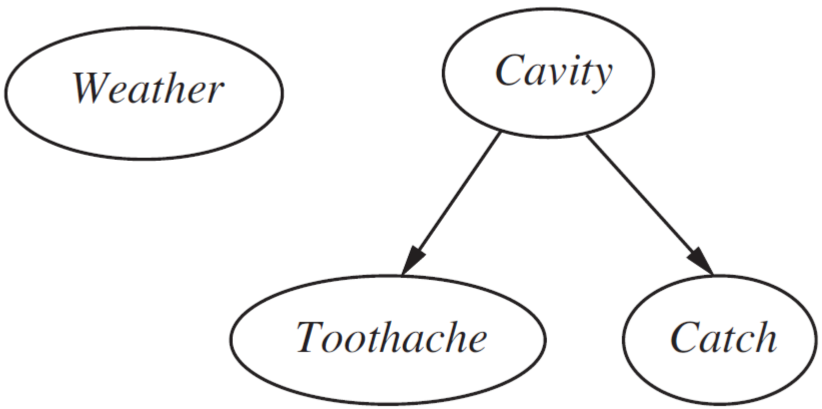
Bayesian Networks

Contents

The Syntax of Bayesian Networks

Example:



- *Weather* is independent of the other variables
- *Toothache* and *Catch* are conditionally independent given *Cavity*

$$\mathbf{P}(Cavity, Toothache, Catch)$$
$$= \mathbf{P}(Cavity)\,\mathbf{P}(Toothache \mid Cavity)\,\mathbf{P}(Catch \mid Cavity, Toothache)$$
$$= \mathbf{P}(Cavity)\,\mathbf{P}(Toothache \mid Cavity)\,\mathbf{P}(Catch \mid Cavity)$$

The Syntax of Bayesian Networks

- A Bayesian network is a simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions
- Syntax:
  - A set of nodes, one per variable
  - A directed, acyclic graph (link ≈ "directly influences")
  - A conditional distribution for each node given its parents:

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

- In the simplest case, conditional distribution is represented as a conditional probability table (CPT) giving the distribution over X_i for each combination of parent values
- Topology of network encodes conditional independence assertions

The Syntax of Bayesian Networks

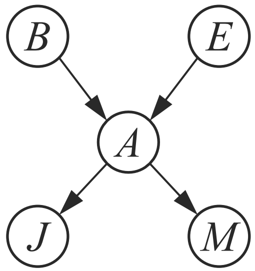
Example:

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: $Burglar$, $Earthquake$, $Alarm$, $JohnCalls$, $MaryCalls$

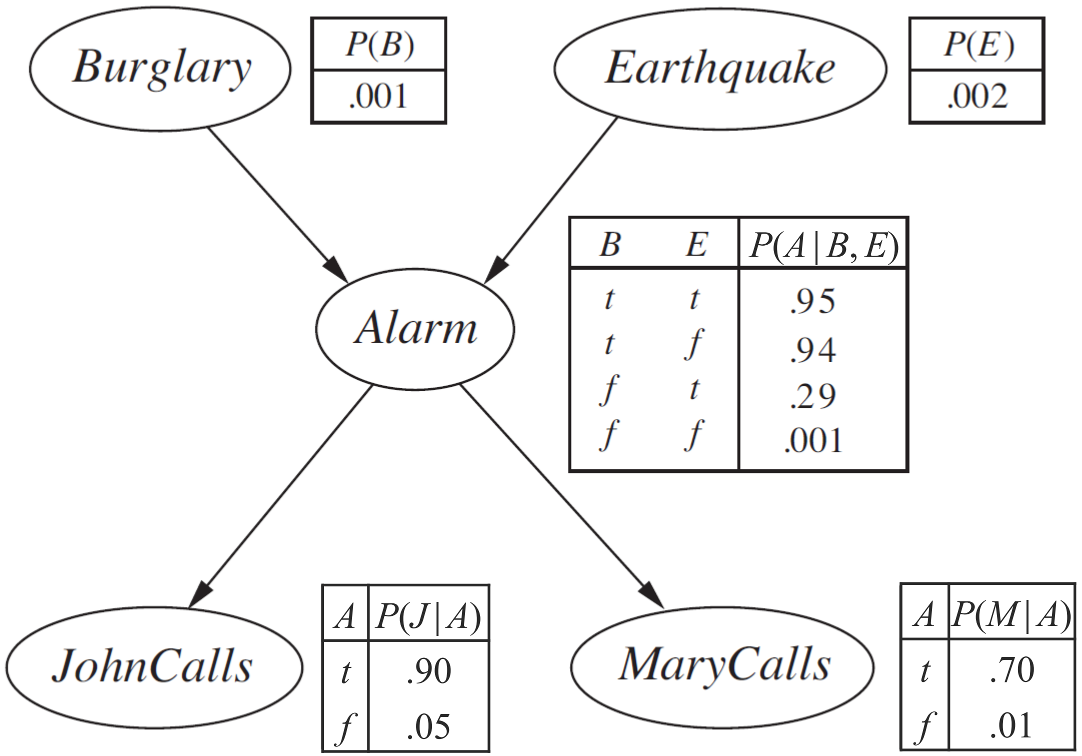
Network topology reflects "causal" knowledge:

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



The Syntax of Bayesian Networks

Example:



