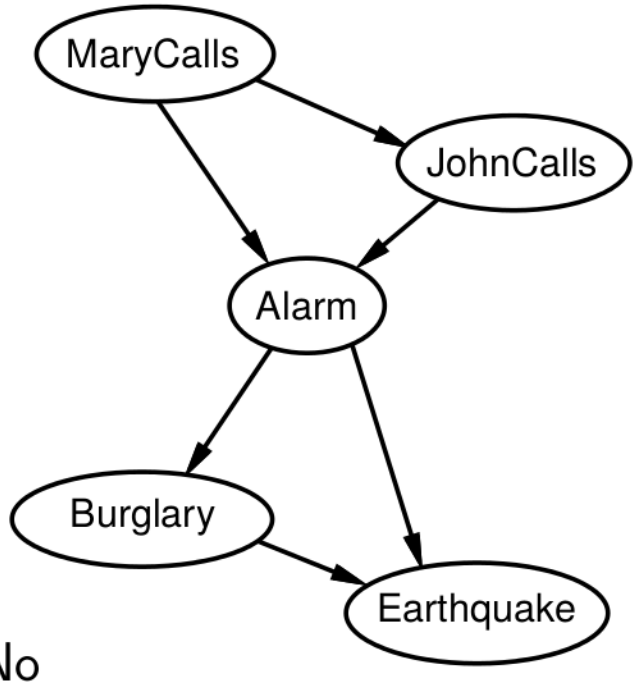
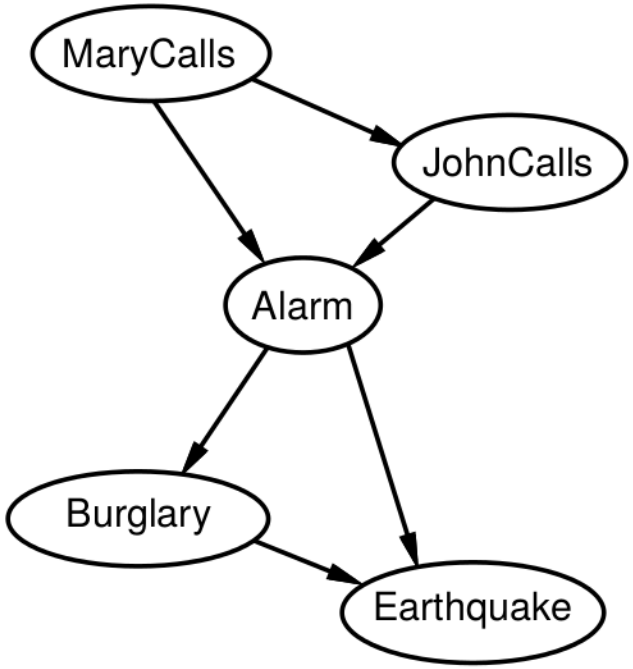
Example

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



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

Example contd.



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

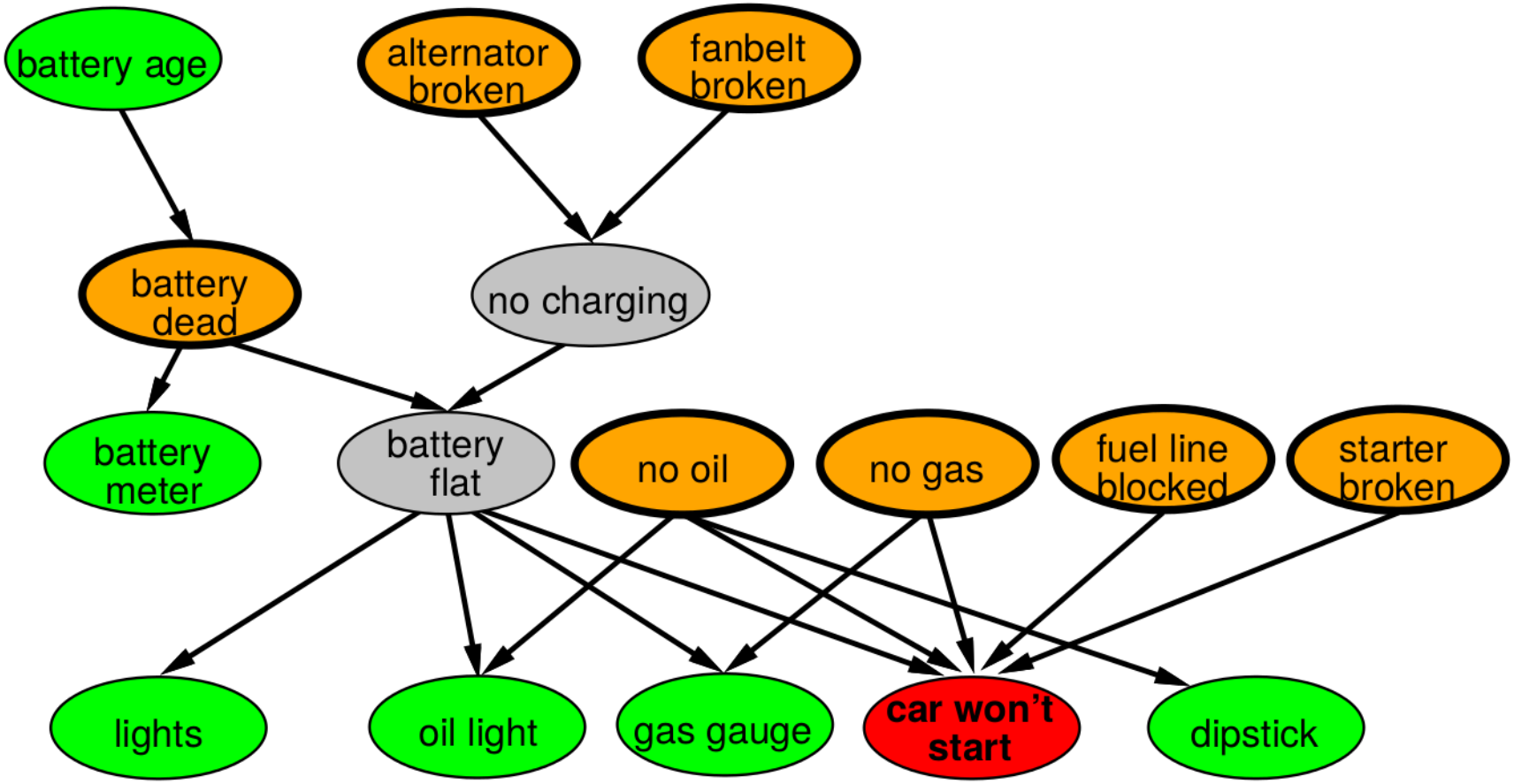
Network is less compact: $1 + 2 + 4 + 2 + 4 = 13$ numbers needed

Example: Car diagnosis

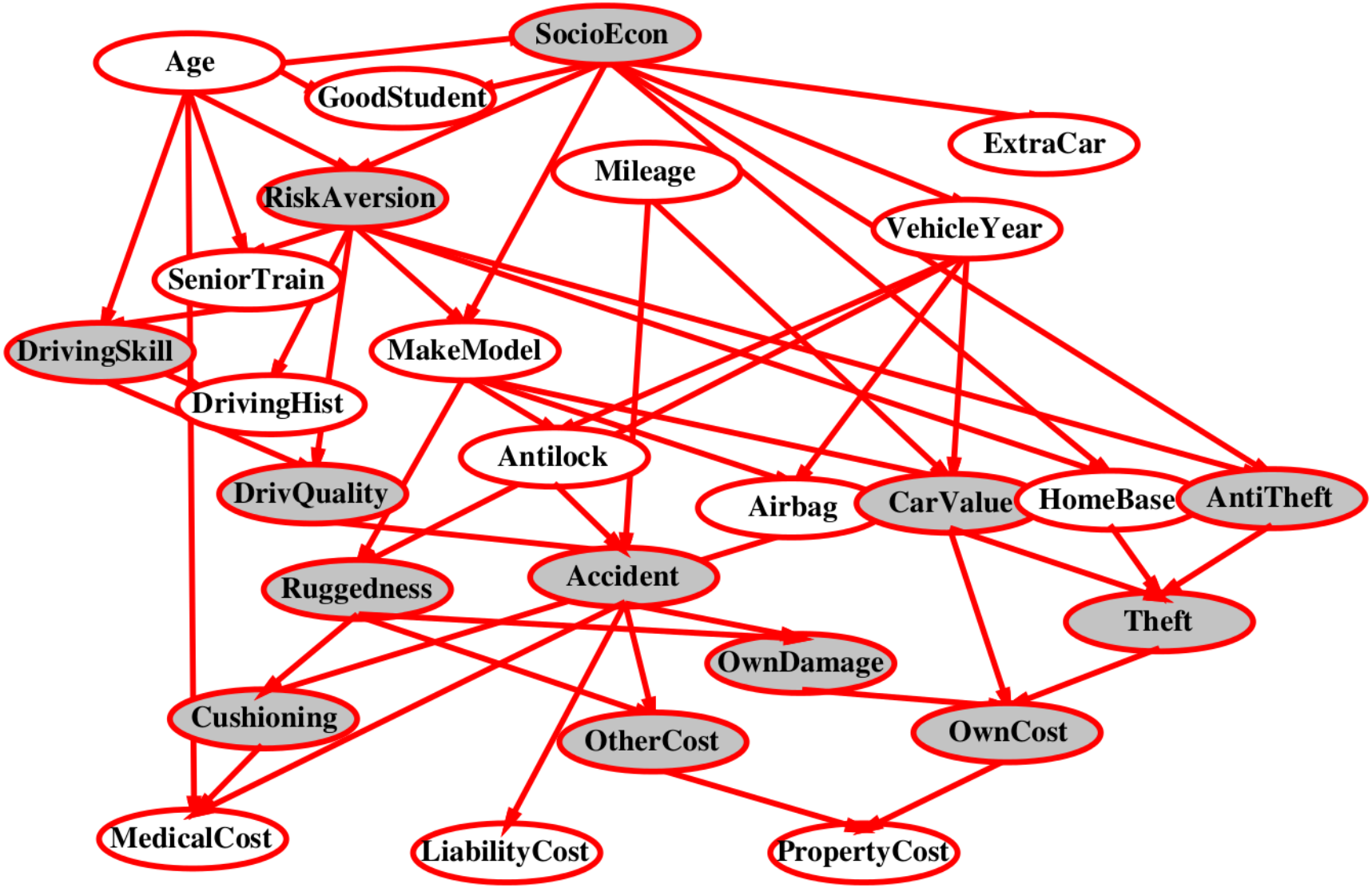
Initial evidence: car won't start

Testable variables (green), "broken, so fix it" variables (orange)

