PROBLEMS BY SEARCHING (CHAPTER 3)

CHAPTER 4: INFORMED SEARCH (CHAPTER 3)

CHAPTER 5: LOGICAL AGENT (CHAPTER 7)

CHAPTER 6: FIRST-ORDER LOGIC (CHAPTER 8, 9)

CHAPTER 7: QUANTIFYING UNCERTAINTY(CHAPTER 13)

CHAPTER 8: PROBABILISTIC REASONING (CHAPTER 14)

CHAPTER 9: LEARNING FROM EXAMPLES (CHAPTER 18)

CHAPTER 8: PROBABILISTIC REASONING

8.1 Representing Knowledge In An Uncertain Domain

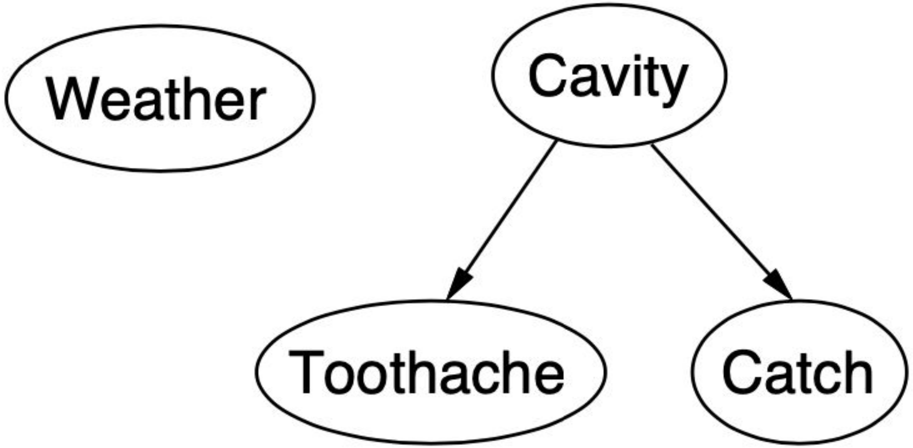
8.1 Representing Knowledge In An Uncertain Domain

- A simple, graphical notation for conditional independence assertions
- Syntax:
  - a set of nodes, one per variable
  - a directed, acyclic graph (link ≈ “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

8.1 Example

Topology of network encodes conditional independence assertions:



Weather is independent of the other variables

Toothache and Catch are conditionally independent given Cavity

8.1 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:

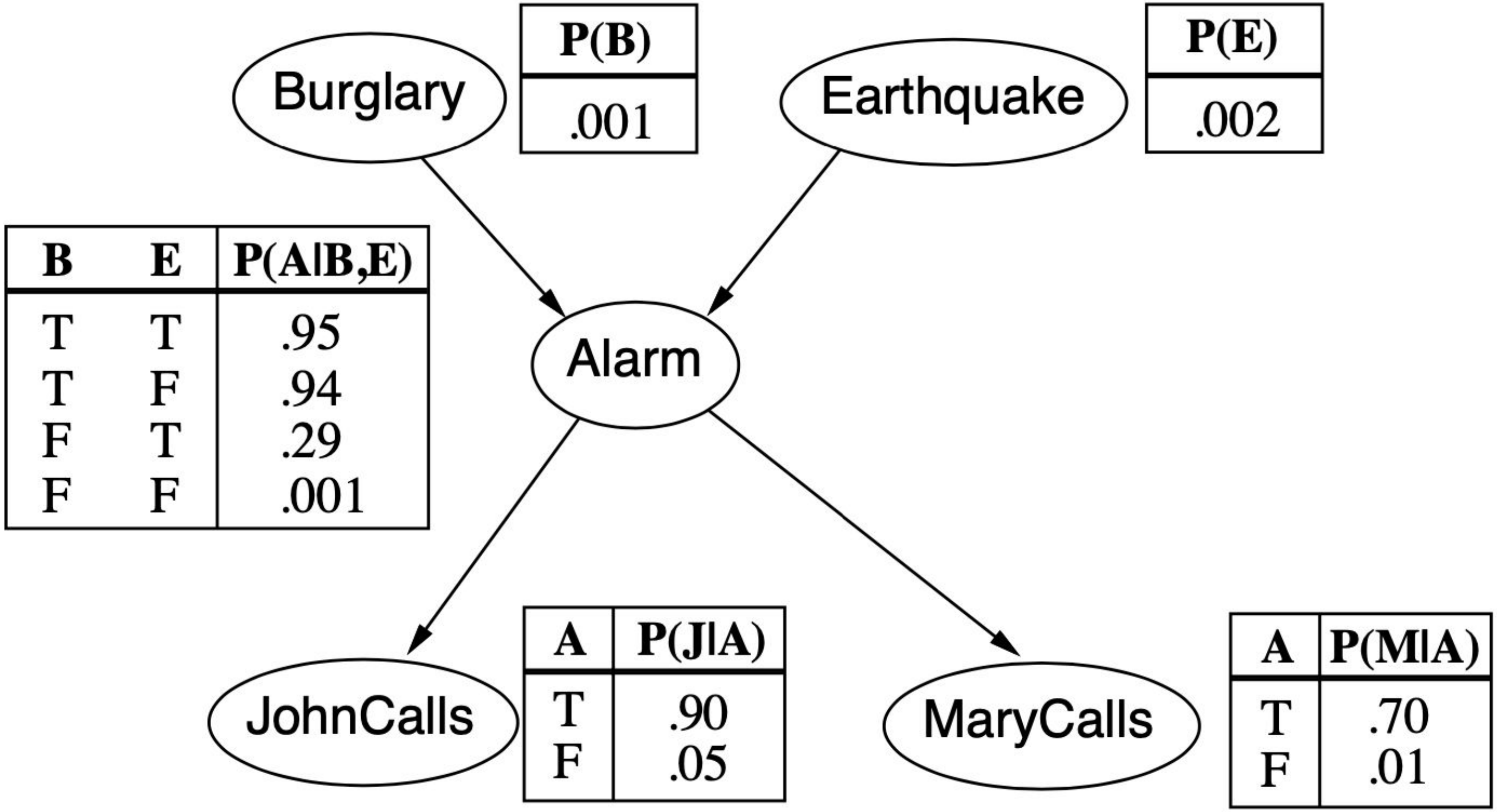
– A burglar can set the alarm off

– An earthquake can set the alarm off

– The alarm can cause Mary to call

– The alarm can cause John to call

8.1 Example



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

8.2 The Semantics Of Bayesian Networks

8.2 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 do

    add X_i to the network

    select parents from $X_1, \ldots, X_{i-1}$ such that

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

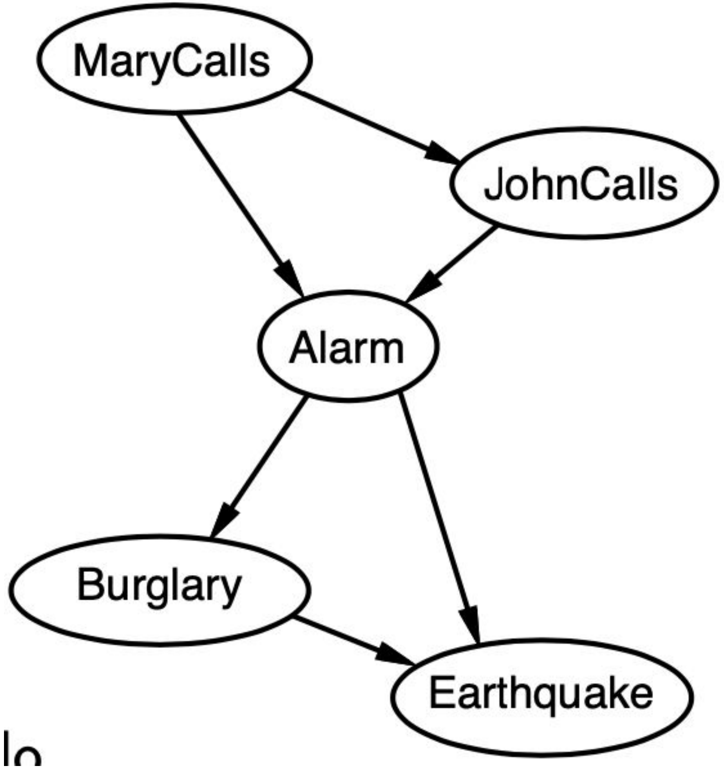
    CPTs: write down the **c**onditional **p**robability **t**able, $P(X_i|Parents(X_i))$

