

Lecture 9: Uncertainty

Previously...



Search

- Path-based search

- Iterative improvement search

Logic

- Propositional Logic

- First Order Logic (FOL)

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 (KCBS traffic reports)
- 3) uncertainty in action outcomes (flat tire, etc.)
- 4) immense complexity of modelling and predicting traffic

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 etc."

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

Methods for handling uncertainty



Default or nonmonotonic logic:

Assume my car does not have a flat tire

Assume A_{25} works unless contradicted by evidence

Issues: What assumptions are reasonable? How to handle contradiction?

Rules with fudge factors:

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

$\textit{Sprinkler} \mapsto_{0.99} \textit{WetGrass}$

$\textit{WetGrass} \mapsto_{0.7} \textit{Rain}$

Issues: Problems with combination, e.g., *Sprinkler* causes *Rain*??

Probability

Given the available evidence,

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

Mahaviracarya (9th C.), Cardano (1565) theory of gambling

Probability



Probability



Probabilistic assertions **summarize** effects of
 laziness: failure to enumerate exceptions, qualifications, etc.
 ignorance: lack of relevant facts, initial conditions, etc.

Subjective or 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 status $KB \models \alpha$, not truth.)

Probability

car-goat-door
guess



after your choice,
I will open one from the
rest doors that has goat



Will you change your
choice?

Making decisions under uncertainty



Suppose I believe the following:

$$P(A_{25} \text{ gets me there on time} | \dots) = 0.04$$

$$P(A_{90} \text{ gets me there on time} | \dots) = 0.70$$

$$P(A_{120} \text{ gets me there on time} | \dots) = 0.95$$

$$P(A_{1440} \text{ gets me there on time} | \dots) = 0.9999$$

Which action to choose?

Depends on my **preferences** for missing flight vs. airport cuisine, etc.

Utility theory is used to represent and infer preferences

Decision theory = utility theory + probability theory

Probability basics



Begin with a set Ω —the sample space

e.g., 6 possible rolls of a die.

$\omega \in \Omega$ is a sample point/possible world/atomic event

A probability space or probability model is a sample space with an assignment $P(\omega)$ for every $\omega \in \Omega$ s.t.

$$0 \leq P(\omega) \leq 1$$

$$\sum_{\omega} P(\omega) = 1$$

e.g., $P(1) = P(2) = P(3) = P(4) = P(5) = P(6) = 1/6$.

An event A is any subset of Ω

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

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, e.g., the reals or Booleans

e.g., $Odd(1) = true$.

P induces a **probability distribution** for any r.v. X :

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

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

Propositions



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

Given Boolean random variables 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 $B(\omega) = true$

Often in AI applications, the sample points are **defined** by the values of a set of random variables, i.e., the sample space is the Cartesian product of the ranges of the variables

With Boolean variables, sample point = propositional logic model

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

Proposition = disjunction of atomic events in which it is true

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

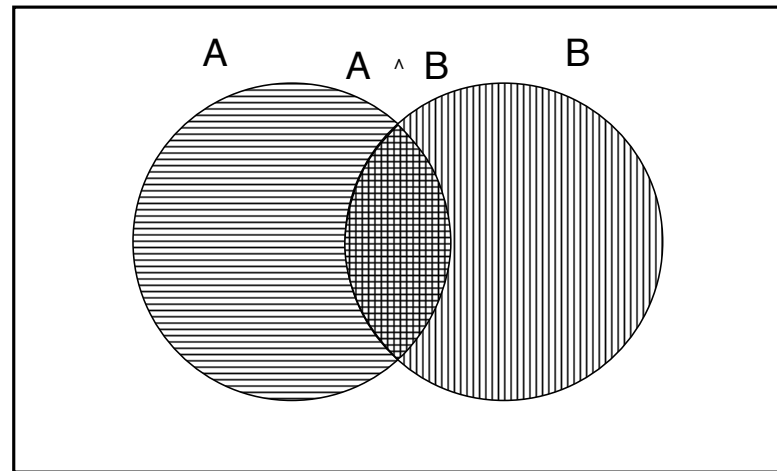
$\Rightarrow P(a \vee b) = P(\neg a \wedge b) + P(a \wedge \neg b) + P(a \wedge b)$

Why use probability?

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

E.g., $P(a \vee b) = P(a) + P(b) - P(a \wedge b)$

True



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 outcome.

Syntax for propositions



Propositional or Boolean random variables

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

Cavity = true is a proposition, also written *cavity*

Discrete random variables (finite or infinite)

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

Weather = rain is a proposition

Values must be exhaustive and mutually exclusive

Continuous random variables (bounded or unbounded)

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

Arbitrary Boolean combinations of basic propositions

Prior probability

Prior or unconditional probabilities of propositions

e.g., $P(\textit{Cavity} = \textit{true}) = 0.1$ and $P(\textit{Weather} = \textit{sunny}) = 0.72$

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

Probability distribution gives values for all possible assignments:

$\mathbf{P}(\textit{Weather}) = \langle 0.72, 0.1, 0.08, 0.1 \rangle$ (normalized, i.e., sums to 1)

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)

$\mathbf{P}(\textit{Weather}, \textit{Cavity}) =$ a 4×2 matrix of values:

<i>Weather =</i>	<i>sunny</i>	<i>rain</i>	<i>cloudy</i>	<i>snow</i>
<i>Cavity = true</i>	0.144	0.02	0.016	0.02
<i>Cavity = false</i>	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

Conditional probability

Conditional or posterior probabilities

e.g., $P(\text{cavity}|\text{toothache}) = 0.8$

i.e., **given that *toothache* is all I know**

NOT “if *toothache* then 80% chance of *cavity*”

(Notation for conditional distributions:

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

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

$P(\text{cavity}|\text{toothache}, \text{cavity}) = 1$

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.,

$P(\text{cavity}|\text{toothache}, \text{49ersWin}) = P(\text{cavity}|\text{toothache}) = 0.8$

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

Conditional probability



Definition of conditional probability:

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

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.,

$$\mathbf{P}(\textit{Weather}, \textit{Cavity}) = \mathbf{P}(\textit{Weather}|\textit{Cavity})\mathbf{P}(\textit{Cavity})$$

(View as a 4×2 set of equations, **not** matrix mult.)

Chain rule is derived by successive application of product rule:

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

Inference by enumeration

Start with the joint distribution:

	<i>toothache</i>		\neg <i>toothache</i>	
	<i>catch</i>	\neg <i>catch</i>	<i>catch</i>	\neg <i>catch</i>
<i>cavity</i>	.108	.012	.072	.008
\neg <i>cavity</i>	.016	.064	.144	.576

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

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

Inference by enumeration

Start with the joint distribution:

	<i>toothache</i>		\neg <i>toothache</i>	
	<i>catch</i>	\neg <i>catch</i>	<i>catch</i>	\neg <i>catch</i>
<i>cavity</i>	.108	.012	.072	.008
\neg <i>cavity</i>	.016	.064	.144	.576

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

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

$$P(\textit{toothache}) = 0.108 + 0.012 + 0.016 + 0.064 = 0.2$$

Inference by enumeration

Start with the joint distribution:

	<i>toothache</i>		\neg <i>toothache</i>	
	<i>catch</i>	\neg <i>catch</i>	<i>catch</i>	\neg <i>catch</i>
<i>cavity</i>	.108	.012	.072	.008
\neg <i>cavity</i>	.016	.064	.144	.576

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

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

$$P(\text{cavity} \vee \text{toothache}) = 0.108 + 0.012 + 0.072 + 0.008 + 0.016 + 0.064 = 0.28$$

Inference by enumeration

Start with the joint distribution:

	<i>toothache</i>		\neg <i>toothache</i>	
	<i>catch</i>	\neg <i>catch</i>	<i>catch</i>	\neg <i>catch</i>
<i>cavity</i>	.108	.012	.072	.008
\neg <i>cavity</i>	.016	.064	.144	.576

Can also compute conditional probabilities:

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

Normalization

	<i>toothache</i>		\neg <i>toothache</i>	
	<i>catch</i>	\neg <i>catch</i>	<i>catch</i>	\neg <i>catch</i>
<i>cavity</i>	.108	.012	.072	.008
\neg <i>cavity</i>	.016	.064	.144	.576

Denominator can be viewed as a normalization constant α

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

General 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. Typically, we want 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 $\mathbf{H} = \mathbf{X} - \mathbf{Y} - \mathbf{E}$

Then the required summation of joint entries is done by summing out the hidden variables:

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

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

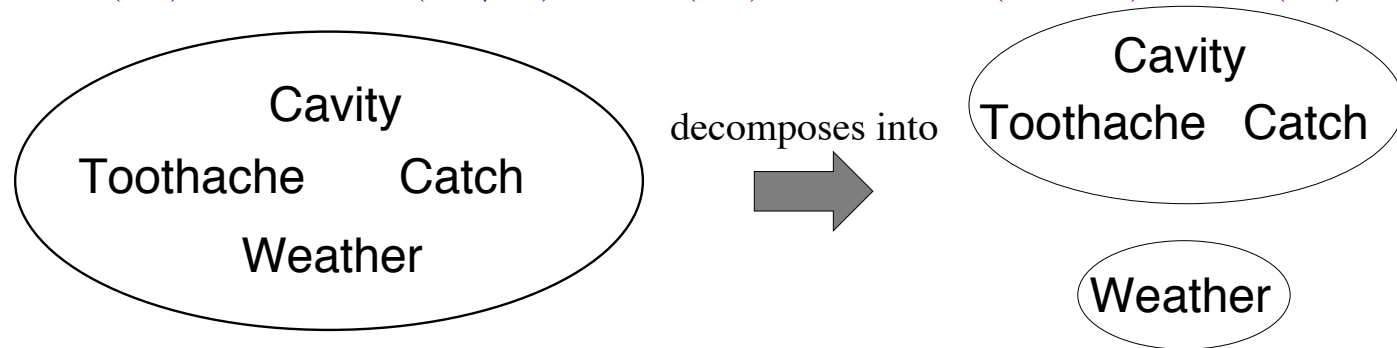
Obvious problems:

- 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

$$\mathbf{P}(A|B) = \mathbf{P}(A) \quad \text{or} \quad \mathbf{P}(B|A) = \mathbf{P}(B) \quad \text{or} \quad \mathbf{P}(A, B) = \mathbf{P}(A)\mathbf{P}(B)$$



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

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

Absolute independence 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:

$$\begin{aligned} & \mathbf{P}(Toothache, Catch, Cavity) \\ &= \mathbf{P}(Toothache|Catch, Cavity)\mathbf{P}(Catch, Cavity) \\ &= \mathbf{P}(Toothache|Catch, Cavity)\mathbf{P}(Catch|Cavity)\mathbf{P}(Cavity) \\ &= \mathbf{P}(Toothache|Cavity)\mathbf{P}(Catch|Cavity)\mathbf{P}(Cavity) \end{aligned}$$

I.e., $2 + 2 + 1 = 5$ independent numbers (equations 1 and 2 remove 2)

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 uncertain environments.

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:

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

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

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

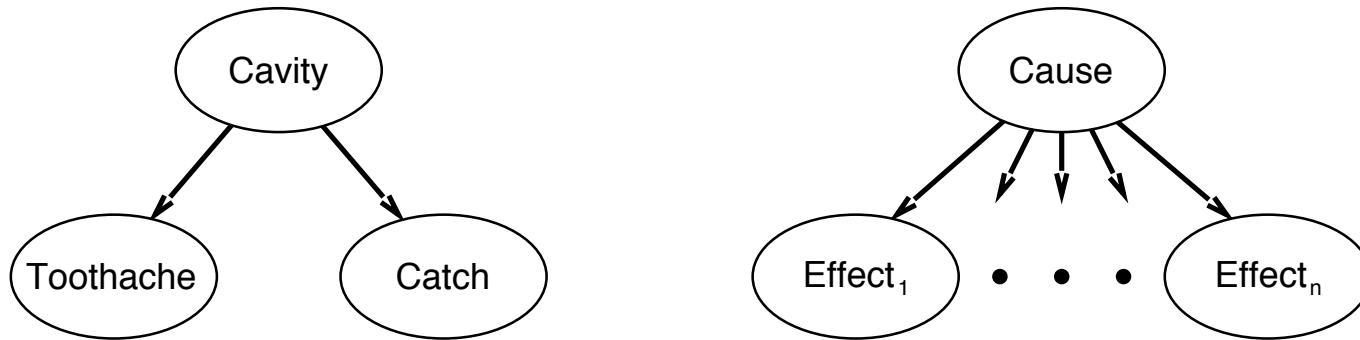
Note: posterior probability of meningitis still very small!