Hidden variables (gray) ensure sparse structure, reduce parameters



Example: Car insurance

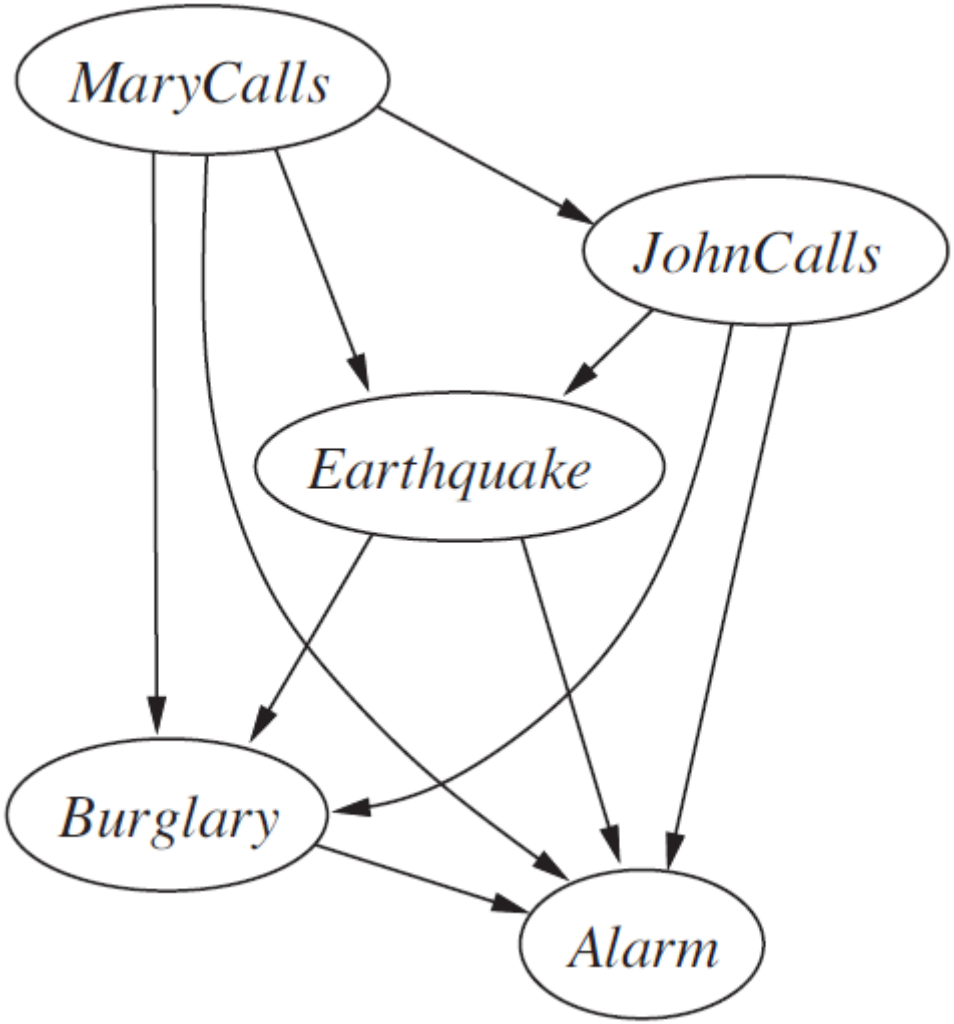


Constructing Bayesian Networks

- Quiz: MaryCalls, JohnCalls, Earthquake, Burglary, Alarm

Constructing Bayesian Networks

- MaryCalls, JohnCalls, Earthquake, Burglary, Alarm



Variable elimination algorithm

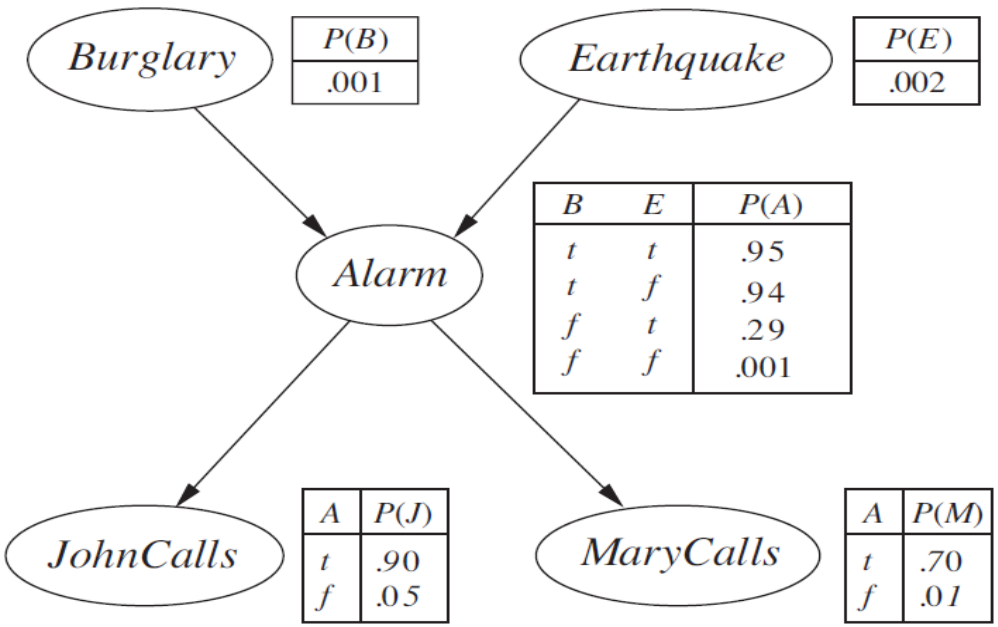
$$\mathbf{P}(B \mid j,m) = \alpha \underbrace{\mathbf{P}(B)}_{\mathbf{f}_1(B)} \sum_e \underbrace{P(e)}_{\mathbf{f}_2(E)} \sum_a \underbrace{\mathbf{P}(a \mid B,e)}_{\mathbf{f}_3(A,B,E)} \underbrace{P(j \mid a)}_{\mathbf{f}_4(A)} \underbrace{P(m \mid a)}_{\mathbf{f}_5(A)}$$

- Annotate each part of expression with the name of the corresponding **factor**
- Each factor is a matrix indexed by the values of its argument variables
- $f_4(A)$ and $f_5(A)$ corresponding to P(j | a) and P(m| a) depend just on A because J and M are fixed by the query.

$$\mathbf{f}_4(A) = \begin{pmatrix} P(j \mid a) \\ P(j \mid \neg a) \end{pmatrix} = \begin{pmatrix} 0.90 \\ 0.05 \end{pmatrix} \qquad \mathbf{f}_5(A) = \begin{pmatrix} P(m \mid a) \\ P(m \mid \neg a) \end{pmatrix} = \begin{pmatrix} 0.70 \\ 0.01 \end{pmatrix}$$

Bayesian Networks

- Compact (# of parameters)



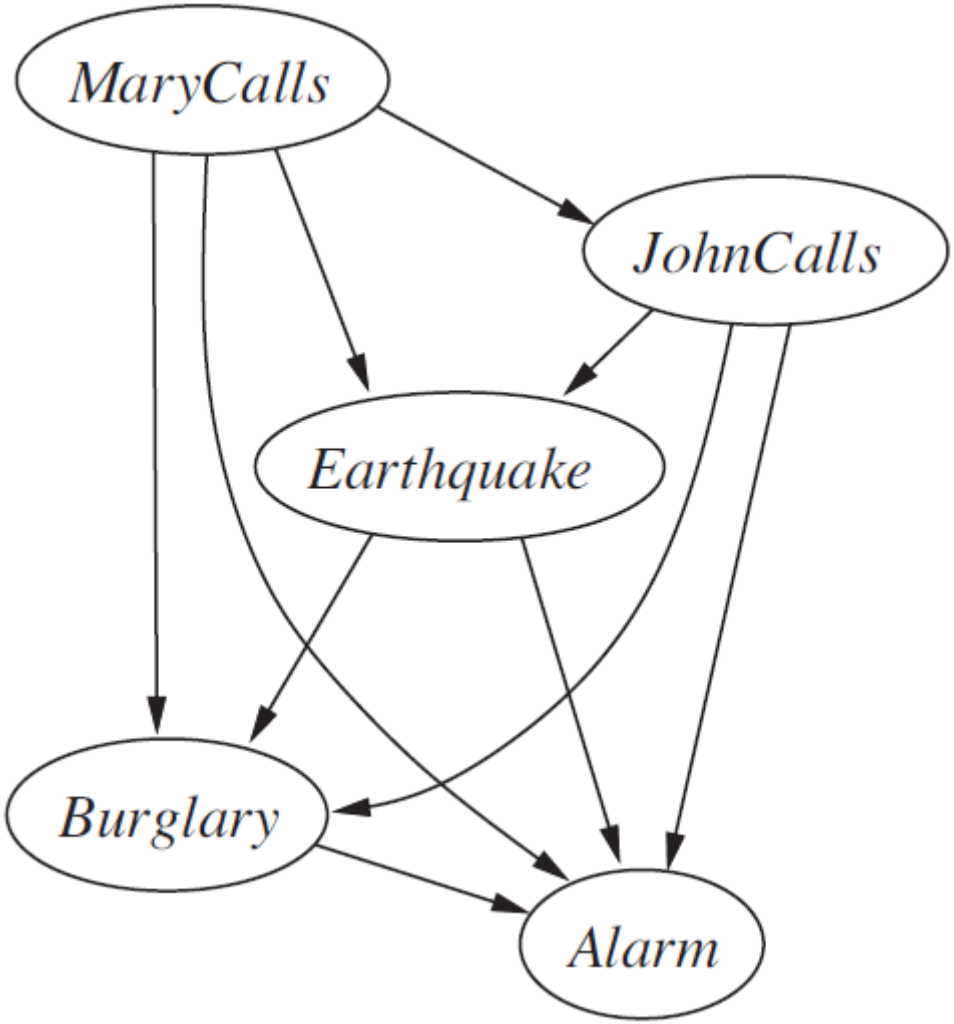
$$\mathbf{P}(X_1, \ldots, X_n) = \prod_{i=1}^{n} \mathbf{P}(X_i | X_1, \ldots, X_{i-1}) \quad \text{(chain rule)}$$
$$= \prod_{i=1}^{n} \mathbf{P}(X_i | Parents(X_i)) \quad \text{(by construction)}$$

Constructing Bayesian Networks

- MaryCalls, JohnCalls, Earthquake, Burglary, Alarm

Constructing Bayesian Networks

- MaryCalls, JohnCalls, Earthquake, Burglary, Alarm

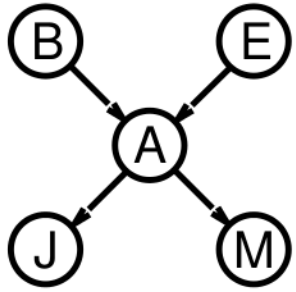


Inference by enumeration

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

Simple query on the burglary network:

$$\begin{aligned}&\mathbf{P}(B|j,m)\\&= \mathbf{P}(B,j,m)/P(j,m)\\&= \alpha\mathbf{P}(B,j,m)\\&= \alpha \sum_e \sum_a \mathbf{P}(B,e,a,j,m)\end{aligned}$$



Rewrite full joint entries using product of CPT entries:

$$\begin{aligned}&\mathbf{P}(B|j,m)\\&= \alpha \sum_e \sum_a \mathbf{P}(B)P(e)\mathbf{P}(a|B,e)P(j|a)P(m|a)\\&= \alpha\mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B,e)P(j|a)P(m|a)\end{aligned}$$