$P(B)$
.001

$P(E)$
.002

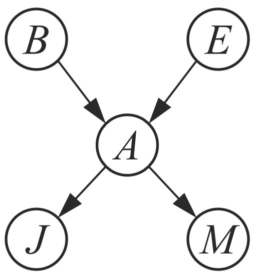
B	E	$P(A \mid B, E)$
t	t	.95
t	f	.94
f	t	.29
f	f	.001

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

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

The Syntax of Bayesian Networks

- Compactness:
  - A CPT for Boolean X_i with k Boolean parents has 2^k rows for the combinations of parent values
  - Each row requires one 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 the full joint distribution



  - For burglary net, $1 + 1 + 4 + 2 + 2 = 10$ numbers (vs. $2^5 - 1 = 31$)

The Semantics of Bayesian Networks

- "Numerical" semantics defines the full joint distribution as the product of the local conditional distributions:

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

(by chain rule & conditional independence)

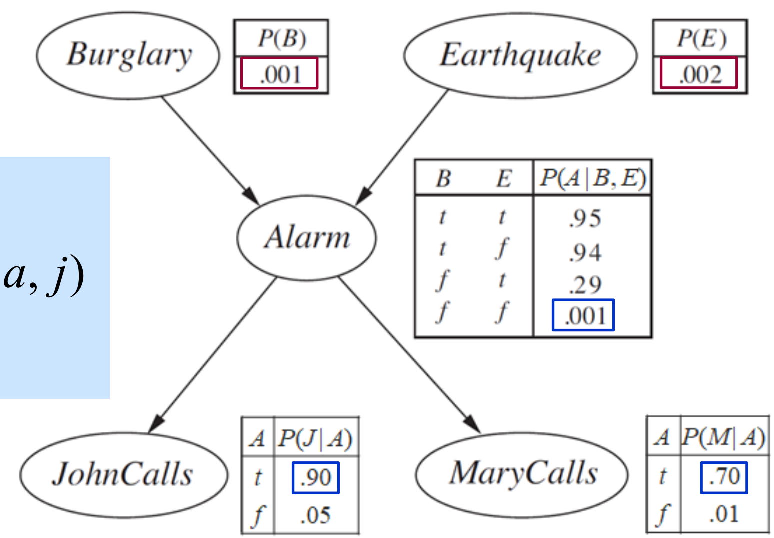
e.g., $P(j \wedge m \wedge a \wedge \neg b \wedge \neg e)$

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

$$= 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998$$

$$\approx 0.00063$$

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



Constructing Bayesian Networks

- Need a method such that a series of locally testable assertions of conditional independence guarantees the required numerical semantics

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

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

$$\mathbf{P}(X_1, X_2, X_3, X_4) = \mathbf{P}(X_1)\,\mathbf{P}(X_2 \mid X_1)\,\mathbf{P}(X_3 \mid X_1, X_2)\,\mathbf{P}(X_4 \mid X_1, X_2, X_3)$$

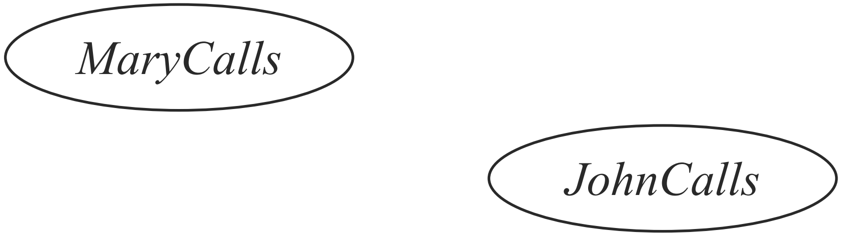
This choice of parents guarantees the numerical semantics:

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

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

Constructing Bayesian Networks

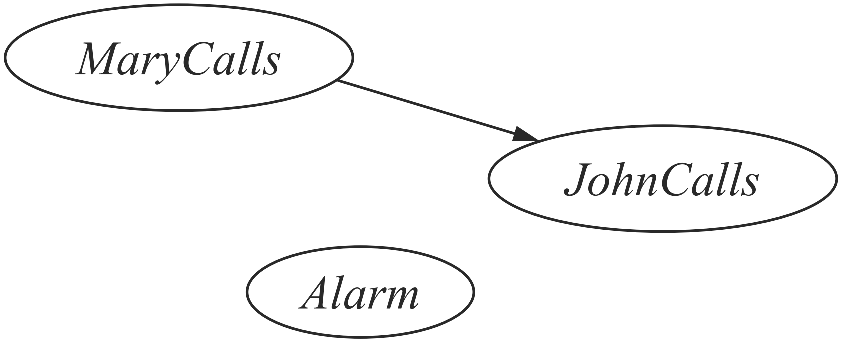
Example: Suppose we choose the ordering M, J, A, B, E



$\mathbf{P}(J \mid M) = \mathbf{P}(J)$?

