Computing with Probabilities: Law of Total Probability

Law of Total Probability (aka "summing out" or marginalization)

$$P(a) = \Sigma_b \; P(a, b)$$
$$= \Sigma_b \; P(a \mid b) \, P(b) \qquad \text{where B is any random variable}$$

Why is this useful?

given a joint distribution (e.g., P(a,b,c,d)) we can obtain any "marginal" probability (e.g., P(b)) by summing out the other variables, e.g.,

$$P(b) = \Sigma_a \, \Sigma_c \, \Sigma_d \; P(a, b, c, d)$$

Less obvious: we can also compute <u>any conditional probability of interest</u> given a joint distribution, e.g.,

$$P(c \mid b) = \Sigma_a \, \Sigma_d \; P(a, c, d \mid b)$$
$$= 1 \,/\, P(b) \;\; \Sigma_a \, \Sigma_d \; P(a, c, d, b)$$

where $1 / P(b)$ is just a normalization constant

Thus, the joint distribution contains the information we need to compute any probability of interest.

We can always write

$$P(a, b, c, \ldots z) = P(a \mid b, c, \ldots. z)\, P(b, c, \ldots z)$$

(by definition of joint probability)

Repeatedly applying this idea, we can write

$$P(a, b, c, \ldots z) = P(a \mid b, c, \ldots. z)\, P(b \mid c, .. z)\, P(c \mid .. z) .. P(z)$$

This factorization holds for any ordering of the variables

This is the chain rule for probabilities

- 2 random variables A and B are conditionally independent given C iff

  $$P(a, b \mid c) = P(a \mid c)\, P(b \mid c) \quad \text{for all values } a, b, c$$

- More intuitive (equivalent) conditional formulation
  - A and B are conditionally independent given C iff

    $$P(a \mid b, c) = P(a \mid c) \quad \text{OR} \quad P(b \mid a, c)\, P(b \mid c), \quad \text{for all values } a, b, c$$

  - Intuitive interpretation:

    $P(a \mid b, c) = P(a \mid c)$ tells us that learning about b, given that we already know c, provides no change in our probability for a,

    i.e., b contains no information about a beyond what c provides

- Can generalize to more than 2 random variables
  - E.g., K different symptom variables X1, X2, … XK, and C = disease
  - $P(X1, X2, \ldots XK \mid C) = \prod P(Xi \mid C)$
  - Also known as the naïve Bayes assumption

Bayesian Networks

- A Bayesian network specifies a joint distribution in a structured form
- Represent dependence/independence via a directed graph
  - Nodes = random variables
  - Edges = direct dependence
- Structure of the graph ⇔ Conditional independence relations

In general,

$$p(X_1, X_2, \ldots X_N) = \prod p(X_i \mid \text{parents}(X_i))$$

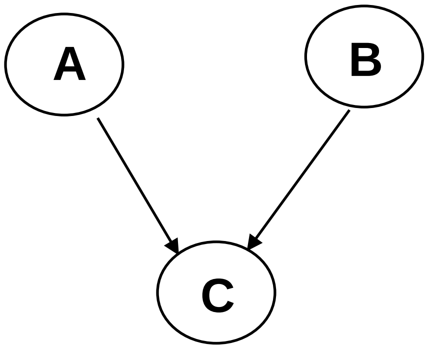
The full joint distribution

The graph-structured approximation

- Requires that graph is acyclic (no directed cycles)
- 2 components to a Bayesian network
  - The graph structure (conditional independence assumptions)
  - The numerical probabilities (for each variable given its parents)

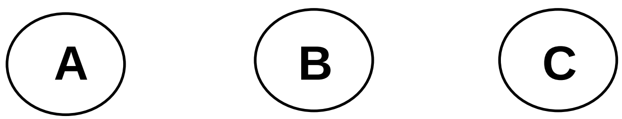
Example of a simple Bayesian network

$$p(A,B,C) = p(C|A,B)p(A)p(B) \longleftrightarrow$$



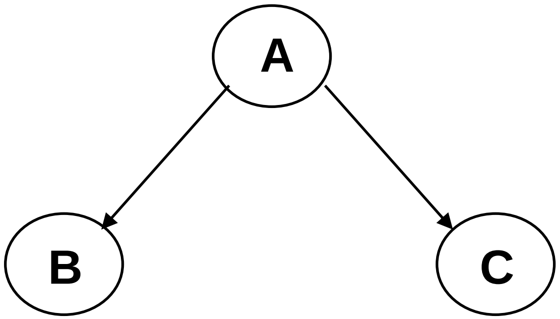
- Probability model has simple factored form
- Directed edges => direct dependence
- Absence of an edge => conditional independence
- Also known as belief networks, graphical models, causal networks
- Other formulations, e.g., undirected graphical models

Examples of 3-way Bayesian Networks



Marginal Independence:
p(A,B,C) = p(A) p(B) p(C)

Examples of 3-way Bayesian Networks

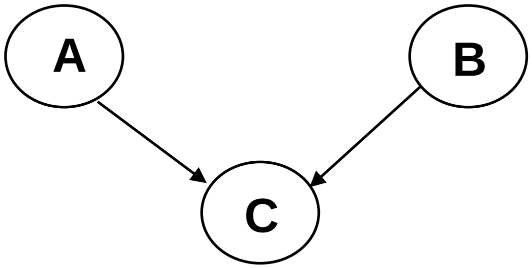


Conditionally independent effects:
p(A,B,C) = p(B|A)p(C|A)p(A)

B and C are conditionally independent
Given A

e.g., A is a disease, and we model
B and C as conditionally independent
symptoms given A

Examples of 3-way Bayesian Networks

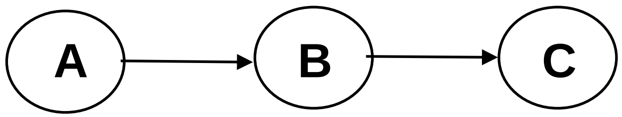


Independent Causes:
$p(A,B,C) = p(C|A,B)p(A)p(B)$

"Explaining away" effect:
Given C, observing A makes B less likely
e.g., earthquake/burglary/alarm example

A and B are (marginally) independent
but become dependent once C is known

Examples of 3-way Bayesian Networks



Markov dependence:
p(A,B,C) = p(C|B) p(B|A)p(A)

Example

- Consider the following 5 binary variables:
  - B = a burglary occurs at your house
  - E = an earthquake occurs at your house
  - A = the alarm goes off
  - J = John calls to report the alarm
  - M = Mary calls to report the alarm

  - What is P(B | M, J) ? (for example)

  - We can use the full joint distribution to answer this question
    - Requires $2^5 = 32$ probabilities

    - Can we use prior domain knowledge to come up with a Bayesian network that requires fewer probabilities?