Recursive depth-first enumeration: $O(n)$ space, $O(d^n)$ time

Enumeration algorithm

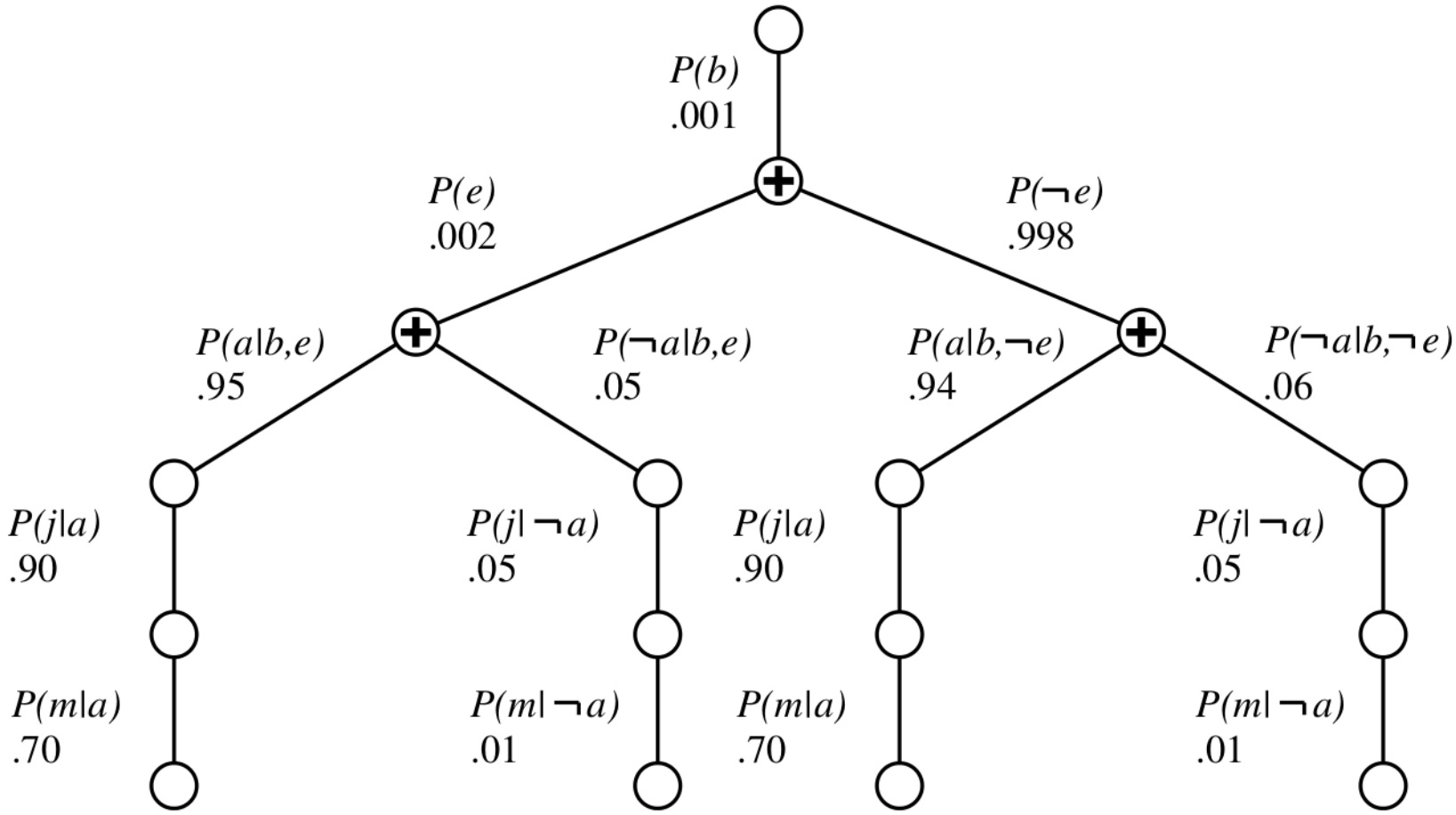
function ENUMERATION-ASK($X, \mathbf{e}, bn$) **returns** a distribution over X
  inputs: X, the query variable
    $\mathbf{e}$, observed values for variables $\mathbf{E}$
    bn, a Bayesian network with variables $\{X\} \cup \mathbf{E} \cup \mathbf{Y}$

  $\mathbf{Q}(X) \leftarrow$ a distribution over X, initially empty
  for each value x_i of X **do**
    extend $\mathbf{e}$ with value x_i for X
    $\mathbf{Q}(x_i) \leftarrow$ ENUMERATE-ALL(VARS[bn], $\mathbf{e}$)
  return NORMALIZE($\mathbf{Q}(X)$)

---

function ENUMERATE-ALL($vars, \mathbf{e}$) **returns** a real number
  if EMPTY?($vars$) **then return** 1.0
  $Y \leftarrow$ FIRST($vars$)
  if Y has value y in $\mathbf{e}$
    then return $P(y \mid Pa(Y)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}$)
    else return $\Sigma_y \; P(y \mid Pa(Y)) \times$ ENUMERATE-ALL(REST($vars$), $\mathbf{e}_y$)
      where $\mathbf{e}_y$ is $\mathbf{e}$ extended with $Y = y$

Evaluation tree



Enumeration is inefficient: repeated computation

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

Inference by variable elimination

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

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

Variable elimination: Basic operations

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

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

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

Variable elimination algorithm

$$\mathbf{P}(B \mid j,m) = \alpha \underbrace{\mathbf{P}(B)}_{\mathbf{f}_1(B)} \sum_e \underbrace{P(e)}_{\mathbf{f}_2(E)} \sum_a \underbrace{\mathbf{P}(a \mid B,e)}_{\mathbf{f}_3(A,B,E)} \underbrace{P(j \mid a)}_{\mathbf{f}_4(A)} \underbrace{P(m \mid a)}_{\mathbf{f}_5(A)}$$

- Annotate each part of expression with the name of the corresponding **factor**
- Each factor is a matrix indexed by the values of its argument variables
- $f_4(A)$ and $f_5(A)$ corresponding to P(j | a) and P(m| a) depend just on A because J and M are fixed by the query.

$$\mathbf{f}_4(A) = \begin{pmatrix} P(j \mid a) \\ P(j \mid \neg a) \end{pmatrix} = \begin{pmatrix} 0.90 \\ 0.05 \end{pmatrix} \qquad \mathbf{f}_5(A) = \begin{pmatrix} P(m \mid a) \\ P(m \mid \neg a) \end{pmatrix} = \begin{pmatrix} 0.70 \\ 0.01 \end{pmatrix}$$

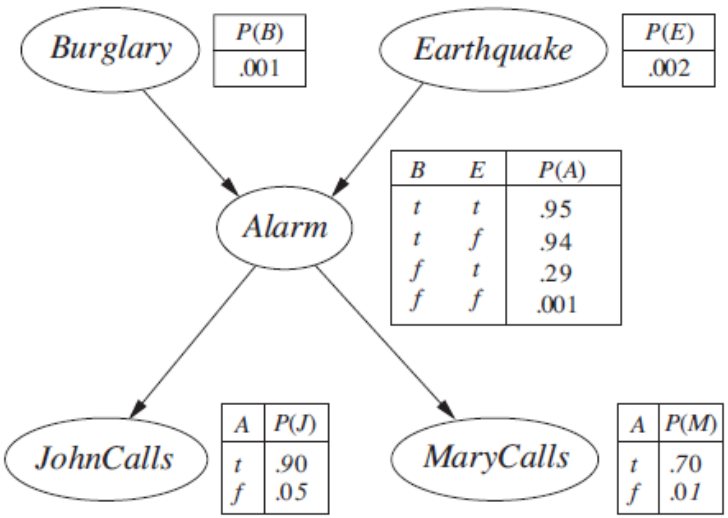
Variable elimination algorithm

$$\mathbf{P}(B \mid j,m) = \alpha \underbrace{\mathbf{P}(B)}_{\mathbf{f}_1(B)} \sum_e \underbrace{P(e)}_{\mathbf{f}_2(E)} \sum_a \underbrace{\mathbf{P}(a \mid B,e)}_{\mathbf{f}_3(A,B,E)} \underbrace{P(j \mid a)}_{\mathbf{f}_4(A)} \underbrace{P(m \mid a)}_{\mathbf{f}_5(A)}$$

$$\mathbf{f}_4(A) = \begin{pmatrix} P(j \mid a) \\ P(j \mid \neg a) \end{pmatrix} = \begin{pmatrix} 0.90 \\ 0.05 \end{pmatrix} \qquad \mathbf{f}_5(A) = \begin{pmatrix} P(m \mid a) \\ P(m \mid \neg a) \end{pmatrix} = \begin{pmatrix} 0.70 \\ 0.01 \end{pmatrix}$$

- $f_3(A,B,E)$ will be a 2×2×2 matrix

A	B	E	p(A|B,E)
T	T	T	0.95
T	T	F	0.94
T	F	T	0.29
T	F	F	0.01
F	T	T	0.05
F	T	F	0.06
F	F	T	0.71
F	F	F	0.999



Variable elimination algorithm

$$\mathbf{P}(B \mid j, m) = \alpha \underbrace{\mathbf{P}(B)}_{\mathbf{f}_1(B)} \sum_e \underbrace{P(e)}_{\mathbf{f}_2(E)} \sum_a \underbrace{\mathbf{P}(a \mid B, e)}_{\mathbf{f}_3(A,B,E)} \underbrace{P(j \mid a)}_{\mathbf{f}_4(A)} \underbrace{P(m \mid a)}_{\mathbf{f}_5(A)}$$

$$\mathbf{P}(B \mid j, m) = \alpha \mathbf{f}_1(B) \times \sum_e \mathbf{f}_2(E) \times \sum_a \mathbf{f}_3(A, B, E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A)$$

- "x" operator is not ordinary matrix multiplication but instead the **pointwise product** operation
- The pointwise product of two factors f_1 and f_2 yields a new factor f whose variables are the union of the variables in f_1 and f_2 and whose elements are given by the product of the corresponding elements in the two factors.

Variable elimination algorithm

$$\mathbf{P}(B \mid j, m) = \alpha\, \mathbf{f}_1(B) \times \sum_e \mathbf{f}_2(E) \times \sum_a \mathbf{f}_3(A, B, E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A)$$

- Pointwise product operation:

A	B	$\mathbf{f}_1(A,B)$	B	C	$\mathbf{f}_2(B,C)$	A	B	C	$\mathbf{f}_3(A,B,C)$
T	T	.3	T	T	.2	T	T	T	$.3 \times .2 = .06$
T	F	.7	T	F	.8	T	T	F	$.3 \times .8 = .24$
F	T	.9	F	T	.6	T	F	T	$.7 \times .6 = .42$
F	F	.1	F	F	.4	T	F	F	$.7 \times .4 = .28$
						F	T	T	$.9 \times .2 = .18$
						F	T	F	$.9 \times .8 = .72$
						F	F	T	$.1 \times .6 = .06$
						F	F	F	$.1 \times .4 = .04$

Figure 14.10 Illustrating pointwise multiplication: $\mathbf{f}_1(A, B) \times \mathbf{f}_2(B, C) = \mathbf{f}_3(A, B, C)$.