Bayes' Rule and conditional independence

$$\begin{aligned} & \mathbf{P}(Cavity|toothache \wedge catch) \\ &= \alpha \mathbf{P}(toothache \wedge catch|Cavity)\mathbf{P}(Cavity) \\ &= \alpha \mathbf{P}(toothache|Cavity)\mathbf{P}(catch|Cavity)\mathbf{P}(Cavity) \end{aligned}$$

This is an example of a **naive Bayes** model:

$$\mathbf{P}(Cause, Effect_1, \dots, Effect_n) = \mathbf{P}(Cause)\prod_i \mathbf{P}(Effect_i|Cause)$$



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

Bayesian networks

Bayesian networks

A simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions

Syntax:

- a set of nodes, one per variable

- a directed, acyclic graph (link \approx “directly influences”)

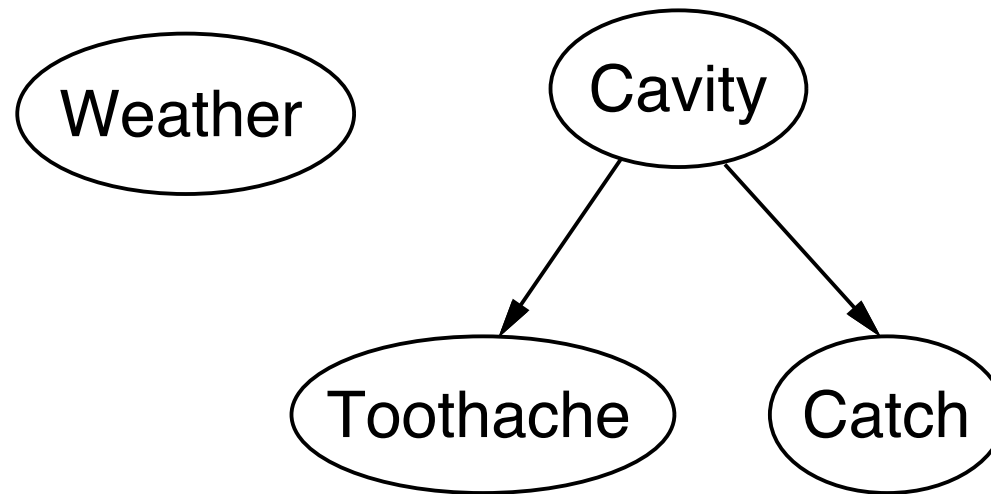
- a conditional distribution for each node given its parents:

$$P(X_i | Parents(X_i))$$

In the simplest case, conditional distribution represented as a **conditional probability table** (CPT) giving the distribution over X_i for each combination of parent values

Example

Topology of network encodes conditional independence assertions:



Weather is independent of the other variables

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

Example

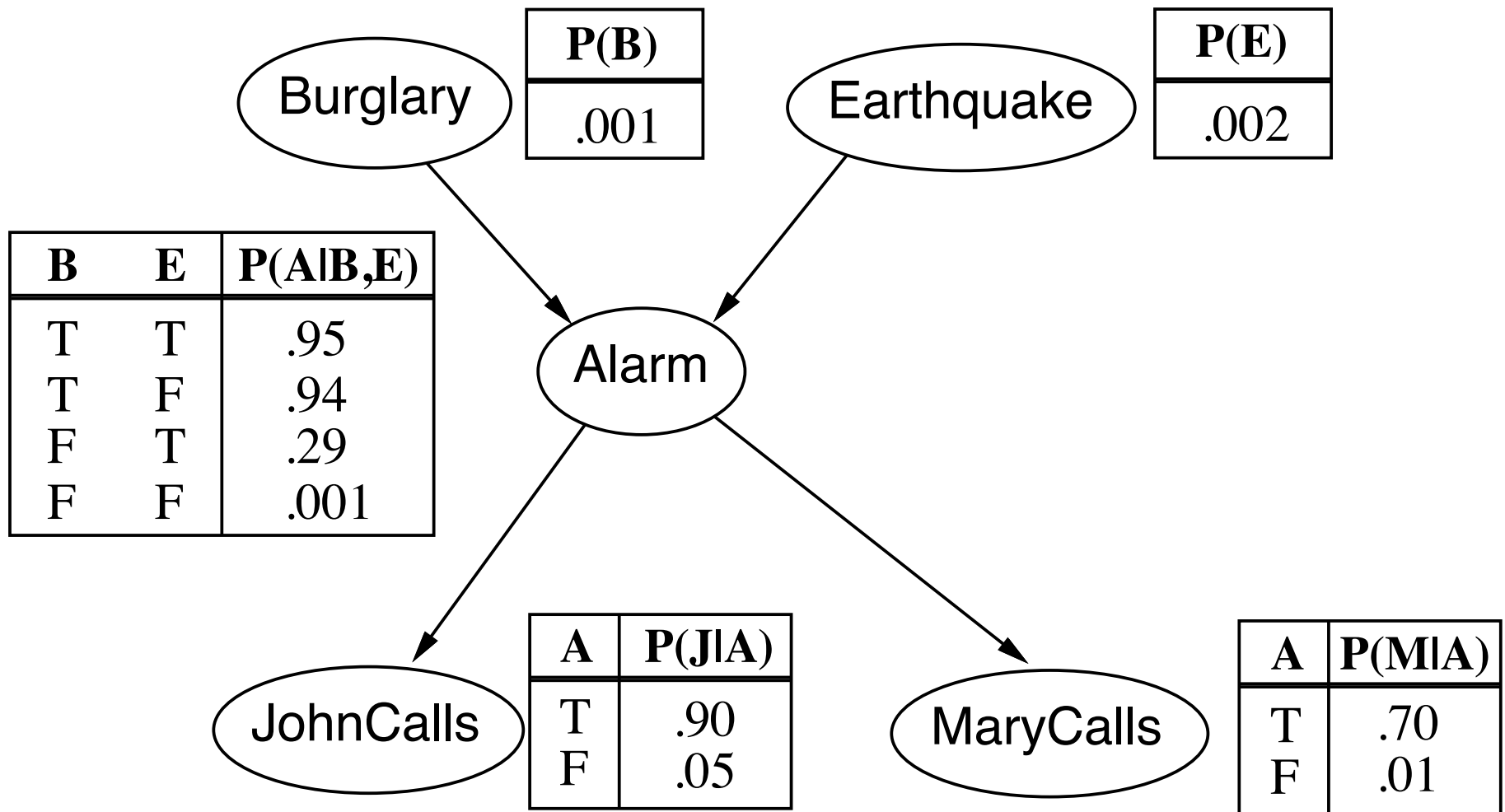
I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes. Is there a burglar?

Variables: *Burglar*, *Earthquake*, *Alarm*, *JohnCalls*, *MaryCalls*

Network topology reflects “causal” knowledge:

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

Example



Compactness

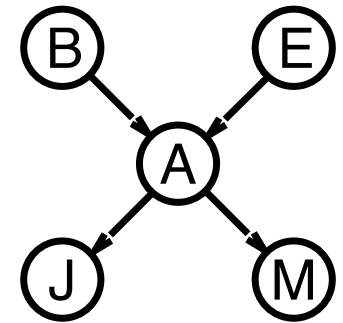
A CPT for Boolean X_i with k Boolean parents has 2^k rows for the combinations of parent values

Each row requires one number p for $X_i = \text{true}$ (the number for $X_i = \text{false}$ 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. $2^5 - 1 = 31$)



Global semantics

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

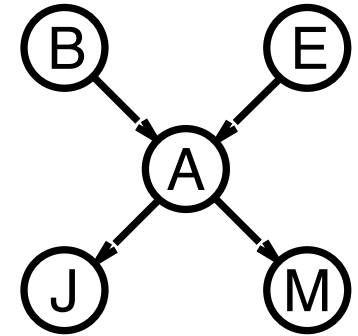
$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | \text{parents}(X_i))$$

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

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

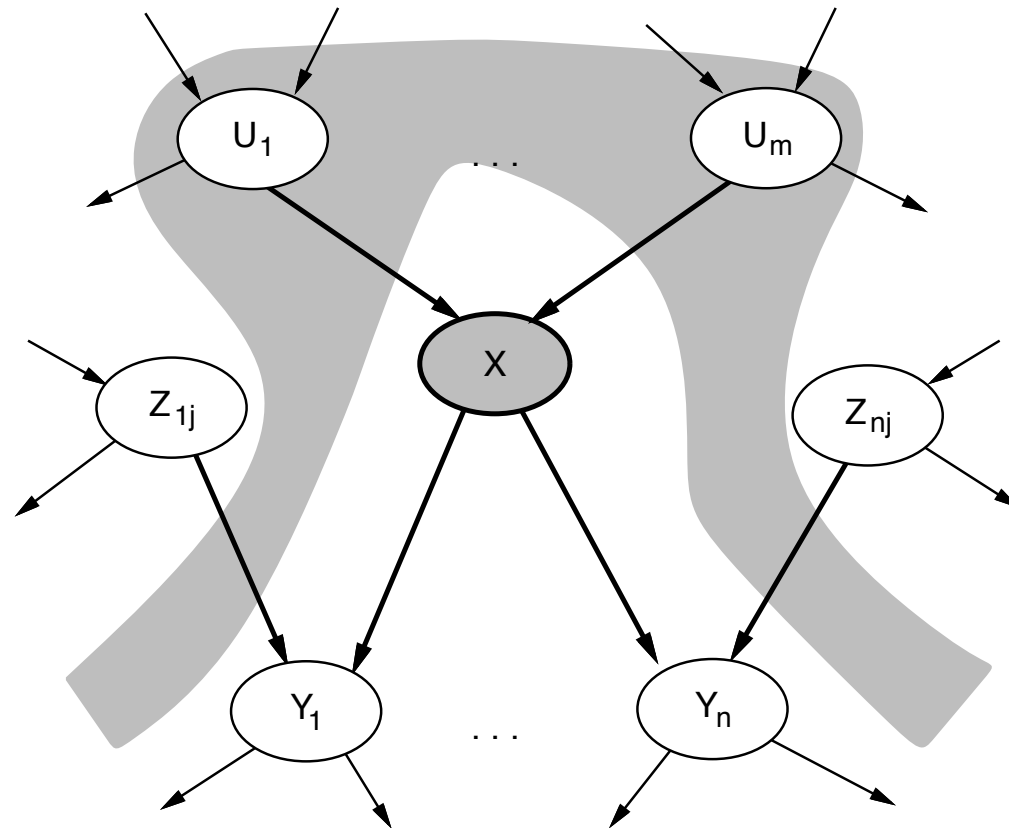
$$= 0.9 \times 0.7 \times 0.001 \times 0.999 \times 0.998$$

$$\approx 0.00063$$



Local semantics

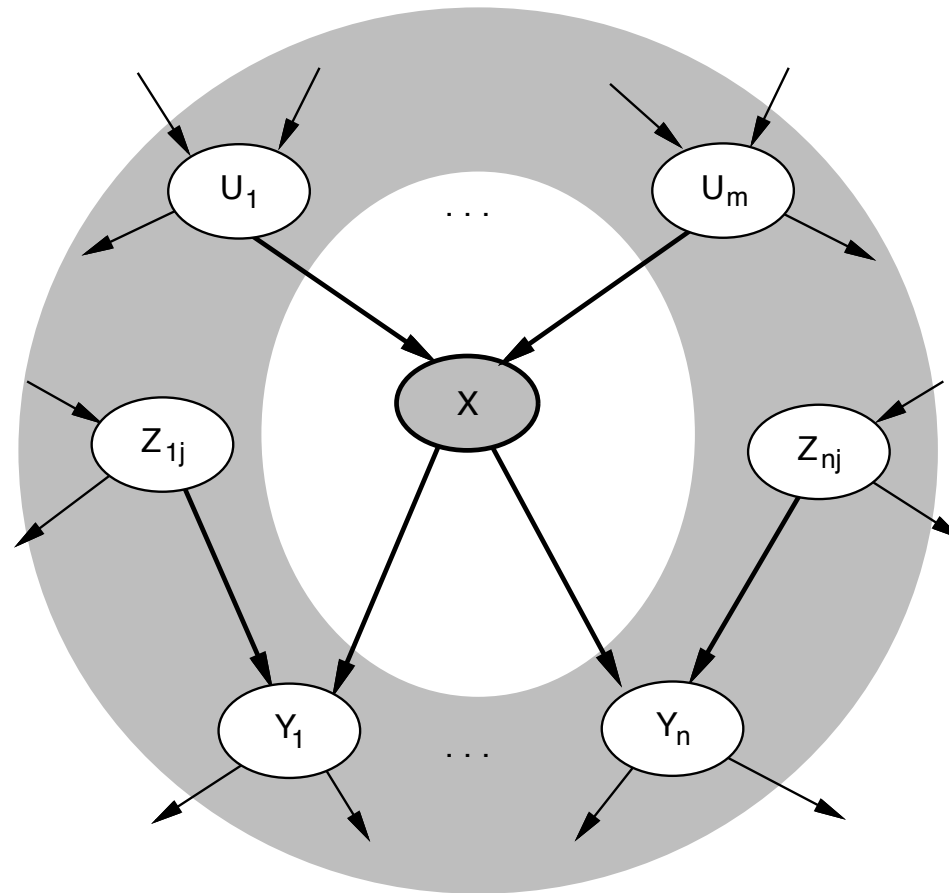
Local semantics: each node is conditionally independent of its nondescendants given its parents