Constructing a Bayesian Network: Step 1

- Order the variables in terms of causality (may be a partial order)

  e.g., {E, B} -> {A} -> {J, M}

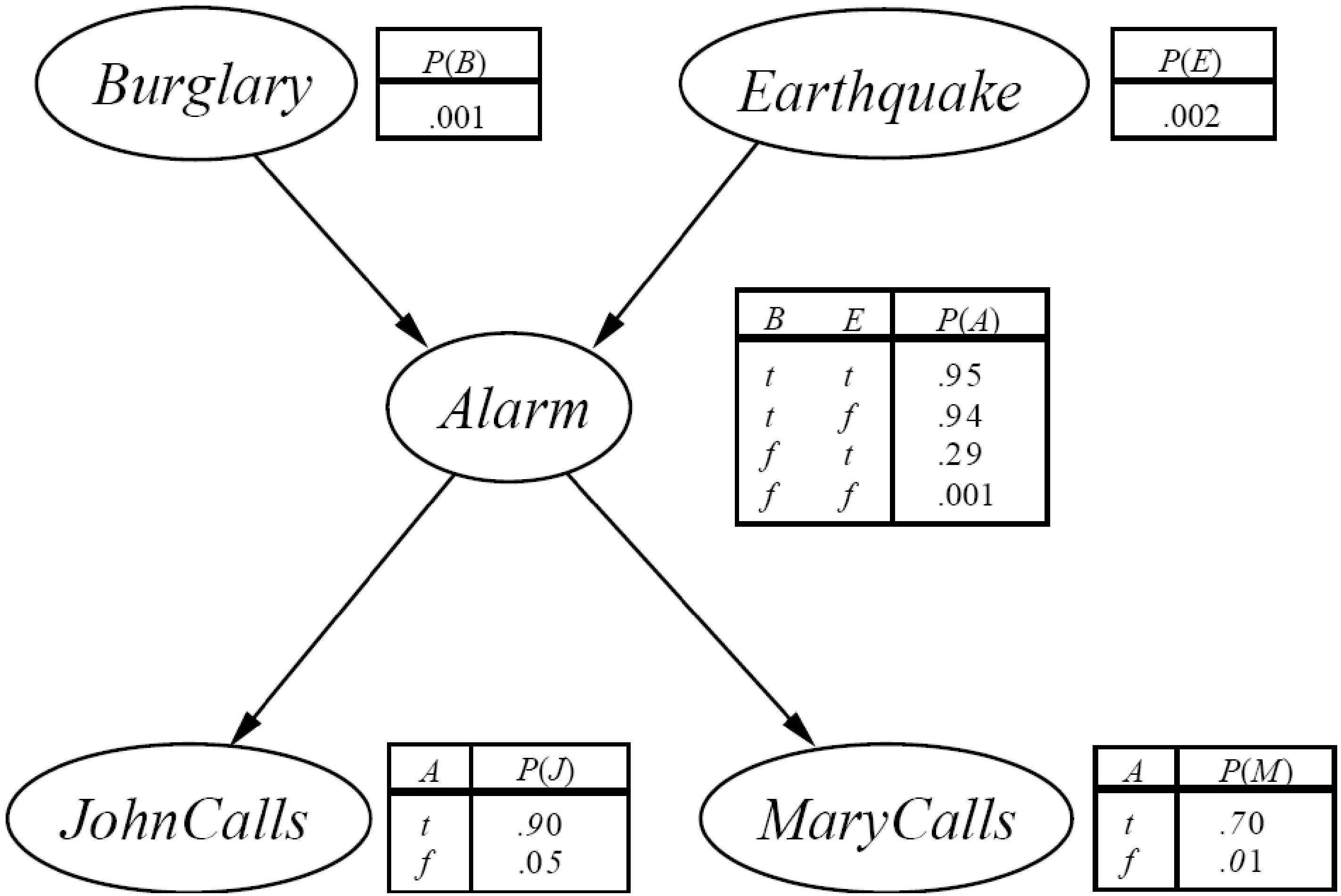
- P(J, M, A, E, B) = P(J, M | A, E, B) P(A| E, B) P(E, B)

  ~ P(J, M | A) P(A| E, B) P(E) P(B)

  ~ P(J | A) P(M | A) P(A| E, B) P(E) P(B)

These CI assumptions are reflected in the graph structure of the Bayesian network

The Resulting Bayesian Network



$P(B)$
.001

$P(E)$
.002

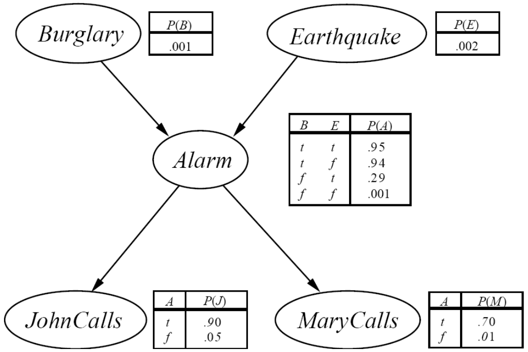
B	E	$P(A)$
t	t	.95
t	f	.94
f	t	.29
f	f	.001