Variable elimination algorithm

$$\mathbf{P}(B \mid j, m) = \alpha\, \mathbf{f}_1(B) \times \sum_e \mathbf{f}_2(E) \times \sum_a \mathbf{f}_3(A, B, E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A)$$

- Summation operation: Summing out a variable from a product of factors is done by adding up the submatrices formed by fixing the variable to each of its values in turn

$$\begin{aligned} \mathbf{f}(B, C) &= \sum_a \mathbf{f}_3(A, B, C) = \mathbf{f}_3(a, B, C) + \mathbf{f}_3(\neg a, B, C) \\ &= \begin{pmatrix} .06 & .24 \\ .42 & .28 \end{pmatrix} + \begin{pmatrix} .18 & .72 \\ .06 & .04 \end{pmatrix} = \begin{pmatrix} .24 & .96 \\ .48 & .32 \end{pmatrix} . \end{aligned}$$

Variable elimination algorithm

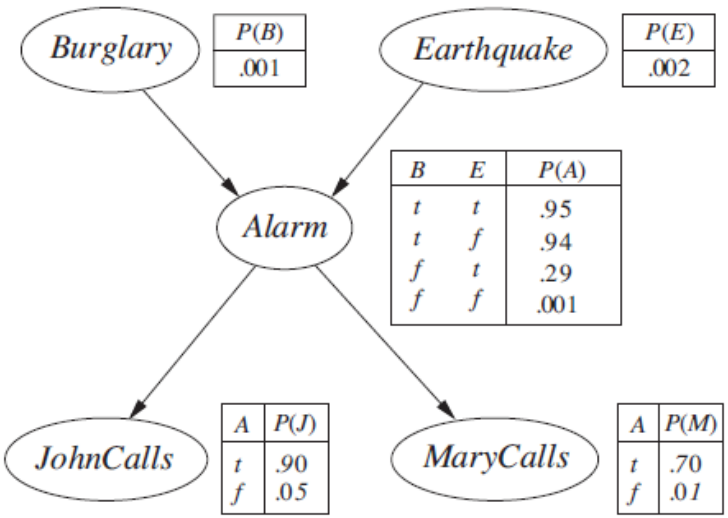
$$\mathbf{P}(B \mid j,m) = \alpha \underbrace{\mathbf{P}(B)}_{\mathbf{f}_1(B)} \sum_e \underbrace{P(e)}_{\mathbf{f}_2(E)} \sum_a \underbrace{\mathbf{P}(a \mid B,e)}_{\mathbf{f}_3(A,B,E)} \underbrace{P(j \mid a)}_{\mathbf{f}_4(A)} \underbrace{P(m \mid a)}_{\mathbf{f}_5(A)}$$

Quiz

$$\mathbf{P}(B \mid j,m) = \alpha \mathbf{f}_1(B) \times \sum_e \mathbf{f}_2(E) \times \sum_a \mathbf{f}_3(A,B,E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A)$$

$$\mathbf{f}_4(A) = \begin{pmatrix} P(j \mid a) \\ P(j \mid \neg a) \end{pmatrix} = \begin{pmatrix} 0.90 \\ 0.05 \end{pmatrix} \qquad \mathbf{f}_5(A) = \begin{pmatrix} P(m \mid a) \\ P(m \mid \neg a) \end{pmatrix} = \begin{pmatrix} 0.70 \\ 0.01 \end{pmatrix}$$

A	B	E	f_3(A,B,E)
T	T	T	0.95
T	T	F	0.94
T	F	T	0.29
T	F	F	0.01
F	T	T	0.05
F	T	F	0.06
F	F	T	0.71
F	F	F	0.999



Variable elimination algorithm

$$\mathbf{P}(B \mid j, m) = \alpha\, \mathbf{f}_1(B) \times \sum_e \mathbf{f}_2(E) \times \sum_a \mathbf{f}_3(A, B, E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A)$$

- The trick to notice is that any factor that does not depend on the variable to be summed out can be moved outside the summation.

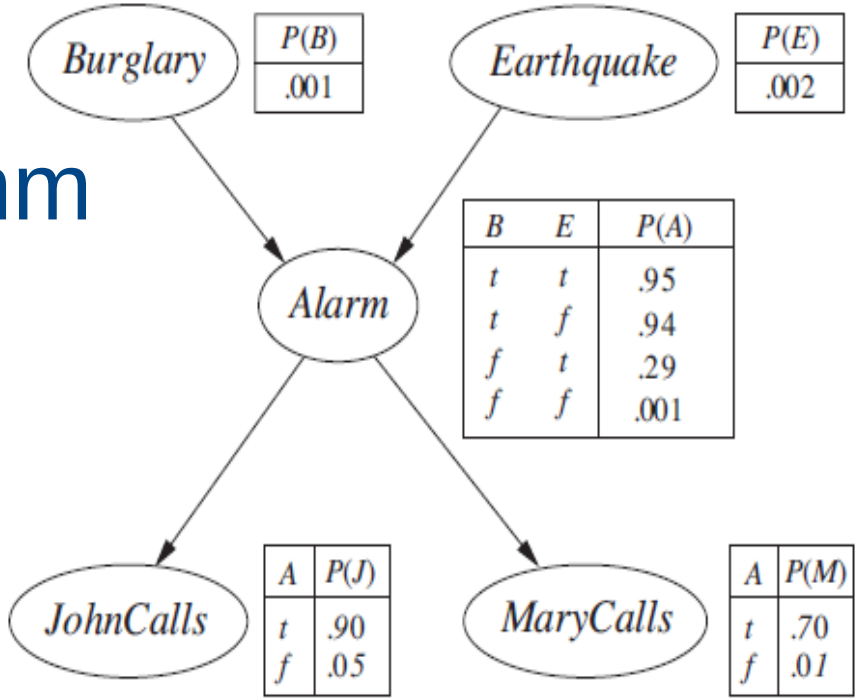
$$\sum_e \mathbf{f}_2(E) \times \mathbf{f}_3(A, B, E) \times \mathbf{f}_4(A) \times \mathbf{f}_5(A) = \mathbf{f}_4(A) \times \mathbf{f}_5(A) \times \sum_e \mathbf{f}_2(E) \times \mathbf{f}_3(A, B, E)$$

- Different orderings cause different intermediate factors to be generated during the calculation

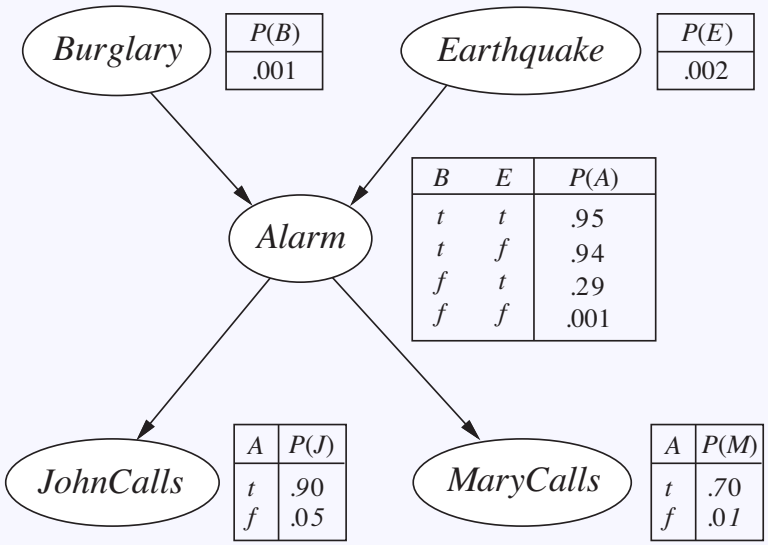
$$\mathbf{P}(B \mid j, m) = \alpha\, \mathbf{f}_1(B) \times \sum_a \mathbf{f}_4(A) \times \mathbf{f}_5(A) \times \sum_e \mathbf{f}_2(E) \times \mathbf{f}_3(A, B, E)$$

- Idea: Eliminate whichever variable minimizes the size of the next factor to be constructed.

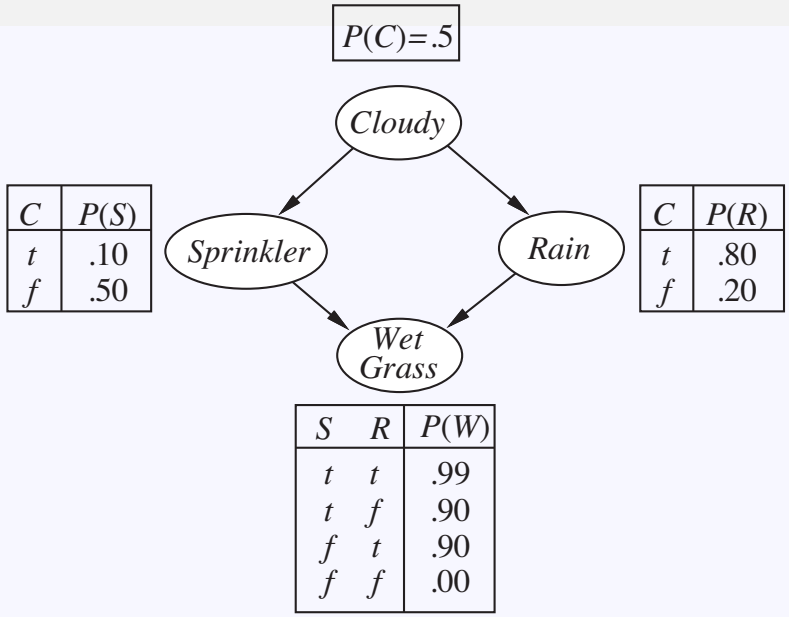
Variable elimination algorithm



Complexity of Exact Inference



Singly connected network



Multiply connected network

- Singly connected networks (or polytrees) are networks where there is at most one undirected path between any two nodes in the networks.
- The time and space complexity of exact inference in polytrees is **linear** in the number of CPT entries.
- For multiply connected networks, variable elimination can have **exponential** time and space complexity in the worst case.

Approximate Inference Methods

Since exact inference is intractable in large networks, we consider approximate inference methods that are much faster.

- Monte Carlo
  - Direct sampling methods
  - Markov chain sampling
- Variational methods
- Loopy propagation

Direct sampling methods

What is sampling?

Sampling consists in generating a finite number of samples (values) from a known probability distribution.

Example

Sampling from a Bernoulli distribution $P(Coin) =< 0.5, 0.5 >$, where $Coin \in \{heads, tails\}$, consists in flipping a coin a number of times and observing the results, e.g. $\{heads, tails, tails, heads, tails, \dots\}$.

Why do we use sampling methods?

Sampling is often used to compute $\mathbb{E}[f(x)]$, where x is a random variable and $\mathbb{E}[f(x)]$ cannot be computed in a closed form (or efficiently).

Example

To compute $\mathbb{E}[\sqrt{|x|}]$ where $x \sim \mathcal{N}(0,1)$ (standard normal distribution), we generate samples $\{0.0591, 1.7971, 0.2641, 0.8717, -1.4462\}$, and get

$$\mathbb{E}[\sqrt{x}] \approx \frac{\sqrt{|0.0591|}+\sqrt{|1.7971|}+\sqrt{|0.2641|}+\sqrt{|0.8717|}+\sqrt{|-1.4462|}}{5} \approx 0.85$$

Direct sampling methods

Sample events from a network that has no evidence associated with it.