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

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

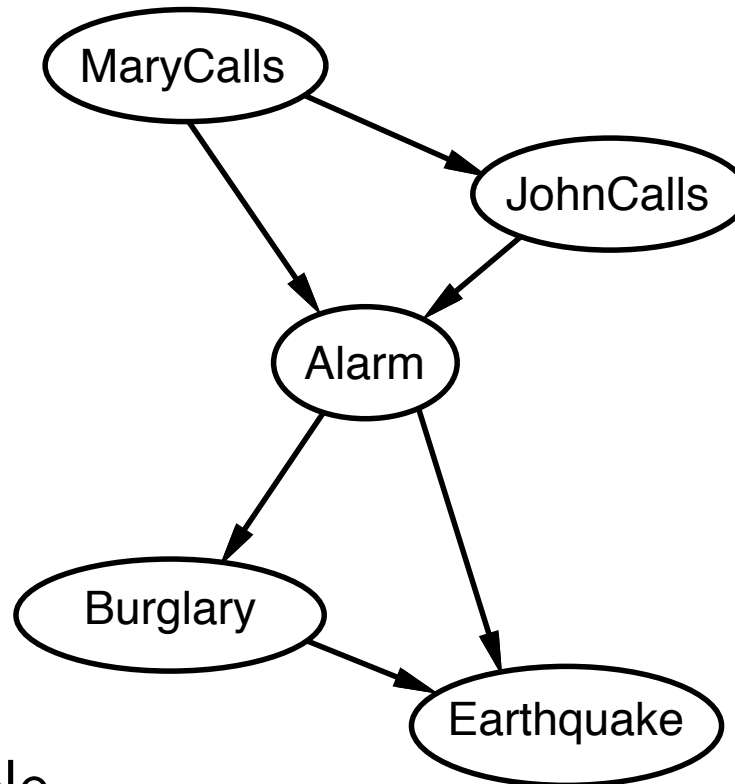
1. Choose an ordering of variables X_1, \dots, X_n
2. For $i = 1$ to n
 - add X_i to the network
 - select parents from X_1, \dots, X_{i-1} such that
$$\mathbf{P}(X_i | \text{Parents}(X_i)) = \mathbf{P}(X_i | X_1, \dots, X_{i-1})$$

This choice of parents guarantees the global semantics:

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

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

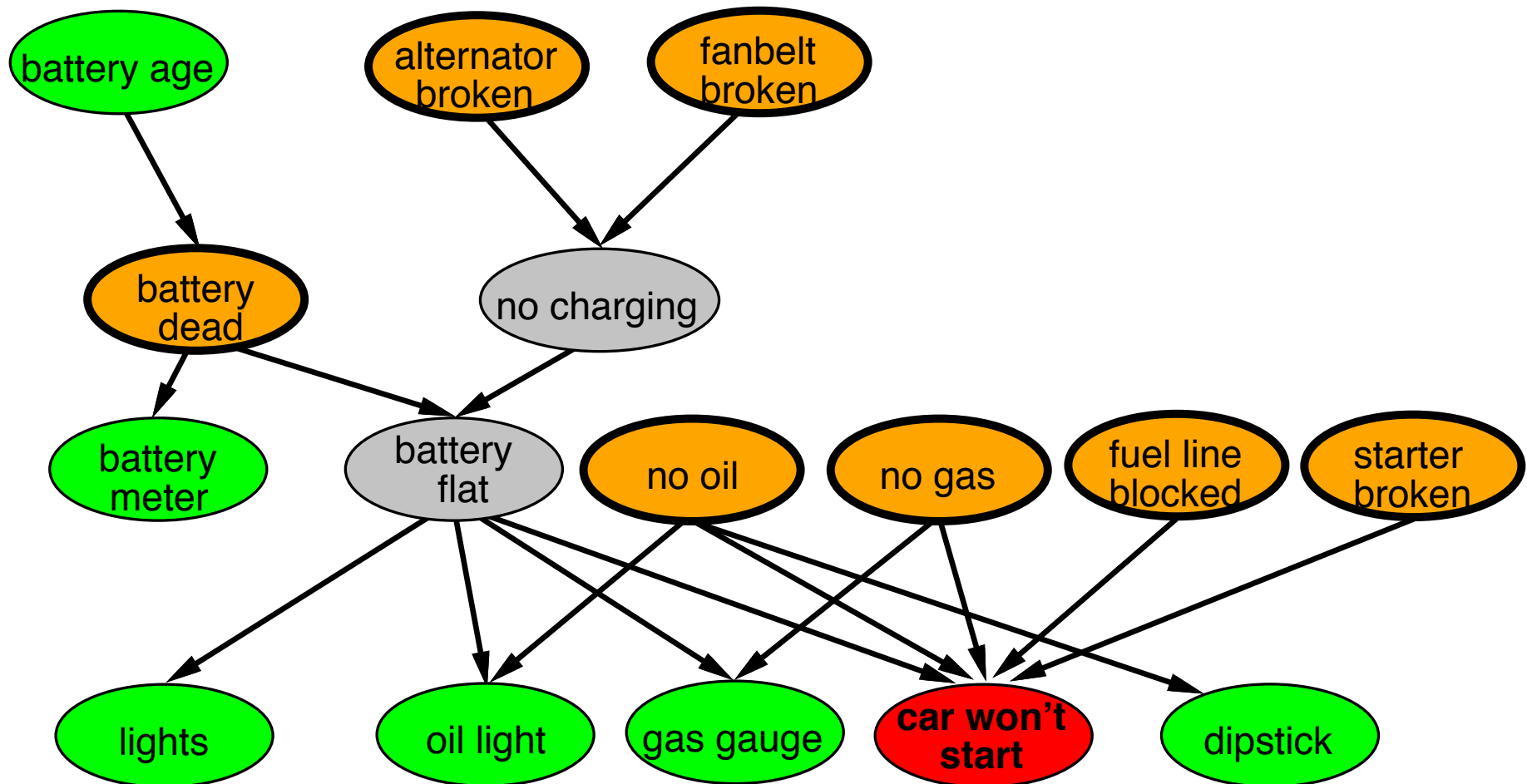
$P(E|B, A, J, M) = P(E|A, B)$? Yes

Example: 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



Compact conditional distributions



CPT grows exponentially with number of parents

CPT becomes infinite with continuous-valued parent or child

Solution: **canonical** distributions that are defined compactly

Deterministic nodes are the simplest case:

$$X = f(\text{Parents}(X)) \text{ for some function } f$$

E.g., Boolean functions

$$\textit{NorthAmerican} \Leftrightarrow \textit{Canadian} \vee \textit{US} \vee \textit{Mexican}$$

E.g., numerical relationships among continuous variables

$$\frac{\partial \textit{Level}}{\partial t} = \textit{inflow} + \textit{precipitation} - \textit{outflow} - \textit{evaporation}$$

Compact conditional distributions contd.



Noisy-OR distributions model multiple noninteracting causes

- 1) Parents $U_1 \dots U_k$ include all causes (can add leak node)
- 2) Independent failure probability q_i for each cause alone

$$\Rightarrow P(X|U_1 \dots U_j, \neg U_{j+1} \dots \neg U_k) = 1 - \prod_{i=1}^j q_i$$

<i>Cold</i>	<i>Flu</i>	<i>Malaria</i>	$P(\text{Fever})$	$P(\neg \text{Fever})$
F	F	F	0.0	1.0
F	F	T	0.9	0.1
F	T	F	0.8	0.2
F	T	T	0.98	$0.02 = 0.2 \times 0.1$
T	F	F	0.4	0.6
T	F	T	0.94	$0.06 = 0.6 \times 0.1$
T	T	F	0.88	$0.12 = 0.6 \times 0.2$
T	T	T	0.988	$0.012 = 0.6 \times 0.2 \times 0.1$

Number of parameters **linear** in number of parents

Inference in Bayesian networks

Inference tasks



Simple queries: compute posterior marginal $\mathbf{P}(X_i|\mathbf{E} = \mathbf{e})$

e.g., $P(\text{NoGas}|\text{Gauge} = \text{empty}, \text{Lights} = \text{on}, \text{Starts} = \text{false})$

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 $P(\text{outcome}|\text{action}, \text{evidence})$

Value of information: which evidence to seek next?

Sensitivity analysis: which probability values are most critical?

Explanation: why do I need a new starter motor?

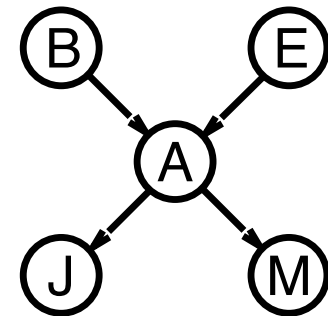
Exact inference

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 $\{X\} \cup \mathbf{E} \cup \mathbf{Y}$

$Q(X) \leftarrow$ a distribution over X , initially empty

for each value x_i of X **do**

 extend \mathbf{e} with value x_i for X

$Q(x_i) \leftarrow$ ENUMERATE-ALL(VARS[bn], \mathbf{e})

return NORMALIZE($Q(X)$)

function ENUMERATE-ALL($vars, \mathbf{e}$) **returns** a real number

if EMPTY?($vars$) **then return** 1.0

$Y \leftarrow$ FIRST($vars$)

