Uncertain Knowledge and Reasoning

8

8 Uncertain Knowledge and Reasoning

8.1 Uncertainty

8.2 Probability
- Syntax and semantics • Inference
- Independence • Bayes' rule

8.3 Bayesian networks

8.4 Probabilistic reasoning$^+$

8.5 Dynamic Bayesian networks$^+$

8.6 Causal Inference*

8.7 Probabilistic programming*

8.8 Probabilistic logic*

Uncertainty

Let action A_t = leave for airport t minutes before flight
Will A_t get me there on time?

Problems

1) partial observability (road state, other drivers' plans, etc.)
2) noisy sensors (traffic radio)
3) uncertainty in action outcomes (flat tire, etc.), etc.

Hence a purely logical approach either

1) risks falsehood: "A_{25} will get me there on time"

or 2) leads to conclusions that are too weak for decision making

"A_{25} will get me there on time if there's no accident on the bridge and it doesn't rain and my tires remain intact etc."

(A_{1440} might reasonably be said to get me there on time
but I'd have to stay overnight in the airport ...)

Uncertainty knowledge representation and reasoning

Nonmonotonic logic

Assume A_{25} works unless contradicted by evidence

Issues: How to handle quantitation? Reasonable assumptions?

Rules with fudge factors:

$A_{25} \mapsto_{0.3} AtAirportOnTime$

$Sprinkler \mapsto_{0.99} WetGrass$

$WetGrass \mapsto_{0.7} Rain$

Issues: problems with combination, e.g., $Sprinkler$ causes $Rain$?

Fuzzy logic handles **degree of truth** NOT uncertainty e.g.,

$WetGrass$ is true to degree 0.2

Probability

Given the available evidence,

A_{25} will get me there on time with probability 0.04

Qualitative vs. quantitative $\Rightarrow$ Logic vs. probability $\Leftarrow$ Prob. logics

Probability

Probabilistic assertions **summarize** effects of

laziness: failure to enumerate exceptions, qualifications, etc.

ignorance: lack of relevant facts, initial conditions, etc.

Subjective (posterior, conditional, Bayesian) probability

Probabilities relate propositions to one's own state of knowledge

e.g., $P(A_{25}|\text{no reported accidents}) = 0.06$

These are **not** claims of a "probabilistic tendency" in the current situation

(but might be learned from past experience of similar situations)

Probabilities of propositions change with new evidence

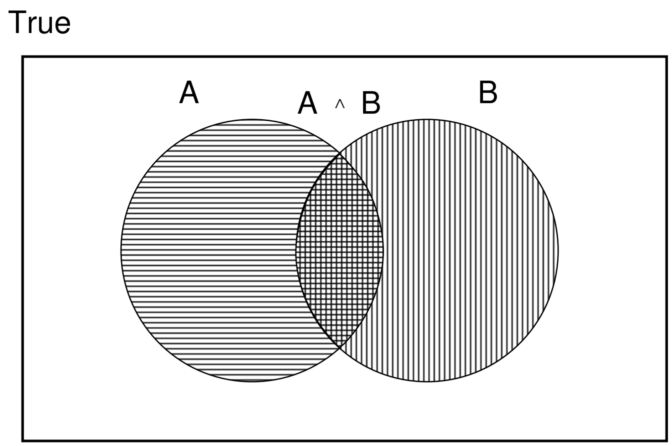
e.g., $P(A_{25}|\text{no reported accidents, 5 a.m.}) = 0.15$

(Analogous to logical entailment $KB \models \alpha$, not truth but nonmonotonic in nature)

Why use probability?

The definitions imply that certain logically related events must have related probabilities

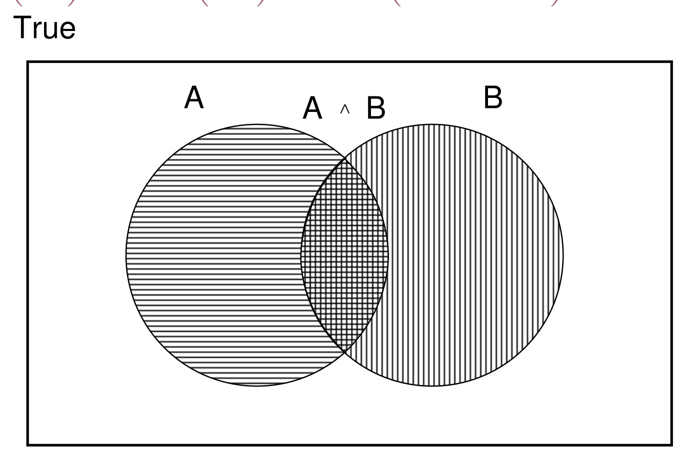
E.g., $P(a \lor b) = P(a) + P(b) - P(a \land b)$



de Finetti (1931): an agent who bets according to probabilities that violate these axioms can be forced to bet so as to lose money regardless of the outcome

Axioms of probability

For any propositions A, B
1. $0 \leq P(A) \leq 1$
2. $P(True) = 1$ and $P(False) = 0$
3. $P(A \vee B) = P(A) + P(B) - P(A \wedge B)$



A probability is a measure over a set of events that satisfies three axioms $\Rightarrow$ probability theory is analogous to logical theory (axioms)

e.g., $P(\neg a) = 1 - P(a)$ is derived from the axioms

$P(a \vee b) = P(a) + P(b) - P(a \wedge b)$ (inclusion-exclusion principle)

Syntax and semantics

Traditional probability theory has informal language
needs to be formalized for agents

Begin with a set Ω — sample space
e.g., 6 possible rolls of a die
$\omega \in \Omega$ is a sample point (outcome/possible world/atomic event/data)
A probability space or probability model is a sample space
with an assignment $P(\omega)$ for every $\omega \in \Omega$ s.t.
(1) $0 \leq P(\omega) \leq 1$
(2) $\Sigma_\omega P(\omega) = 1$
e.g., $P(1) = P(2) = P(3) = P(4) = P(5) = P(6) = 1/6$
An event A is any subset of Ω

$$P(A) = \Sigma_{\{\omega \in A\}} P(\omega)$$

e.g., $P(\text{die roll} < 4) = P(1) + P(2) + P(3) = 1/6 + 1/6 + 1/6 = 1/2$

Random variables

A random variable is a function from sample points to some **range**

– Booleans (propositions)

e.g., $Cavity$ (do I have a cavity?)

$Cavity = true$ is a proposition, also written $Cavity$

– Discrete (finite or infinite)

e.g., $Weather$ is one of $\langle sunny, rain, cloudy, snow \rangle$

$Weather = rain$ is a proposition

Values must be exhaustive and mutually exclusive

– Continuous or real (bounded or unbounded)

e.g., $Temp = 21.6$; also allow, e.g., $Temp < 22.0$

Arbitrary Boolean combinations of basic propositions

Probability distribution

P induces a (prob.) distribution for any r.v. (random variable) X

$$P(X=x_i) = \Sigma_{\{\omega : X(\omega)=x_i\}} P(\omega)$$

gives values for all possible assignments

E.g., $P(Odd=true) = P(1)+P(3)+P(5) = 1/6+1/6+1/6 = 1/2$

The probability of a proposition $Odd=true$ as the sum of the probabilities of worlds in which it holds

Propositions

Think of a proposition as the event (set of sample points)
where the proposition is true

Given Boolean r.v.s A and B

event a = set of sample points where $A(\omega) = true$
event $\neg a$ = set of sample points where $A(\omega) = false$
event $a \wedge b$ = points where $A(\omega) = true$ and $B(\omega) = true$

The sample points are defined by the values of a set of r.v.s
i.e., the sample space is Cartesian product of the ranges of the r.v.s

Propositions

For Boolean r.v.s

sample point (possible world) = propositional logic model

e.g., $A = true$, $B = false$, or $a \wedge \neg b$

A possible world is defined to be an assignment of values to all of the r.v.s under consideration

– possible worlds are mutually exclusive and exhaustive, why??

Proposition = disjunction of atomic events (clausal form)

e.g., $(a \vee b) \equiv (\neg a \wedge b) \vee (a \wedge \neg b) \vee (a \wedge b)$

$\Rightarrow P(a \vee b) = P(\neg a \wedge b) + P(a \wedge \neg b) + P(a \wedge b)$

For any proposition ϕ, the possible world (model) ω where it is true

$\omega \models \phi$

Hint: (propositional) logic + probability $\Rightarrow$ probabilistic logic

Prior probability

Prior (unconditional probabilities) of propositions

e.g., $P(Cavity = true) = 0.1$

$P(Weather = sunny) = 0.72$

correspond to belief prior to the arrival of any (new) evidence

Probability distribution gives values for all possible assignments

$\mathbf{P}(Weather) = \langle 0.72, 0.1, 0.08, 0.1 \rangle$

(normalized, i.e., sums to 1)

Joint probability distribution

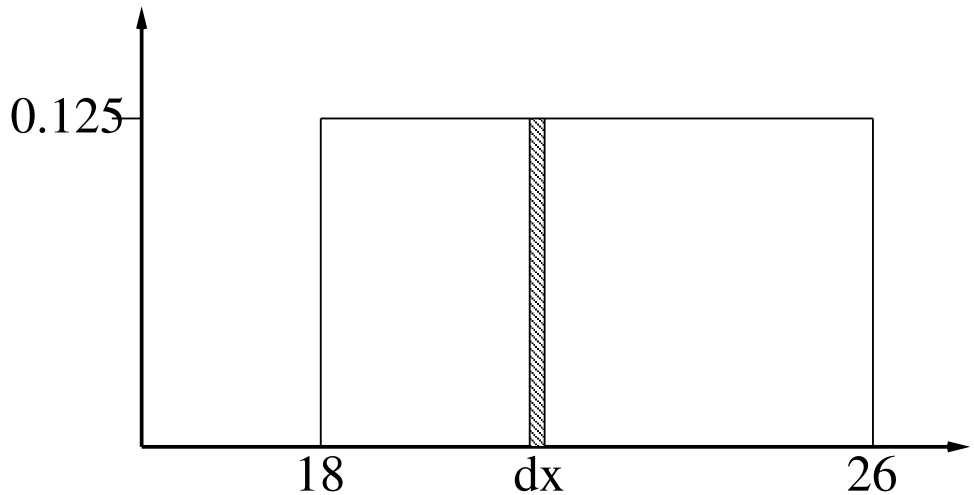
Joint probability distribution for a set of r.v.s gives the probability of every atomic event on those r.v.s
(i.e., every sample point)
$\mathbf{P}(Weather, Cavity) =$ a 4×2 matrix of values

$Weather =$	$sunny$	$rain$	$cloudy$	$snow$
$Cavity = true$	0.144	0.02	0.016	0.02
$Cavity = false$	0.576	0.08	0.064	0.08

Every question about a domain can be answered by the joint distribution because every event is a sum of sample points

Probability for continuous variables#

Express distribution as a parameterized function of value
$P(X=x) = U[18,26](x) =$ uniform density between 18 and 26



Here P is a density; integrates to 1
$P(X=20.5) = 0.125$ really means

$$\lim_{dx\to 0} P(20.5 \le X \le 20.5 + dx)/dx = 0.125$$

Conditional probability

Conditional (posterior) probabilities

e.g., $P(cavity|toothache) = 0.8$

i.e., given evidence that $toothache$ is all I know

NOT "if $toothache$ then 80% chance of $cavity$"

Full joint (conditional probability) distribution for all of the r.v.s

$\mathbf{P}(Cavity|Toothache) =$ 2-element vector of 2-element vectors

If we know more, e.g., $cavity$ is also given, then we have

$P(cavity|toothache, cavity) = 1$

Note: the less specific belief remains valid after more evidence arrives, but is not always useful

New evidence may be irrelevant, allowing simplification, e.g.,

$P(cavity|toothache, 49ersWin) = P(cavity|toothache) = 0.8$

This kind of inference, sanctioned by domain knowledge, is crucial

Conditional probability

Defn. of conditional probability by unconditional probabilities

$$P(a|b) = \frac{P(a \wedge b)}{P(b)} \text{ if } P(b) \neq 0$$

Product rule gives an alternative formulation
$P(a \wedge b) = P(a|b)P(b) = P(b|a)P(a)$

A general version holds for whole distributions, e.g.,
$\mathbf{P}(Weather, Cavity) = \mathbf{P}(Weather|Cavity)\mathbf{P}(Cavity)$
(View as a 4×2 set of equations, **not** matrix mult.)

Chain rule is derived by successive application of product rule

$$\begin{aligned}\mathbf{P}(X_1, \ldots, X_n) &= \mathbf{P}(X_1, \ldots, X_{n-1})\, \mathbf{P}(X_n|X_1, \ldots, X_{n-1}) \\ &= \mathbf{P}(X_1, \ldots, X_{n-2})\, \mathbf{P}(X_{n-1}|X_1, \ldots, X_{n-2})\, \mathbf{P}(X_n|X_1, \ldots, X_{n-1}) \\ &= \ldots \\ &= \prod_{i=1}^{n} \mathbf{P}(X_i|X_1, \ldots, X_{i-1})\end{aligned}$$

Inference

Probabilistic inference is the computation of posterior probabilities for query propositions given observed evidence
where the full joint distribution can be viewed as the KB
from which answers to all questions may be derived

Start with the joint distribution

	toothache		$\neg$ *toothache*	
	catch	$\neg$ *catch*	*catch*	$\neg$ *catch*
cavity	**.108**	**.012**	**.072**	**.008**
$\neg$ *cavity*	**.016**	**.064**	**.144**	**.576**

For any proposition ϕ, sum the atomic events where it is true
$P(\phi) = \Sigma_{\omega:\omega\models\phi} P(\omega)$

Inference by enumeration

Start with the joint distribution

	toothache		*¬ toothache*	
	catch	*¬ catch*	*catch*	*¬ catch*
cavity	**.108**	**.012**	**.072**	**.008**
¬ cavity	**.016**	**.064**	**.144**	**.576**

For any proposition ϕ, sum the atomic events where it is true
$P(\phi) = \Sigma_{\omega:\omega\models\phi} P(\omega)$

E.g., $P(toothache) = 0.108 + 0.012 + 0.016 + 0.064 = 0.2$

Inference by enumeration

Start with the joint distribution

	toothache		¬ *toothache*	
	catch	¬ *catch*	*catch*	¬ *catch*
cavity	**.108**	**.012**	**.072**	**.008**
¬ *cavity*	**.016**	**.064**	**.144**	**.576**

For any proposition ϕ, sum the atomic events where it is true

$P(\phi) = \Sigma_{\omega:\omega \models \phi} P(\omega)$

E.g., $P(cavity \lor toothache) = 0.108 + 0.012 + 0.072 + 0.008 + 0.016 + 0.064 = 0.28$

Inference by enumeration

Start with the joint distribution

	toothache		¬ *toothache*	
	catch	¬ *catch*	*catch*	¬ *catch*
cavity	**.108**	**.012**	**.072**	**.008**
¬ *cavity*	**.016**	**.064**	**.144**	**.576**

Can also compute conditional probabilities

$$
\begin{aligned}
P(\neg cavity|toothache) &= \frac{P(\neg cavity \wedge toothache)}{P(toothache)} \\
&= \frac{0.016 + 0.064}{0.108 + 0.012 + 0.016 + 0.064} = 0.4
\end{aligned}
$$

Normalization

	toothache		*¬ toothache*	
	catch	*¬ catch*	*catch*	*¬ catch*
cavity	**.108**	**.012**	**.072**	**.008**
¬ cavity	**.016**	**.064**	**.144**	**.576**

Denominator can be viewed as a normalization constant α

$$\mathbf{P}(Cavity|toothache) = \alpha\,\mathbf{P}(Cavity, toothache)$$
$$= \alpha\,[\mathbf{P}(Cavity, toothache, catch) + \mathbf{P}(Cavity, toothache, \neg catch)]$$
$$= \alpha\,[\langle 0.108, 0.016\rangle + \langle 0.012, 0.064\rangle]$$
$$= \alpha\,\langle 0.12, 0.08\rangle = \langle 0.6, 0.4\rangle$$

Idea: compute distribution on query variable
by fixing evidence variables and summing over hidden variables

Inference by enumeration contd.

