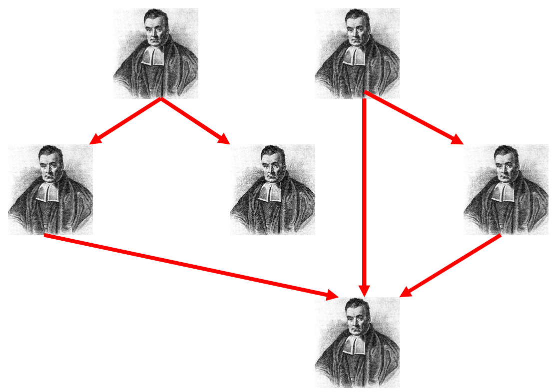
CSE 473

Chapter 14
Bayesian Networks

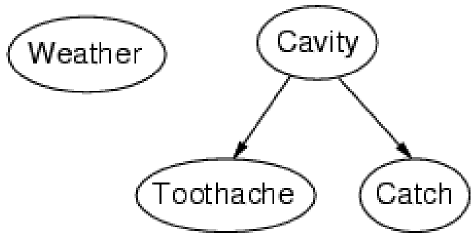


What are Bayesian networks?

- Simple, graphical notation for conditional independence assertions
- Allows compact specification of full joint distributions
- Syntax:
  - a set of nodes, one per random variable
  - a directed, acyclic graph (link ≈ "directly influences")
  - a conditional distribution for each node given its parents:
    P $(X_i \mid \text{Parents}(X_i))$
- For discrete variables, conditional distribution = **conditional probability table (CPT)** = distribution over X_i for each combination of parent values

Back at the Dentist's

- Topology of network encodes conditional independence assertions:



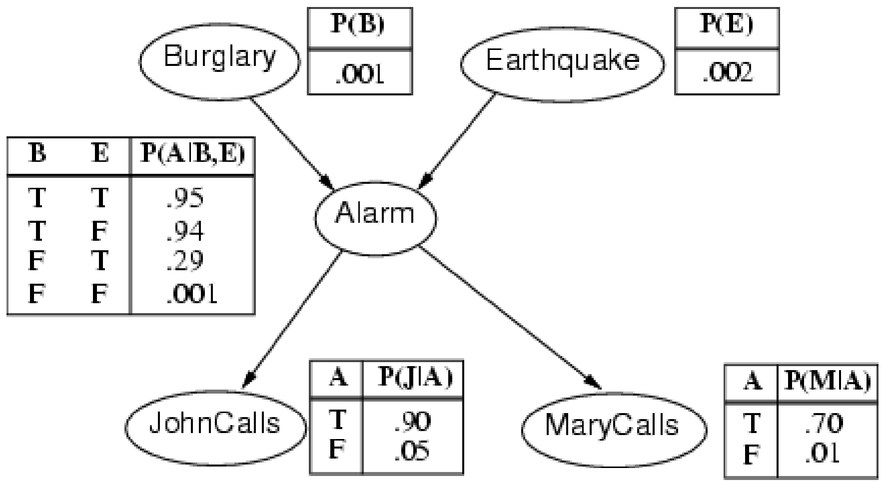
- *Weather* is independent of the other variables
- *Toothache* and *Catch* are <u>conditionally independent</u> of each other given *Cavity*

Example 2: Burglars and Earthquakes

- You are at a "Done with 473" party at a friend's.
- Neighbor John calls to say your home alarm is ringing (but neighbor Mary doesn't).
- Sometimes your alarm is set off by minor earthquakes.
- Question: Is your home being burglarized?
- Variables: *Burglary*, *Earthquake*, *Alarm*, *JohnCalls*, *MaryCalls*
- Network topology reflects "causal" knowledge:

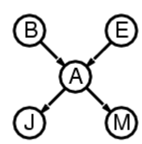
  A burglar can set the alarm off
  An earthquake can set the alarm off
  The alarm can cause Mary to call
  The alarm can cause John to call

Burglars and Earthquakes



Compact Representation of Probabilities in Bayesian Networks

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



- Each row requires 1 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 full joint distribution
- For our network, 1+1+4+2+2 = 10 numbers (vs. 2^5-1 = 31)

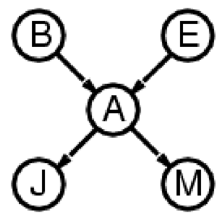
Semantics

Full joint distribution is defined as product of local conditional distributions:

$$\mathbf{P}(X_1, \ldots, X_n) = \pi_{i=1}^{n} \mathbf{P}(X_i \mid Parents(X_i))$$

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

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



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

This choice of parents guarantees:

$\mathbf{P}(X_1, \ldots, X_n) = \mathbf{P}(X_n \mid X_1, \ldots, X_{n-1})\, \mathbf{P}(X_1, \ldots, X_{n-1})$

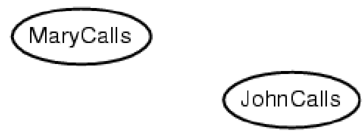
$= \mathbf{P}(X_n \mid X_1, \ldots, X_{n-1})\, \mathbf{P}(X_{n-1} \mid X_1, \ldots, X_{n-2})\, \mathbf{P}(X_1, \ldots, X_{n-2})$