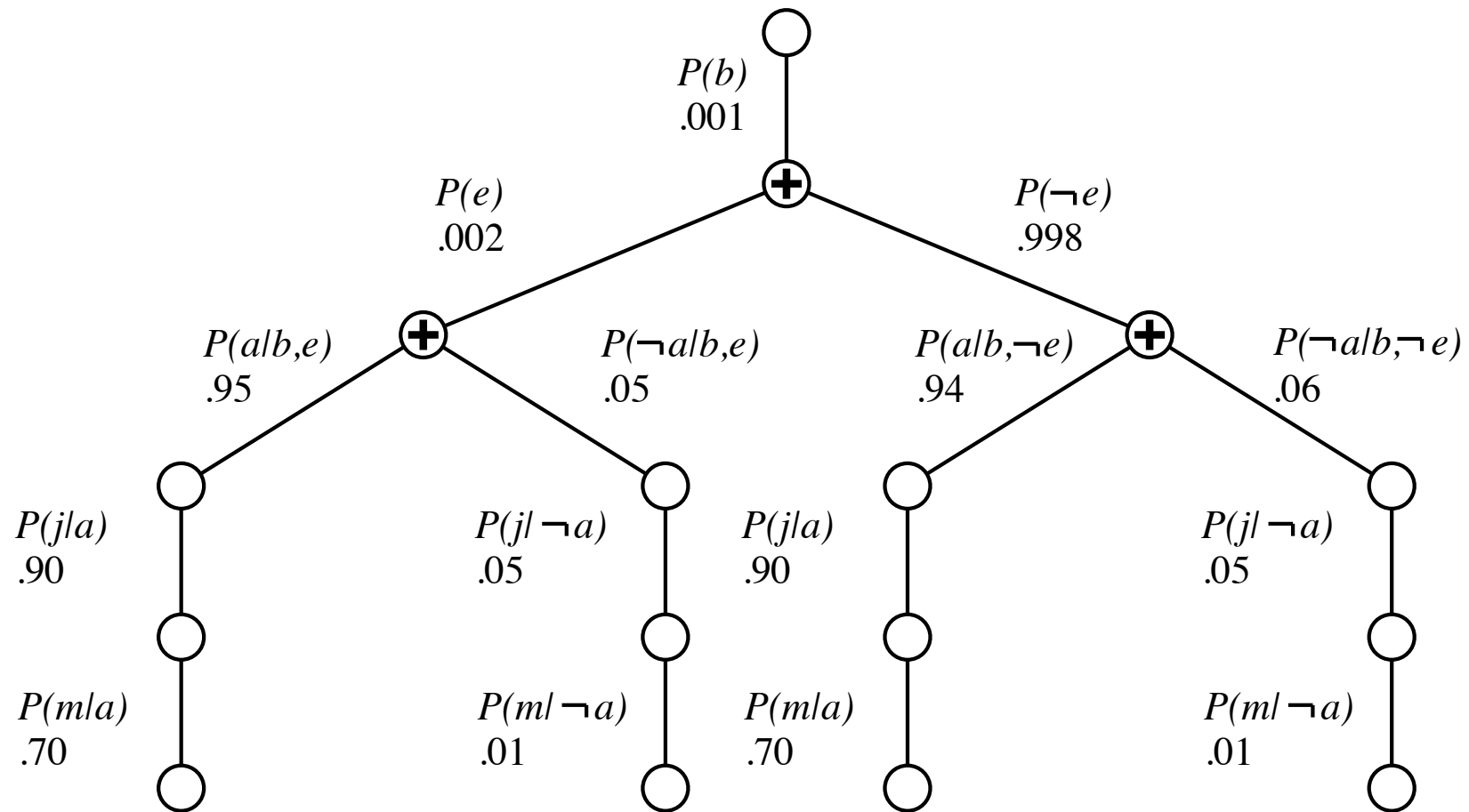
if Y has value y in \mathbf{e}

then return $P(y \mid Pa(Y)) \times$ ENUMERATE-ALL(REST($vars$), \mathbf{e})

else return $\sum_y P(y \mid Pa(Y)) \times$ 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

$$\begin{aligned} \mathbf{P}(B|j, m) &= \alpha \underbrace{\mathbf{P}(B)}_B \underbrace{\sum_e P(e)}_E \underbrace{\sum_a \mathbf{P}(a|B, e)}_A \underbrace{P(j|a)}_J \underbrace{P(m|a)}_M \\ &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) P(j|a) f_M(a) \\ &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a \mathbf{P}(a|B, e) f_J(a) f_M(a) \\ &= \alpha \mathbf{P}(B) \sum_e P(e) \sum_a f_A(a, b, e) f_J(a) f_M(a) \\ &= \alpha \mathbf{P}(B) \sum_e P(e) f_{\bar{A}JM}(b, e) \text{ (sum out } A) \\ &= \alpha \mathbf{P}(B) f_{\bar{E}\bar{A}JM}(b) \text{ (sum out } E) \\ &= \alpha f_B(b) \times f_{\bar{E}\bar{A}JM}(b) \end{aligned}$$

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, \dots, f_i do not depend on X

Pointwise product of factors f_1 and f_2 :

$$\begin{aligned} f_1(x_1, \dots, x_j, y_1, \dots, y_k) \times f_2(y_1, \dots, y_k, z_1, \dots, z_l) \\ = f(x_1, \dots, x_j, y_1, \dots, y_k, z_1, \dots, z_l) \end{aligned}$$

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

Variable elimination algorithm



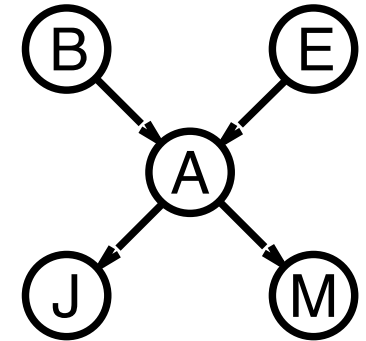
```
function ELIMINATION-ASK( $X, e, bn$ ) returns a distribution over  $X$   
  inputs:  $X$ , the query variable  
            $e$ , evidence specified as an event  
            $bn$ , a belief network specifying joint distribution  $\mathbf{P}(X_1, \dots, X_n)$   
  
   $factors \leftarrow []$ ;  $vars \leftarrow \text{REVERSE}(\text{VARS}[bn])$   
  for each  $var$  in  $vars$  do  
     $factors \leftarrow [\text{MAKE-FACTOR}(var, e) | factors]$   
    if  $var$  is a hidden variable then  $factors \leftarrow \text{SUM-OUT}(var, factors)$   
  return  $\text{NORMALIZE}(\text{POINTWISE-PRODUCT}(factors))$ 
```

Irrelevant variables

Consider the query $P(\text{JohnCalls} | \text{Burglary} = \text{true})$

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

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



Thm 1: Y is irrelevant unless $Y \in \text{Ancestors}(\{X\} \cup \mathbf{E})$

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

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

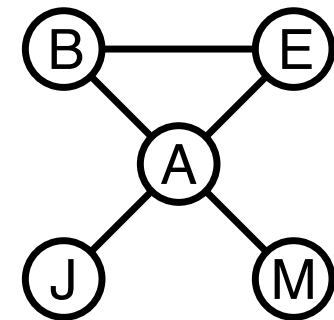
Irrelevant variables contd.

Defn: moral graph of Bayes net: marry all parents and drop arrows

Defn: **A** is m-separated from **B** by **C** iff separated by **C** in the moral graph

Thm 2: **Y** is irrelevant if m-separated from **X** by **E**

For $P(\text{JohnCalls} | \text{Alarm} = \text{true})$, both *Burglary* and *Earthquake* are irrelevant



Complexity of exact inference

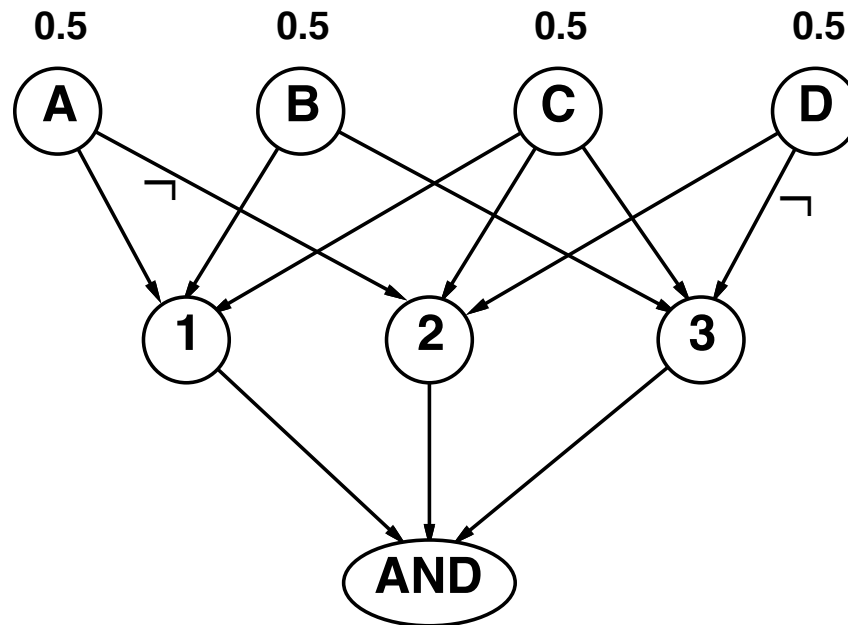
Singly connected networks (or **polytrees**):

- 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. $A \vee B \vee C$
2. $C \vee D \vee \neg A$
3. $B \vee C \vee \neg D$



Approximate inference

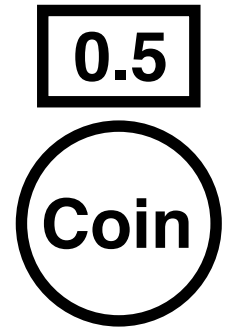
Inference by stochastic simulation

Basic idea:

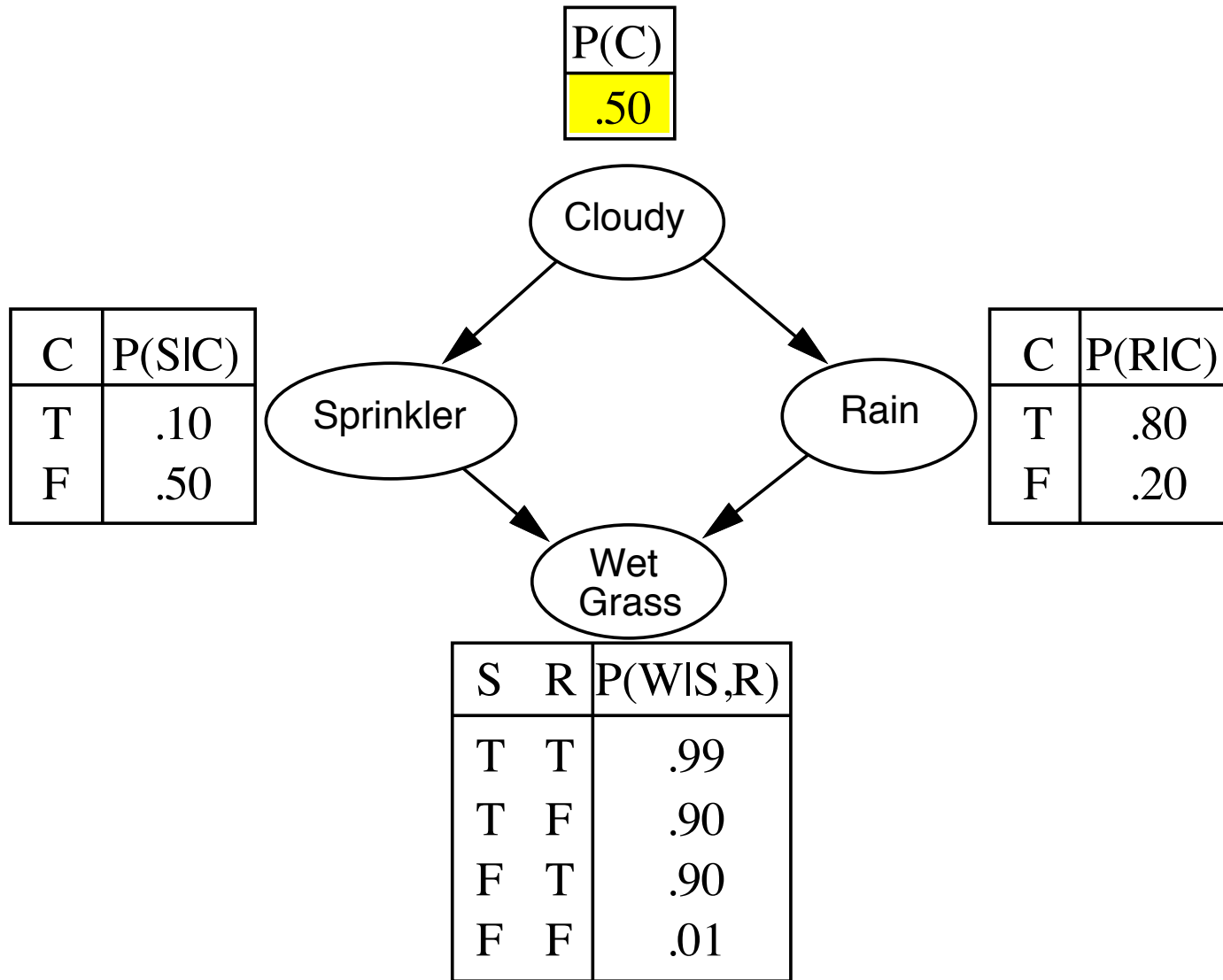
- 1) Draw N samples from a sampling distribution S
- 2) Compute an approximate posterior probability \hat{P}
- 3) Show this converges to the true probability P