Let $\mathbf{X}$ be all the variables. ASK
the posterior joint distribution of the query variables $\mathbf{Y}$
given specific values $\mathbf{e}$ for the evidence variables $\mathbf{E}$

Let the hidden variables be $\mathbf{H} = \mathbf{X} - \mathbf{Y} - \mathbf{E}$
$\Rightarrow$ the required summation of joint entries is done by summing out the hidden variables:

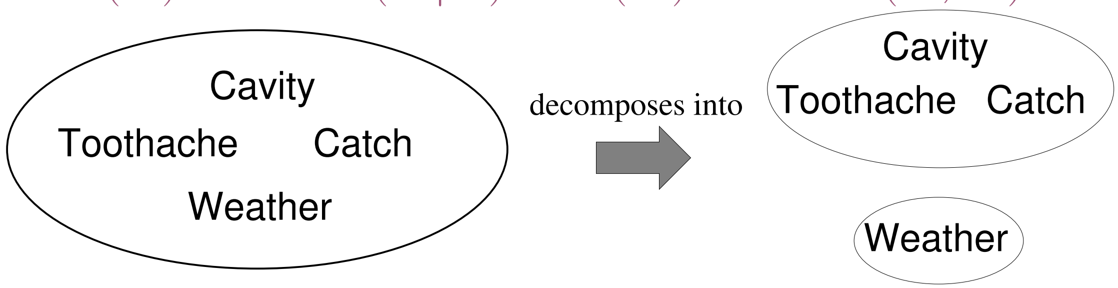
$$\mathbf{P}(\mathbf{Y}|\mathbf{E}=\mathbf{e}) = \alpha\mathbf{P}(\mathbf{Y},\mathbf{E}=\mathbf{e}) = \alpha\Sigma_{\mathbf{h}}\mathbf{P}(\mathbf{Y},\mathbf{E}=\mathbf{e},\mathbf{H}=\mathbf{h})$$

The terms in the summation are joint entries because $\mathbf{Y}$, $\mathbf{E}$, and $\mathbf{H}$ together exhaust the set of random variables

Problems
1) Worst-case time complexity $O(d^n)$ where d is the largest arity
2) Space complexity $O(d^n)$ to store the joint distribution
3) How to find the numbers for $O(d^n)$ entries?

Independence

A and B are independent iff
$\mathbf{P}(A|B) = \mathbf{P}(A)$ or $\mathbf{P}(B|A) = \mathbf{P}(B)$ or $\mathbf{P}(A, B) = \mathbf{P}(A)\mathbf{P}(B)$



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

32 entries reduced to 12; for n independent biased coins, $2^n \to n$

Absolute independence is powerful but rare

Dentistry is a large field with hundreds of variables,
none of which are independent. What to do?

Conditional independence

$\mathbf{P}(Toothache, Cavity, Catch)$ has $2^3 - 1 = 7$ independent entries

If I have a cavity, the probability that the probe catches in it doesn't depend on whether I have a toothache

(1) $P(catch|toothache, cavity) = P(catch|cavity)$

The same independence holds if I haven't got a cavity

(2) $P(catch|toothache, \neg cavity) = P(catch|\neg cavity)$

$Catch$ is conditionally independent of $Toothache$ given $Cavity$

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

Equivalent statements

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

Conditional independence

Write out full joint distribution using chain rule

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

i.e., $2 + 2 + 1 = 5$ independent numbers

In most cases, the use of conditional independence reduces the size of the representation of the joint distribution from exponential in n to linear in n

Conditional independence is our most basic and robust form of knowledge about uncertainty

Bayes' rule

Product rule $P(a \wedge b) = P(a|b)P(b) = P(b|a)P(a)$

$$\Rightarrow \text{Bayes' rule } P(a|b) = \frac{P(b|a)P(a)}{P(b)}$$

or in distribution form

$$\mathbf{P}(Y|X) = \frac{\mathbf{P}(X|Y)\mathbf{P}(Y)}{\mathbf{P}(X)} = \alpha\mathbf{P}(X|Y)\mathbf{P}(Y)$$

Useful for assessing diagnostic probability from causal probability

$$P(Cause|Effect) = \frac{P(Effect|Cause)P(Cause)}{P(Effect)}$$

E.g., let M be meningitis, S be stiff neck

$$P(m|s) = \frac{P(s|m)P(m)}{P(s)} = \frac{0.8 \times 0.0001}{0.1} = 0.0008$$

Naive Bayes

Bayes' rule and conditional independence

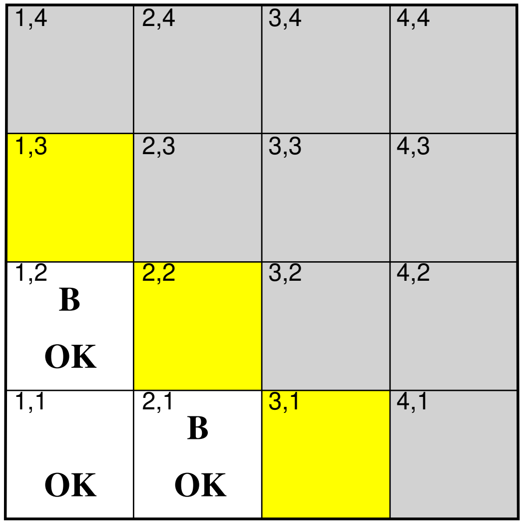
$$\mathbf{P}(Cavity|toothache \wedge catch)$$
$$= \alpha\, \mathbf{P}(toothache \wedge catch|Cavity)\mathbf{P}(Cavity)$$
$$= \alpha\, \mathbf{P}(toothache|Cavity)\mathbf{P}(catch|Cavity)\mathbf{P}(Cavity)$$

This is an example of a naive Bayes model (Bayesian classifier)

$$\mathbf{P}(Cause, Effect_1, \ldots, Effect_n) = \mathbf{P}(Cause)\Pi_i \mathbf{P}(Effect_i|Cause)$$

Total number of parameters is **linear** in n

Example: Wumpus World



$P_{ij} = true$ iff $[i, j]$ contains a pit

$B_{ij} = true$ iff $[i, j]$ is breezy

Include only $B_{1,1}, B_{1,2}, B_{2,1}$ in the probability model

Specifying the probability model

The full joint distribution is $\mathbf{P}(P_{1,1}, \ldots, P_{4,4}, B_{1,1}, B_{1,2}, B_{2,1})$

Apply product rule: $\mathbf{P}(B_{1,1}, B_{1,2}, B_{2,1} \mid P_{1,1}, \ldots, P_{4,4})\mathbf{P}(P_{1,1}, \ldots, P_{4,4})$

(Do it this way to get $P(Effect|Cause)$)

First term: 1 if pits are adjacent to breezes, 0 otherwise

Second term: pits are placed randomly, probability 0.2 per square:

$$\mathbf{P}(P_{1,1}, \ldots, P_{4,4}) = \Pi_{i,j=1,1}^{4,4}\mathbf{P}(P_{i,j}) = 0.2^n \times 0.8^{16-n}$$

for n pits

Observations and query

We know the following facts:

$b = \neg b_{1,1} \wedge b_{1,2} \wedge b_{2,1}$
$known = \neg p_{1,1} \wedge \neg p_{1,2} \wedge \neg p_{2,1}$

Query is $\mathbf{P}(P_{1,3}|known, b)$

Define $Unknown = P_{ij}$s other than $P_{1,3}$ and $Known$

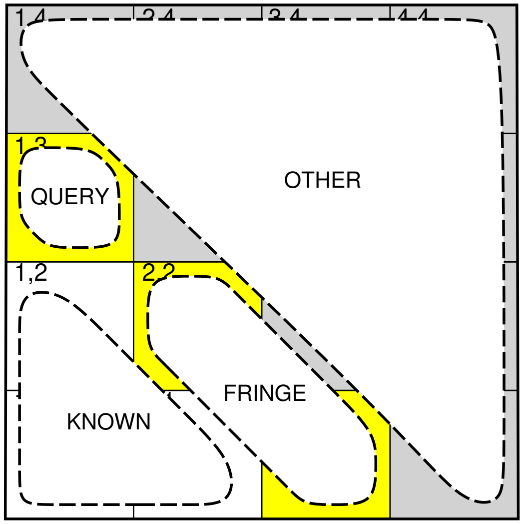
For inference by enumeration, we have

$$\mathbf{P}(P_{1,3}|known, b) = \alpha \Sigma_{unknown} \mathbf{P}(P_{1,3}, unknown, known, b)$$

Grows exponentially with number of squares

Using conditional independence

Basic insight: observations are conditionally independent of other hidden squares, given neighboring hidden squares



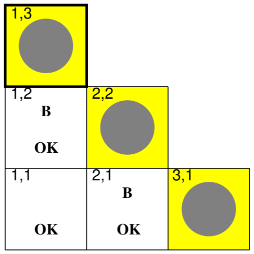
Define $Unknown = Fringe \cup Other$
$\mathbf{P}(b|P_{1,3}, Known, Unknown) = \mathbf{P}(b|P_{1,3}, Known, Fringe)$

Manipulate the query into a form where we can use this

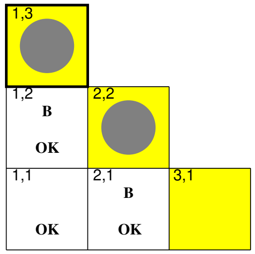
Using conditional independence#

$$
\begin{aligned}
\mathbf{P}(P_{1,3}|known, b) = \alpha \sum_{unknown} \mathbf{P}(P_{1,3}, unknown, known, b) \\
&= \alpha \sum_{unknown} \mathbf{P}(b|P_{1,3}, known, unknown)\mathbf{P}(P_{1,3}, known, unknown) \\
&= \alpha \sum_{fringe} \sum_{other} \mathbf{P}(b|known, P_{1,3}, fringe, other)\mathbf{P}(P_{1,3}, known, fringe, other) \\
&= \alpha \sum_{fringe} \sum_{other} \mathbf{P}(b|known, P_{1,3}, fringe)\mathbf{P}(P_{1,3}, known, fringe, other) \\
&= \alpha \sum_{fringe} \mathbf{P}(b|known, P_{1,3}, fringe) \sum_{other} \mathbf{P}(P_{1,3}, known, fringe, other) \\
&= \alpha \sum_{fringe} \mathbf{P}(b|known, P_{1,3}, fringe) \sum_{other} \mathbf{P}(P_{1,3})P(known)P(fringe)P(other) \\
&= \alpha\, P(known)\mathbf{P}(P_{1,3}) \sum_{fringe} \mathbf{P}(b|known, P_{1,3}, fringe)P(fringe) \sum_{other} P(other) \\
&= \alpha'\, \mathbf{P}(P_{1,3}) \sum_{fringe} \mathbf{P}(b|known, P_{1,3}, fringe)P(fringe)
\end{aligned}
$$

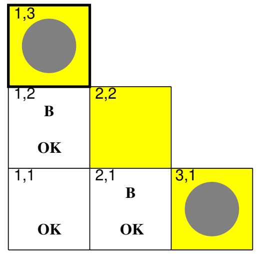
Using conditional independence



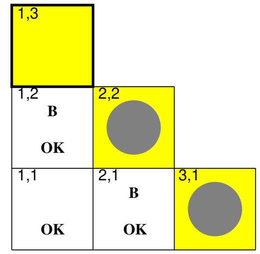
0.2 x 0.2 = 0.04



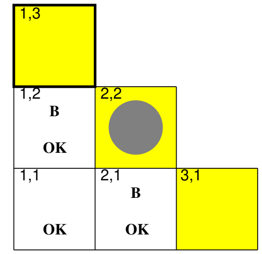
0.2 x 0.8 = 0.16



0.8 x 0.2 = 0.16



0.2 x 0.2 = 0.04



0.2 x 0.8 = 0.16

$$\begin{aligned}\mathbf{P}(P_{1,3}|known, b) &= \alpha' \langle 0.2(0.04+0.16+0.16),\ 0.8(0.04+0.16)\rangle \\ &\approx \langle 0.31, 0.69\rangle\end{aligned}$$

$$\mathbf{P}(P_{2,2}|known, b) \approx \langle 0.86, 0.14\rangle$$

Bayesian networks

BNs: a graphical notation for conditional independence assertions
and hence for compact specification of full joint distributions
    alias Probabilistic Graphical Models (PGMs)

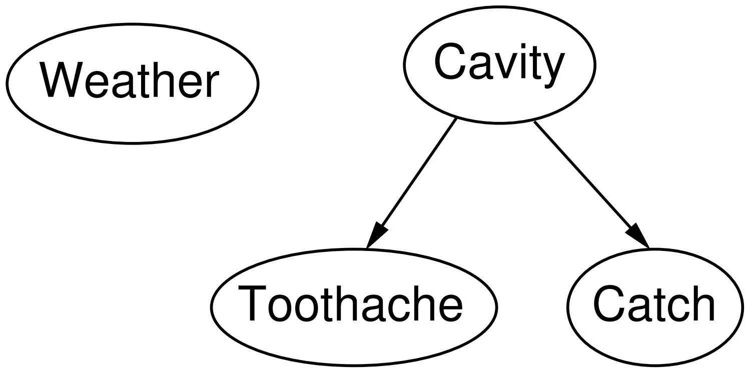
Syntax

a set of nodes, one per variable
a directed acyclic graph (DAG, link $\rightarrow$ "directly influences")
a conditional distribution for each node given its parents
$\mathbf{P}(X_i|Parents(X_i))$