Constructing Bayesian Networks

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

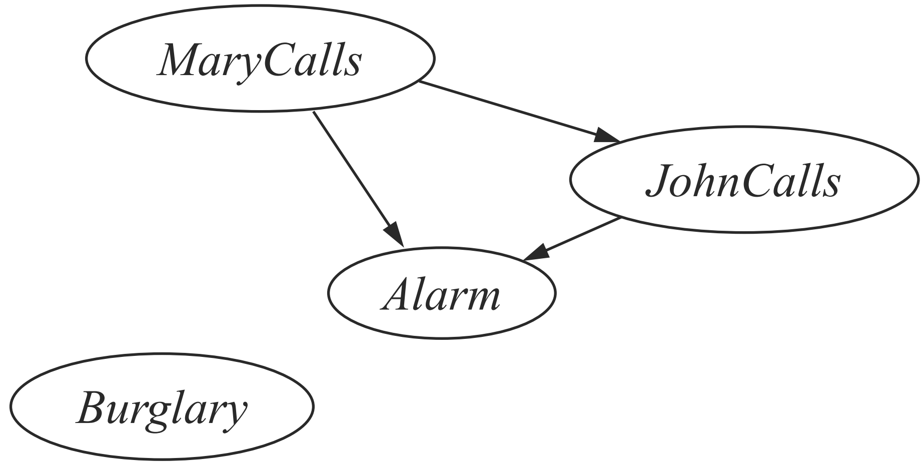


$\mathbf{P}(J \mid M) = \mathbf{P}(J)$? No

$\mathbf{P}(A \mid J, M) = \mathbf{P}(A \mid J)$? $\mathbf{P}(A \mid J, M) = \mathbf{P}(A)$?

Constructing Bayesian Networks

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



$\mathbf{P}(J \mid M) = \mathbf{P}(J)$? No

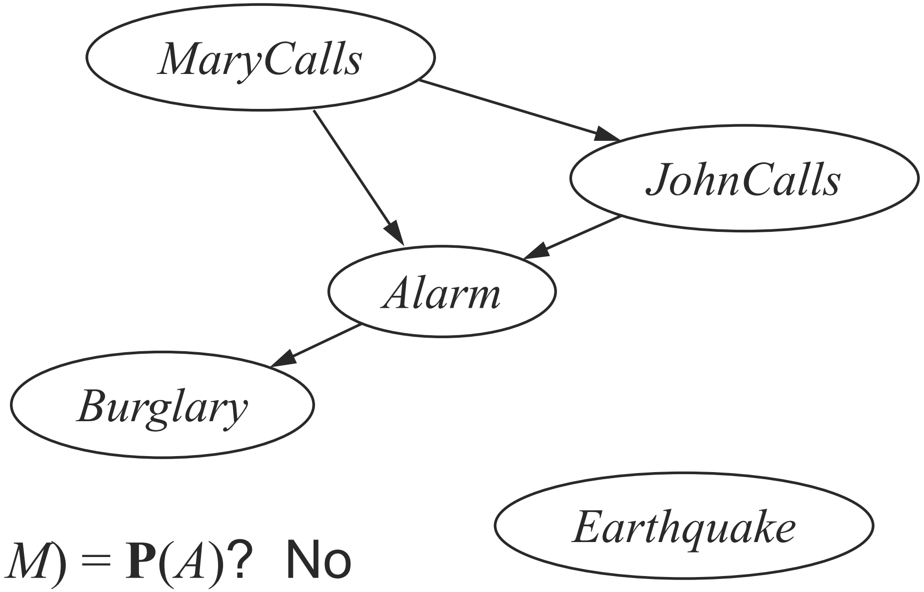
$\mathbf{P}(A \mid J, M) = \mathbf{P}(A \mid J)$? $\mathbf{P}(A \mid J, M) = \mathbf{P}(A)$? No

$\mathbf{P}(B \mid A, J, M) = \mathbf{P}(B \mid A)$?

$\mathbf{P}(B \mid A, J, M) = \mathbf{P}(B)$?

Constructing Bayesian Networks

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



$\mathbf{P}(J \mid M) = \mathbf{P}(J)$? No

$\mathbf{P}(A \mid J, M) = \mathbf{P}(A \mid J)$? $\mathbf{P}(A \mid J, M) = \mathbf{P}(A)$? No

$\mathbf{P}(B \mid A, J, M) = \mathbf{P}(B \mid A)$? Yes

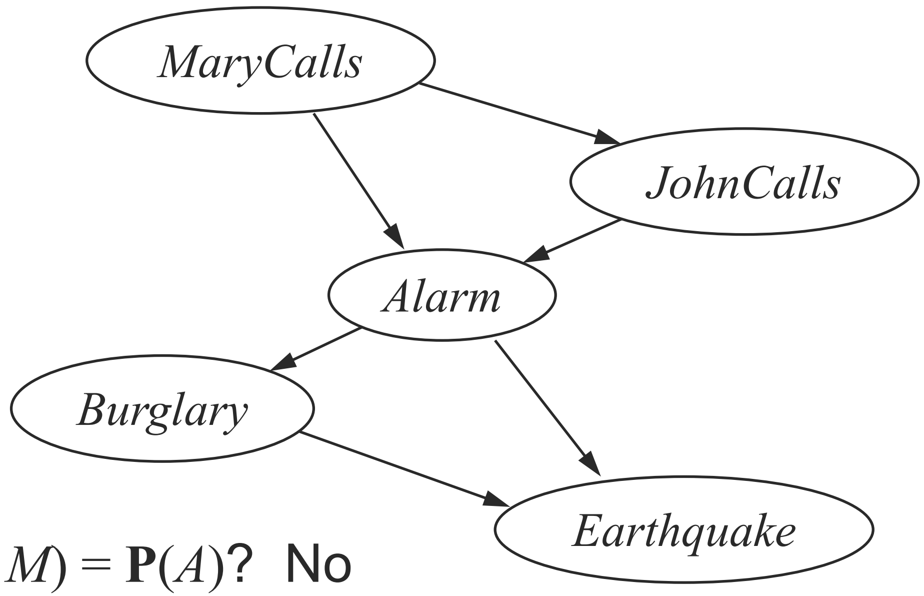
$\mathbf{P}(B \mid A, J, M) = \mathbf{P}(B)$? No

$\mathbf{P}(E \mid B, A, J, M) = \mathbf{P}(E \mid A)$?