Outline:

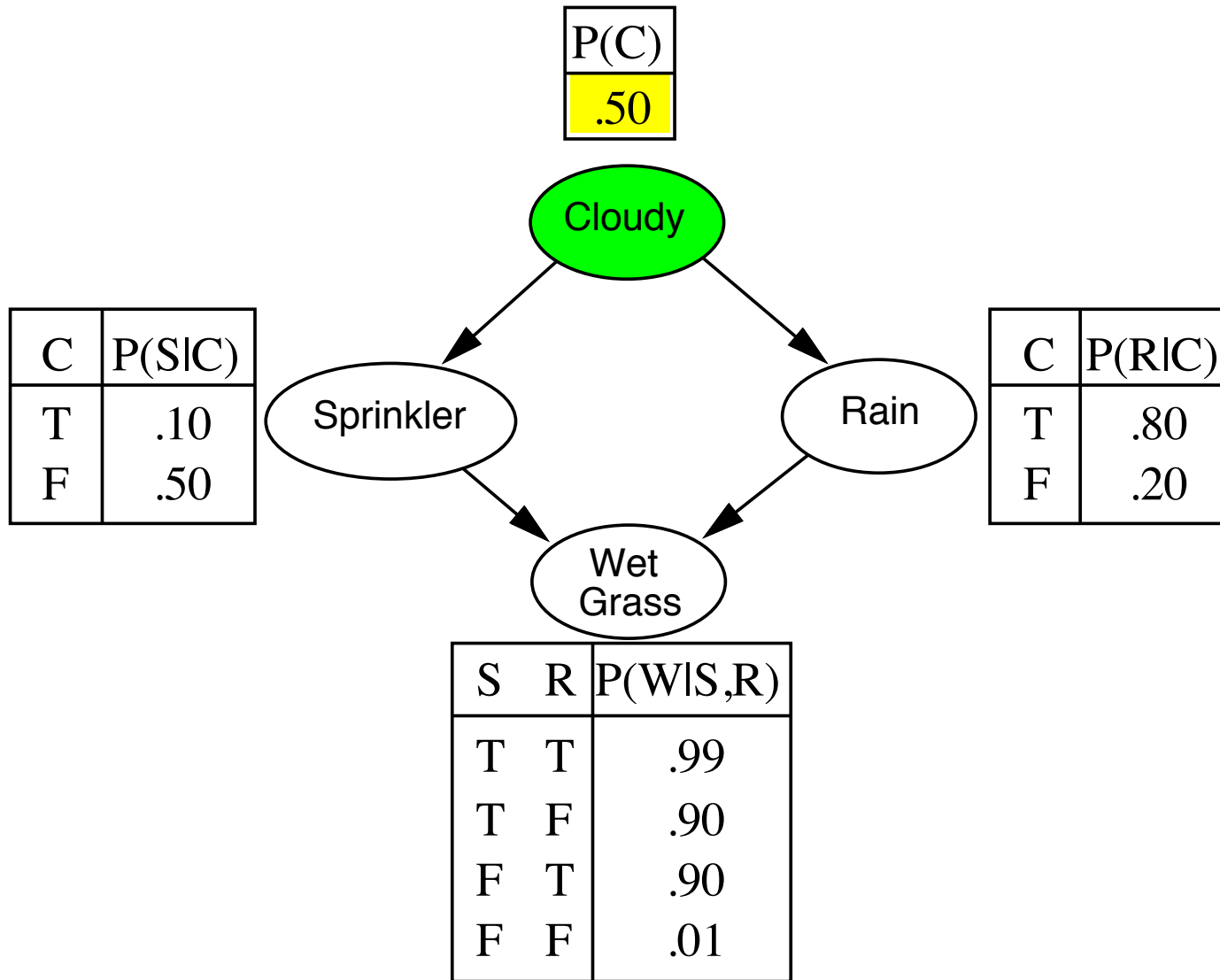
- Sampling from an empty network
- Rejection sampling: reject samples disagreeing with evidence
- Likelihood weighting: use evidence to weight samples
- Markov chain Monte Carlo (MCMC): sample from a stochastic process whose stationary distribution is the true posterior