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

Example: Bayesian networks

Topology of the network encodes conditional independence assertions



$Weather$ is independent of the other variables

$Toothache$ and $Catch$ are conditionally independent given $Cavity$

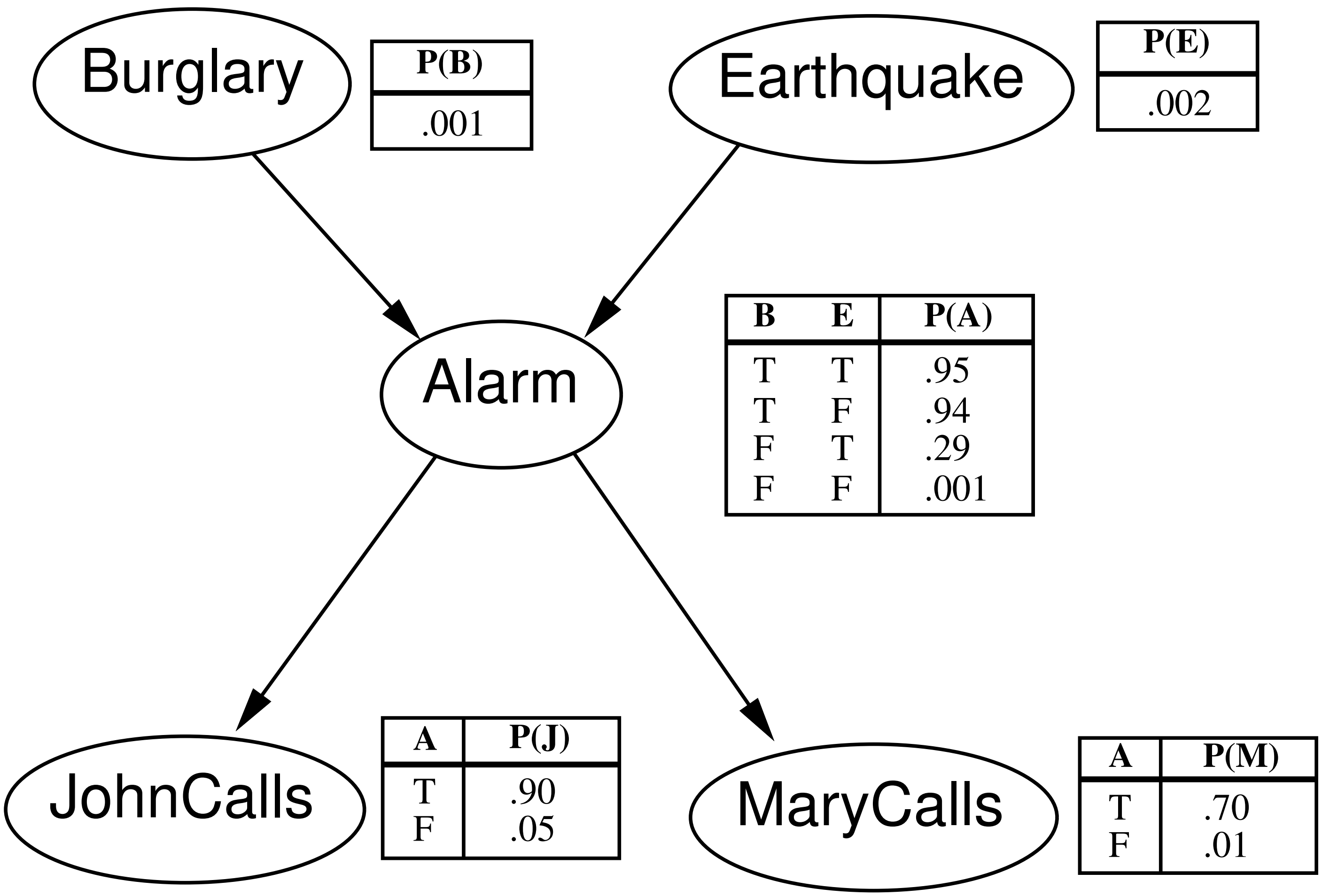
Example: burglary network

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

Example: burglary network



P(B)
.001

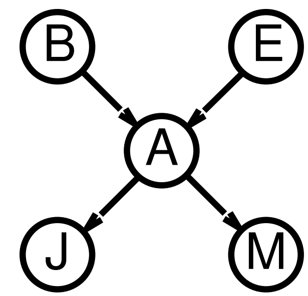
P(E)
.002

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

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. $2^5 - 1 = 31$)

In certain cases (assumptions of conditional independency), BNs make
$O(2^n) \Rightarrow O(kn)$ (NP $\Rightarrow$ P !)

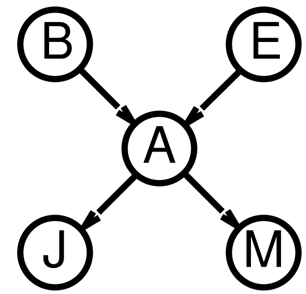
Global semantics$^+$

Global semantics defines the full joint distribution
as the product of the local conditional distributions

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

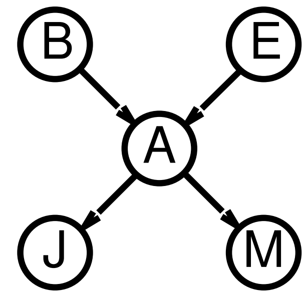
e.g., $P(j \land m \land a \land \neg b \land \neg e)$

$$=$$



Global semantics$^+$

Global semantics defines the full joint distribution
as the product of the local conditional distributions



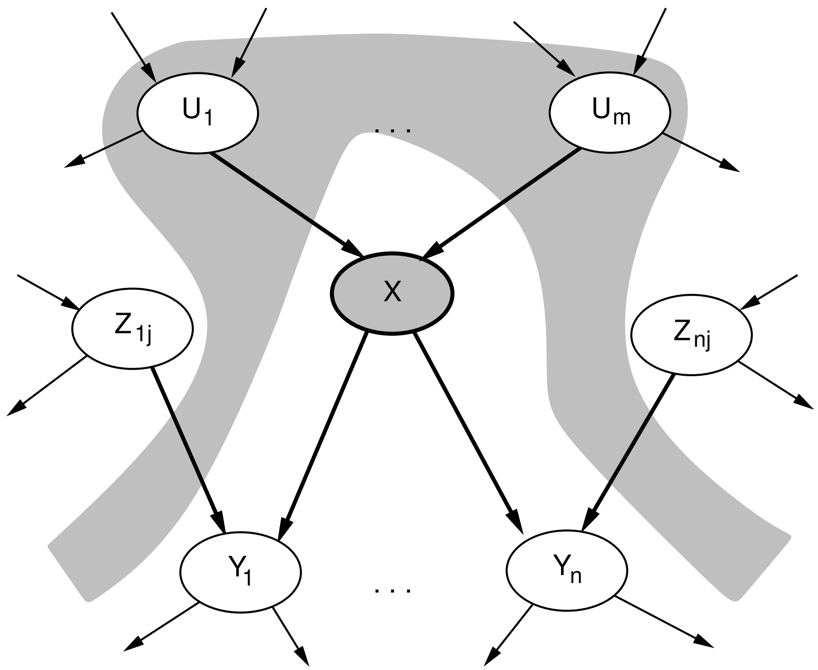
$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i))$$

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

$$\begin{aligned}
&= P(j|a)P(m|a)P(a|\neg b, \neg e)P(\neg b)P(\neg e) \\
&= 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998 \\
&\approx 0.00063
\end{aligned}$$

Local semantics$^+$

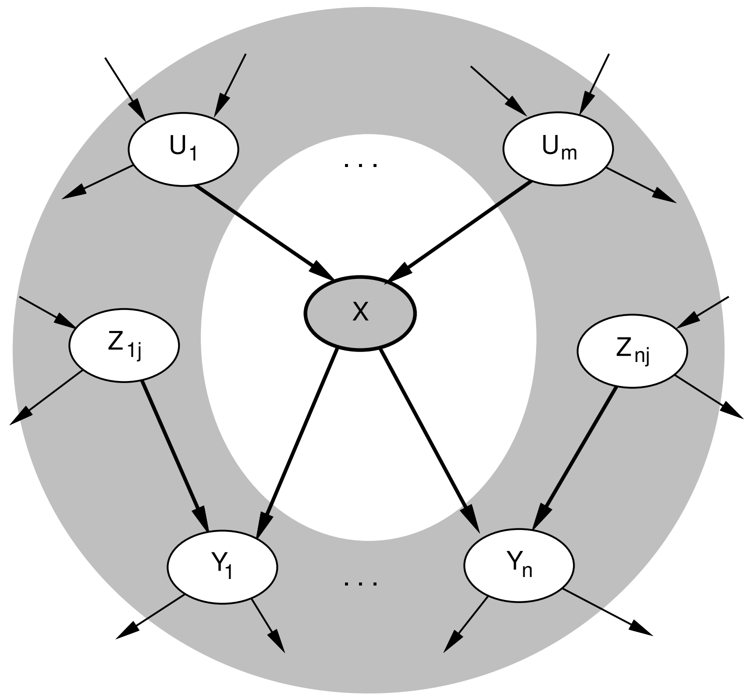
Local semantics: each node is conditionally independent of its nondescendants ($Z_{i,j}$) given its parents (U_i in the gray area)



Theorem: Local semantics $\Leftrightarrow$ global semantics

Markov blanket$^+$

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



Constructing Bayesian networks

Algorithm: 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
   add X_i to the network
   select parents from $X_1, \ldots, X_{i-1}$ such that
   $$\mathbf{P}(X_i | Parents(X_i)) = \mathbf{P}(X_i | X_1,\, \ldots,\, X_{i-1})$$

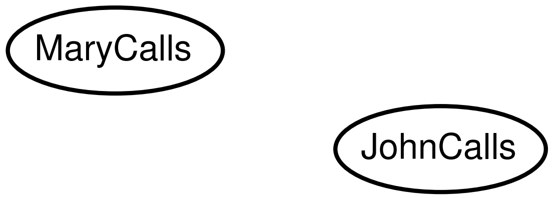
This choice of parents guarantees the global semantics:

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

Each node is conditionally independent of its other predecessors in the node (partial) ordering, given its parents

Example: burglary network

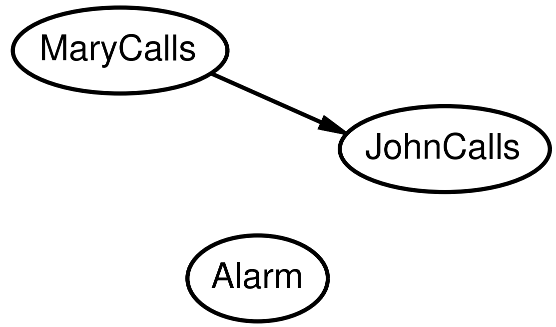
Suppose we choose the ordering M, J, A, B, E



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

Example: burglary network

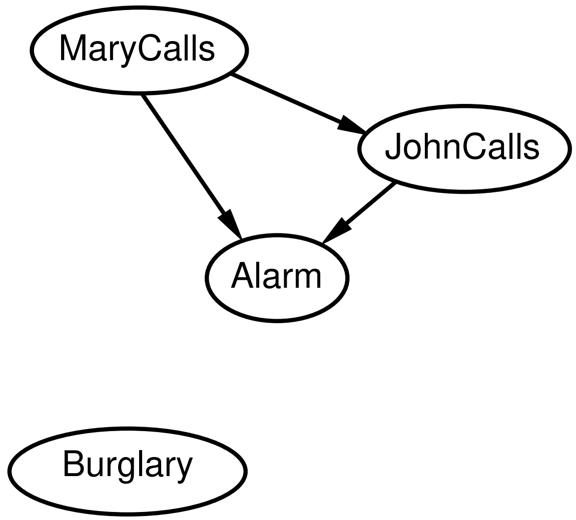
Suppose we choose the ordering M, J, A, B, E



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

Example: burglary network

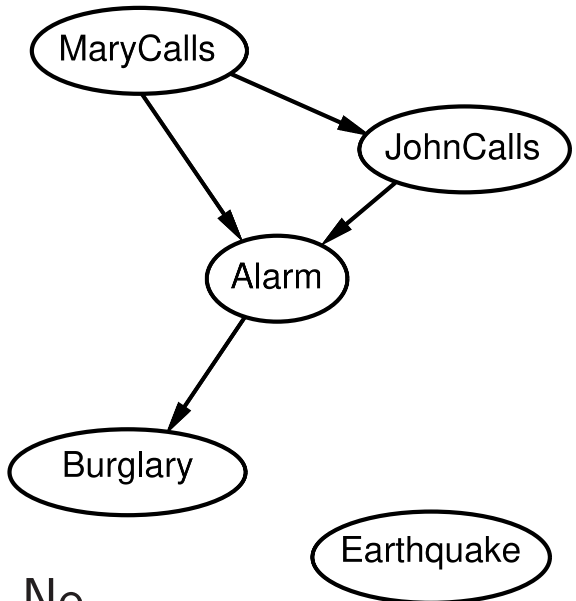
Suppose we choose the ordering M, J, A, B, E



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

Example: burglary network

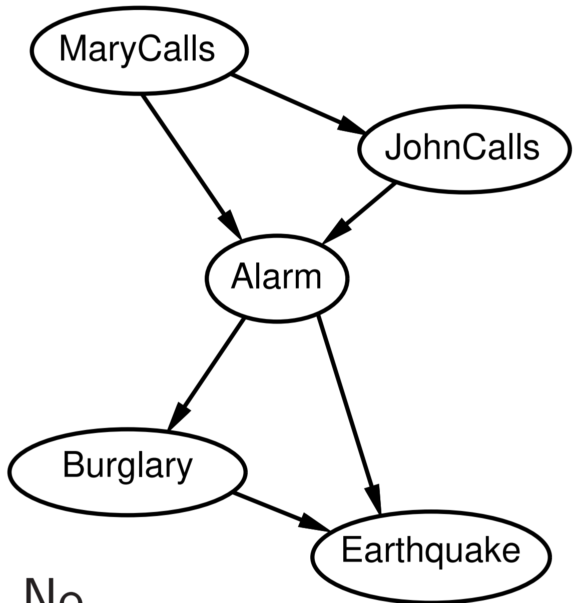
Suppose we choose the ordering M, J, A, B, E



$P(J|M) = P(J)$? 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
$P(E|B,A,J,M) = P(E|A)$?
$P(E|B,A,J,M) = P(E|A,B)$?