$= \pi_{i=1}^{n} \mathbf{P}(X_i \mid X_1, \ldots, X_{i-1})$ (chain rule)

$= \pi_{i=1}^{n} \mathbf{P}(X_i \mid Parents(X_i))$ (by construction)

Example

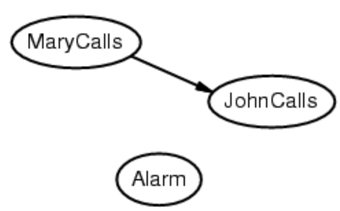
- Suppose we choose the ordering *M, J, A, B, E*



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

Example

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

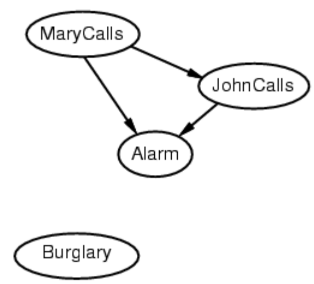


P(J | M) = P(J)? **No**

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

Example

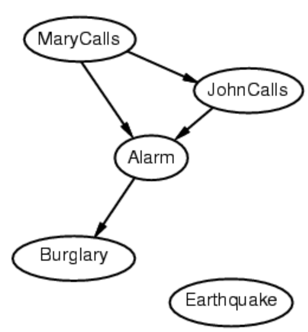
- Suppose we choose the ordering *M, J, A, B, E*



$P(J \mid M) = P(J)$? **No**
$P(A \mid J, M) = P(A \mid J)$? **No** $P(A \mid J, M) = P(A)$? **No**
$P(B \mid A, J, M) = P(B)$?
$P(B \mid A, J, M) = P(B \mid A)$?

Example

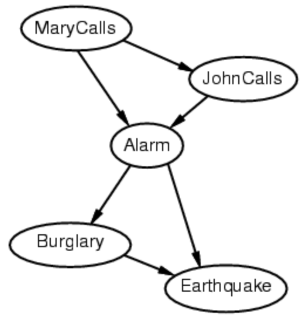
- Suppose we choose the ordering *M, J, A, B, E*



$P(J \mid M) = P(J)$? **No**
$P(A \mid J, M) = P(A \mid J)$? **No** $P(A \mid J, M) = P(A)$? **No**
$P(B \mid A, J, M) = P(B)$? **No**
$P(B \mid A, J, M) = P(B \mid A)$? **Yes**
$P(E \mid B, A, J, M) = P(E \mid A)$?
$P(E \mid B, A, J, M) = P(E \mid A, B)$?

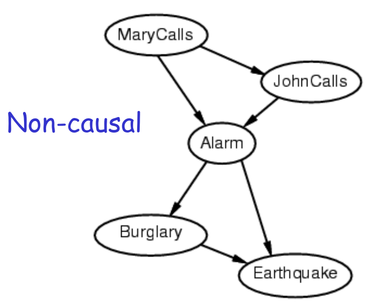
Example

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

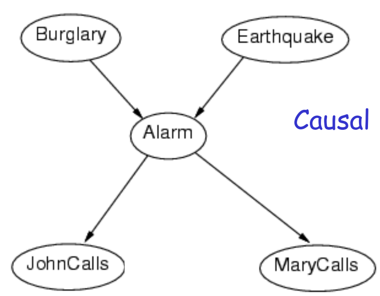


$P(J \mid M) = P(J)$? No
$P(A \mid J, M) = P(A \mid J)$? No $P(A \mid J, M) = P(A)$? No
$P(B \mid A, J, M) = P(B)$? No
$P(B \mid A, J, M) = P(B \mid A)$? Yes
$P(E \mid B, A, J, M) = P(E \mid A)$? No
$P(E \mid B, A, J, M) = P(E \mid A, B)$? Yes

Example contd.



vs.



- Deciding conditional independence is hard in non-causal directions
- Causal models and conditional independence seem hardwired for humans! (recent Cog Sci research)
- Non-causal network is less compact: 1 + 2 + 4 + 2 + 4 = 13 numbers (vs. 1+1+4+2+2 = 10 numbers)

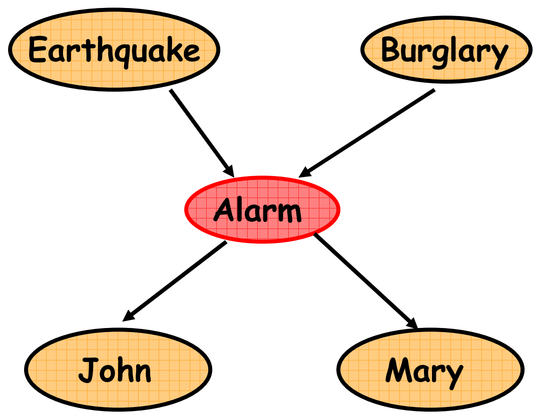
Lesson: Add nodes representing "root causes" first, then the variables they influence, and so on.

Keep it causal, baby!