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

8.2 Example

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

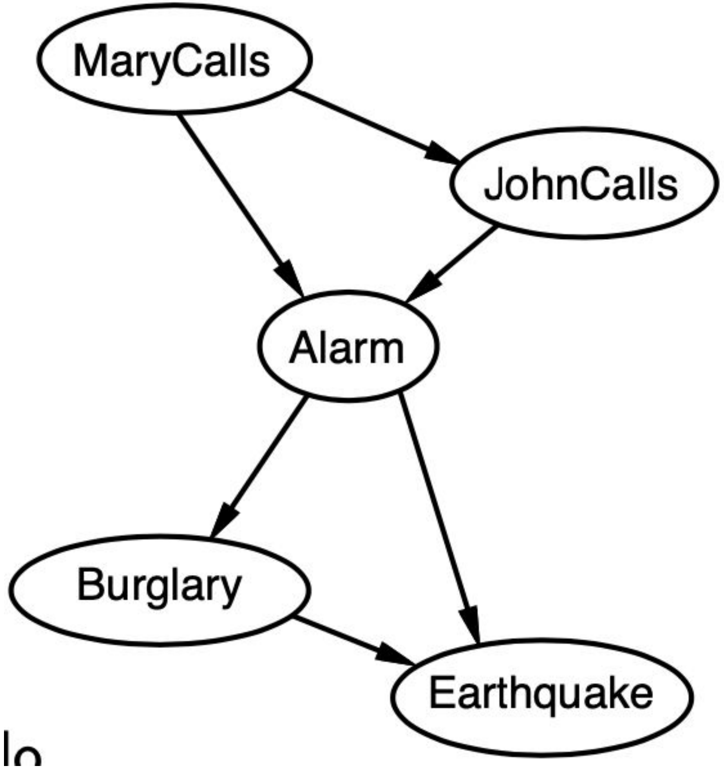
- Adding MaryCalls: No parents.
- Adding JohnCalls: If Mary calls (~ the alarm has gone off), which would make it more likely that John calls.
  => JohnCalls needs MaryCalls as a parent.
- Adding Alarm: Clearly, if both call, it is more likely that the alarm has gone off if just one or neither calls
  => MaryCalls and JohnCalls as parents.
- Adding Burglary: If we know the alarm state, then the call from John or Mary might give us information about our phone ringing, but not about burglary:
  P(Burglary | Alarm, JohnCalls, MaryCalls)
  = P(Burglary | Alarm) .
  => Alarm as parent.



8.2 Example

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

- 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
  => Alarm and Burglary as parents.



8.3 Exact Inference In Bayesian Networks

8.2 Inference by enumeration

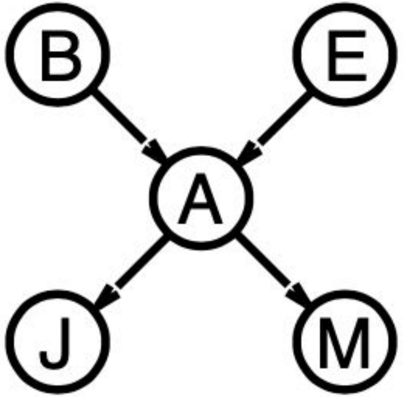
- Simple queries: compute the posterior probability distribution $P(X_i|E = e)$
  e.g., P(NoGas|Gauge=empty, Lights=on, Starts=false)
- A query can be answered using a Bayesian network by computing sums of products of conditional probabilities from the network
  Compute P(Burglary|JohnCalls=true,MaryCalls=true),
  the hidden variables for this query are Earthquake and Alarm