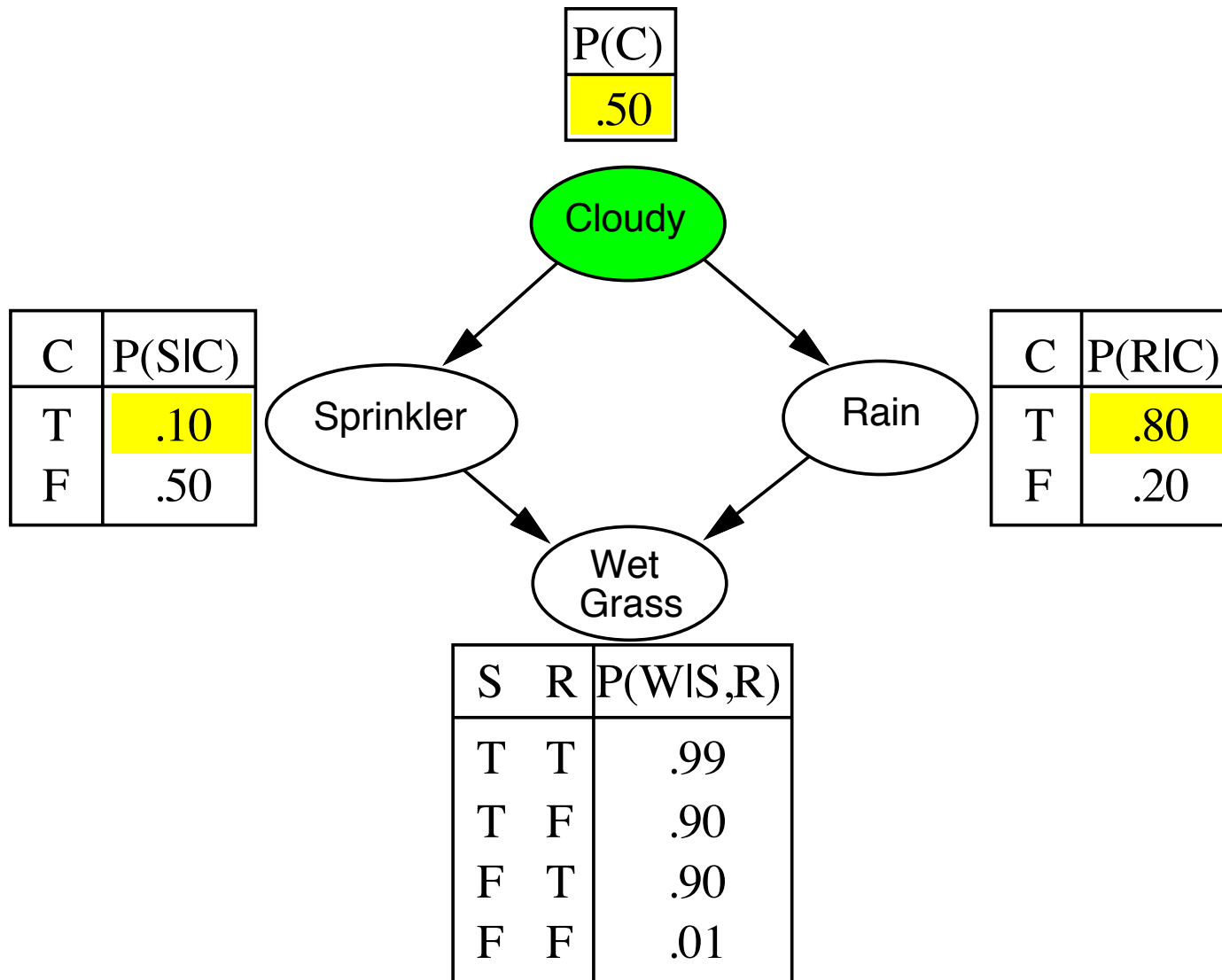
Sampling from an empty network



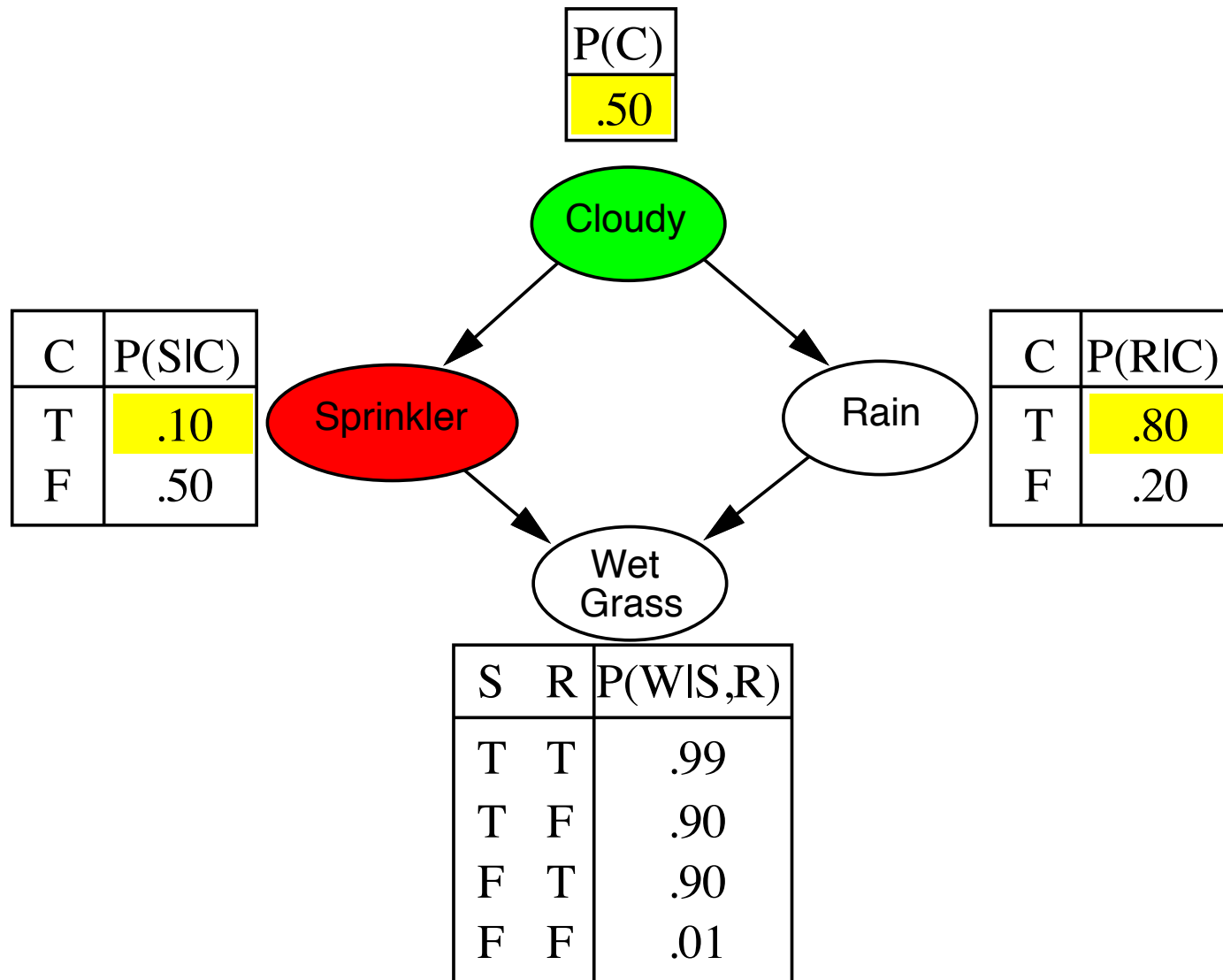
Sampling from an empty network



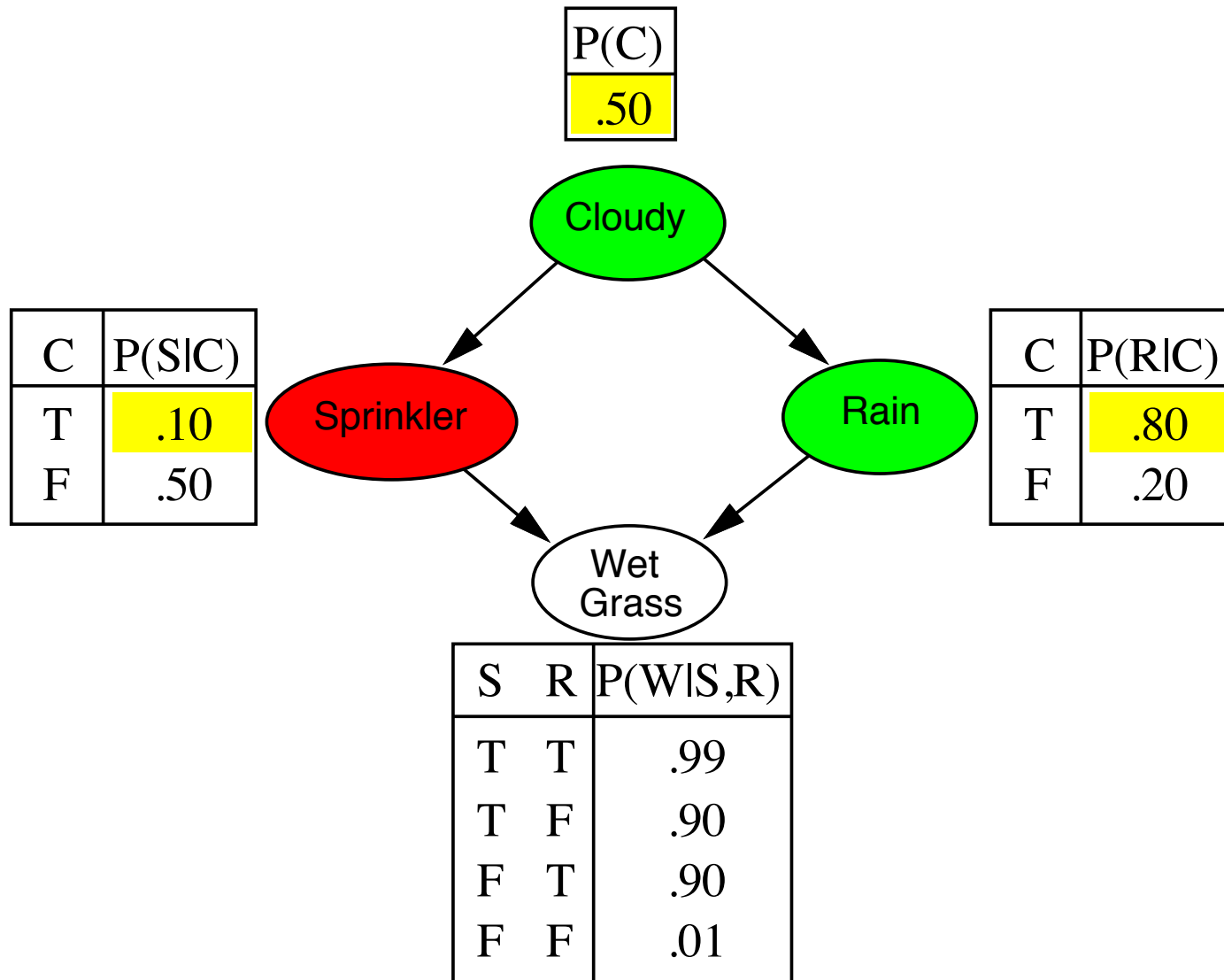
Sampling from an empty network



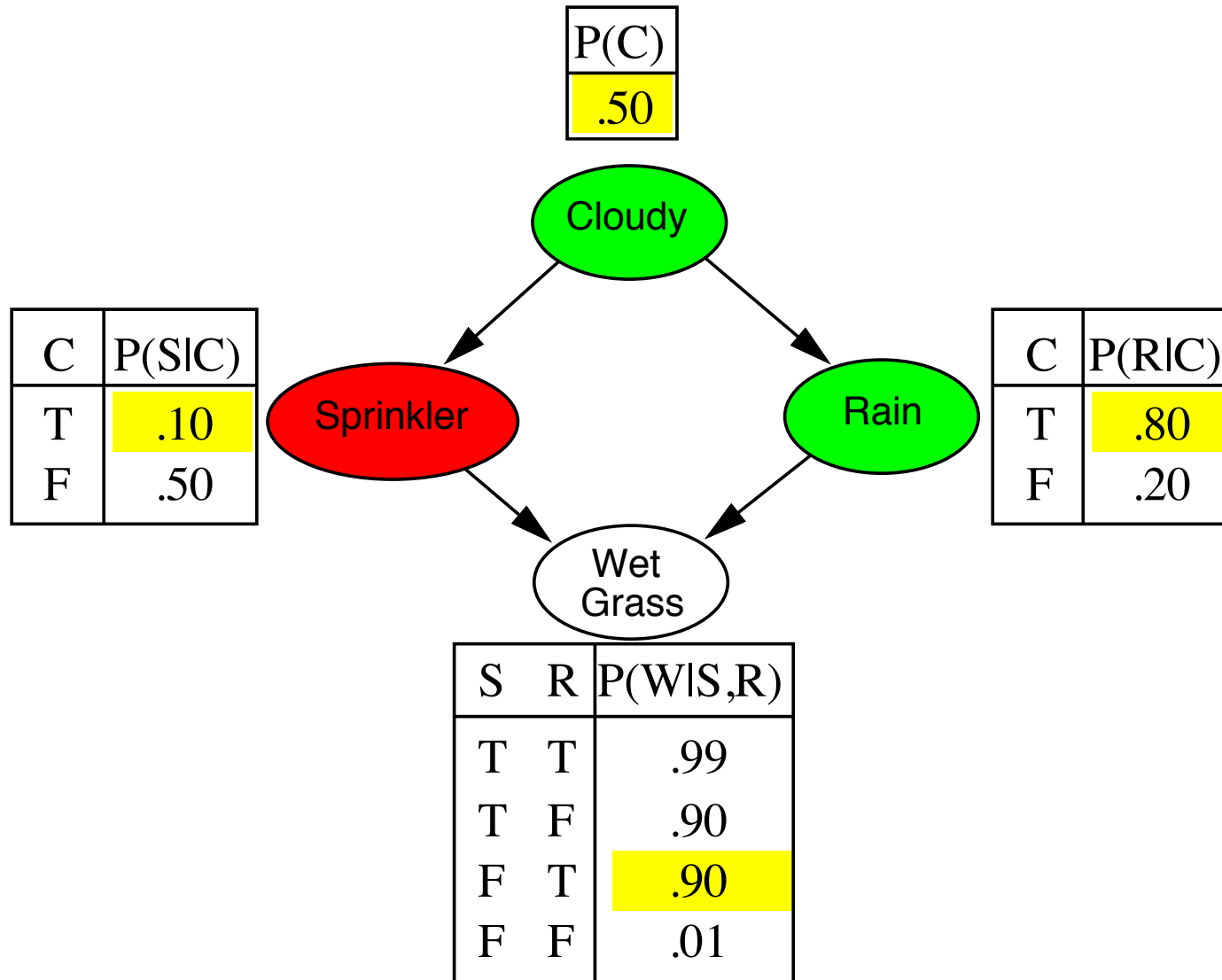
Sampling from an empty network



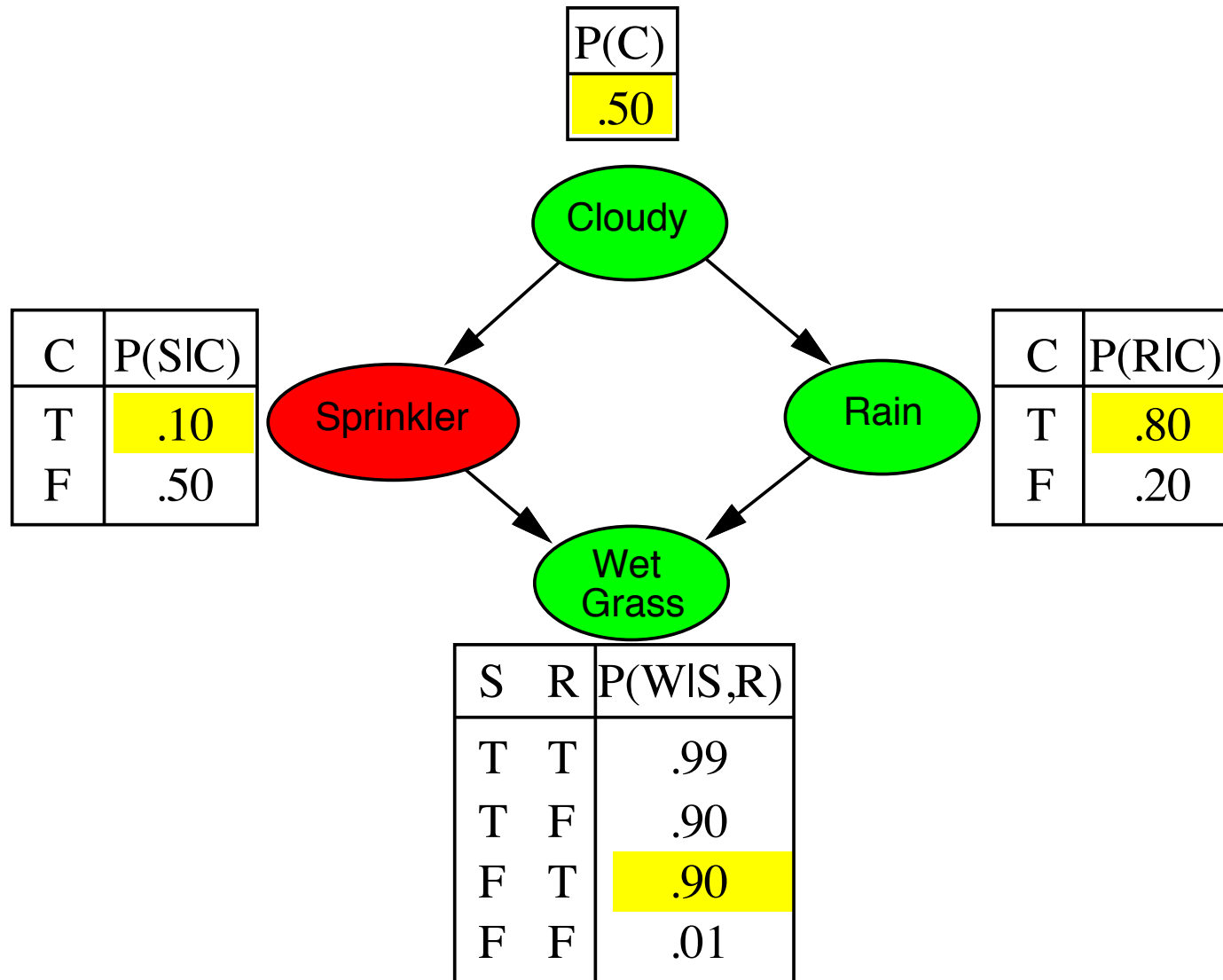
Sampling from an empty network



Sampling from an empty network



Sampling from an empty network



Sampling from an empty network

```
function PRIOR-SAMPLE(bn) returns an event sampled from bn  
inputs: bn, a belief network specifying joint distribution  $\mathbf{P}(X_1, \dots, X_n)$   
x  $\leftarrow$  an event with n elements  
for i = 1 to n do  
    xi  $\leftarrow$  a random sample from  $\mathbf{P}(X_i \mid \text{parents}(X_i))$   
        given the values of Parents(Xi) in x  
return x
```

Sampling from an empty network contd.