A	$P(J)$
t	.90
f	.05

A	$P(M)$
t	.70
f	.01

Constructing this Bayesian Network: Step 2

- P(J, M, A, E, B) =
  P(J | A) P(M | A) P(A | E, B) P(E) P(B)

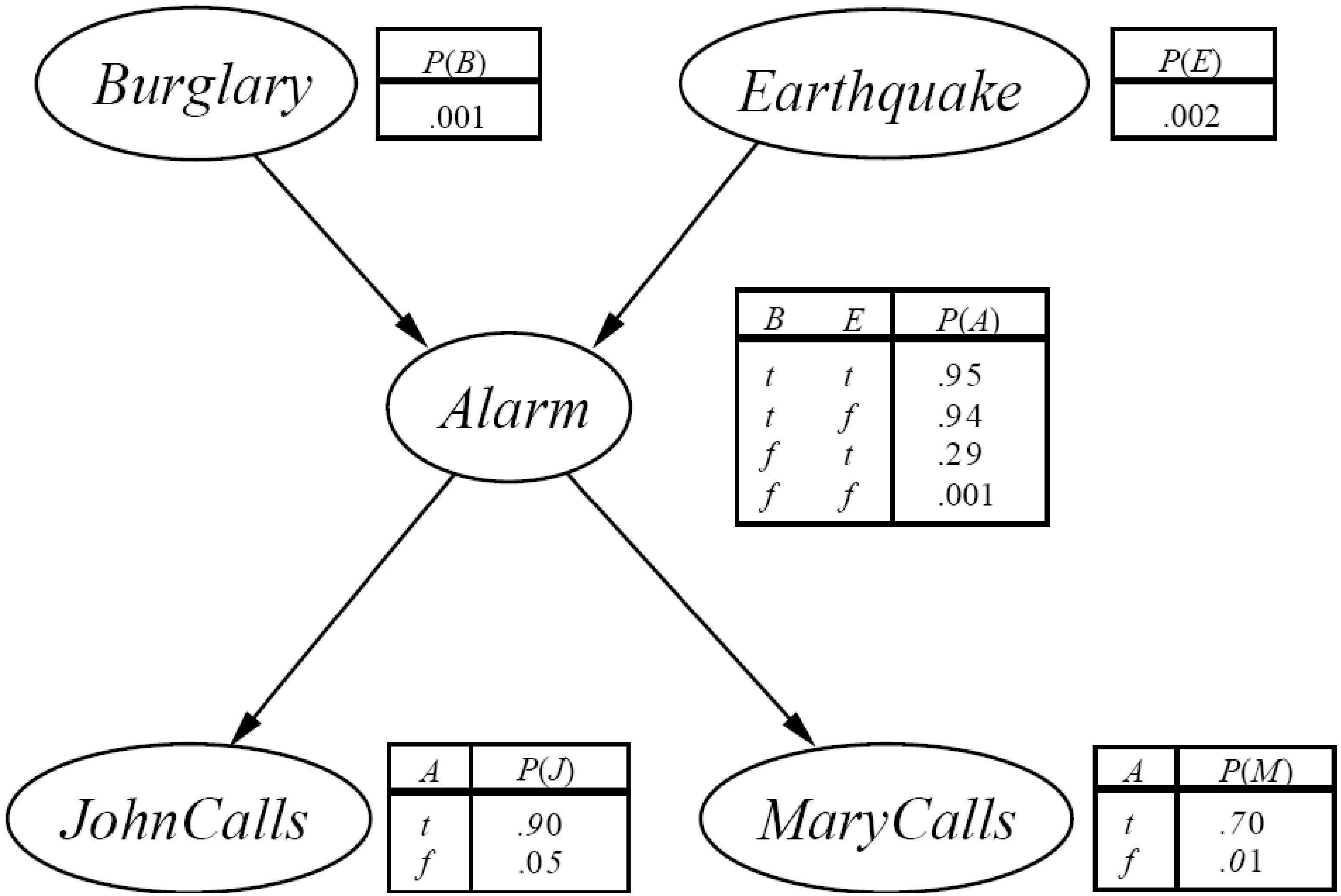


- There are 3 conditional probability tables (CPDs) to be determined: P(J | A), P(M | A), P(A | E, B)
  - Requiring 2 + 2 + 4 = 8 probabilities

- And 2 marginal probabilities P(E), P(B) -> 2 more probabilities

- Where do these probabilities come from?
  - Expert knowledge
  - From data (relative frequency estimates)
  - Or a combination of both - see discussion in Section 20.1 and 20.2 (optional)