$\mathbf{P}(E \mid B, A, J, M) = \mathbf{P}(E \mid A, B)$?

Constructing Bayesian Networks

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



$\mathbf{P}(J \mid M) = \mathbf{P}(J)$? No

$\mathbf{P}(A \mid J, M) = \mathbf{P}(A \mid J)$? $\mathbf{P}(A \mid J, M) = \mathbf{P}(A)$? No

$\mathbf{P}(B \mid A, J, M) = \mathbf{P}(B \mid A)$? Yes

$\mathbf{P}(B \mid A, J, M) = \mathbf{P}(B)$? No

$\mathbf{P}(E \mid B, A, J, M) = \mathbf{P}(E \mid A)$? No

$\mathbf{P}(E \mid B, A, J, M) = \mathbf{P}(E \mid A, B)$? Yes

Constructing Bayesian Networks

- Deciding conditional independence is hard in noncausal directions
  - Causal models and conditional independence seem hardwired for humans!
  - Assessing conditional probabilities is hard in noncausal directions
  - Network is less compact: $1 + 2 + 4 + 2 + 4 = 13$ numbers needed

$$\begin{aligned} P(m,j,a,b,e) &= P(m)P(j \mid m)P(a \mid j,m)P(b \mid a,j,m)P(e \mid b,a,j,m) \\ &= P(m)P(j \mid m)P(a \mid j,m)P(b \mid a)P(e \mid b,a) \end{aligned}$$

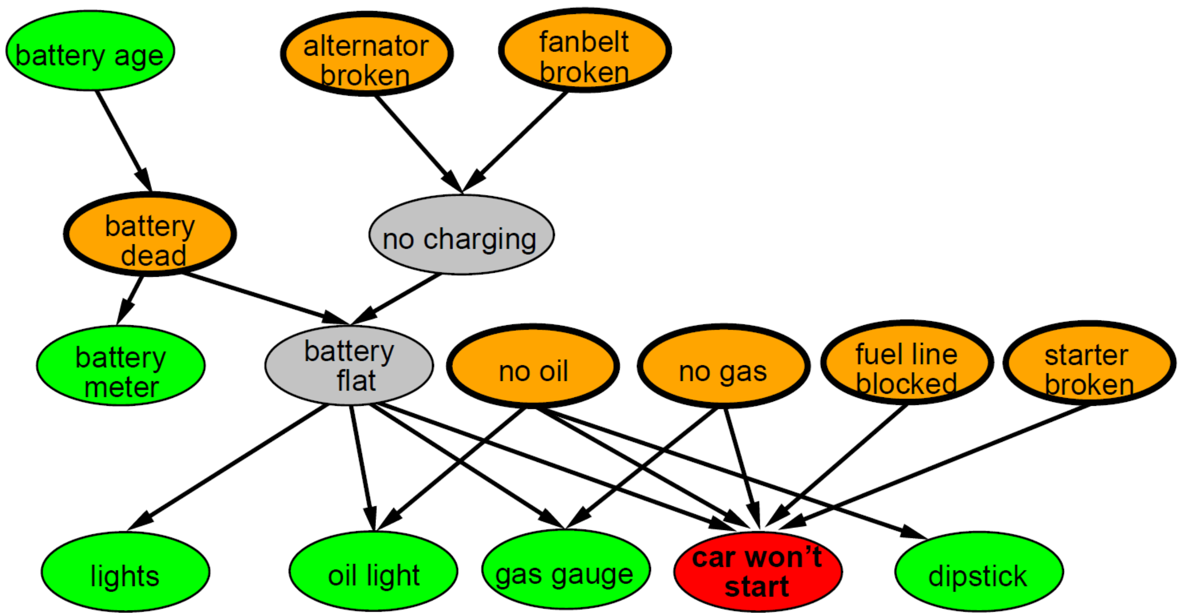
compared to 10 numbers with the right variable ordering

$$\begin{aligned} P(e,b,a,j,m) &= P(e)P(b \mid e)P(a \mid e,b)P(j \mid e,b,a)P(m \mid e,b,a,j) \\ &= P(e)P(b)P(a \mid e,b)P(j \mid a)P(m \mid a) \end{aligned}$$