Probability that PRIORSAMPLE generates a particular event

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

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

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

That is, estimates derived from PRIORSAMPLE are **consistent**

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

Conditional Probability: Rejection sampling



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

```
function REJECTION-SAMPLING( $X, \mathbf{e}, bn, N$ ) returns an estimate of  $P(X|\mathbf{e})$ 
  local variables:  $\mathbf{N}$ , a vector of counts over  $X$ , initially zero
  for  $j = 1$  to  $N$  do
     $\mathbf{x} \leftarrow$  PRIOR-SAMPLE( $bn$ )
    if  $\mathbf{x}$  is consistent with  $\mathbf{e}$  then
       $\mathbf{N}[x] \leftarrow \mathbf{N}[x] + 1$  where  $x$  is the value of  $X$  in  $\mathbf{x}$ 
  return NORMALIZE( $\mathbf{N}[X]$ )
```

E.g., estimate $\mathbf{P}(Rain|Sprinkler = true)$ using 100 samples

27 samples have $Sprinkler = true$

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

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

Similar to a basic real-world empirical estimation procedure

Analysis of rejection sampling



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

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!

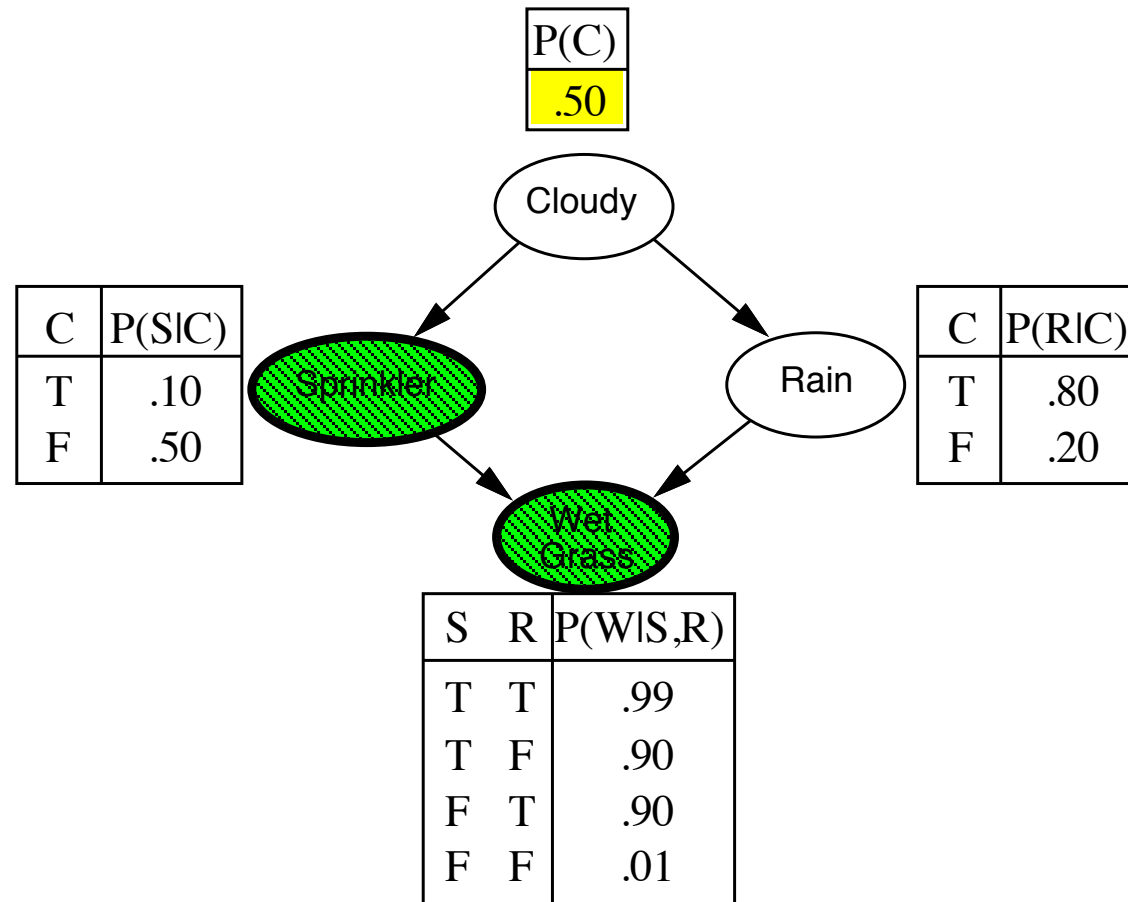
Likelihood weighting

Idea: fix evidence variables, sample only nonevidence variables, and weight each sample by the likelihood it accords the evidence

```
function LIKELIHOOD-WEIGHTING( $X, \mathbf{e}, bn, N$ ) returns an estimate of  $P(X|\mathbf{e})$   
  local variables:  $\mathbf{W}$ , a vector of weighted counts over  $X$ , initially zero  
  for  $j = 1$  to  $N$  do  
     $\mathbf{x}, w \leftarrow$  WEIGHTED-SAMPLE( $bn$ )  
     $\mathbf{W}[x] \leftarrow \mathbf{W}[x] + w$  where  $x$  is the value of  $X$  in  $\mathbf{x}$   
  return NORMALIZE( $\mathbf{W}[X]$ )
```

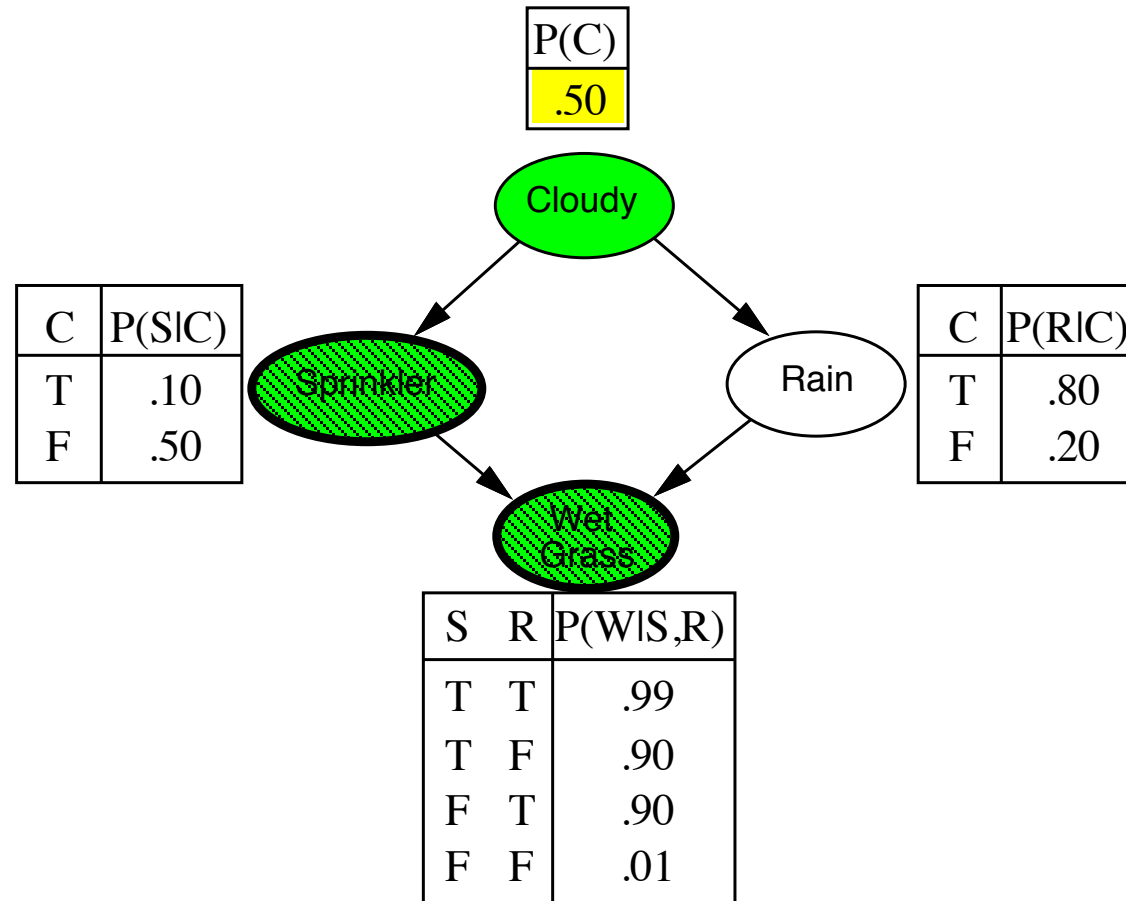
```
function WEIGHTED-SAMPLE( $bn, \mathbf{e}$ ) returns an event and a weight  
   $\mathbf{x} \leftarrow$  an event with  $n$  elements;  $w \leftarrow 1$   
  for  $i = 1$  to  $n$  do  
    if  $X_i$  has a value  $x_i$  in  $\mathbf{e}$   
      then  $w \leftarrow w \times P(X_i = x_i \mid \text{parents}(X_i))$   
      else  $x_i \leftarrow$  a random sample from  $\mathbf{P}(X_i \mid \text{parents}(X_i))$   
  return  $\mathbf{x}, w$ 
```