The Bayesian network



$P(B)$
.001

$P(E)$
.002

B	E	$P(A)$
t	t	.95
t	f	.94
f	t	.29
f	f	.001

A	$P(J)$
t	.90
f	.05

A	$P(M)$
t	.70
f	.01

Number of Probabilities in Bayesian Networks

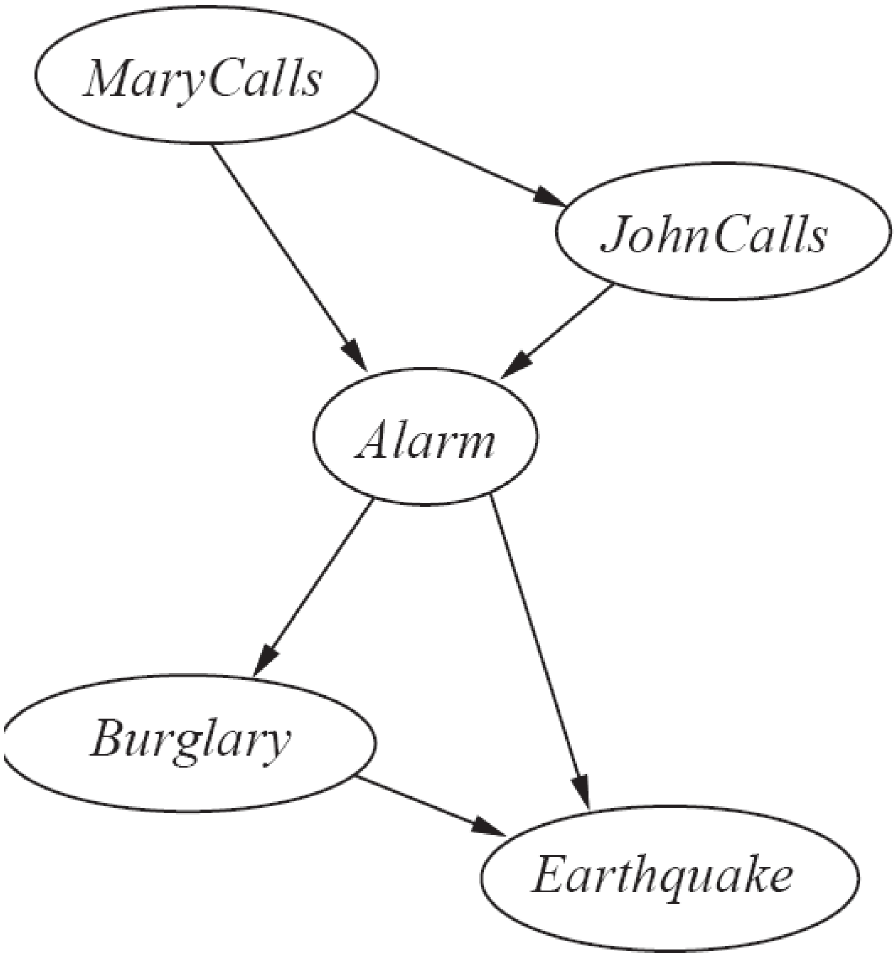
- Consider n binary variables

- Unconstrained joint distribution requires $O(2^n)$ probabilities

- If we have a Bayesian network, with a maximum of k parents for any node, then we need $O(n\ 2^k)$ probabilities

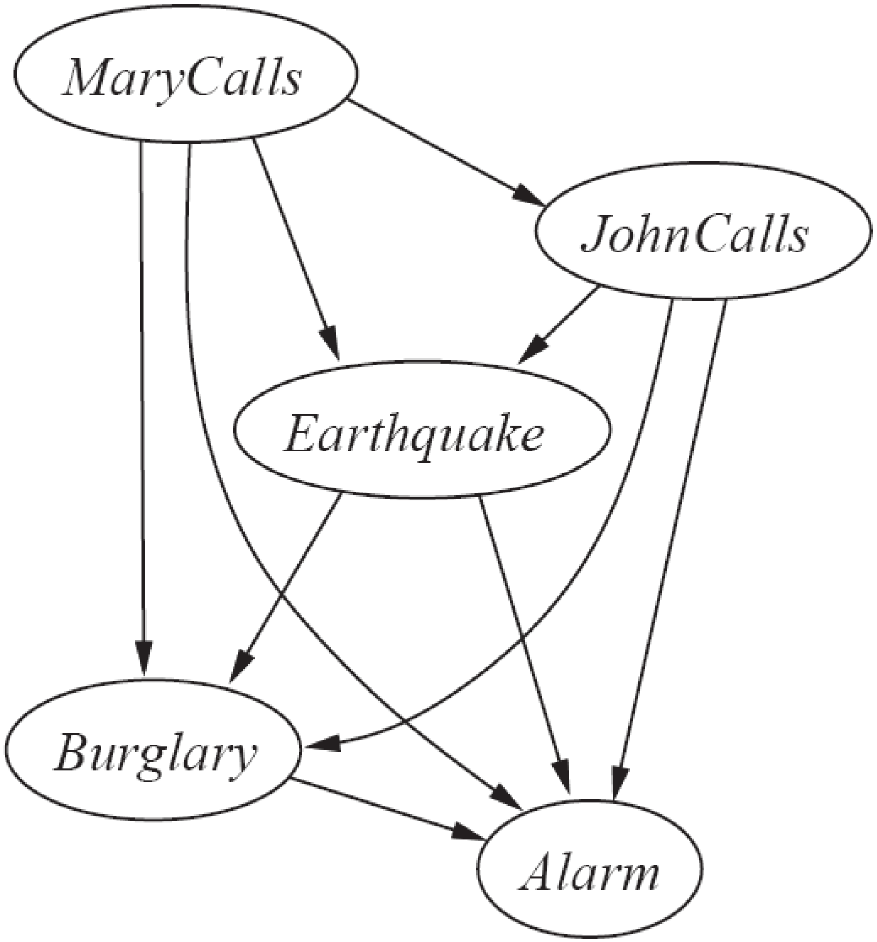
- Example
  - Full unconstrained joint distribution
    - $n = 30$: need 10^9 probabilities for full joint distribution
  - Bayesian network
    - $n = 30$, $k = 4$: need 480 probabilities

The Bayesian Network from a different Variable Ordering



(a)

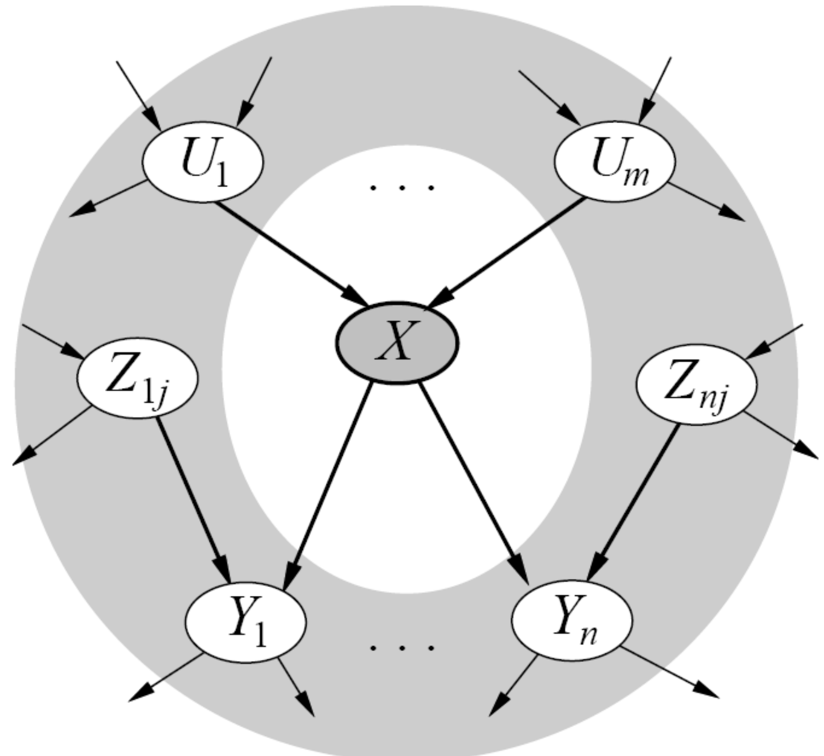
The Bayesian Network from a different Variable Ordering