Constructing Bayesian Networks

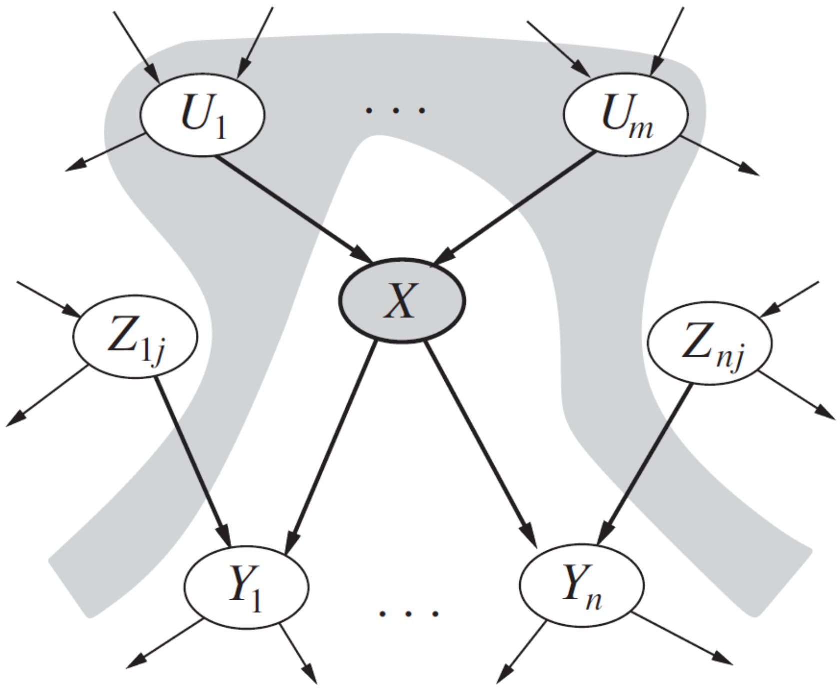
Example: Car diagnosis

- Initial evidence: car won't start
- Testable variables (green), “broken, so fix it” variables (orange)
- Hidden variables (gray) ensure sparse structure, reduce parameters



Conditional Independence Relations in Bayesian Networks

- Topological semantics: each node is conditionally independent of its non-descendants given its parents



$$\mathbf{P}(X \mid Parents(X)) = \mathbf{P}(X \mid Parents(X), ND(X))$$

Conditional Independence Relations in Bayesian Networks

<u>Example</u>:

$$\mathbf{P}(B \mid a, c) = \alpha \mathbf{P}(a, B, c) = \alpha \textstyle\sum_d \mathbf{P}(a, B, c, d)$$

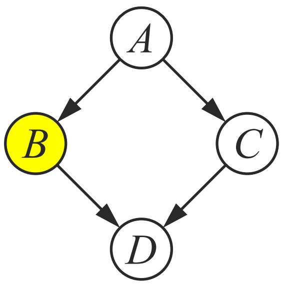
$$= \alpha \textstyle\sum_d P(a)\, \mathbf{P}(B \mid a)\, P(c \mid a)\, \mathbf{P}(d \mid B, c)$$

$$= \alpha P(a)\, \mathbf{P}(B \mid a)\, P(c \mid a) \textstyle\sum_d \mathbf{P}(d \mid B, c)$$

$$= \alpha P(a)\, \mathbf{P}(B \mid a)\, P(c \mid a)$$

$$= \alpha' \mathbf{P}(B \mid a)$$

$$= \mathbf{P}(B \mid a)$$



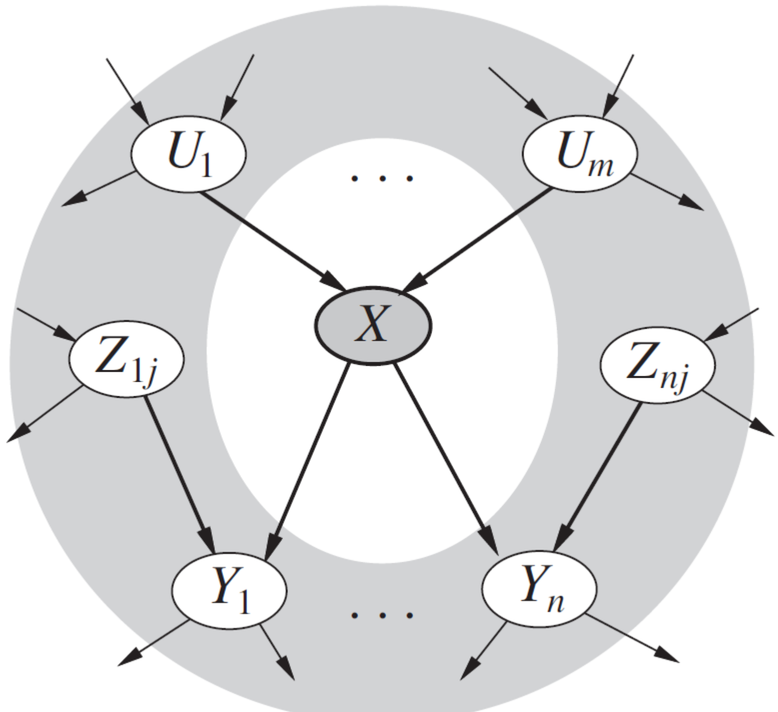
$$\mathbf{P}(B \mid a, c, d) = \alpha \mathbf{P}(a, B, c, d)$$

$$= \alpha P(a)\, \mathbf{P}(B \mid a)\, P(c \mid a)\, \mathbf{P}(d \mid B, c)$$

$$= \alpha' \mathbf{P}(B \mid a)\, \mathbf{P}(d \mid B, c)$$

Conditional Independence Relations in Bayesian Networks

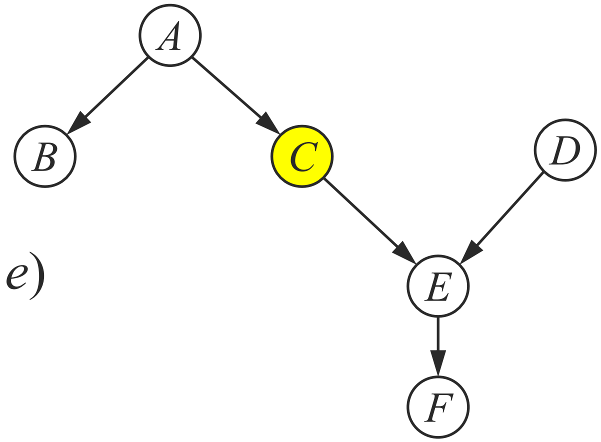
- Each node is conditionally independent of all others given its Markov blanket: parents + children + children's parents