Each variable is sampled in turn, in topological order.

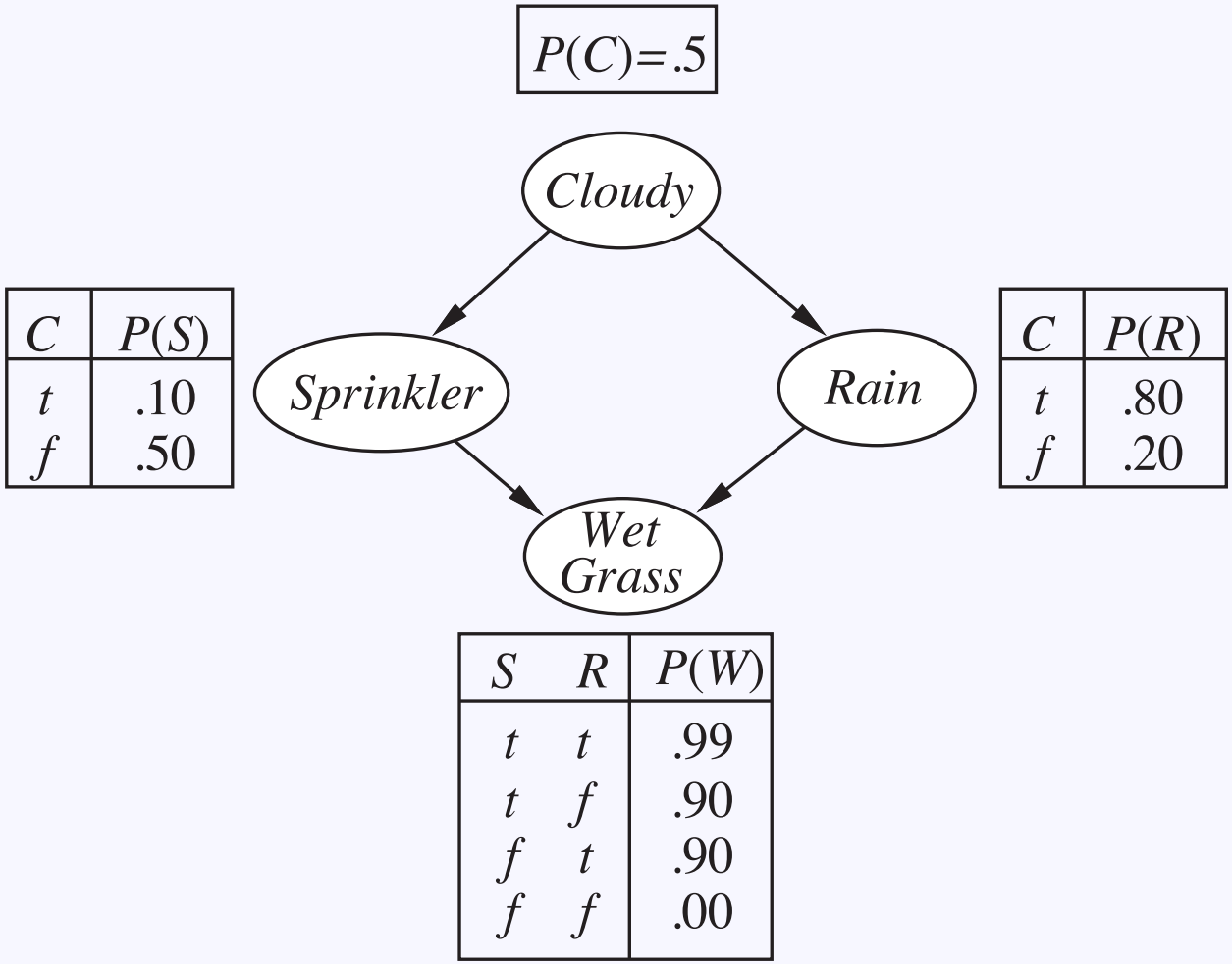
The probability distribution from which the value is sampled is conditioned on the values already assigned to the variable's parents.

function PRIOR-SAMPLE(bn) **returns** an event sampled from the prior specified by bn
  inputs: bn, a Bayesian network specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$

  $\mathbf{x} \leftarrow$ an event with n elements
  foreach variable X_i **in** $X_1, \ldots, X_n$ **do**
    $\mathbf{x}[i] \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
  return $\mathbf{x}$

Generate several samples $\mathbf{x}$ and calculate the frequency of each instance.

Example


P(C)=.5
Cloudy
Sprinkler
Rain
Wet Grass
C P(S)
t .10
f .50
C P(R)
t .80
f .20
S R P(W)
t t .99
t f .90
f t .90
f f .00

Example

1. Sample from $P(Cloudy) =< 0.5, 0.5 >$, value is true.
2. Sample from $P(Sprinkler \mid Cloudy = true) =< 0.1, 0.9 >$, value is false.
3. Sample from $P(Rain \mid Cloudy = true) =< 0.8, 0.2 >$, value is true.
4. Sample from $P(WetGrass \mid Sprinkler = false, Rain = true) =< 0.9, 0.1 >$, value is true.

Rejection sampling

Direct sampling is useful for estimating a joint probability $P(x_1, x_2, \ldots, x_n)$ when there is no evidence (no known value for any variable).

The same idea can be used to estimate $P(\mathbf{X} \mid \mathbf{e})$, where $\mathbf{X}$ is any variable and $\mathbf{e}$ is an evidence (the value(s) of certain variable(s)).

1. Generate samples from the prior distribution specified by the network.
2. Reject all those that do not match the evidence.
3. Estimate $\hat{P}(x| \mathbf{e})$ is obtained by counting the number of samples where $\mathbf{X} = x$.

$$\hat{P}(x| \mathbf{e}) = \frac{\text{number of samples } (x, \mathbf{e})}{\text{number of samples } (\mathbf{e})}$$

Example of rejection sampling

- We wish to estimate $P(Rain \mid Sprinkler = true)$, using 100 samples.
- Of the 100 samples,
  - 73 have $Sprinkler = false$ and are rejected,
  - 27 have $Sprinkler = true$. Of the 27,
    - 8 have $Rain = true$,
    - and 19 have $Rain = false$.
- Thus,

$$P(Rain|Sprinkler = true) \approx normalize(< 8, 19 >) =< 0.296, 0.704 > .$$

Rejection sampling

function REJECTION-SAMPLING(X, **e**, bn, N) **returns** an estimate of $\mathbf{P}(X|\mathbf{e})$
  inputs: X, the query variable
    e, observed values for variables **E**
    bn, a Bayesian network
    N, the total number of samples to be generated
  local variables: **N**, a vector of counts for each value of X, initially zero

  for $j = 1$ to N **do**
    x $\leftarrow$ PRIOR-SAMPLE(bn)
    if **x** is consistent with **e** **then**
      N[x] $\leftarrow$ **N**[x]+1 where x is the value of X in **x**
  return NORMALIZE(**N**)

Likelihood weighting

Rejection sampling is not efficient because it wastes a lot of samples (all the samples that do not agree with the provided evidence).

Can we simply force the evidence variables to agree with the provided values, and sample only the non-evidence variables?

Temporal probability models

Chapter 15, Sections 1–5

Independence

- Two variables are *independent* in a joint distribution if:

$$P(X,Y) = P(X)P(Y)$$

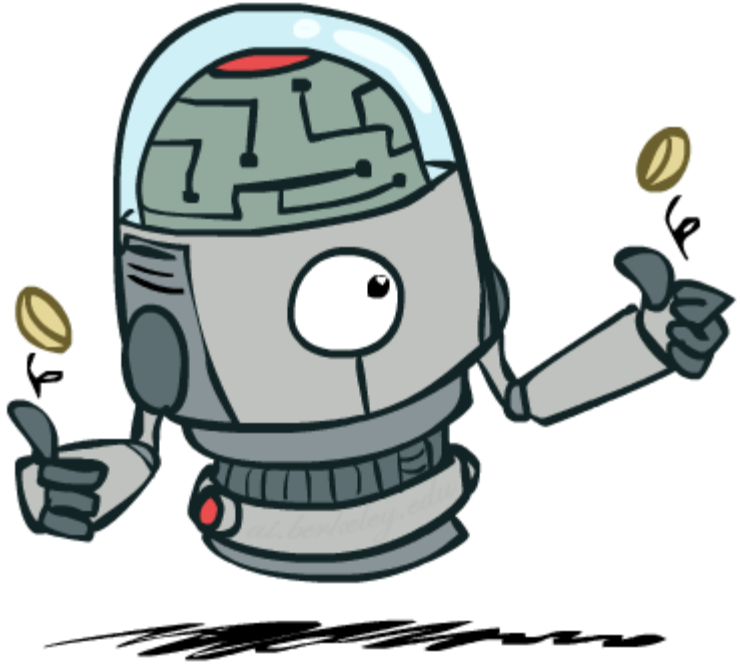
$$X \perp\!\!\!\perp Y$$

$$\forall x, y \, P(x,y) = P(x)P(y)$$

  - Says the joint distribution *factors* into a product of two simple ones
  - Usually variables aren't independent!

- Can use independence as a *modeling assumption*
  - Independence can be a simplifying assumption
  - *Empirical* joint distributions: at best "close" to independent
  - What could we assume for {Weather, Traffic, Cavity}?

- Independence is like something from CSPs: what?



Example: Independence?