(b)

Given a graph, can we “read off” conditional independencies?

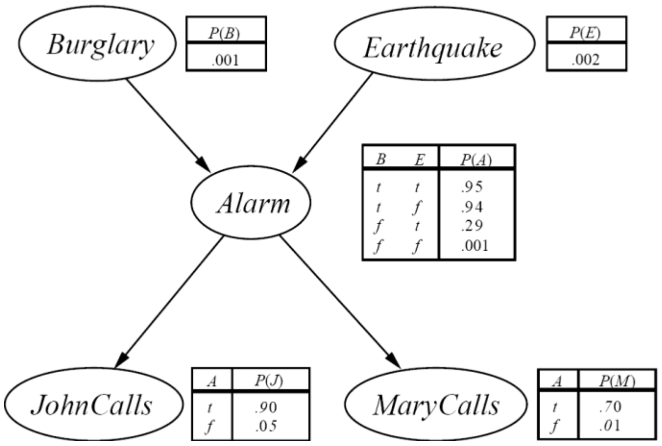
A node is conditionally independent of all other nodes in the network given its Markov blanket (in gray)



Inference (Reasoning) in Bayesian Networks

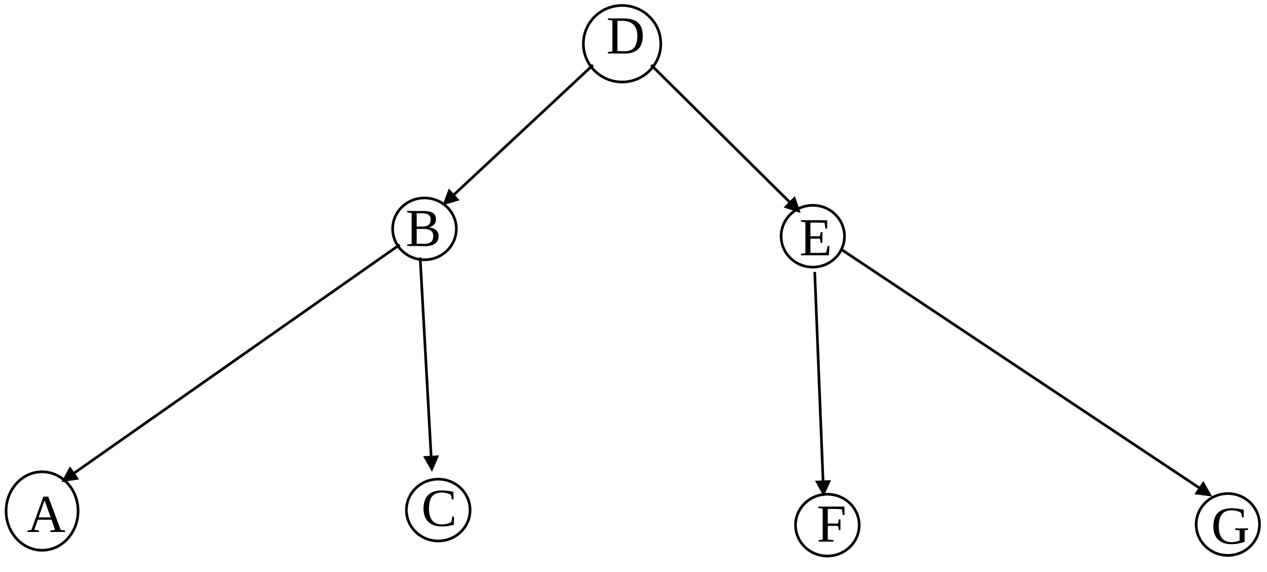
- Consider answering a query in a Bayesian Network
  - Q = set of query variables
  - e = evidence (set of instantiated variable-value pairs)
  - Inference = computation of conditional distribution P(Q | e)

- Examples
  - P(burglary | alarm)
  - P(earthquake | JCalls, MCalls)
  - P(JCalls, MCalls | burglary, earthquake)