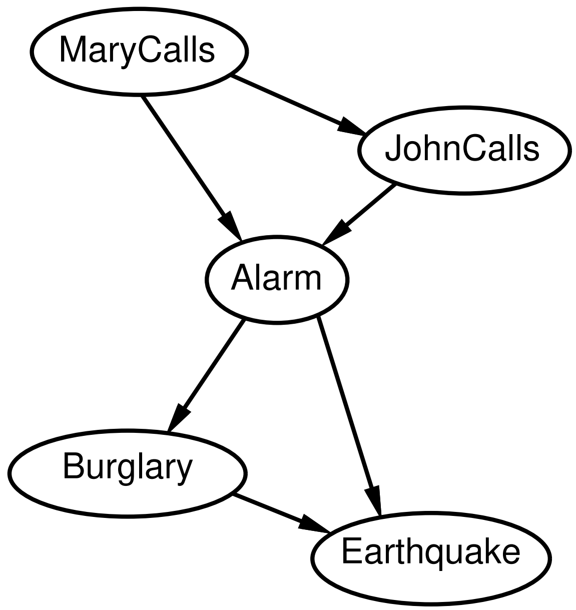
Example: burglary network

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: burglary network



Assessing conditional probabilities is hard in noncausal directions
The network can be far more compact than the full joint distribution
But, this network is less compact: $1+2+4+2+4=13$
(due to the ordering of the variables)

Probabilistic reasoning+

- Exact inference
  enumeration
  variable elimination

- Approximate inference*
  stochastic simulation
  Markov chain Monte Carlo

Reasoning tasks in BNs (PGMs)#

Simple queries: compute posterior marginal $\mathbf{P}(X_i|\mathbf{E}=\mathbf{e})$
e.g., $P(NoGas|Gauge=empty, Lights=on, Starts=false)$

Conjunctive queries: $\mathbf{P}(X_i, X_j|\mathbf{E}=\mathbf{e}) = \mathbf{P}(X_i|\mathbf{E}=\mathbf{e})\mathbf{P}(X_j|X_i, \mathbf{E}=\mathbf{e})$

Optimal decisions: decision networks include utility information
probabilistic inference required for $P(outcome|action, evidence)$

Value of information: which evidence to seek next?

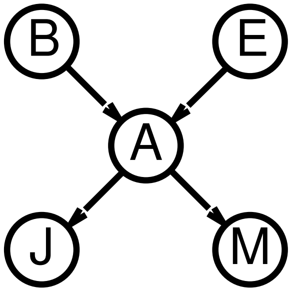
Sensitivity analysis: which probability values are most critical?

Explanation/Causal inference: why do I need a nucleic acid detection (for coronavirus)?

Inference by enumeration

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

Simple query on the burglary network



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

Rewrite full joint entries using product of CPT entries

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

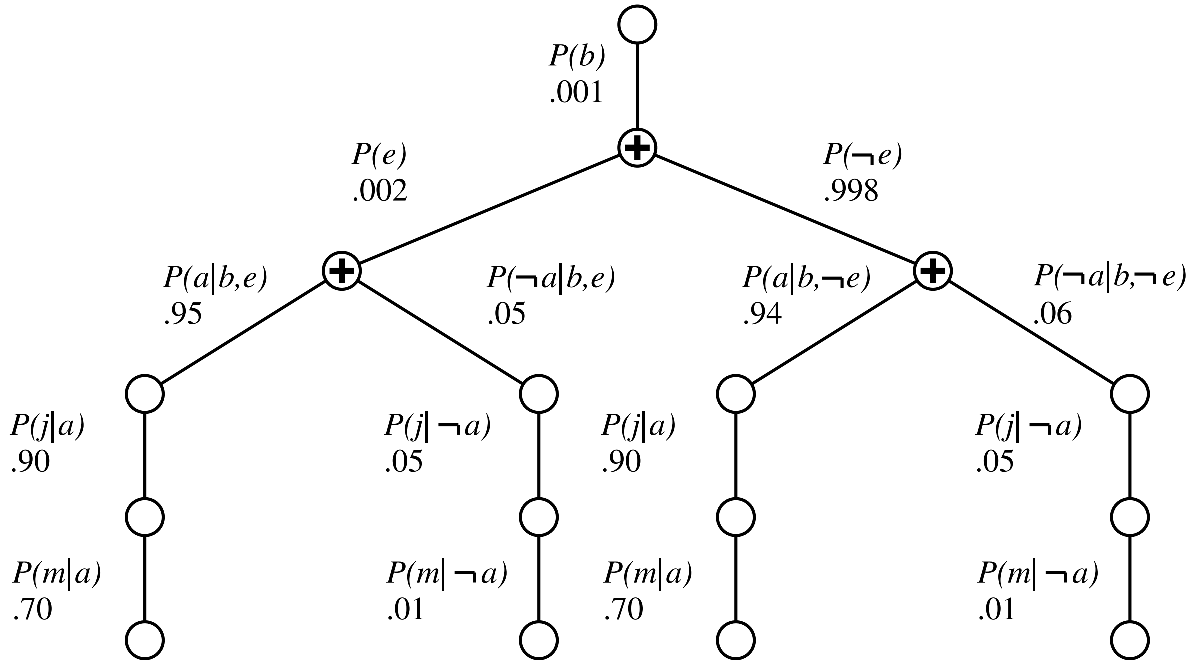
Recursive depth-first enumeration: $O(n)$ space, $O(d^n)$ time

Enumeration algorithm#

```
def ENUMERATION-ASK(X,e,bn)
inputs: X, the query variable
        e, observed values for variables E
        bn, a Bayes net with variables vars

   Q(X) ← a distribution over X, initially empty
   for each value x_i of X do
        Q(x_i) ← ENUMERATE-ALL(vars, e_{x_i})
           where e_{x_i} is e extended with X = x_i
   return NORMALIZE(Q(X))  // a distribution over X
def ENUMERATE-ALL(vars,e)
   if EMPTY?(vars) then return 1.0
   V ← FIRST(vars)
   if V is an evidence variable with value v in e
      then return P(v | parents(V)) × ENUMERATE-ALL(REST(vars),e)
      else return Σ_v P(v | parents(V)) × ENUMERATE-ALL(REST(vars),e_v)
         where e_v is e extended with V = v
```

Evaluation tree

Summing at the "+" nodes



Enumeration is inefficient: repeated computation
e.g., computes $P(j|a)P(m|a)$ for each value of e
improved by eliminating repeated variables

Inference by variable elimination

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

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

Variable elimination: Basic operations

Summing out a variable from a product of factors
move any constant factors outside the summation
add up submatrices in the 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

$$f_1(x_1, \ldots, x_j, y_1, \ldots, y_k) \times f_2(y_1, \ldots, y_k, z_1, \ldots, z_l)$$
$$= f(x_1, \ldots, x_j, y_1, \ldots, y_k, z_1, \ldots, z_l)$$

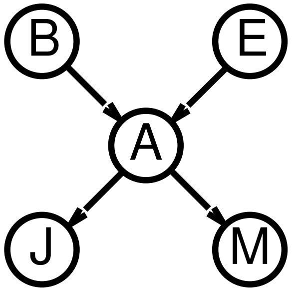
e.g., $f_1(a, b) \times f_2(b, c) = f(a, b, c)$

Variable elimination algorithm#

```
def ELIMINATION-ASK(X,e,bn)
    inputs: X, the query variable
            e, observed values for variables E
            bn, a Bayes net with variables vars

    factors ← [ ]
    for each var in ORDER(vars) do
        factors ← [MAKE-FACTOR(V, e)] + |factors
        if V is a hidden variable then factors ← SUM-OUT(V,factors)
    return NORMALIZE(POINTWISEPRODUCT(factors))
```

Irrelevant variables*

Consider the query $P(JohnCalls|Burglary = true)$



$$P(J|b) = \alpha P(b) \sum_e P(e) \sum_a P(a|b,e) P(J|a) \sum_m P(m|a)$$

Sum over m is identically 1; M is **irrelevant** to the query

Theorem: Y is irrelevant unless $Y \in Ancestors(\{X\} \cup \mathbf{E})$

Here, $X = JohnCalls$, $\mathbf{E} = \{Burglary\}$, and
$Ancestors(\{X\} \cup \mathbf{E}) = \{Alarm, Earthquake\}$
so $MaryCalls$ is irrelevant

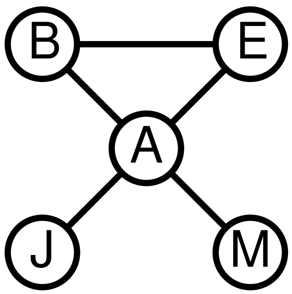
(Compare this to backward chaining from the query in Horn clause KBs)

Irrelevant variables*

Defn: moral graph of BN: marry all parents and drop arrows

Defn: $\mathbf{A}$ is m-**separated** from $\mathbf{B}$ by $\mathbf{C}$ iff separated by $\mathbf{C}$ in the moral graph

Theorem: Y is irrelevant if m-separated from X by $\mathbf{E}$



For $P(JohnCalls|Alarm = true)$, both $Burglary$ and $Earthquake$ are irrelevant

Complexity of exact inference

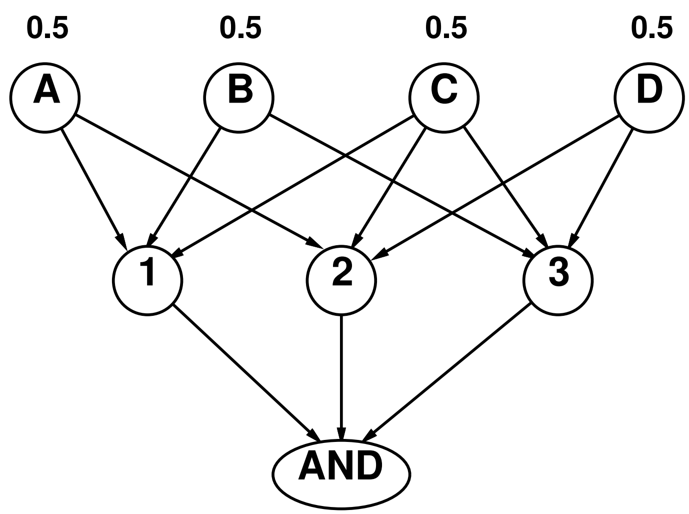
Singly connected networks (or polytrees)
– any two nodes are connected by at most one (undirected) path
– time and space cost of variable elimination are $O(d^k n)$

Multiply connected networks
– can reduce 3SAT to exact inference $\Rightarrow$ NP-hard
– equivalent to **counting** 3SAT models $\Rightarrow$ #P-complete

1. A v B v C

2. C v D v ~A

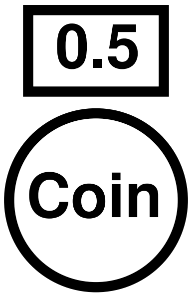
3. B v C v ~D



Inference by stochastic simulation*

Idea

1) Draw N samples from a sampling distribution S
2) Compute an approximate posterior probability $\hat{P}$
3) Show this converges to the true probability P



Methods

– Sampling from an empty network
– Rejection sampling: reject samples disagreeing with evidence
– Likelihood weighting: use evidence to weight samples
– Markov chain Monte Carlo (MCMC)
  sample from a stochastic process
  whose stationary distribution is the true posterior

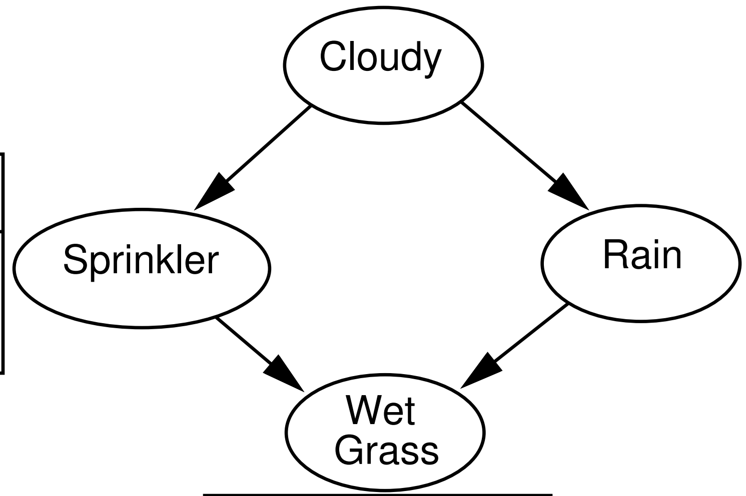
Sampling from an empty network

Direct sampling from a network that has no evidence associated
(sampling each variable in turn, in topological order)

def PRIOR-SAMPLE(bn)
  inputs: bn, a BN specifying joint distribution $\mathbf{P}(X_1, \ldots, X_n)$ $\mathbf{P}(X_1, \ldots, X_n)$
  $\mathbf{x} \leftarrow$ an event with n elements
  for each 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 $\mathbf{x}$ // an event sampled from the prior specified by bn