$$\mathbf{P}(X \mid Mb(X)) = \mathbf{P}(X \mid Mb(X), AllOthers)$$

Conditional Independence Relations in Bayesian Networks

<u>Example</u>:



$$\mathbf{P}(C \mid a, b, d, e, f)$$
$$= \alpha\, \mathbf{P}(a, b, C, d, e, f)$$
$$= \alpha\, P(a)\, P(b \mid a)\, \mathbf{P}(C \mid a)\, P(d)\, \mathbf{P}(e \mid C, d)\, P(f \mid e)$$
$$= \alpha'\, \mathbf{P}(C \mid a)\, \mathbf{P}(e \mid C, d)$$

$$\mathbf{P}(C \mid a, d, e) = \alpha\, \mathbf{P}(a, C, d, e) = \alpha \textstyle\sum_b \sum_f \mathbf{P}(a, b, C, d, e, f)$$
$$= \alpha \textstyle\sum_b \sum_f P(a)\, P(b \mid a)\, \mathbf{P}(C \mid a)\, P(d)\, \mathbf{P}(e \mid C, d)\, P(f \mid e)$$
$$= \alpha\, P(a)\, \mathbf{P}(C \mid a)\, P(d)\, \mathbf{P}(e \mid C, d) \textstyle\sum_b \sum_f P(b \mid a)\, P(f \mid e)$$
$$= \alpha'\, \mathbf{P}(C \mid a)\, \mathbf{P}(e \mid C, d) \textstyle\sum_b P(b \mid a) \sum_f P(f \mid e)$$
$$= \alpha'\, \mathbf{P}(C \mid a)\, \mathbf{P}(e \mid C, d)$$

Compact Conditional Distributions

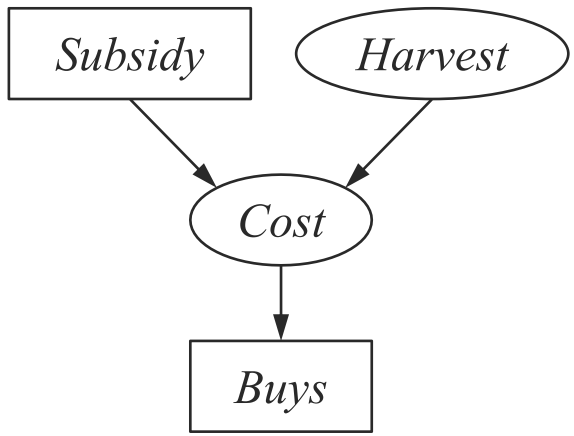
- Noisy-OR distributions model multiple non-interacting causes
  - Parents $U_1, \ldots, U_k$ include all causes (can add leak node)
  - Negated causes $\neg U_i$ do not have any influence on X
  - Independent failure probability q_i for each cause alone
    - ➡ $P(\neg x \mid u_1, \ldots u_j, \neg u_{j+1}, \ldots, \neg u_k) = \prod_{i=1}^{j} P(\neg x \mid u_i) = \prod_{i=1}^{j} q_i$

$Cold$	Flu	$Malaria$	$P(Fever)$	$P(\neg Fever)$
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	$0.02 = 0.2 \times 0.1$
T	F	F	0.4	**0.6**
T	F	T	0.94	$0.06 = 0.6 \times 0.1$
T	T	F	0.88	$0.12 = 0.6 \times 0.2$
T	T	T	0.988	$0.012 = 0.6 \times 0.2 \times 0.1$

Number of parameters linear in number of parents

Hybrid (discrete + continuous) Networks

- Discrete (*Subsidy* and *Buys*); continuous (*Harvest* and *Cost*)



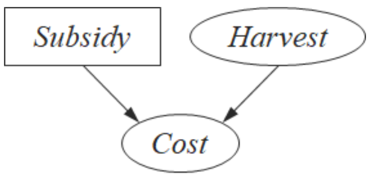
Option 1: discretization—possibly large errors, large CPTs

Option 2: finitely parameterized canonical families

1) Continuous variable, discrete + continuous parents (e.g., *Cost*)
2) Discrete variable, continuous parents (e.g., *Buys*)

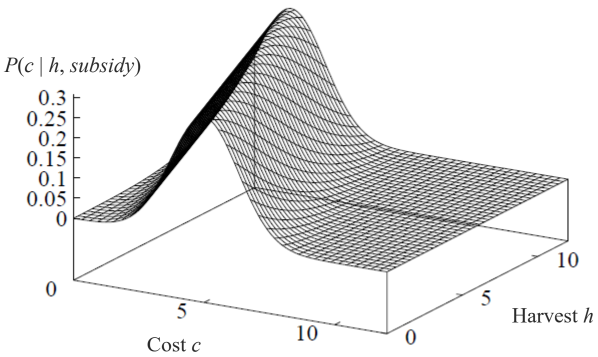
Continuous Child Variable

- Need one conditional density function for child variable given continuous parents, for each possible assignment to discrete parents