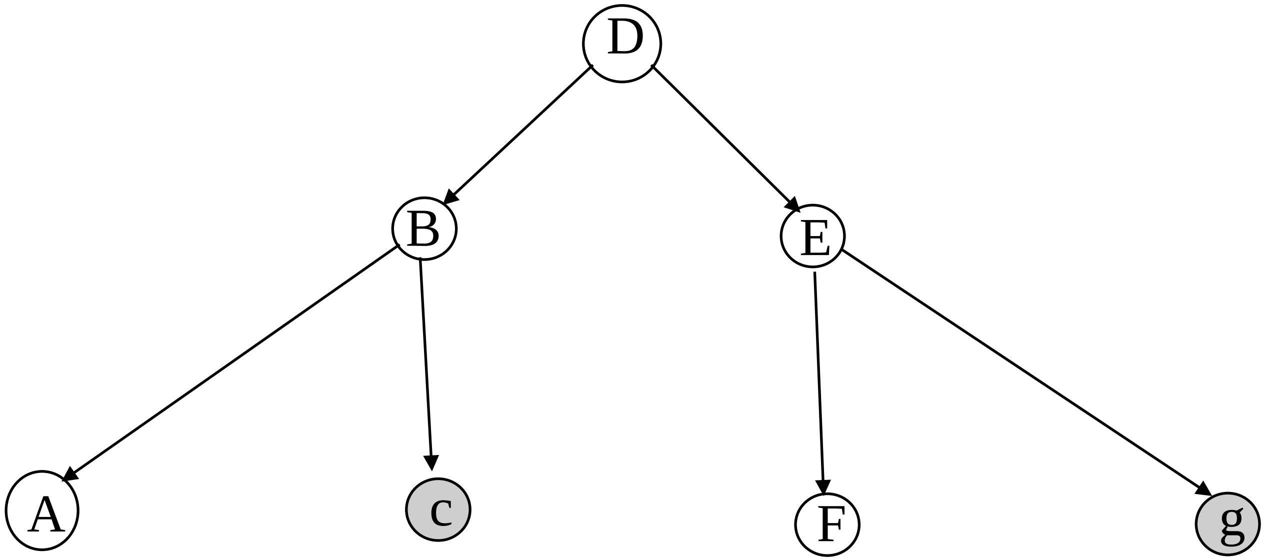
- Can we use the structure of the Bayesian Network to answer such queries efficiently? Answer = yes
  - Generally speaking, complexity is inversely proportional to sparsity of graph

Example: Tree-Structured Bayesian Network



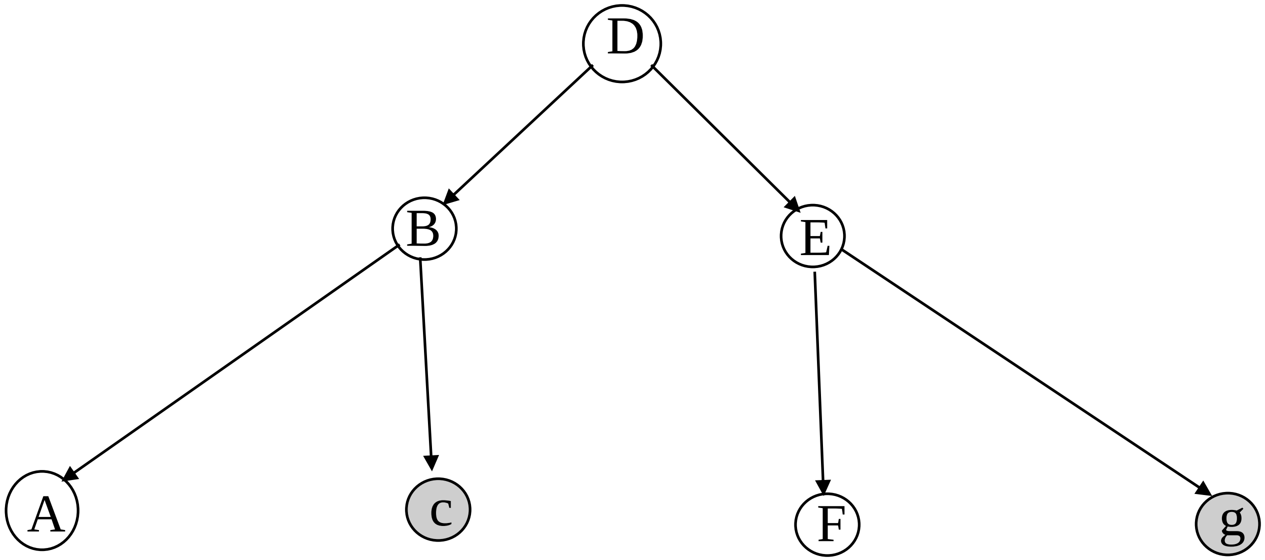
p(a, b, c, d, e, f, g) is modeled as p(a|b)p(c|b)p(f|e)p(g|e)p(b|d)p(e|d)p(d)

Example



Say we want to compute p(a | c, g)