Example: prior sampling

P(C)
.50

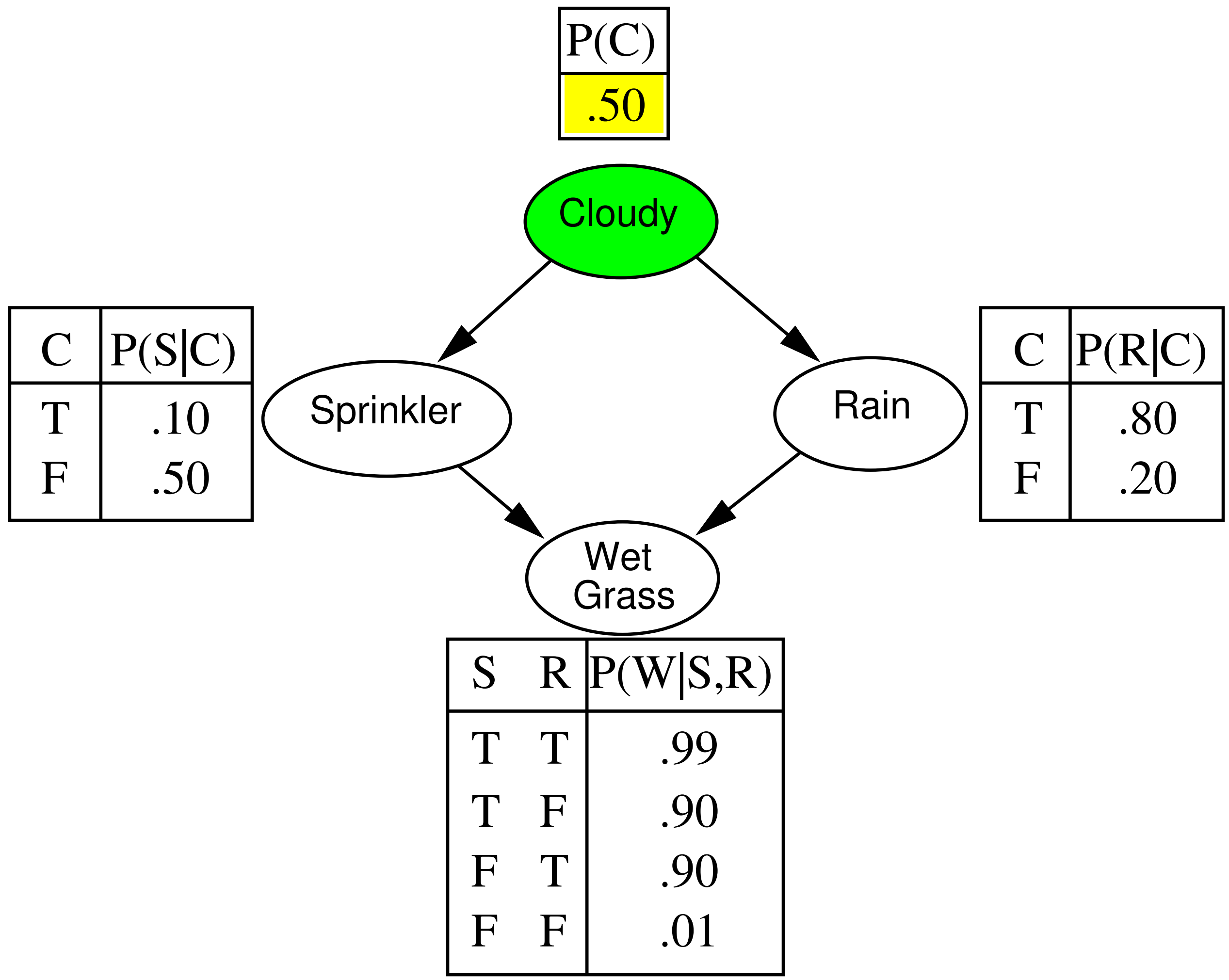


C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Example: prior sampling



P(C)
.50

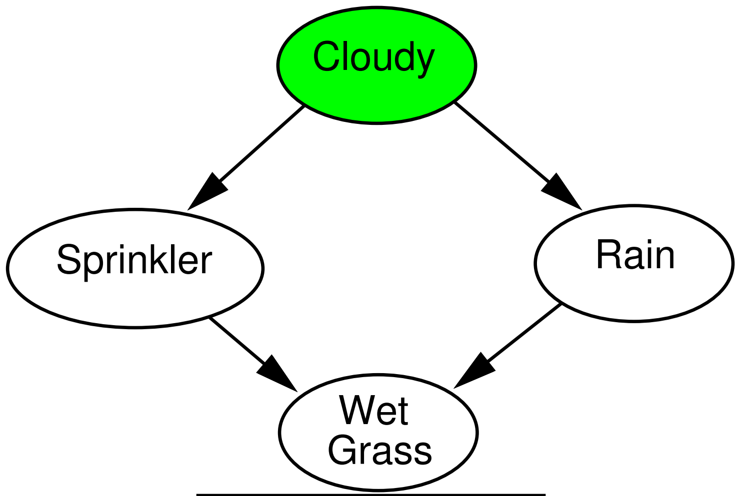
C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Example: prior sampling

P(C)
.50

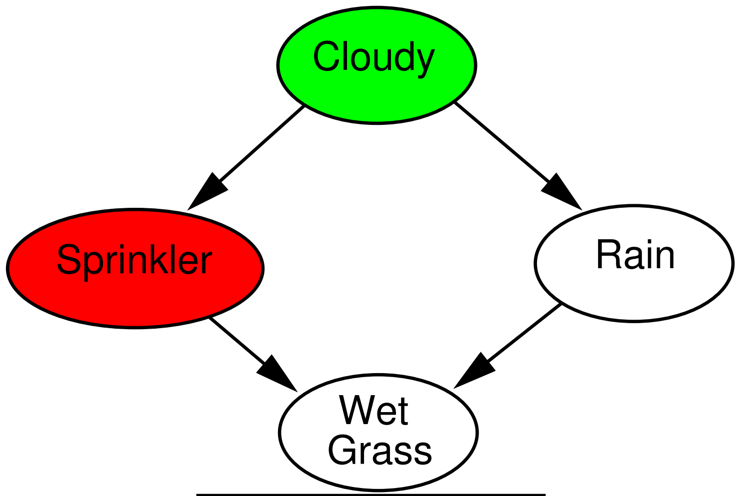


C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Example: prior sampling



P(C)
.50

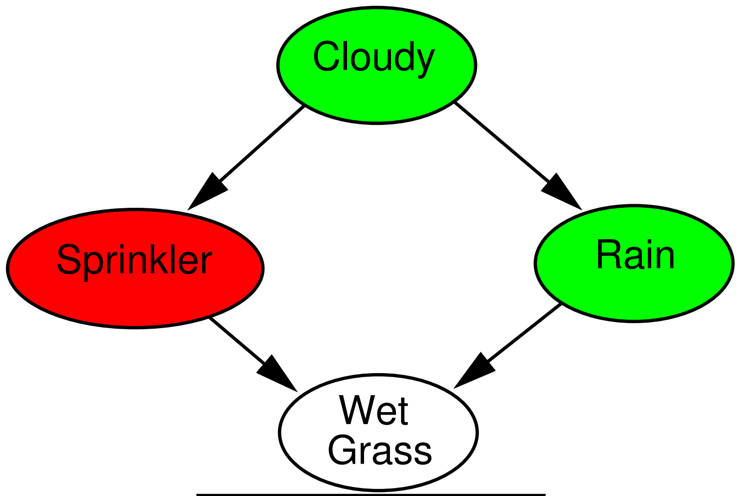
C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Example: prior sampling

P(C)
.50

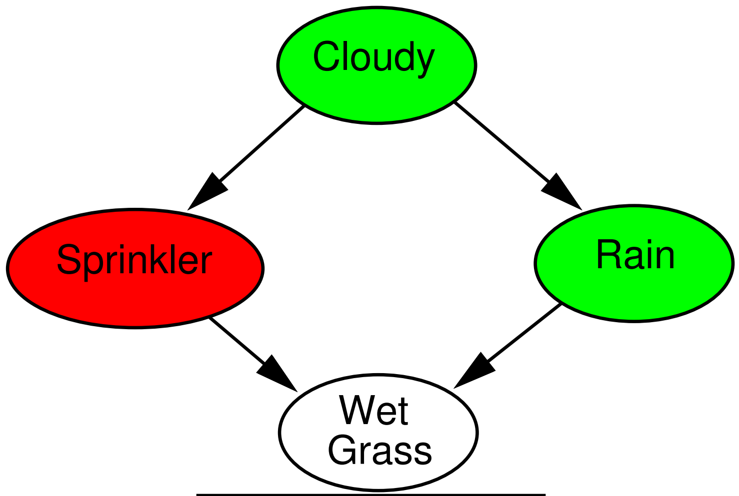


C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Example: prior sampling



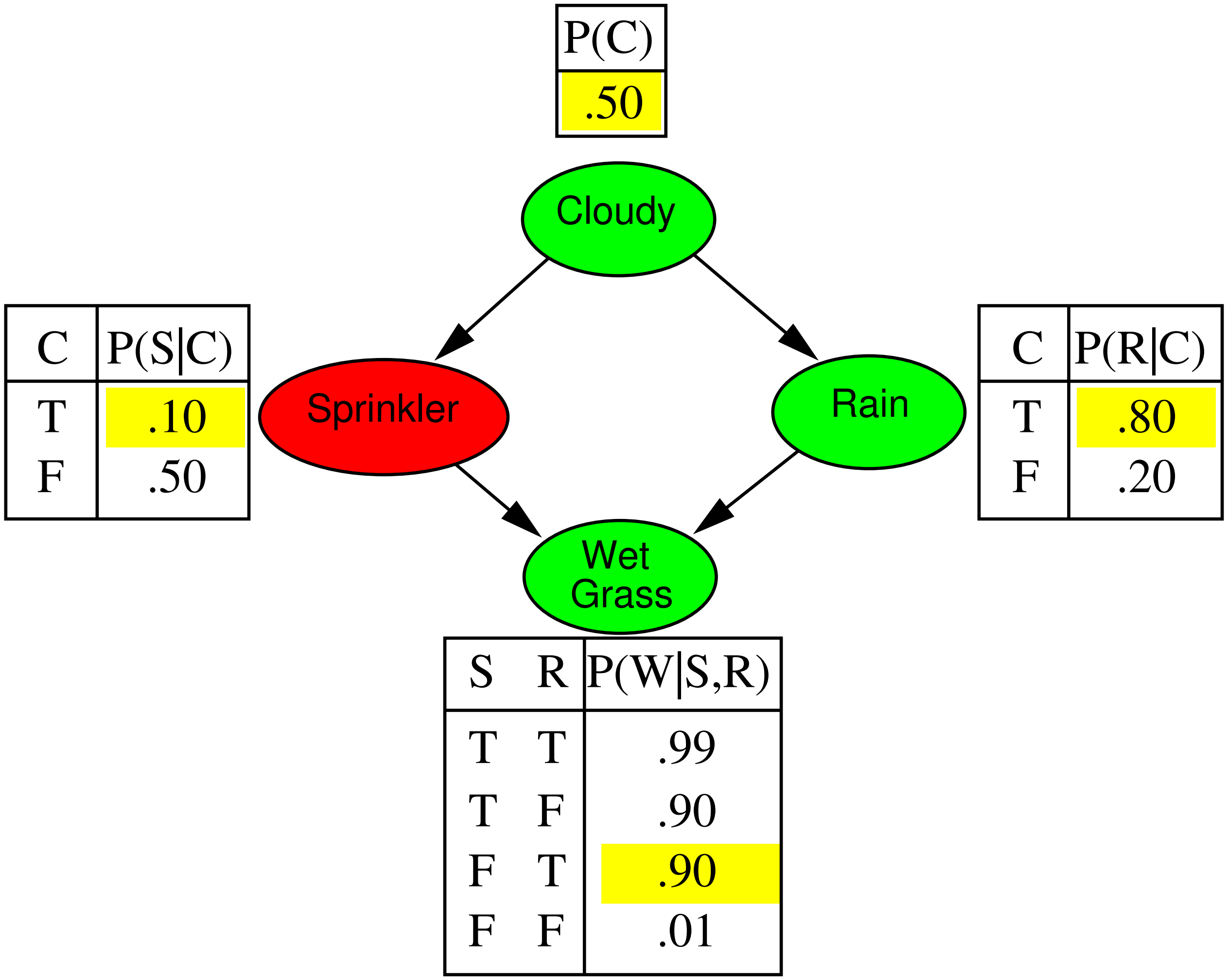
P(C)
.50

C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

Example: prior sampling



Sampling from an empty network contd.

Probability that PriorSample generates a particular event

$S_{PS}(x_1 \dots x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i)) = P(x_1 \dots x_n)$

i.e., the true prior probability

E.g., $S_{PS}(t, f, t, t) = 0.5 \times 0.9 \times 0.8 \times 0.9 = 0.324 = P(t, f, t, t)$

Let $N_{PS}(x_1 \dots x_n)$ be the number of samples generated for event $x_1, \dots, x_n$

Then we have

$$\begin{aligned} \lim_{N \to \infty} \hat{P}(x_1, \dots, x_n) &= \lim_{N \to \infty} N_{PS}(x_1, \dots, x_n)/N \\ &= S_{PS}(x_1, \dots, x_n) \\ &= P(x_1 \dots x_n) \end{aligned}$$

That is, estimates derived from PriorSample are consistent

Shorthand: $\hat{P}(x_1, \dots, x_n) \approx P(x_1 \dots x_n)$

Rejection sampling

$\hat{\mathbf{P}}(X|\mathbf{e})$ estimated from samples agreeing with $\mathbf{e}$

```
def REJECTION-SAMPLING(X,e,bn,N)
    inputs: X, the query variable
            e, observed values for variables E
            bn, a BN
            N, the total number of samples to be generated
    local variables: C, a vector of counts for each value of X , initially zero
      for j = 1 to N do
        x ← PRIOR-SAMPLE(bn)
        if x is consistent with e then // do not match the evidence
          C[j] ← C[j]+1 where x_j is the value of X in x
    return NORMALIZE(C)  // an estimate of P(X|e)
```

Example: rejection sampling

Estimate $\mathbf{P}(Rain|Sprinkler = true)$ using 100 samples

27 samples have $Sprinkler = true$

Of these, 8 have $Rain = true$ and 19 have $Rain = false$.

$\hat{\mathbf{P}}(Rain|Sprinkler = true) = \textsc{Normalize}(\langle 8, 19\rangle) = \langle 0.296, 0.704\rangle$

Similar to a basic real-world empirical estimation procedure

Rejection sampling contd.