$P(T)$

T	P
hot	0.5
cold	0.5

$P_1(T, W)$

T	W	P
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

$P(W)$

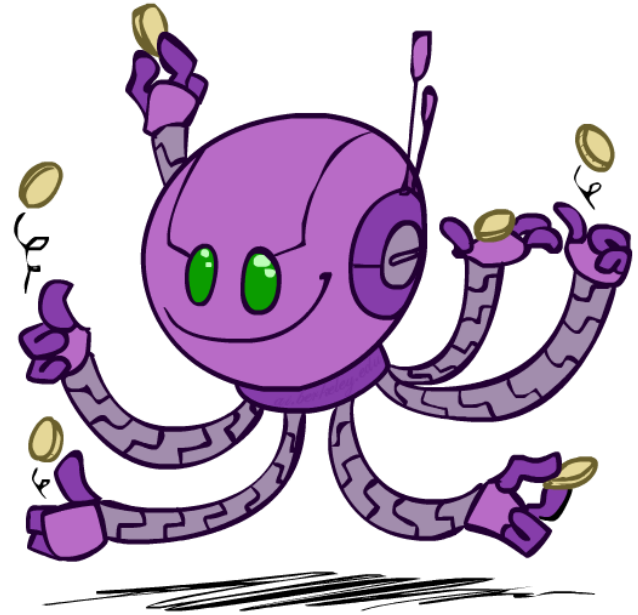
W	P
sun	0.6
rain	0.4

$P_2(T, W) = P(T)P(W)$

T	W	P
hot	sun	0.3
hot	rain	0.2
cold	sun	0.3
cold	rain	0.2

Example: Independence

- N fair, independent coin flips:



$P(X_1)$

H	0.5
T	0.5

$P(X_2)$

H	0.5
T	0.5

$\ldots$

$P(X_n)$

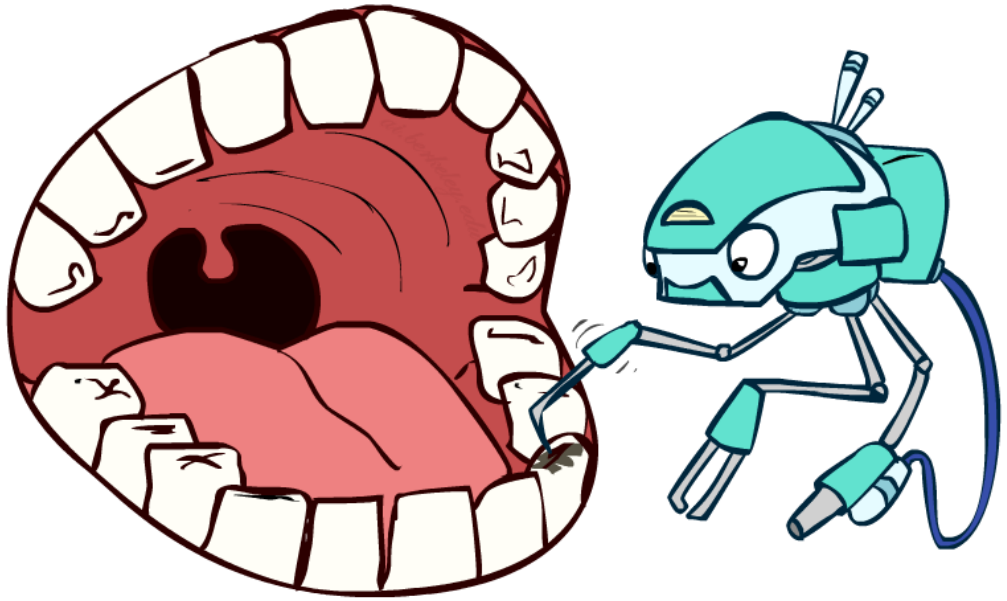
H	0.5
T	0.5

$P(X_1, X_2, \ldots X_n)$

2^n

Conditional Independence

- P(Toothache, Cavity, Catch)
- If I have a cavity, the probability that the probe catches in it doesn't depend on whether I have a toothache:
  - P(+catch | +toothache, +cavity) = P(+catch | +cavity)
- The same independence holds if I don’t have a cavity:
  - P(+catch | +toothache, -cavity) = P(+catch| -cavity)
- Catch is *conditionally independent* of Toothache given Cavity:
  - P(Catch | Toothache, Cavity) = P(Catch | Cavity)



- Equivalent statements:
  - P(Toothache | Catch , Cavity) = P(Toothache | Cavity)
  - P(Toothache, Catch | Cavity) = P(Toothache | Cavity) P(Catch | Cavity)
  - One can be derived from the other easily

Conditional Independence

- Unconditional (absolute) independence very rare (why?)
- *Conditional independence* is our most basic and robust form of knowledge about uncertain environments.
- X is conditionally independent of Y given Z

$$X \perp\!\!\!\perp Y | Z$$

if and only if:

$$\forall x, y, z : P(x, y|z) = P(x|z)P(y|z)$$

or, equivalently, if and only if

$$\forall x, y, z : P(x|z, y) = P(x|z)$$

Probability Recap

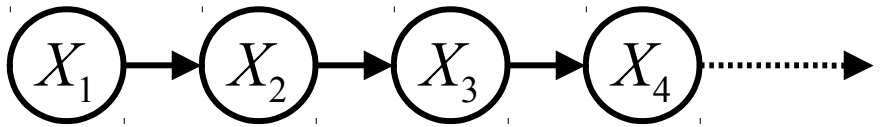
- Conditional probability $$P(x|y) = \frac{P(x,y)}{P(y)}$$
- Product rule $$P(x,y) = P(x|y)P(y)$$
- Chain rule $$\begin{aligned} P(X_1, X_2, \ldots X_n) &= P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)\ldots \\ &= \prod_{i=1}^{n} P(X_i|X_1, \ldots, X_{i-1}) \end{aligned}$$
- X, Y independent if and only if: $$\forall x, y : P(x,y) = P(x)P(y)$$
- X and Y are conditionally independent given Z if and only if: $$X \perp\!\!\!\perp Y|Z$$
$$\forall x, y, z : P(x,y|z) = P(x|z)P(y|z)$$

Reasoning over Time or Space

- Often, we want to reason about a sequence of observations
  - Speech recognition
  - Robot localization
  - User attention
  - Medical monitoring
- Need to introduce time (or space) into our models

Markov Models

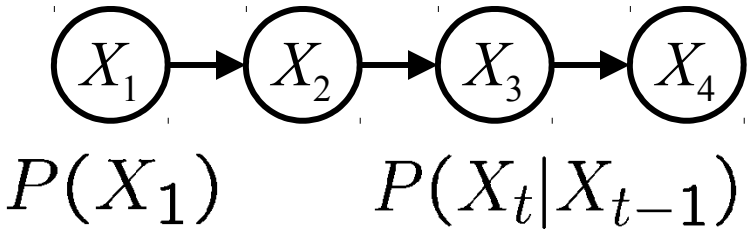
- Future states depend only on the current state not on the events that occurred before it
- Value of X at a given time is called the state



$P(X_1)$ $\quad$ $P(X_t|X_{t-1})$

  - Parameters: called transition probabilities or dynamics, specify how the state evolves over time (also, initial state probabilities)
  - Stationarity assumption: transition probabilities the same at all times

Joint Distribution of a Markov Model



- Joint distribution:

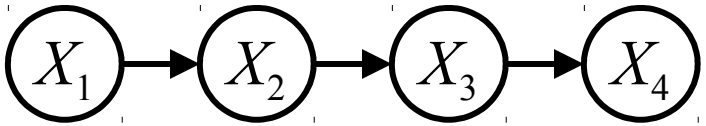
$$P(X_1, X_2, X_3, X_4) = P(X_1)P(X_2|X_1)P(X_3|X_2)P(X_4|X_3)$$

- More generally:

$$P(X_1, X_2, \ldots, X_T) = P(X_1)P(X_2|X_1)P(X_3|X_2)\ldots P(X_T|X_{T-1})$$

$$= P(X_1)\prod_{t=2}^{T} P(X_t|X_{t-1})$$

Chain Rule and Markov Models



- From the chain rule, every joint distribution over X_1, X_2, X_3, X_4 can be written as:

$$P(X_1, X_2, X_3, X_4) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)P(X_4|X_1, X_2, X_3)$$

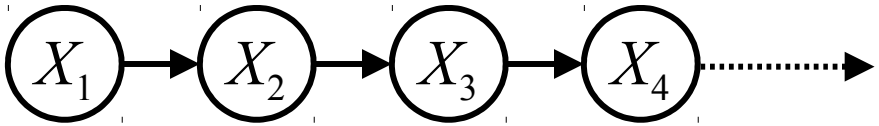
- Assuming that

$$X_3 \perp\!\!\!\perp X_1 \mid X_2 \text{ and } X_4 \perp\!\!\!\perp X_1, X_2 \mid X_3$$

results in the expression posited on the previous slide:

$$P(X_1, X_2, X_3, X_4) = P(X_1)P(X_2|X_1)P(X_3|X_2)P(X_4|X_3)$$

Chain Rule and Markov Models



- From the chain rule, every joint distribution over $X_1, X_2, \ldots, X_T$ can be written as:

$$P(X_1, X_2, \ldots, X_T) = P(X_1) \prod_{t=2}^{T} P(X_t | X_1, X_2, \ldots, X_{t-1})$$

- Assuming that for all *t*:

$$X_t \perp\!\!\!\perp X_1, \ldots, X_{t-2} \mid X_{t-1}$$

gives us the expression posited on the earlier slide:

$$P(X_1, X_2, \ldots, X_T) = P(X_1) \prod_{t=2}^{T} P(X_t | X_{t-1})$$

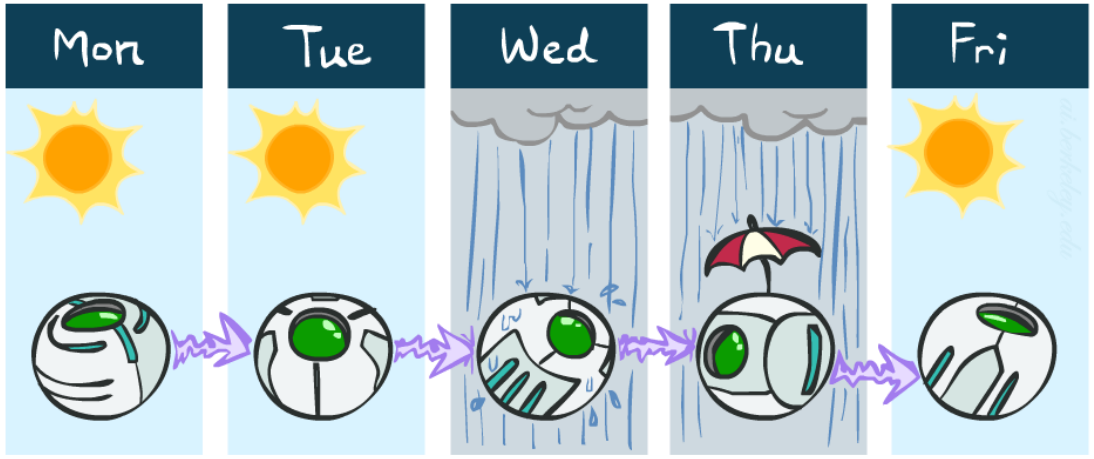
Example Markov Chain: Weather

- States: X = {rain, sun}

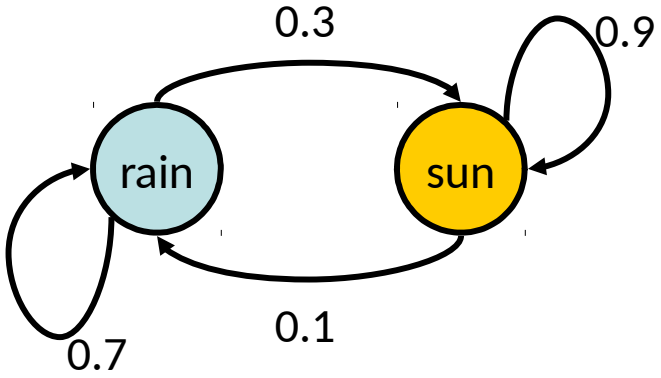
- Initial distribution: 1.0 sun

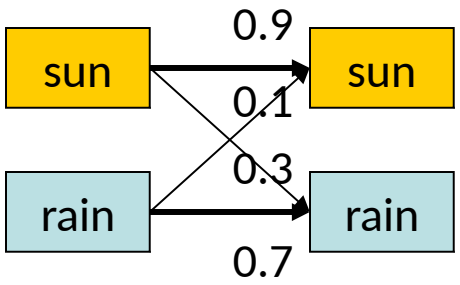
- CPT $P(X_t \mid X_{t-1})$:

X_{t-1}	X_t	$P(X_t \mid X_{t-1})$
sun	sun	0.9
sun	rain	0.1
rain	sun	0.3
rain	rain	0.7



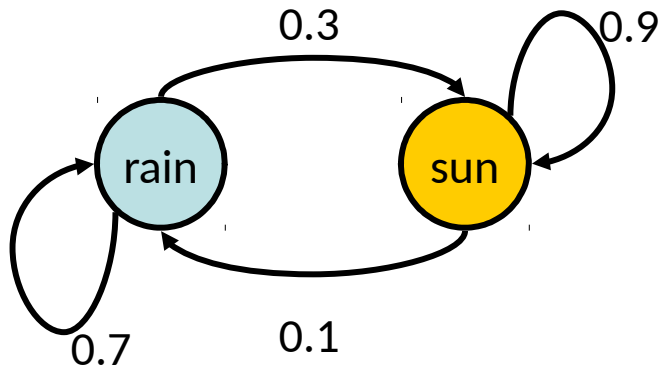
Two new ways of representing the same CPT





Quiz: Example Markov Chain: Weather

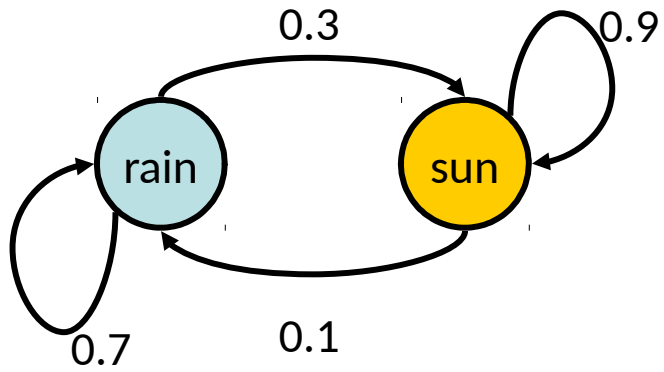
- Initial distribution: 1.0 sun



- What is the probability distribution after one step?
- $P(X_2 = \text{sun}) = ?$

Example Markov Chain: Weather

- Initial distribution: 1.0 sun



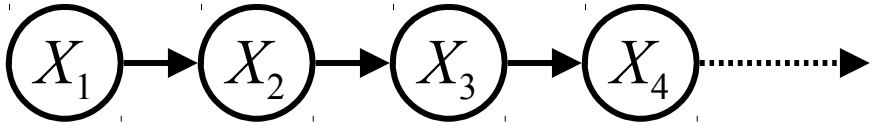
- What is the probability distribution after one step?

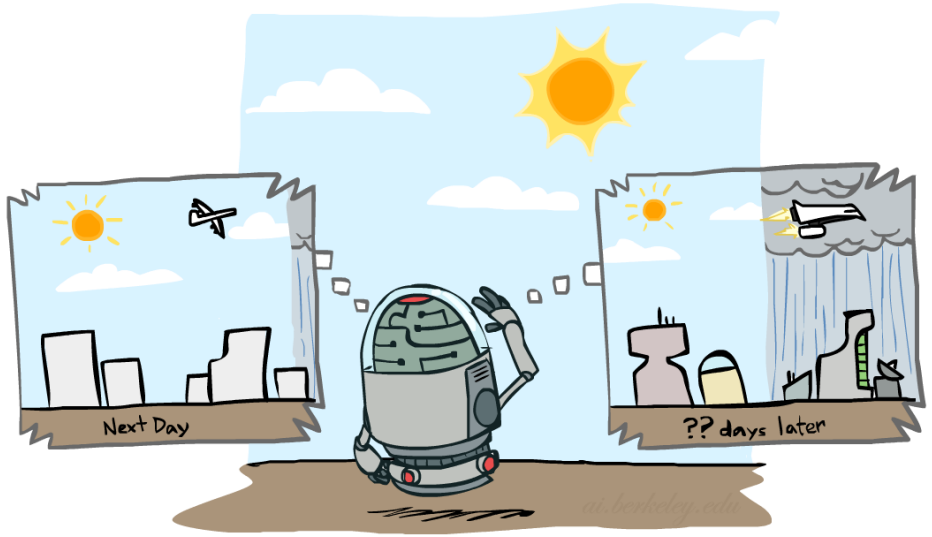
$$P(X_2 = \text{sun}) = P(X_2 = \text{sun}|X_1 = \text{sun})P(X_1 = \text{sun}) + P(X_2 = \text{sun}|X_1 = \text{rain})P(X_1 = \text{rain})$$

$$0.9 \cdot 1.0 + 0.3 \cdot 0.0 = 0.9$$

Mini-Forward Algorithm

- Question: What's P(X) on some day t?





$$P(x_1) = \text{known}$$

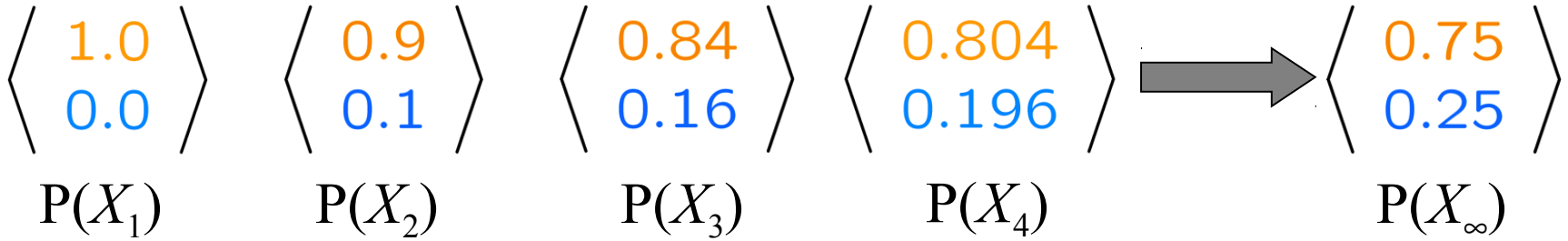
$$P(x_t) = \sum_{x_{t-1}} P(x_{t-1}, x_t)$$

$$= \sum_{x_{t-1}} P(x_t \mid x_{t-1}) P(x_{t-1})$$

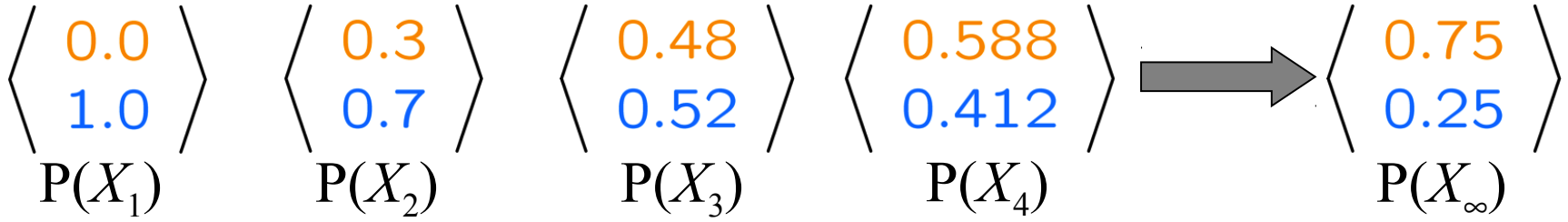
Forward simulation

Example Run of Mini-Forward Algorithm

- From initial observation of sun



- From initial observation of rain



- From yet another initial distribution $P(X_1)$:

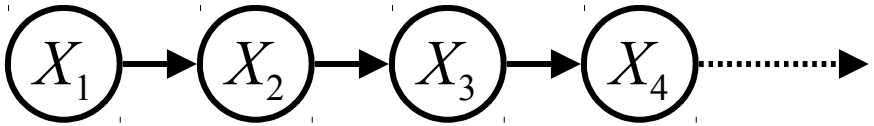
Stationary Distributions

- For most chains:
  - Influence of the initial distribution gets less and less over time.
  - The distribution we end up in is independent of the initial distribution
- Stationary distribution:
  - The distribution we end up with is called the stationary distribution of the chain
  - It satisfies P_∞

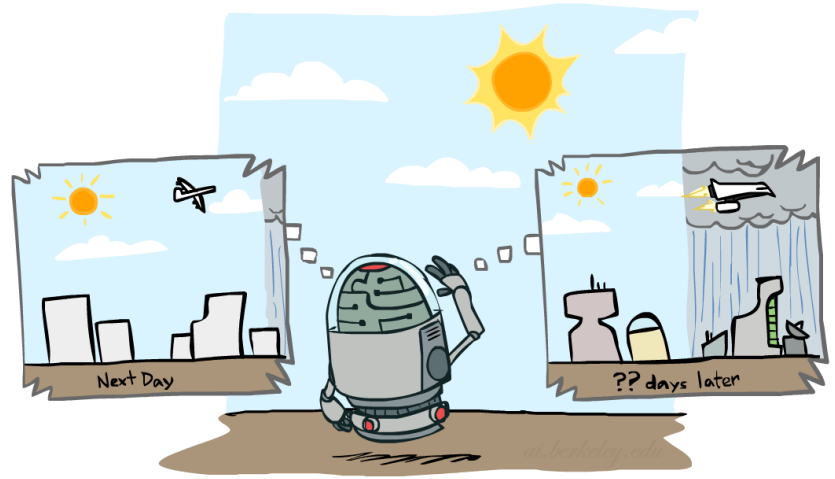
$$P_\infty(X) = P_{\infty+1}(X) = \sum_x P(X|x)P_\infty(x)$$

Quiz: Stationary Distributions

- Question: What's P(X) at time t = infinity?