Probabilistic Inference in BNs

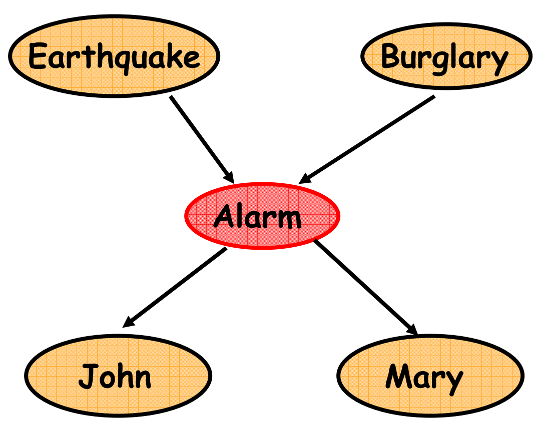
- The graphical independence representation yields efficient inference schemes
- We generally want to compute
  - *P(X|**E**)* where ***E*** is evidence from sensory measurements etc. (known values for variables)
  - Sometimes, may want to compute just *P(X)*
- One simple algorithm:
  - *variable elimination (VE)*

P(B | J=true, M=true)



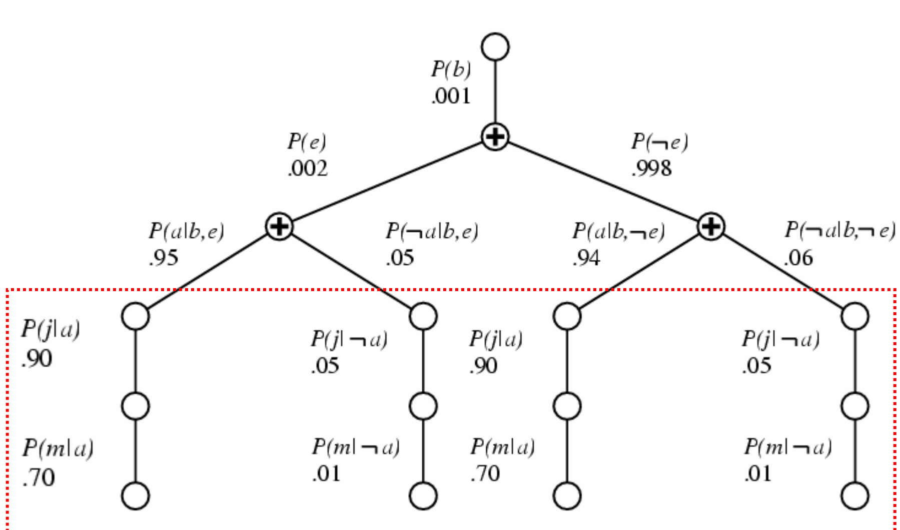
$$P(b|j,m) = \alpha \sum_{e,a} P(b,j,m,e,a)$$

P(B | J=true, M=true)



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

Structure of Computation



Repeated computations ⇒ use dynamic programming?

Variable Elimination

- A *factor* is a function from some set of variables to a specific value: e.g., *f(E,A,Mary)*
  CPTs are factors, e.g., *P(A|E,B)* function of *A,E,B*

- VE works by *eliminating* all variables in turn until there is a factor with only query variable

- To eliminate a variable:
  1. *join* all factors containing that variable (like DBs/SQL), multiplying probabilities
  2. *sum out* the influence of the variable on new factor

Example of VE: P(J)

$P(J)$

$= \Sigma_{M,A,B,E}\ P(J,M,A,B,E)$

$= \Sigma_{M,A,B,E}\ P(J|A)P(M|A)\ P(B)P(A|B,E)P(E)$

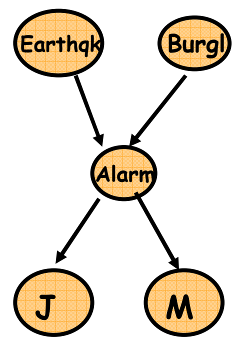
$= \Sigma_A P(J|A)\ \Sigma_M P(M|A)\ \Sigma_B P(B)\ \Sigma_E P(A|B,E)P(E)$

$= \Sigma_A P(J|A)\ \Sigma_M P(M|A)\ \Sigma_B P(B)\ f1(A,B)$

$= \Sigma_A P(N1|A)\ \Sigma_M P(M|A)\ f2(A)$

$= \Sigma_A P(J|A)\ f3(A)$

$= f4(J)$



Other Inference Algorithms

- Direct Sampling:

  Repeat N times:
  - Use random number generator to generate sample values for each node
  - Start with nodes with no parents
  - Condition on sampled parent values for other nodes

  Count frequencies of samples to get an approximation to joint distribution
- Other variants: Rejection sampling, likelihood weighting, Gibbs sampling and other MCMC methods (see text)
- Belief Propagation: A "message passing" algorithm for approximating P(X|evidence) for each node variable X
- Variational Methods: Approximate inference using distributions that are more tractable than original ones

Summary

- Bayesian networks provide a natural way to represent conditional independence
- Network topology + CPTs = compact representation of joint distribution
- Generally easy for domain experts to construct
- BNs allow inference algorithms such as VE that are efficient in many cases

Next Time

- Machine Learning!