$\hat{\mathbf{P}}(X|\mathbf{e}) = \alpha \mathbf{N}_{PS}(X, \mathbf{e})$ (algorithm defn.)
$= \mathbf{N}_{PS}(X, \mathbf{e})/N_{PS}(\mathbf{e})$ (normalized by $N_{PS}(\mathbf{e})$)
$\approx \mathbf{P}(X, \mathbf{e})/P(\mathbf{e})$ (property of PriorSample)
$= \mathbf{P}(X|\mathbf{e})$ (defn. of conditional probability)

Hence rejection sampling returns consistent posterior estimates

Problem: hopelessly expensive if $P(\mathbf{e})$ is small

$P(\mathbf{e})$ drops off exponentially with number of evidence variables

Likelihood weighting

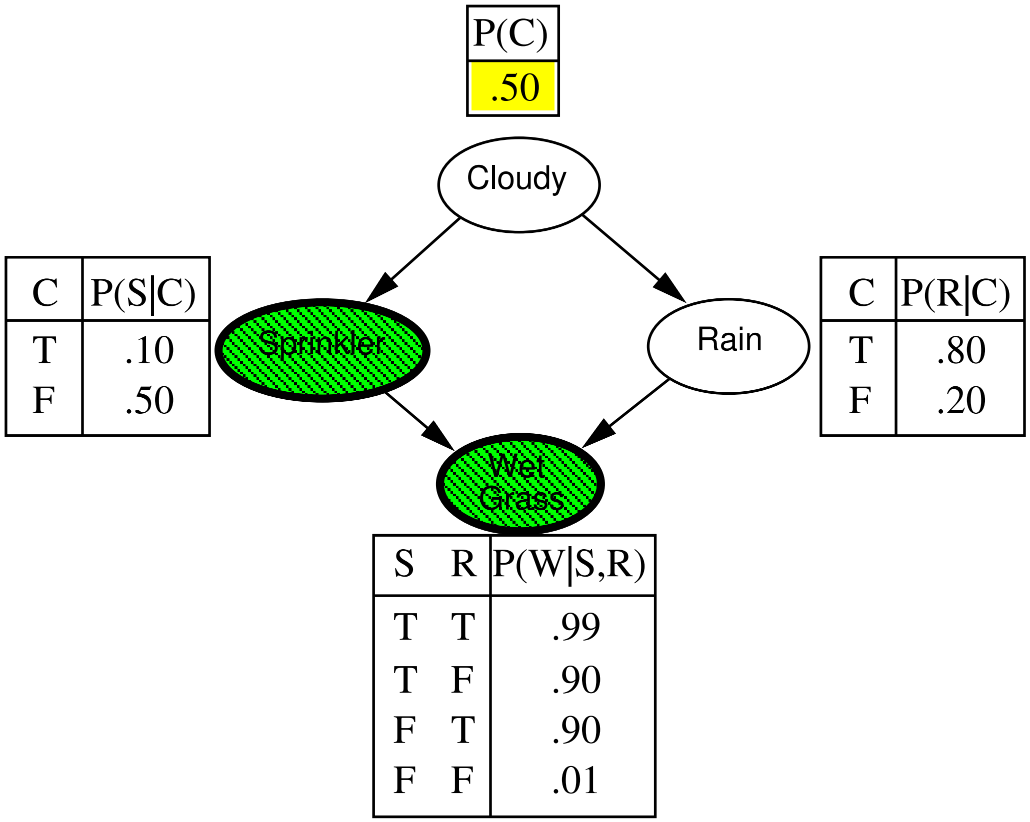
Idea

– fix evidence variables
– sample only nonevidence variables
– weight each sample by the likelihood it accords the evidence

Likelihood weighting

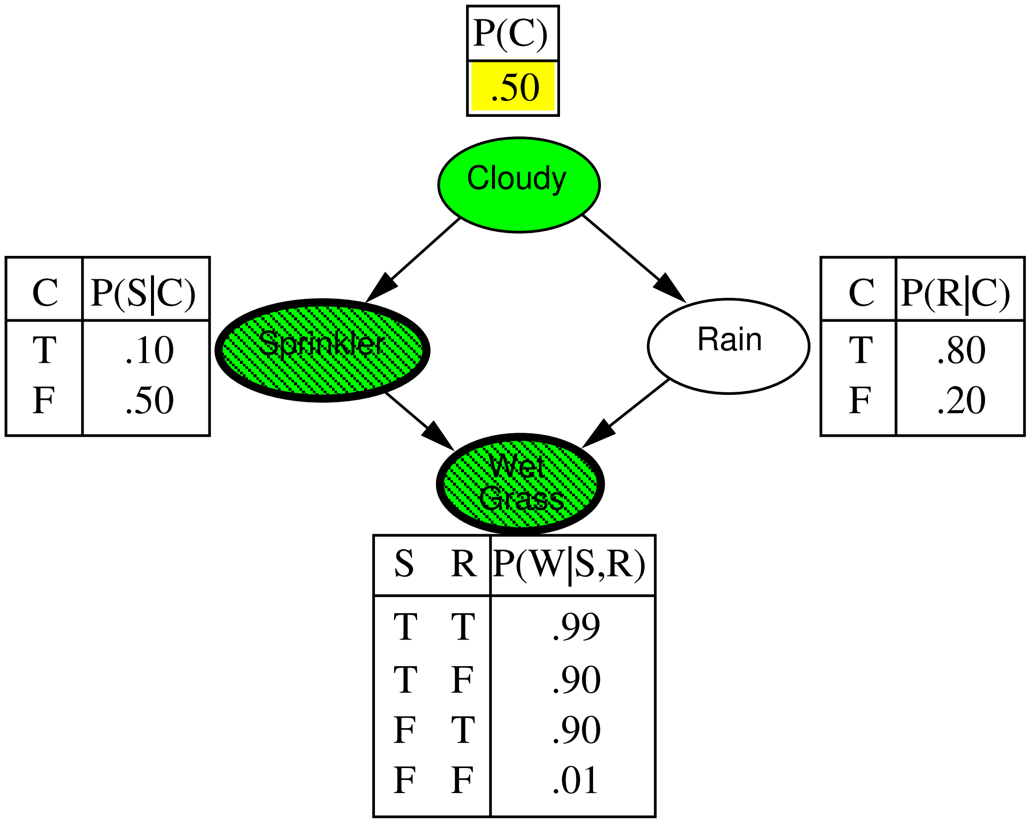
```
def Likelihood-Weighting(X,e,bn,N)
    inputs: X,e, the query variable,observed values for variables E
            bn,N, a BN, the total number of samples to be generated
    local variables: W, a vector of weighted counts for each value of X , initially 0

    for j = 1 to N do
        x,w ← Weighted-Sample(bn,e)
        W[j] ← W[j] + w where x_j is the value of X in x
    return Normalize(W[X])
def Weighted-Sample(bn,e)
    x ← an event with n elements from e; w ← 1
    for i=1 to n do
        if X_i is an evidence variable with value x_i in e
            then w ← w × P(X_i = x_ij | Parents(X_i))
            else x[i] ← a random sample from P(X_i | Parents(X_i))
    return x, w
```

Example: likelihood weighting



$w = 1.0$

Example: likelihood weighting



P(C)
.50

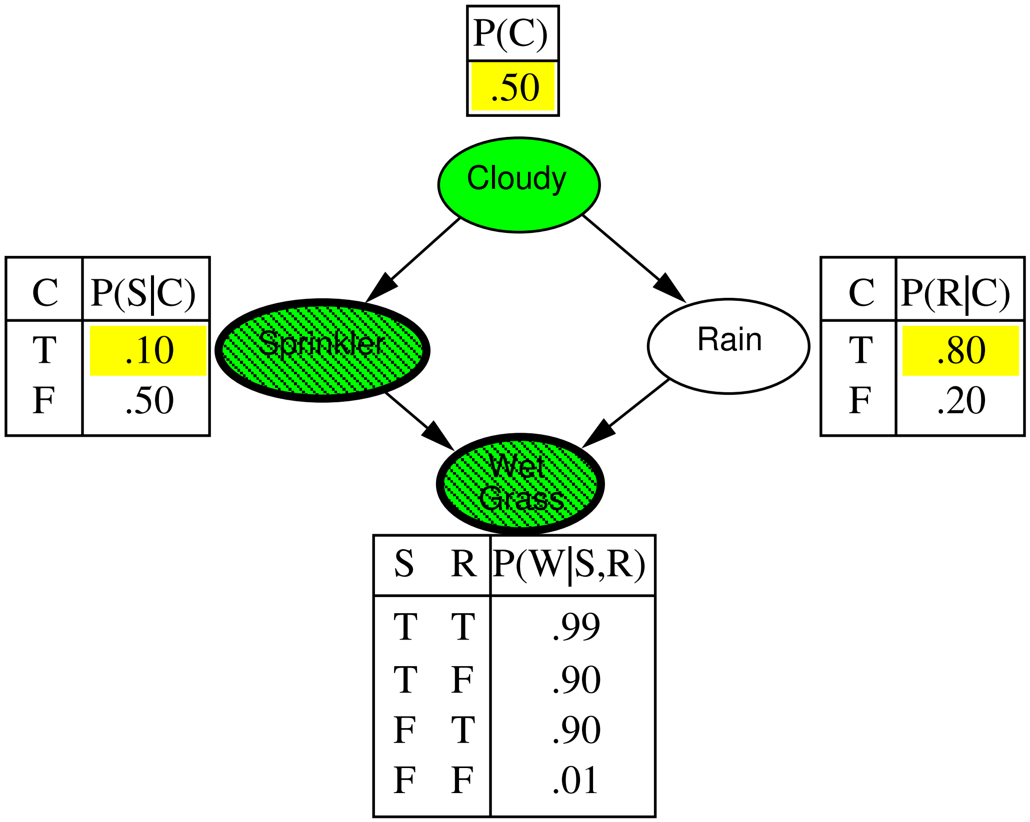
C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

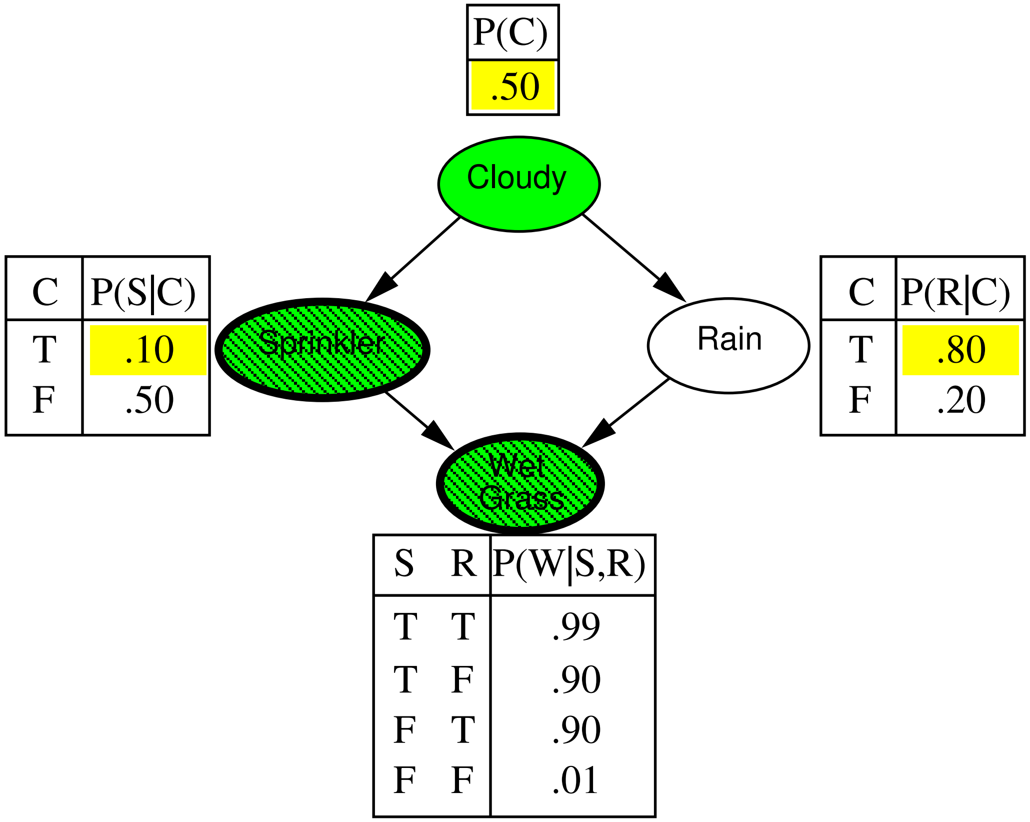
$w = 1.0$

Example: likelihood weighting



$w = 1.0$

Example: likelihood weighting



P(C)
.50

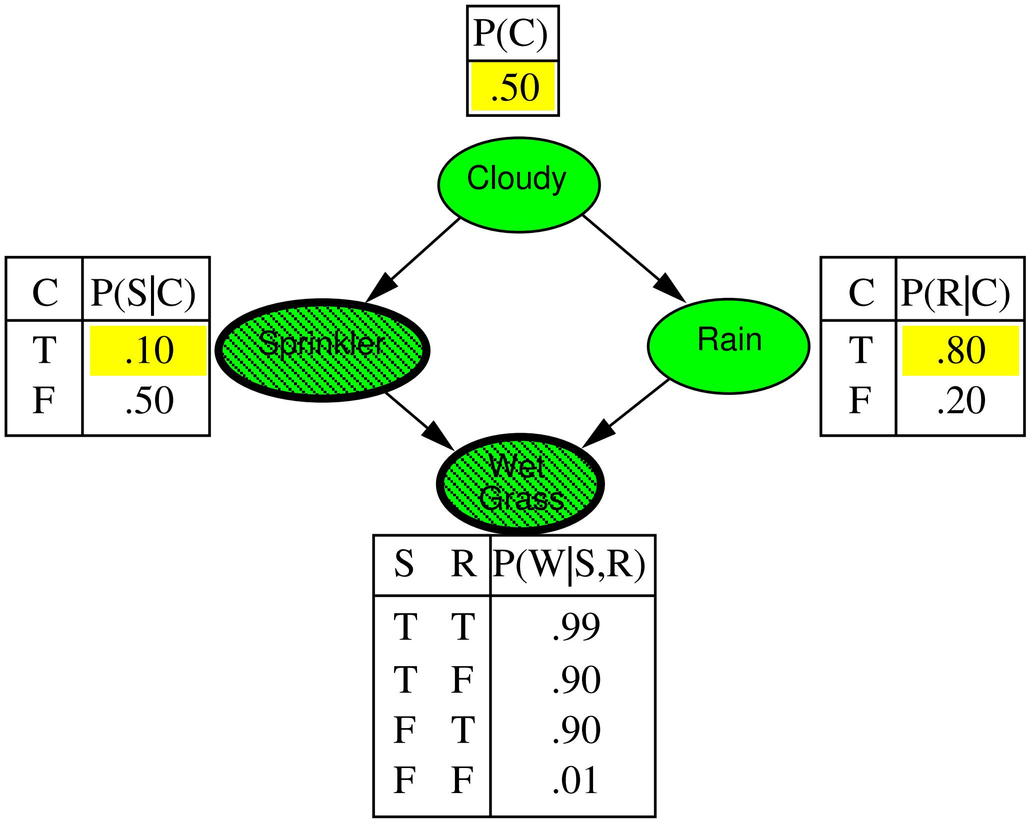
C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