- Most common is the linear Gaussian model, e.g.,:

$$P(c \mid h, subsidy)$$
$$= N(a_t h + b_t,\ \sigma_t)(c)$$
$$= \frac{1}{\sigma_t \sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{c - (a_t h + b_t)}{\sigma_t} \right)^2 \right)$$



- Mean *Cost* varies linearly with *Harvest*, variance is fixed
- Linear variation is unreasonable over the full range

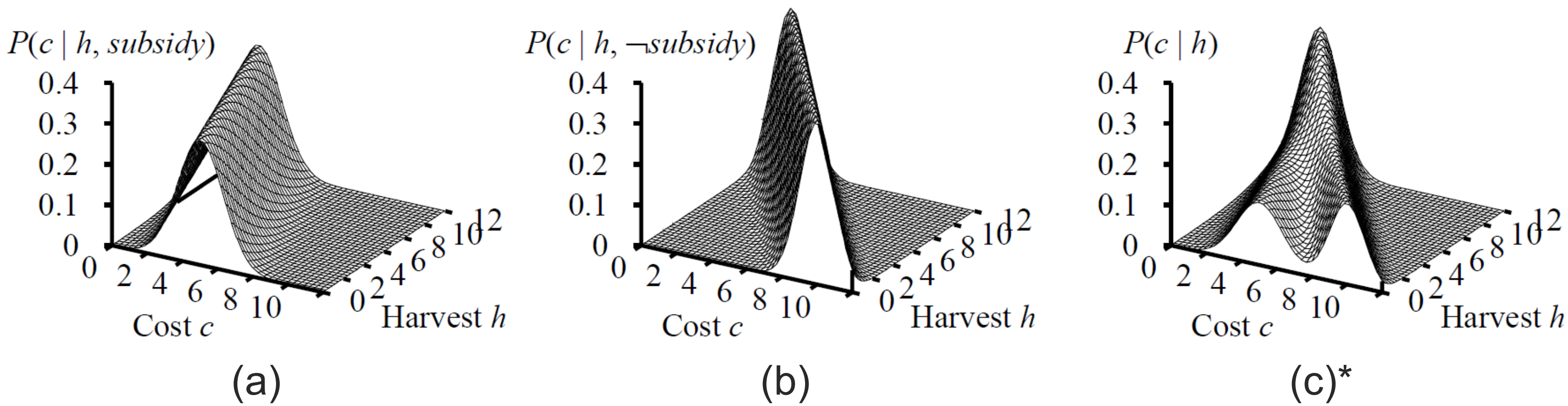
  but works OK if the likely range of *Harvest* is narrow

Continuous Child Variable

- Another distribution for $\neg subsidy$ with different parameters:

$$P(c \mid h, \neg subsidy) = N(a_f h + b_f,\ \sigma_f)(c)$$

$$= \frac{1}{\sigma_f \sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{c-(a_f h + b_f)}{\sigma_f} \right)^2 \right)$$



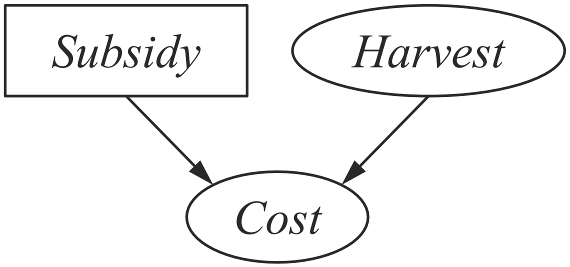
* (c) can be obtained by summing over the two subsidy cases assuming that each has prior probability 0.5

(See the next page)

Continuous Child Variable

<u>Example</u>: Network of only three nodes

$$
\begin{aligned}
\mathbf{P}(C \mid h) &= \alpha\, \mathbf{P}(C, h) \\
&= \alpha \sum_s \mathbf{P}(C, h, s) \\
&= \alpha \sum_s \mathbf{P}(C \mid h, s) P(h) P(s) \\
&= \alpha' \sum_s \mathbf{P}(C \mid h, s) P(s) \\
&= \alpha' \left[\mathbf{P}(C \mid h, s) P(s) + \mathbf{P}(C \mid h, \neg s) P(\neg s) \right]
\end{aligned}
$$



(Note that $\alpha = 1/P(h)$ and $\alpha' = 1$)

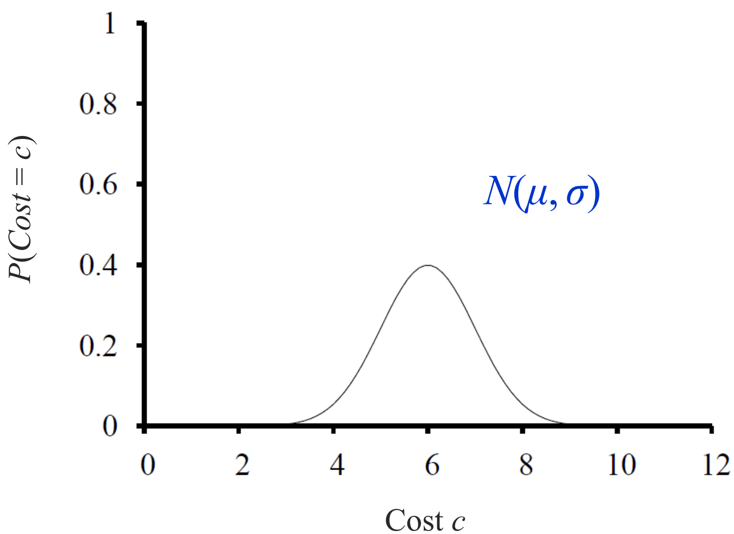
Conditioning rule:
$\mathbf{P}(\mathbf{Y}) = \sum_{\mathbf{z} \in \mathbf{Z}} \mathbf{P}(\mathbf{Y} \mid \mathbf{z}) \mathbf{P}(\mathbf{z})$