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

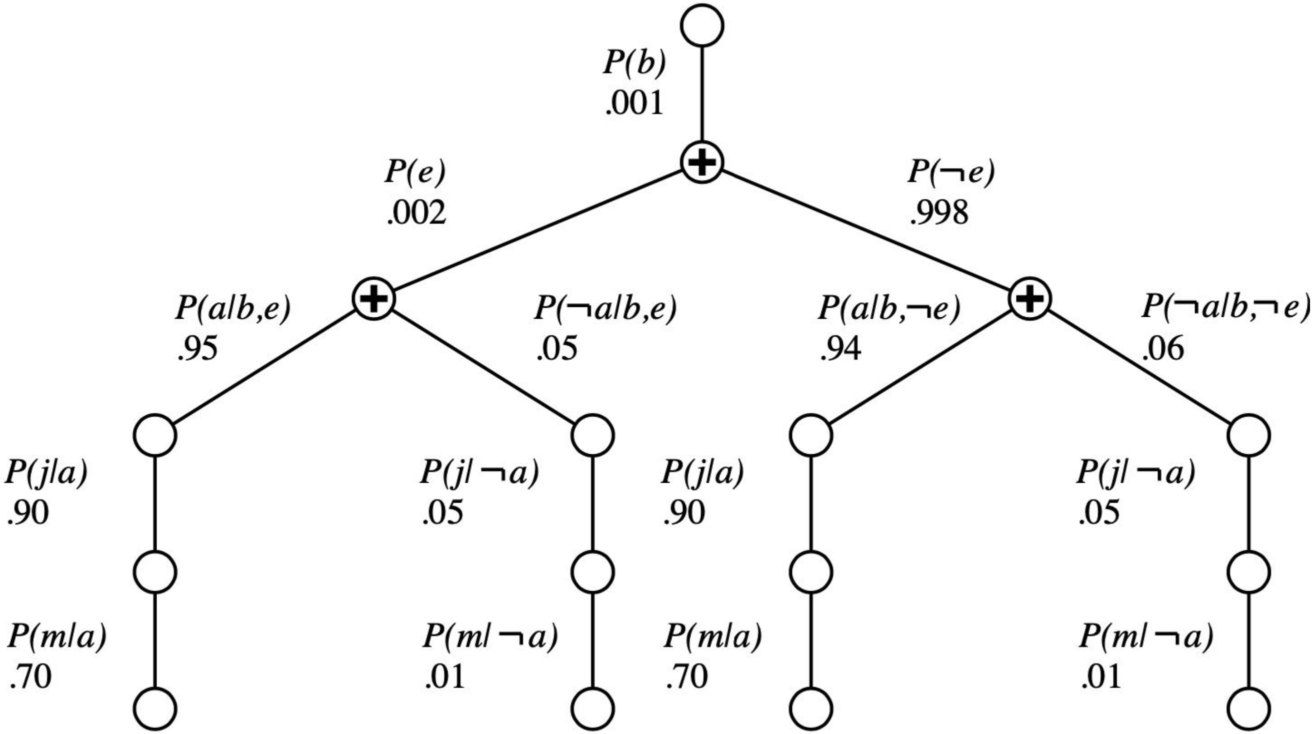
For Burglary = true

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

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



8.2 Inference by enumeration



The evaluation proceeds top down, multiplying values along each path and summing at the "+" nodes. Notice the repetition of the paths for j and m.

8.2 Inference by variable elimination

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

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

8.2 Variable elimination

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

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

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