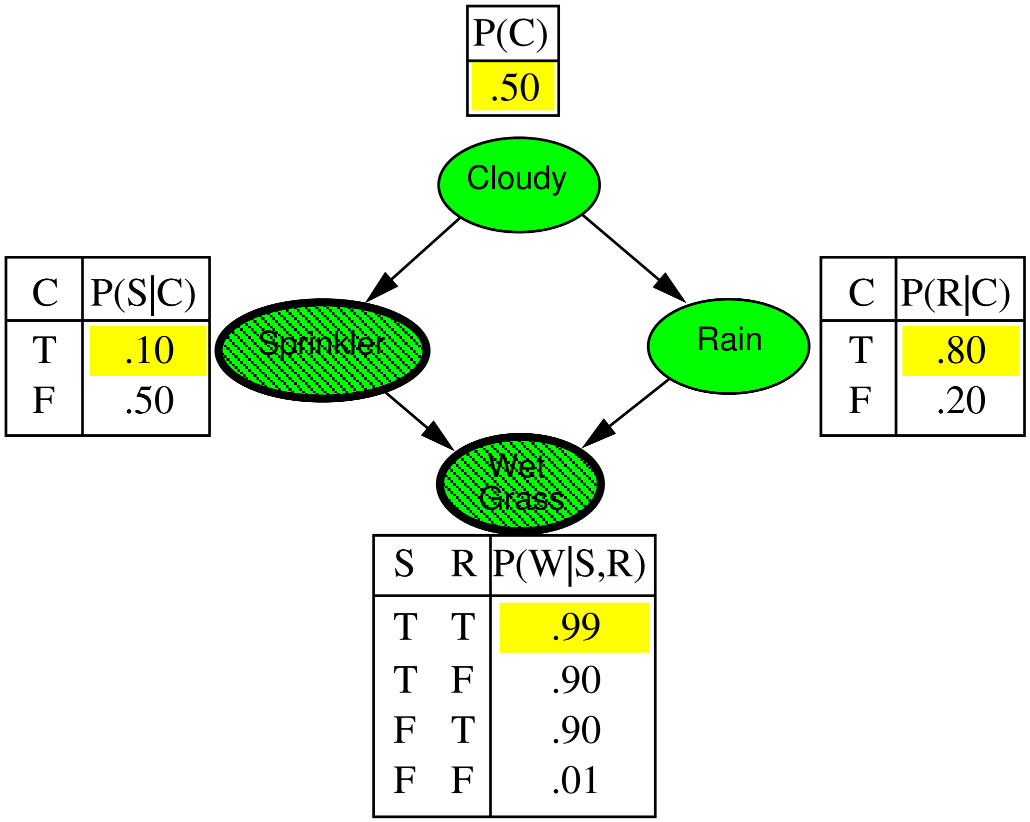
$w = 1.0 \times 0.1$

Example: likelihood weighting



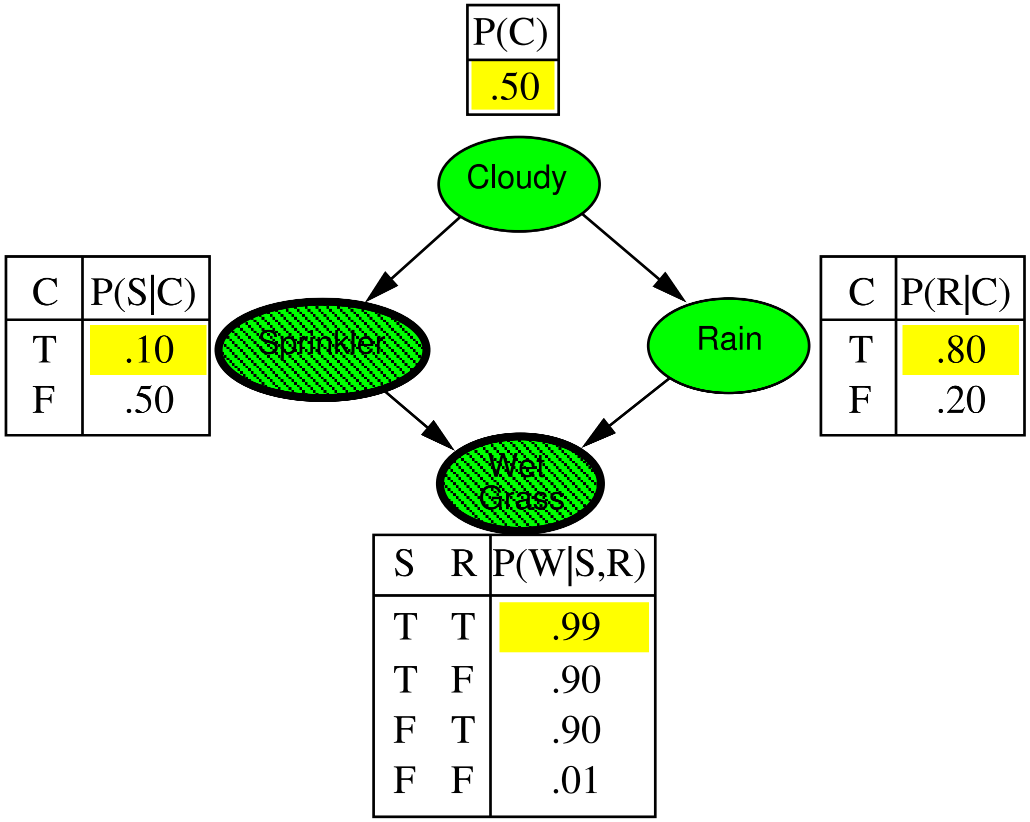
$w = 1.0 \times 0.1$

Example: likelihood weighting



$w = 1.0 \times 0.1$

Example: likelihood weighting



P(C)
.50

C	P(S|C)
T	.10
F	.50

C	P(R|C)
T	.80
F	.20

S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

$$w = 1.0 \times 0.1 \times 0.99 = 0.099$$

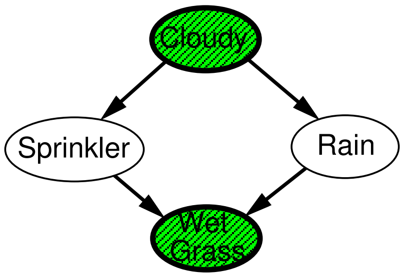
Likelihood weighting contd.

Sampling probability for WeightedSample is

$S_{WS}(\mathbf{z},\mathbf{e}) = \prod_{i=1}^{l} P(z_i|parents(Z_i))$

Note: pays attention to evidence in **ancestors** only

$\Rightarrow$ somewhere "in between" prior and posterior distribution



Weight for a given sample $\mathbf{z},\mathbf{e}$ is

$w(\mathbf{z},\mathbf{e}) = \prod_{i=1}^{m} P(e_i|parents(E_i))$

Weighted sampling probability is

$S_{WS}(\mathbf{z},\mathbf{e})w(\mathbf{z},\mathbf{e})$

$= \prod_{i=1}^{l} P(z_i|parents(Z_i)) \;\; \prod_{i=1}^{m} P(e_i|parents(E_i))$

$= P(\mathbf{z},\mathbf{e})$ (by standard global semantics of network)

Hence likelihood weighting returns consistent estimates
but performance still degrades with many evidence variables
because a few samples have nearly all the total weight

Inference by Markov chain Monte Carlo*

"State" of network = current assignment to all variables
⇒ the next state by making random changes to the current state

Generate the next state by sampling one variable given Markov blanket
recall Markov blanket: parents, children, and children's parents
Sample each variable in turn, keeping evidence fixed

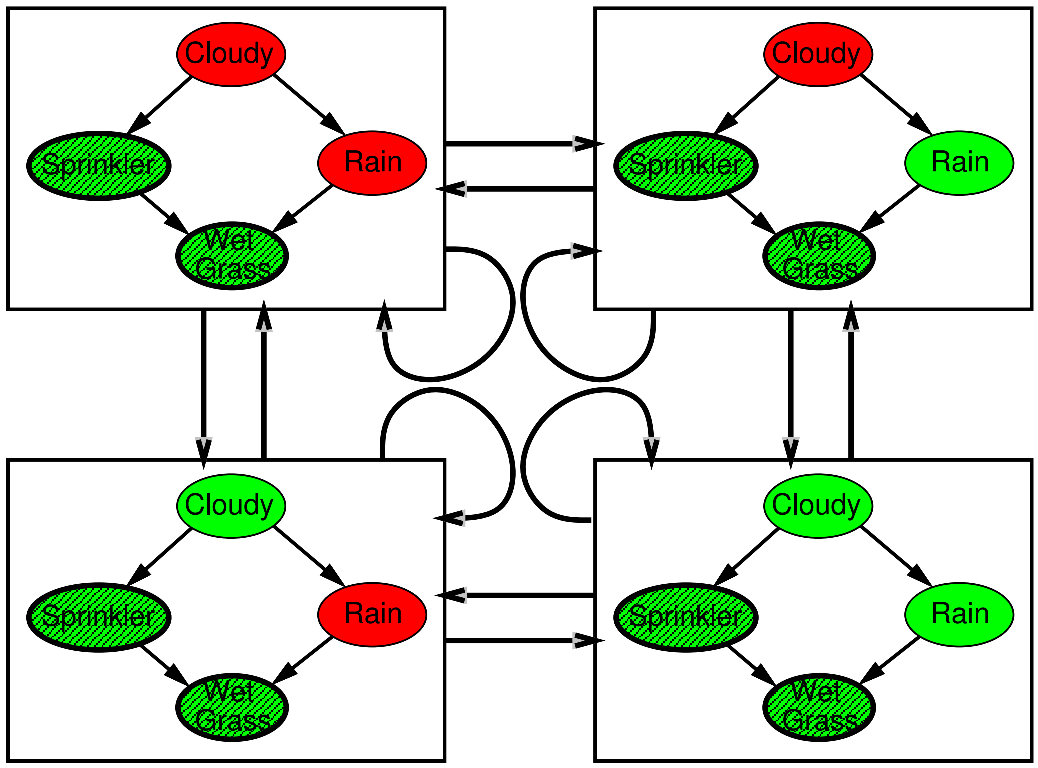
Specific transition probability with which the stochastic process moves from one state to another
defined by conditional distribution given Markov blanket of the variable being sampled

MCMC Gibbs sampling

```
def MCMC-GIBBS-ASK(X,e,bn,N)
    local variables: C, a vector of counts for each value of X, initially zero
                     Z, the nonevidence variables in bn
                     x, the current state of the network, initially copied from e

    initialize x with random values for the variables in Z
    for k = 1 to N do          // Can choose at random
        choose any variable Z_i from Z according to any distribution ρ(i)
        set the value of Z_i in x by sampling from P(Z_i|mb(Z_i)) // Markov blanket
            C[j] ← C[j] + 1 where x_j is the value of X in x
    return NORMALIZE(C)
```

Example: Markov chain

With $Sprinkler = true, WetGrass = true$, there are four states



Wander about for a while

Example: Gibbs sampling

Estimate $\mathbf{P}(Rain|Sprinkler = true, WetGrass = true)$

Sample $Cloudy$ or $Rain$ given its Markov blanket, repeat
Count number of times $Rain$ is true and false in the samples

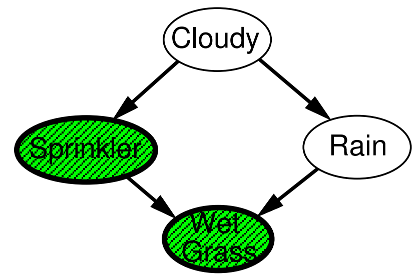
E.g., visit 100 states
31 have $Rain = true$, 69 have $Rain = false$

$\hat{\mathbf{P}}(Rain|Sprinkler = true, WetGrass = true)$
$= \textsc{Normalize}(\langle 31, 69\rangle) = \langle 0.31, 0.69\rangle$

Theorem: chain approaches stationary distribution
long-run fraction of time spent in each state is exactly proportional to its posterior probability

Markov blanket sampling

Markov blanket of $Cloudy$ is
$Sprinkler$ and $Rain$
Markov blanket of $Rain$ is
$Cloudy$, $Sprinkler$, and $WetGrass$



Probability given the Markov blanket is calculated as follows
$$P(x_i'|mb(X_i)) = P(x_i'|parents(X_i))\prod_{Z_j \in Children(X_i)} P(z_j|parents(Z_j))$$

Easily implemented in message-passing parallel systems, brains

Main computational problems:
1) Difficult to tell if convergence has been achieved
2) Can be wasteful if Markov blanket is large
$P(X_i|mb(X_i))$ won't change much (law of large numbers)

Approximate inference

Exact inference by variable elimination:
- polytime on polytrees, NP-hard on general graphs
- space = time, very sensitive to topology

Approximate inference by LW (Likelihood Weighting), MCMC (Markov chain Monte Carlo):
- LW does poorly when there are lots of (downstream) evidence
- LW, MCMC generally insensitive to topology
- Convergence can be very slow with probabilities close to 1 or 0
- Can handle arbitrary combinations of discrete and continuous variables

Dynamic Bayesian networks+

DBNs are BNs that represent **temporal** probability models

Basic idea: copy state and evidence variables for each time step

$\mathbf{X}_t$ = set of unobservable state variables at time t
e.g., $BloodSugar_t$, $StomachContents_t$, etc.