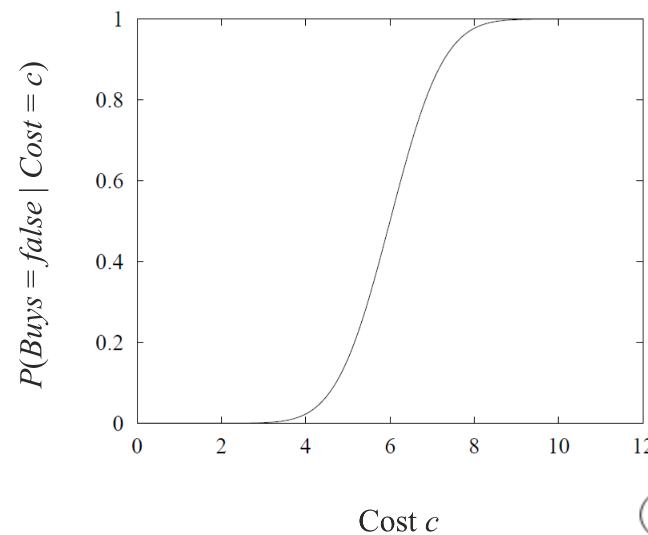
In general (without any independence information)

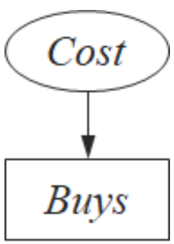
$$
\mathbf{P}(C \mid h) = \textstyle\sum_s \mathbf{P}(C \mid h, s)\, P(s \mid h) \left(= \sum_s \frac{\mathbf{P}(C, h, s)}{P(h, s)} \frac{P(s, h)}{P(h)} = \sum_s \mathbf{P}(C, s \mid h) \right)
$$

Discrete Variable with Continuous Parents

- Probability of *Buys* given *Cost* should be a "soft" threshold:







- Probit distribution uses integral of Gaussian:

$$\Phi(x) = \int_{-\infty}^{x} N(0,1)(x)dx$$

$$P(Buys = false \mid Cost = c) = \Phi\left(\frac{c-\mu}{\sigma}\right)$$

Note that

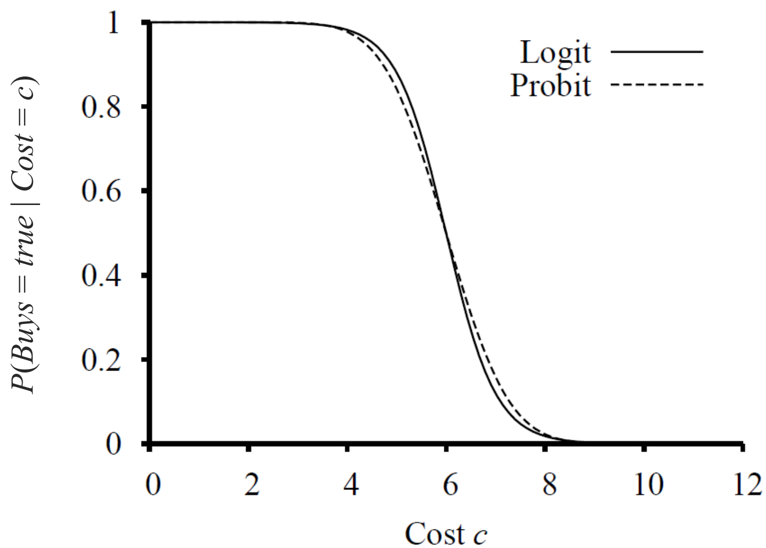
$$P(Buys = true \mid Cost = c) = \Phi\left(\frac{-(c-\mu)}{\sigma}\right)$$

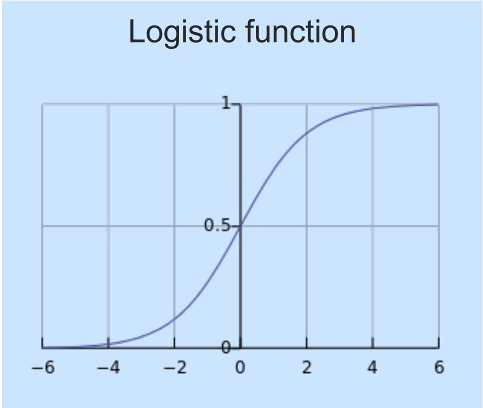
Discrete Variable with Continuous Parents

- Sigmoid (aka logit) distribution uses the logistic function $1/(1+e^{-x})$ to produce a soft threshold:

$$P(Buys = true \mid Cost = c) = \frac{1}{1+\exp\left(-2\frac{-c+\mu}{\sigma}\right)}$$

∵ Sigmoid has much longer tails





- Logit is easier to deal with mathematically than probit