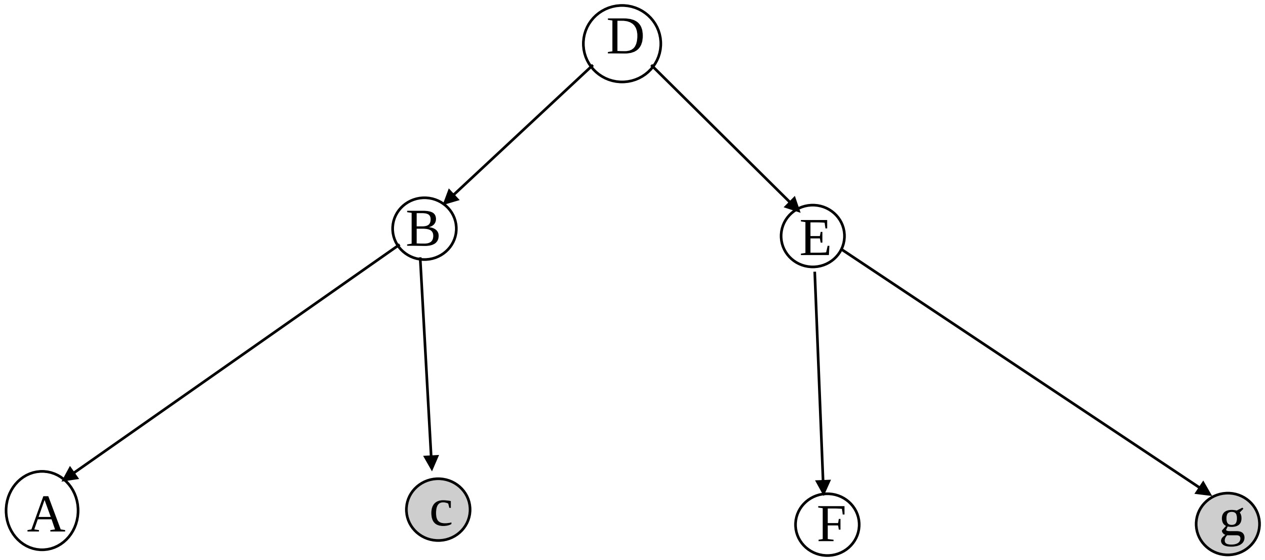
Example



Direct calculation: $p(a|c,g) = \Sigma_{bdef}\, p(a,b,d,e,f \mid c,g)$

Complexity of the sum is $O(m^4)$

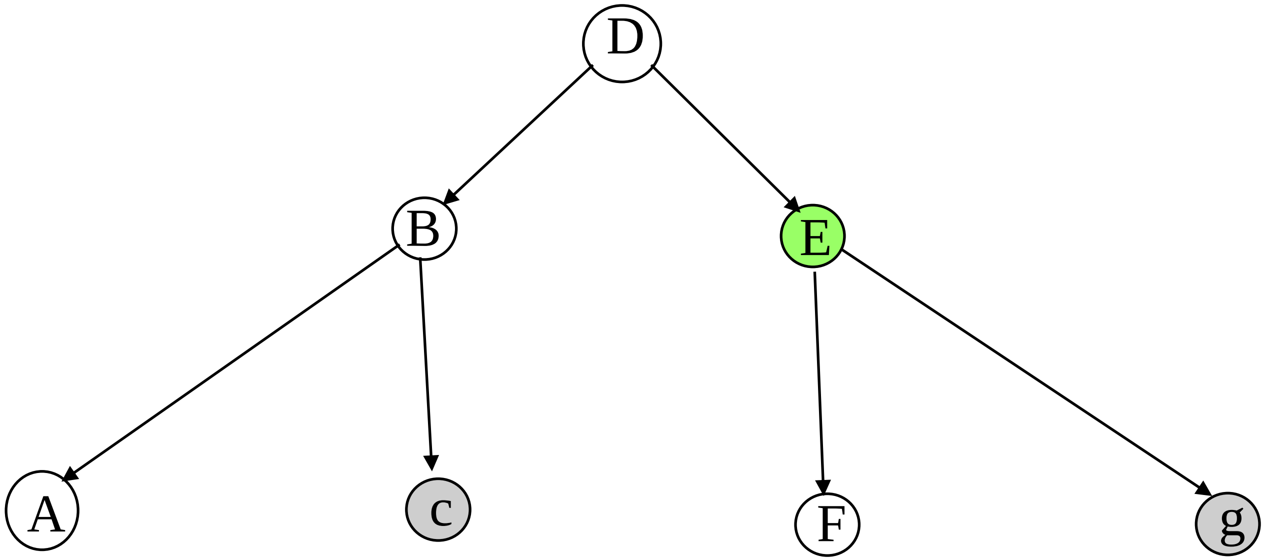
Example



Reordering:

$$\Sigma_d\ p(a|b)\ \Sigma_d\ p(b|d,c)\ \Sigma_e\ p(d|e)\ \Sigma_f\ p(e,f\ |g)$$

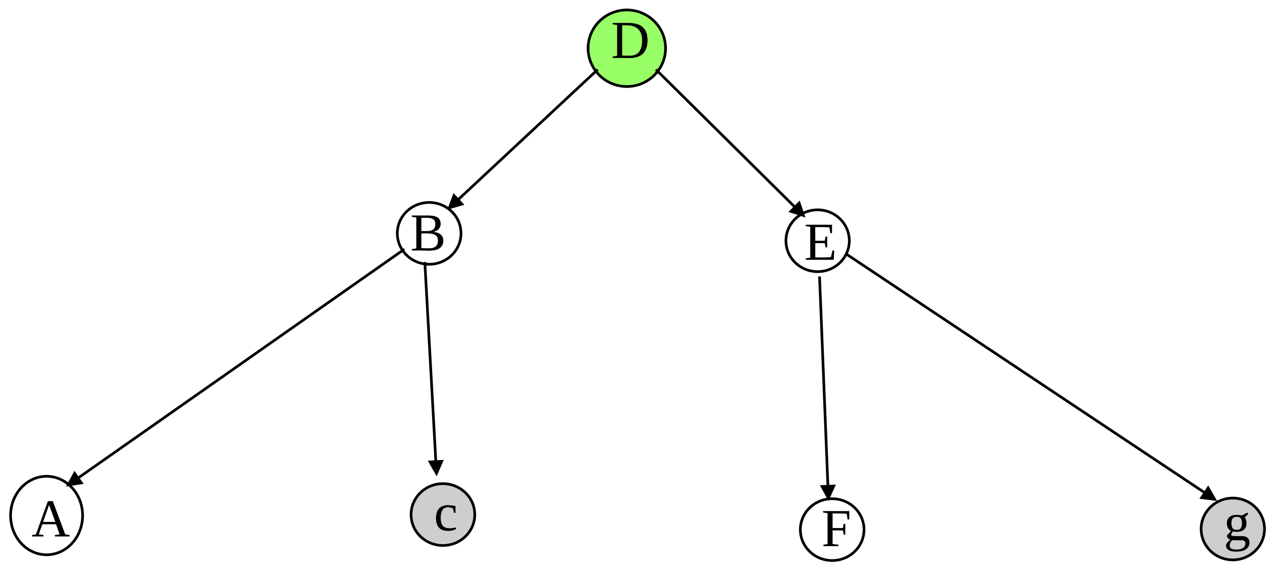
Example



Reordering:

$$\Sigma_b \; p(a|b) \; \Sigma_d \; p(b|d,c) \; \Sigma_e \; p(d|e) \; \underbrace{\Sigma_f \; p(e,f \,|g)}_{p(e|g)}$$