$\mathbf{E}_t$ = set of observable evidence variables at time t
e.g., $MeasuredBloodSugar_t$, $PulseRate_t$, $FoodEaten_t$

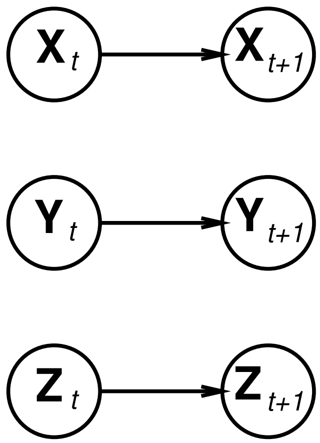
This assumes **discrete time**; step size depends on problem

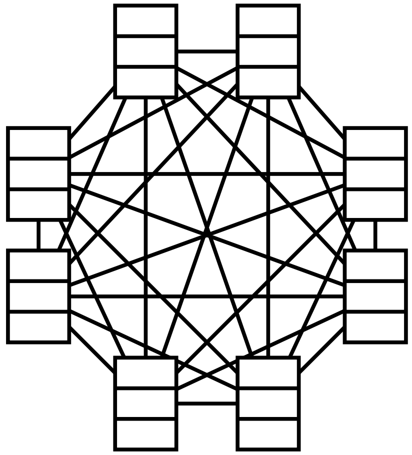
Notation: $\mathbf{X}_{a:b} = \mathbf{X}_a, \mathbf{X}_{a+1}, \ldots, \mathbf{X}_{b-1}, \mathbf{X}_b$

$\mathbf{X}_t$, $\mathbf{E}_t$ contain arbitrarily many variables in a replicated Bayes net

Hidden Markov models

HMMs: single-(state) variable DBNs
every discrete DBN is an HMM
(combine all the state variables in the DBN into a single one)





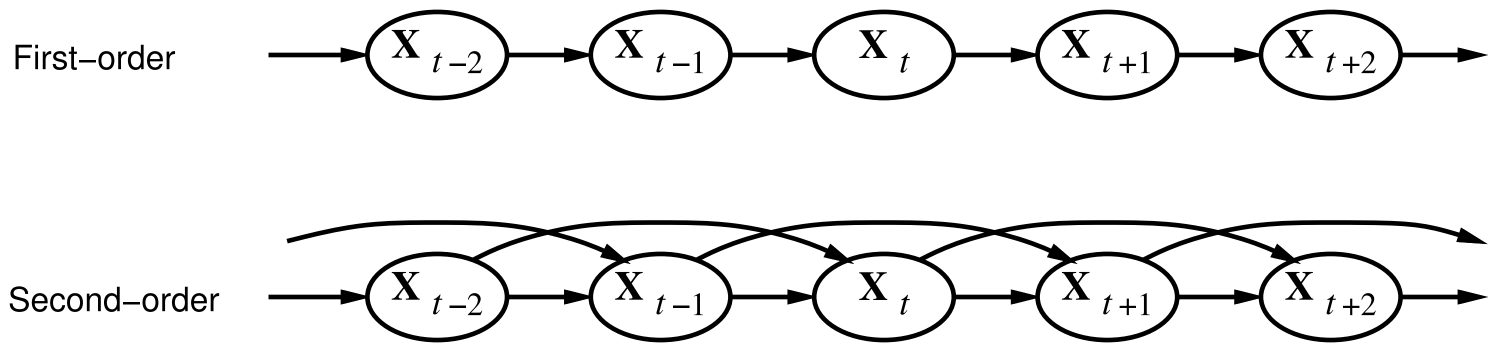
Sparse dependencies $\Rightarrow$ exponentially fewer parameters;
e.g., 20 state variables, three parents each
DBN has $20 \times 2^3 = 160$ parameters, HMM has $2^{20} \times 2^{20} \approx 10^{12}$
(analogous to BNs and full tabulated joint distributions)

Markov processes (Markov chains)

Construct a Bayes net from these variables: parents?

Markov assumption: $\mathbf{X}_t$ depends on **bounded** subset of $\mathbf{X}_{0:t-1}$

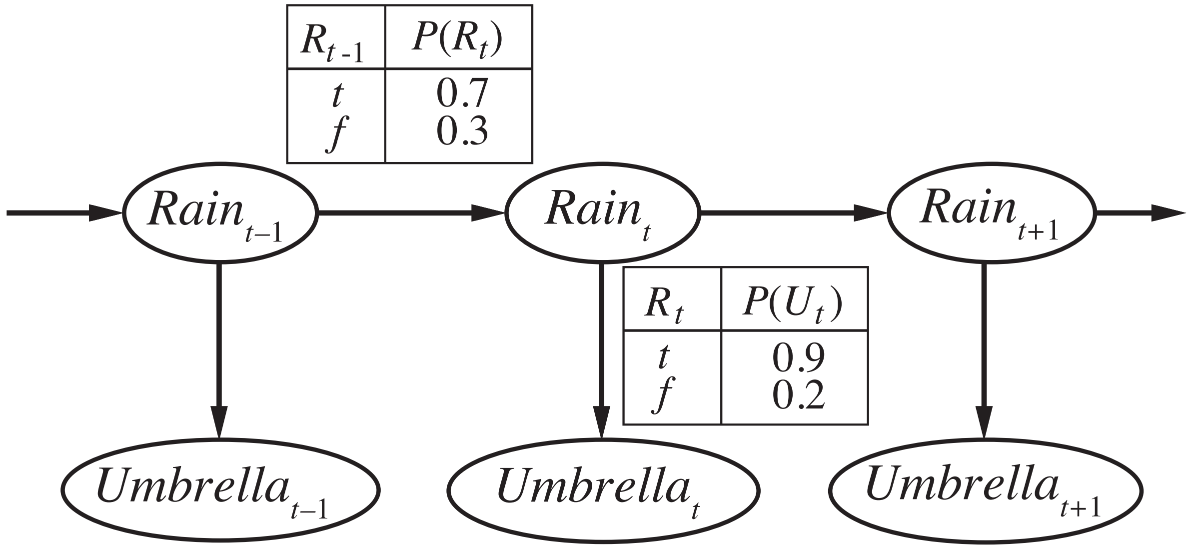
First-order Markov process: $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$
Second-order Markov process: $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{0:t-1}) = \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-2}, \mathbf{X}_{t-1})$



Sensor Markov assumption: $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_{0:t}, \mathbf{E}_{0:t-1}) = \mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$

Stationary process: transition model $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$ and sensor model $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$ fixed for all t

Example: Markov processes



First-order Markov assumption not exactly true in real world

Possible fixes

1. **Increase order** of Markov process
2. **Augment state**, e.g., add $Temp_t$, $Pressure_t$

HMMs

$\mathbf{X}_t$ is a single, discrete variable (usually $\mathbf{E}_t$ is too)
Domain of X_t is $\{1, \ldots, S\}$

Transition matrix $\mathbf{T}_{ij} = P(X_t = j | X_{t-1} = i)$, e.g., $\begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}$

Sensor matrix $\mathbf{O}_t$ for each time step, diagonal elements $P(e_t | X_t = i)$
e.g., with $U_1 = true$, $\mathbf{O}_1 = \begin{pmatrix} 0.9 & 0 \\ 0 & 0.2 \end{pmatrix}$

Forward and backward messages as column vectors

$$\begin{aligned} \mathbf{f}_{1:t+1} &= \alpha \mathbf{O}_{t+1} \mathbf{T}^\top \mathbf{f}_{1:t} \\ \mathbf{b}_{k+1:t} &= \mathbf{T} \mathbf{O}_{k+1} \mathbf{b}_{k+2:t} \end{aligned}$$

Forward-backward algorithm needs time $O(S^2 t)$ and space $O(St)$

Inference tasks in HMMs

Filtering: $\mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$
belief state—input to the decision process of a rational agent

Prediction: $\mathbf{P}(\mathbf{X}_{t+k}|\mathbf{e}_{1:t})$ for $k > 0$
evaluation of possible action sequences;
like filtering without the evidence

Smoothing: $\mathbf{P}(\mathbf{X}_k|\mathbf{e}_{1:t})$ for $0 \leq k < t$
better estimate of past states, essential for learning

Most likely explanation: $\arg\max_{\mathbf{x}_{1:t}} P(\mathbf{x}_{1:t}|\mathbf{e}_{1:t})$
speech recognition, decoding with a noisy channel

Filtering

Aim: devise a **recursive** state estimation algorithm

$$\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) = f(\mathbf{e}_{t+1}, \mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t}))$$

$$\begin{aligned}\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) &= \mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}, \mathbf{e}_{t+1}) \\ &= \alpha\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1}, \mathbf{e}_{1:t})\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t}) \\ &= \alpha\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t})\end{aligned}$$

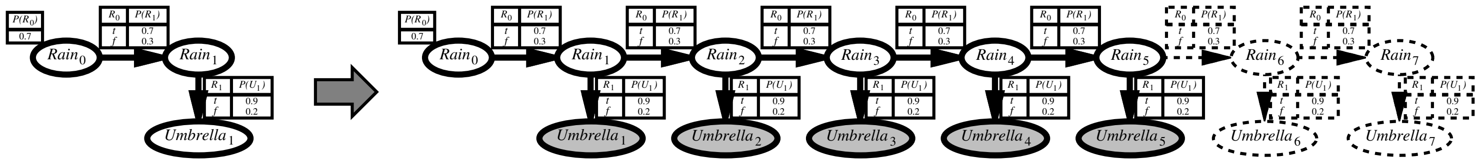
I.e., prediction + estimation. Prediction by summing out $\mathbf{X}_t$

$$\begin{aligned}\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{e}_{1:t+1}) &= \alpha\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})\Sigma_{\mathbf{x}_t}\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t, \mathbf{e}_{1:t})P(\mathbf{x}_t|\mathbf{e}_{1:t}) \\ &= \alpha\mathbf{P}(\mathbf{e}_{t+1}|\mathbf{X}_{t+1})\Sigma_{\mathbf{x}_t}\mathbf{P}(\mathbf{X}_{t+1}|\mathbf{x}_t)P(\mathbf{x}_t|\mathbf{e}_{1:t})\end{aligned}$$

$\mathbf{f}_{1:t+1} = \text{Forward}(\mathbf{f}_{1:t}, \mathbf{e}_{t+1})$ where $\mathbf{f}_{1:t} = \mathbf{P}(\mathbf{X}_t|\mathbf{e}_{1:t})$
Time and space **constant** (independent of t)

Inference in DBNs

Naive method: unroll the network and run any exact algorithm



Problem: inference cost for each update grows with t

Rollup filtering: add slice $t+1$, "sum out" slice t using variable elimination

Largest factor is $O(d^{n+1})$, update cost $O(d^{n+2})$
(cf. HMM update cost $O(d^{2n})$)

Approximate inference by MCMC (Markov chain Monte Carlo) etc.

Causal Inference*

Questions

– Observations: "What is we see A?" (What is?)

$P(y \mid A)$

– Actions: "What if we do A?" (What if?)

$P(y \mid do(A))$

– Counterfactuals: "What if we did things differerently?" (Why?)

$P(y_{A'} \mid A)$

E.g., recall C(limate)-S(prinkler)-R(rain)-W(etness)

"Would the pavement be wet HAD the sprinkler been ON?"

($P(S \mid C) = 1$)

Find if $P(W_{S=1} = 1) = P(W = 1 \mid do(S = 1))$

Can drive counterfactuals from a model

Graphical representations

- Observations $\rightarrow$ Bayesian networks
- Actions $\rightarrow$ Causal Bayesian networks
- Counterfactuals $\rightarrow$ Functional causal diagrams

Hints

– Can reduce the action questions to symbolic calculus

– Can be estimated in polynomial time, complete algorithm (with the independence in the distribution)

Probabilistic programming*

Probability models: defined using executable code in any programming language that incorporates a source of randomness

$\Rightarrow$ Programs as probability models

$\Leftarrow$ probabilistic programming language (PPL)

– all of the expressive power of the underlying language

Computationally universal: they can represent any probability distribution that can be sampled from by a probabilistic Turing machine

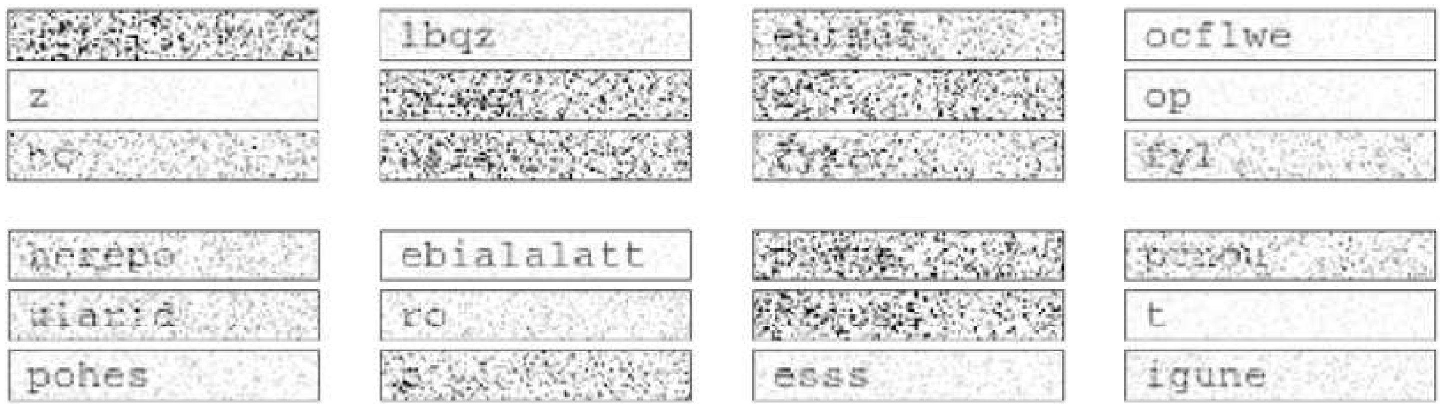
Generative program

```
def GENERATE-IMAGE(())
    letter ← GENERATE-LETTER(10)
    return RENDER-NOISY-IMAGE(letter, 32, 128)
def GENERATE-LETTER(λ)
    n ∼ Poisson(λ)
    for i=1 to n do
        letter ∼ UNIFORM-CHOICE({a, b, c, ···})
    return letter
def RENDER-NOISY-IMAGE(letter, width, height)
    clean-image ← RENDER(letter, width, height, text-top=10, text-left=10)
    noisy-image ← [ ]
    noise-variance ∼ UNIFORM-REAL(0.1,1)
    for row = 1 to width do
        for col = 1 to height do
        noisy-image[row, col] ∼ N(clean-image[row, col],noisy-variance)
    return noisy-image
```

Example: reading text

The program that reads degraded (smudged or blurred) text (or CAPTCHAs), i.e., optical character recognition

– invoking GENERATE-IMAGE 9 times



Probabilistic logic*

Bayesian networks are essentially propositional:
- the set of random variables is fixed and finite
- each variable has a fixed domain of possible values

Probabilistic reasoning can be formalized as probabilistic logic

First-order probabilistic logic combines probability theory with the expressive power of first-order logic

First-order probabilistic logic

Recall: Propositional probabilistic logic

– Proposition $=$ disjunction of atomic events in which it is true

– Possible world (sample point) $\omega =$ propositional logic model (an assignment of values to all of the r.v.s under consideration)

– $\omega \models \phi$: for any proposition ϕ, the ω where it is true

– probability model: a set Ω of possible worlds with a probability $P(\omega)$ for each world ω

First-order probabilistic logic

FOPL

- Probability of any first-order logical sentence ϕ as a sum over the possible worlds where it is true

  $P(\phi) = \Sigma_{\omega:\omega\models\phi} P(\omega)$

- Conditional probabilities $P(\phi|\mathbf{e})$ can be obtained similarly

  ask any question from the probability model

  $\Rightarrow$ (first-order) belief networks

Problem: the set of first-order models is infinite

– the summation could be infeasible

– specifying a complete and consistent distribution over an infinite set of worlds could be very difficult

Analogous to the method of propositionalization for FOL

e.g. relational probability models (RPMs)

Other approaches to uncertain reasoning

- Nonmonotonic reasoning
- Rule-based methods
- Dempster-Shafer theory
- Possibility theory
- Fuzzy logic
- Rough sets