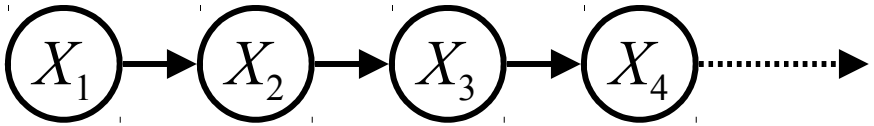
$$P_\infty(sun) = P(sun|sun)P_\infty(sun) + P(sun|rain)P_\infty(rain)$$
$$P_\infty(rain) = P(rain|sun)P_\infty(sun) + P(rain|rain)P_\infty(rain)$$

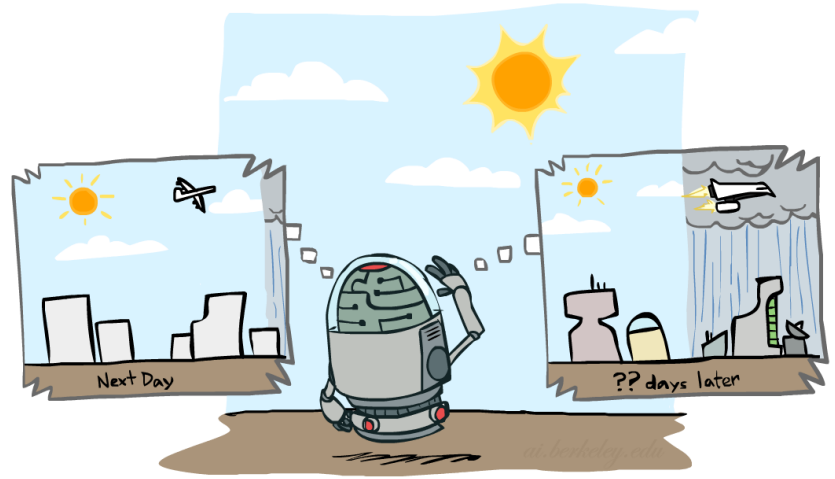


X_{t-1}	X_t	$P(X_t \mid X_{t-1})$
sun	sun	0.9
sun	rain	0.1
rain	sun	0.3
rain	rain	0.7

Quiz: Stationary Distributions

- Question: What's P(X) at time t = infinity?





$$P_\infty(sun) = P(sun|sun)P_\infty(sun) + P(sun|rain)P_\infty(rain)$$
$$P_\infty(rain) = P(rain|sun)P_\infty(sun) + P(rain|rain)P_\infty(rain)$$

$$P_\infty(sun) = 0.9P_\infty(sun) + 0.3P_\infty(rain)$$
$$P_\infty(rain) = 0.1P_\infty(sun) + 0.7P_\infty(rain)$$

$$P_\infty(sun) = 3P_\infty(rain)$$
$$P_\infty(rain) = 1/3P_\infty(sun)$$

Also: $P_\infty(sun) + P_\infty(rain) = 1$

$$P_\infty(sun) = 3/4$$
$$P_\infty(rain) = 1/4$$

X_{t-1}	X_t	$P(X_t \mid X_{t-1})$
sun	sun	0.9
sun	rain	0.1
rain	sun	0.3
rain	rain	0.7

Probability Recap

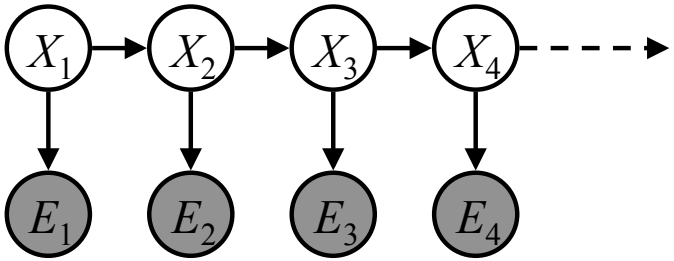
- Conditional probability $P(x|y) = \frac{P(x,y)}{P(y)}$
- Product rule $P(x,y) = P(x|y)P(y)$
- Chain rule $$\begin{aligned} P(X_1, X_2, \ldots X_n) &= P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)\ldots \\ &= \prod_{i=1}^{n} P(X_i|X_1, \ldots, X_{i-1}) \end{aligned}$$
- X, Y independent if and only if: $\forall x, y : P(x,y) = P(x)P(y)$
- X and Y are conditionally independent given Z if and only if: $X \perp\!\!\!\perp Y|Z$

$$\forall x, y, z : P(x,y|z) = P(x|z)P(y|z)$$

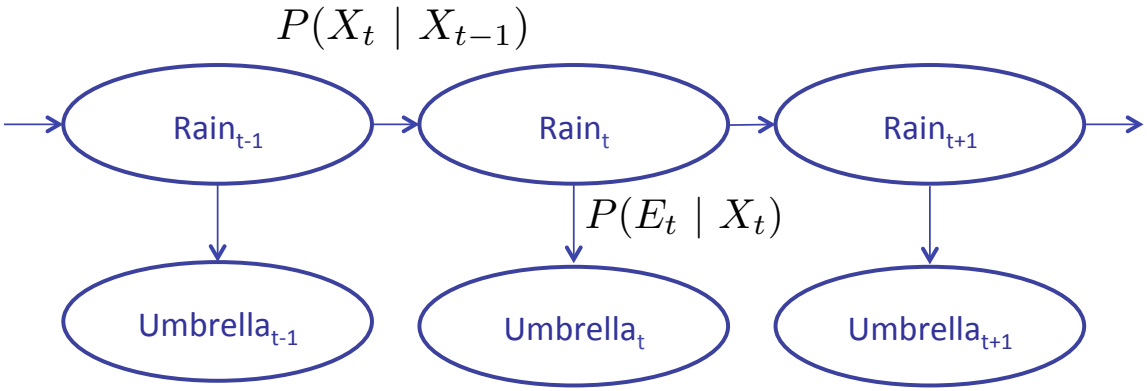
Hidden Markov Models

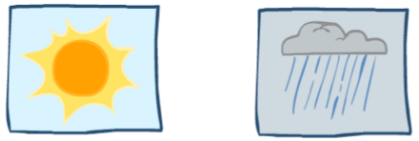
Hidden Markov Models

- Markov chains not so useful for most agents
  - Need observations to update your beliefs
- Hidden Markov models (HMMs)
  - Underlying Markov chain over states X
  - You observe outputs (effects) at each time step



Example: Weather HMM



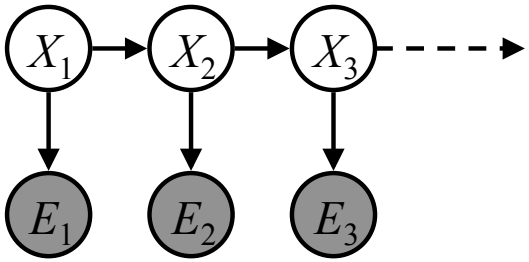


- An HMM is defined by:
  - Initial distribution: $P(X_1)$
  - Transitions: $P(X_t \mid X_{t-1})$
  - Emissions: $P(E_t \mid X_t)$

R_t	R_{t+1}	$P(R_{t+1} \mid R_t)$
+r	+r	0.7
+r	-r	0.3
-r	+r	0.3
-r	-r	0.7

R_t	U_t	$P(U_t \mid R_t)$
+r	+u	0.9
+r	-u	0.1
-r	+u	0.2
-r	-u	0.8

Joint Distribution of an HMM



- Joint distribution:

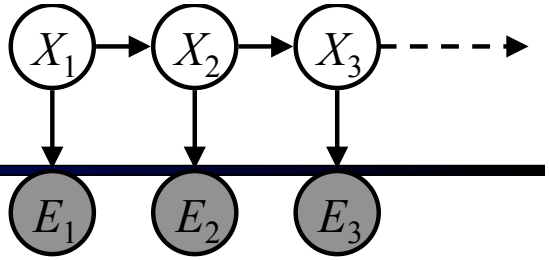
$$P(X_1, E_1, X_2, E_2, X_3, E_3) = P(X_1)P(E_1|X_1)P(X_2|X_1)P(E_2|X_2)P(X_3|X_2)P(E_3|X_3)$$

- More generally:

$$P(X_1, E_1, \ldots, X_T, E_T) = P(X_1)P(E_1|X_1)\prod_{t=2}^{T} P(X_t|X_{t-1})P(E_t|X_t)$$

- Questions to be resolved:
  - Does this indeed define a joint distribution?
  - Can every joint distribution be factored this way, or are we making some assumptions about the joint distribution by using this factorization?

Chain Rule and HMMs



- From the chain rule, *every* joint distribution over $X_1, E_1, X_2, E_2, X_3, E_3$ can be written as:

$$P(X_1, E_1, X_2, E_2, X_3, E_3) = P(X_1)P(E_1|X_1)P(X_2|X_1, E_1)P(E_2|X_1, E_1, X_2)$$
$$P(X_3|X_1, E_1, X_2, E_2)P(E_3|X_1, E_1, X_2, E_2, X_3)$$

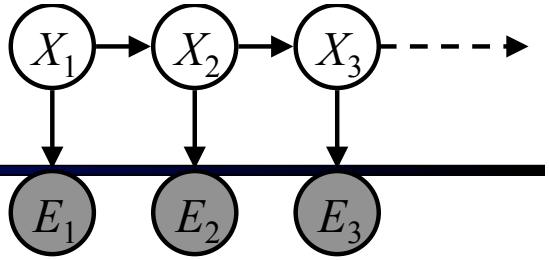
- *Assuming* that

$$X_2 \perp\!\!\!\perp E_1 \mid X_1, \quad E_2 \perp\!\!\!\perp X_1, E_1 \mid X_2, \quad X_3 \perp\!\!\!\perp X_1, E_1, E_2 \mid X_2, \quad E_3 \perp\!\!\!\perp X_1, E_1, X_2, E_2 \mid X_3$$

gives us the expression posited on the previous slide:

$$P(X_1, E_1, X_2, E_2, X_3, E_3) = P(X_1)P(E_1|X_1)P(X_2|X_1)P(E_2|X_2)P(X_3|X_2)P(E_3|X_3)$$

Chain Rule and HMMs



- From the chain rule, *every* joint distribution over $X_1, E_1, \ldots, X_T, E_T$ can be written as:

$$P(X_1, E_1, \ldots, X_T, E_T) = P(X_1)P(E_1|X_1)\prod_{t=2}^{T} P(X_t|X_1, E_1, \ldots, X_{t-1}, E_{t-1})P(E_t|X_1, E_1, \ldots, X_{t-1}, E_{t-1}, X_t)$$

- *Assuming* that for all *t*:
  - State independent of all past states and all past evidence given the previous state, i.e.:

$$X_t \perp\!\!\!\perp X_1, E_1, \ldots, X_{t-2}, E_{t-2}, E_{t-1} \mid X_{t-1}$$

  - Evidence is independent of all past states and all past evidence given the current state, i.e.:

$$E_t \perp\!\!\!\perp X_1, E_1, \ldots, X_{t-2}, E_{t-2}, X_{t-1}, E_{t-1} \mid X_t$$

gives us the expression posited on the earlier slide:

$$P(X_1, E_1, \ldots, X_T, E_T) = P(X_1)P(E_1|X_1)\prod_{t=2}^{T} P(X_t|X_{t-1})P(E_t|X_t)$$

Real HMM Examples

- Speech recognition HMMs:
  - Observations are acoustic signals (continuous valued)
  - States are specific positions in specific words (so, tens of thousands)
- Machine translation HMMs:
  - Observations are words (tens of thousands)
  - States are translation options
- Robot tracking:
  - Observations are range readings (continuous)
  - States are positions on a map (continuous)

Inference in Temporal Models

- **Filtering**: This is the task of computing the belief state—the posterior distribution over the most recent state—given all evidence to date. $\mathbf{P}(\mathbf{X}_t \mid \mathbf{e}_{1:t})$.
  - Umbrella example?
- **Prediction**: This is the task of computing the posterior distribution over the future state, given all evidence to date. $\mathbf{P}(\mathbf{X}_{t+k} \mid \mathbf{e}_{1:t})$ for some k>0. Example?
- **Smoothing**: This is the task of computing the posterior distribution over a past state, given all evidence up to the present. That is, we wish to compute $\mathbf{P}(\mathbf{X}_k \mid \mathbf{e}_{1:t})$ for $0 \leq k < t$.
- **Most likely explanation:** Given a sequence of observations, we might wish to find the sequence of states that is most likely to have generated those observations. $\text{argmax}_{\mathbf{x1:t}} \mathbf{P}(\mathbf{x}_{1:t} \mid \mathbf{e}_{1:t})$.