Example

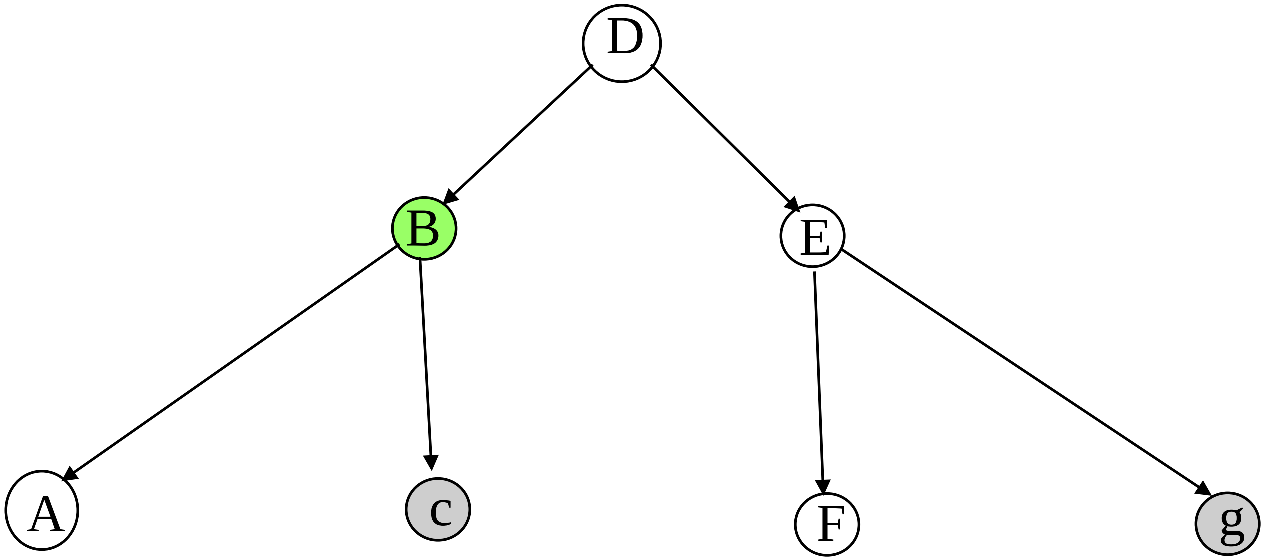


Reordering:

$$\Sigma_b\, p(a|b)\, \Sigma_d\, p(b|d,c)\, \Sigma_e\, p(d|e)\, p(e|g)$$

$p(d|g)$

Example

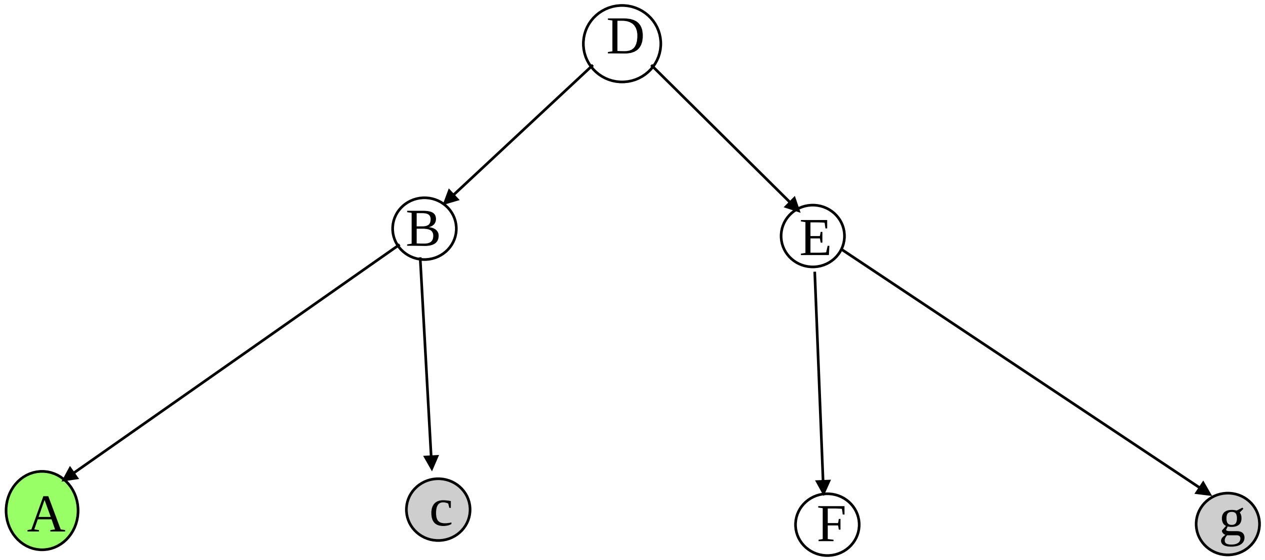


Reordering:

$$\Sigma_b\ p(a|b)\ \Sigma_d\ p(b|d,c)\ p(d|g)$$

$p(b|c,g)$

Example



Reordering:

$\sum_b p(a|b)\, p(b|c,g)$

$p(a|c,g)$

Complexity is O(m), compared to O(m^4)

General Strategy for inference

- Want to compute P(q | e)

Step 1:

$P(q \mid e) = P(q,e)/P(e) = \alpha\, P(q,e)$, since P(e) is constant wrt Q

Step 2:

$P(q,e) = \Sigma_{a..z}\, P(q, e, a, b, \ldots.\, z)$, by the law of total probability

Step 3:

$\Sigma_{a..z}\, P(q, e, a, b, \ldots.\, z) = \Sigma_{a..z}\, \Pi_i\, P(\text{variable } i \mid \text{parents } i)$

(using Bayesian network factoring)

Step 4:

Distribute summations across product terms for efficient computation

Inference Examples

- Examples worked on whiteboard

Complexity of Bayesian Network inference

- Assume the network is a polytree
  - Only a single directed path between any 2 nodes
- Complexity scales as $O(n\ m^{K+1})$
  - n = number of variables
  - m = arity of variables
  - K = maximum number of parents for any node
  - Compare to $O(m^{n-1})$ for brute-force method
- Network is not a polytree?
  - Can cluster variables to render the new graph a tree
  - Very similar to tree methods used for
  - Complexity is $O(n\ m^{W+1})$, where W = num variables in largest cluster