Likelihood weighting example



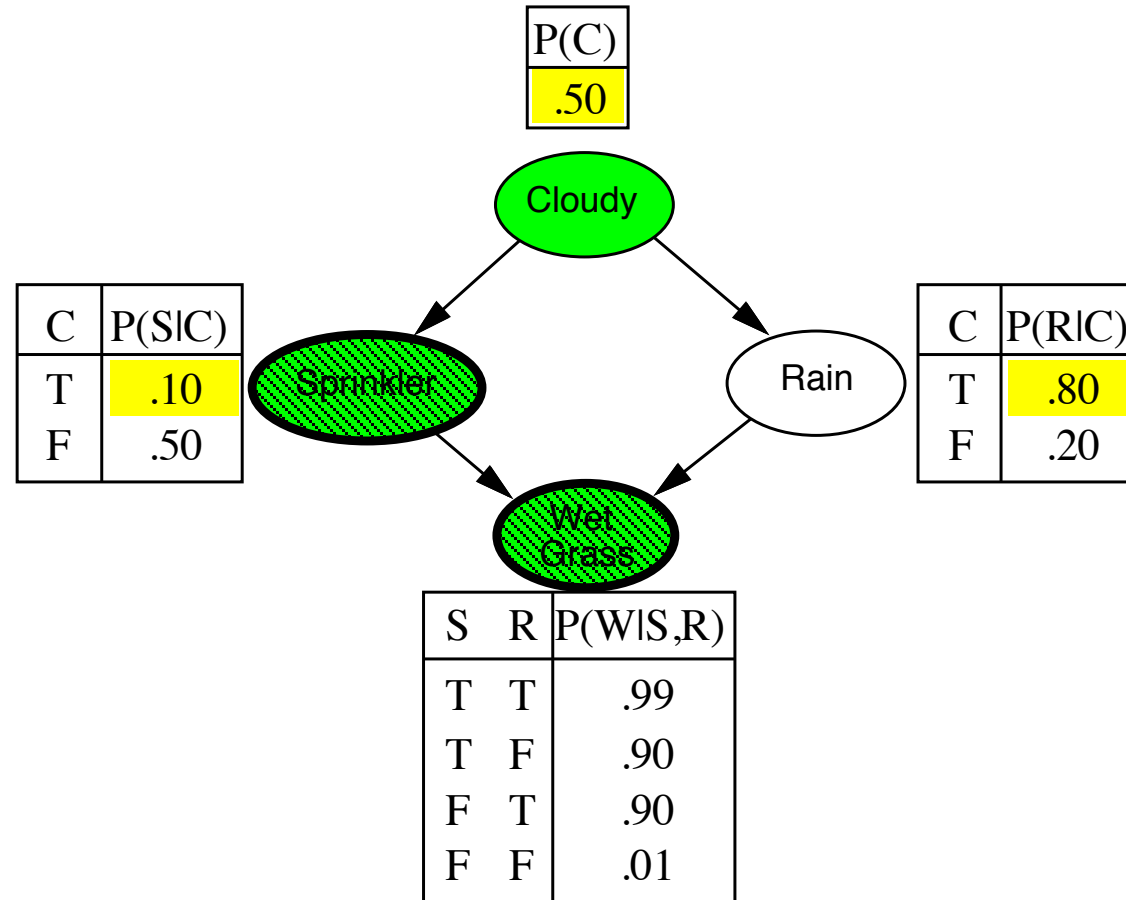
$$w = 1.0$$

Likelihood weighting example



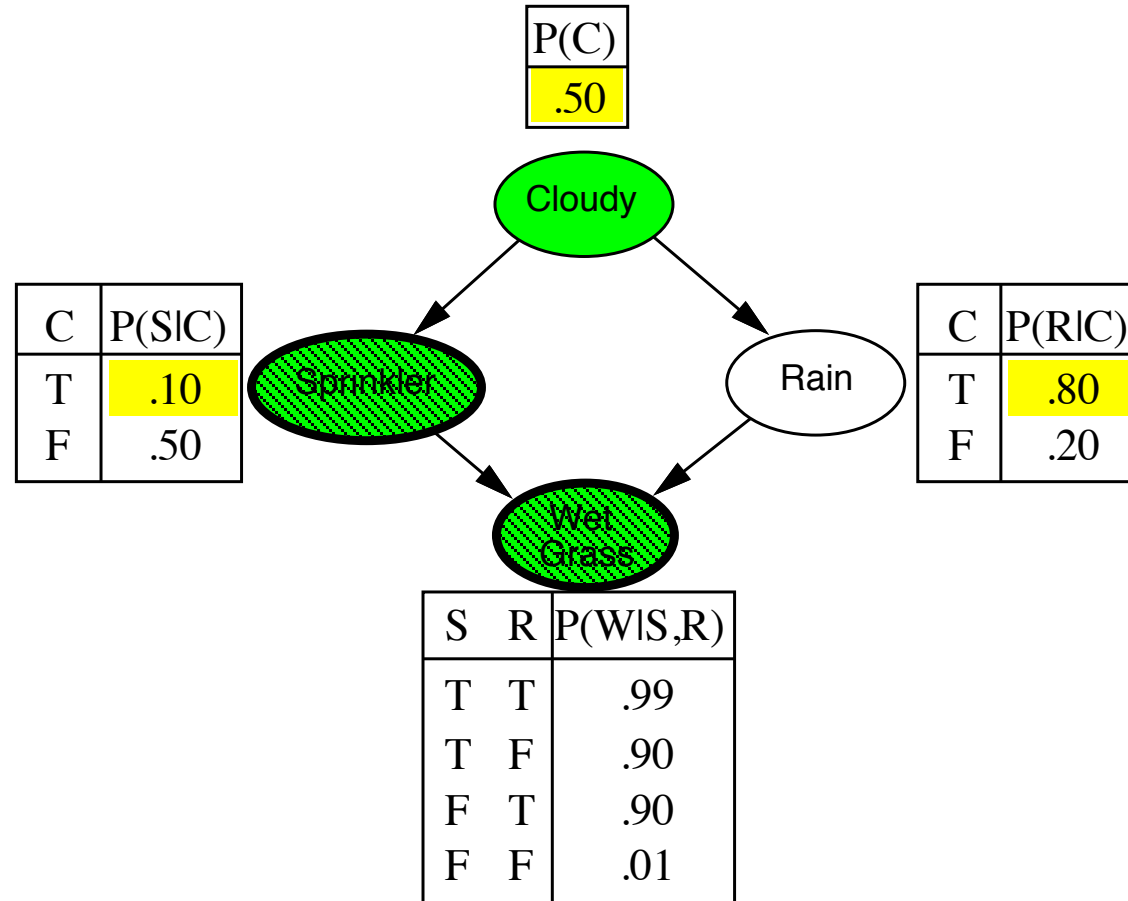
$$w = 1.0$$

Likelihood weighting example



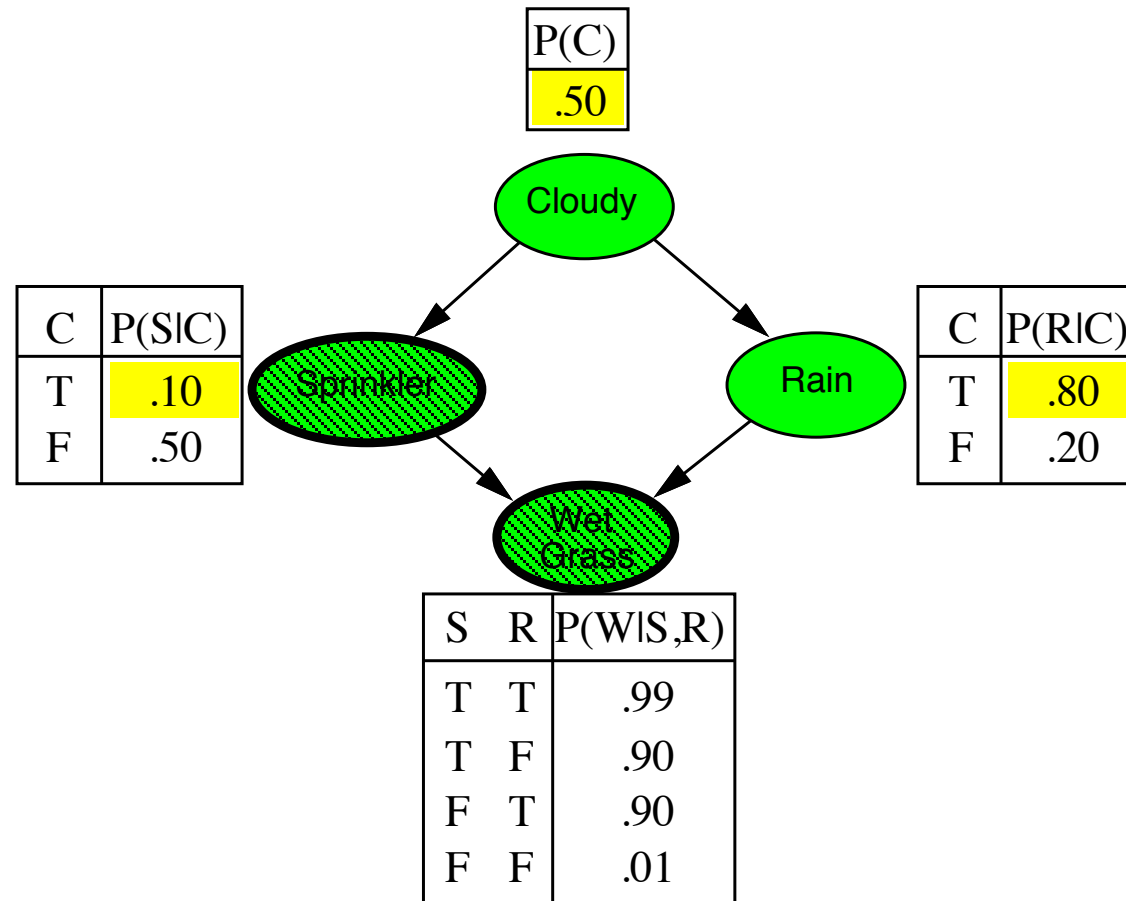
$$w = 1.0$$

Likelihood weighting example



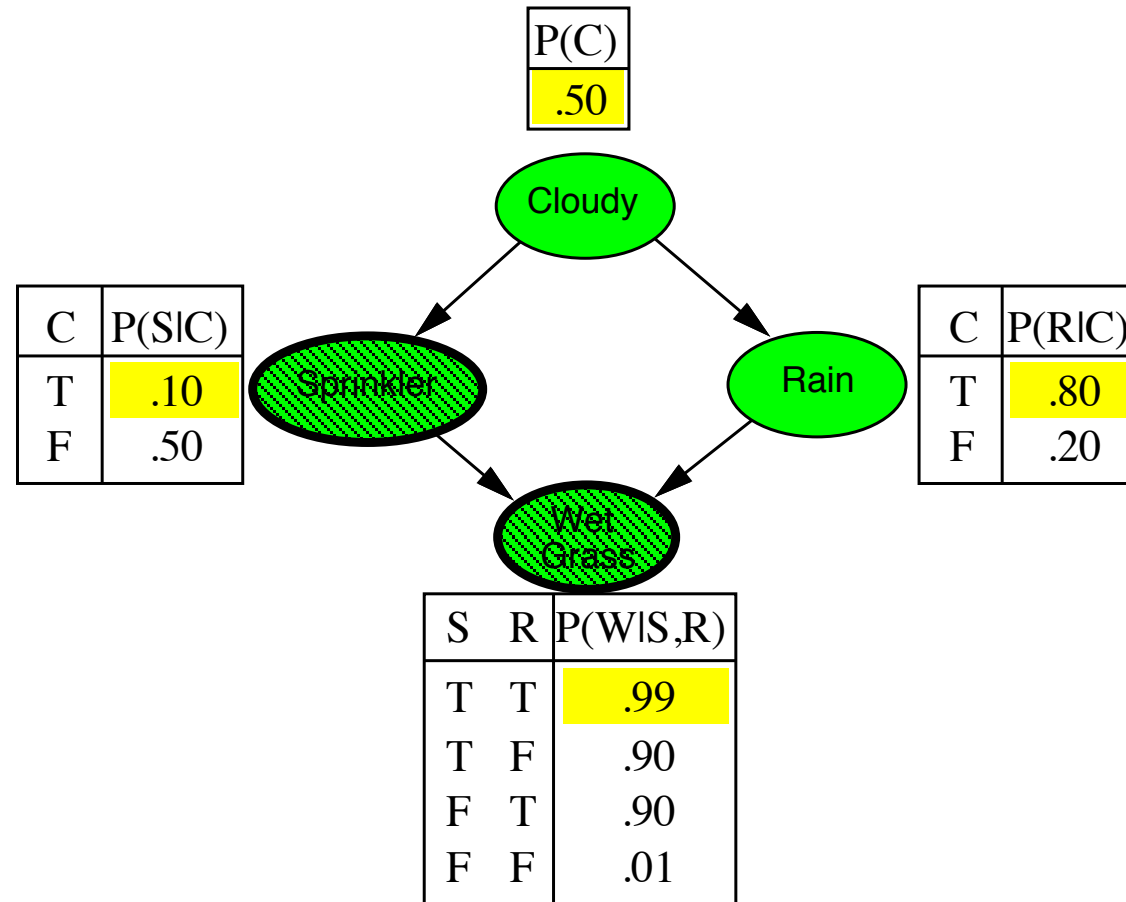
$$w = 1.0 \times 0.1$$

Likelihood weighting example



$$w = 1.0 \times 0.1$$

Likelihood weighting example



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

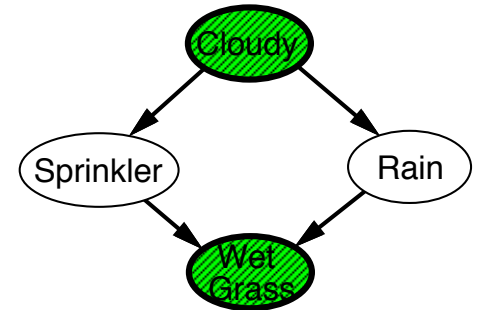
Likelihood weighting analysis

Sampling probability for WEIGHTEDSAMPLE is

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

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

⇒ somewhere “in between” prior and posterior distribution



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

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

Weighted sampling probability is

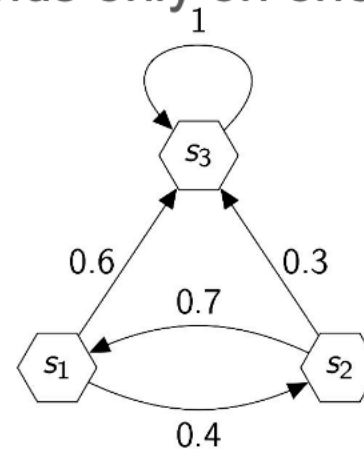
$$\begin{aligned} S_{WS}(\mathbf{z}, \mathbf{e})w(\mathbf{z}, \mathbf{e}) \\ &= \prod_{i=1}^l P(z_i | \text{parents}(Z_i)) \prod_{i=1}^m P(e_i | \text{parents}(E_i)) \\ &= P(\mathbf{z}, \mathbf{e}) \text{ (by standard global semantics of network)} \end{aligned}$$

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

MCMC — Markov Chain

- A triple \langle possibly infinite set \mathbf{S} of possible states, initial distribution over states \mathbf{P}_0 , transition matrix \mathbf{P} (\mathbf{T}) \rangle
- transition matrix - a matrix with probabilities \mathbf{P}_{ij} (\mathbf{T}_{ij}) that being in some state \mathbf{s}_i at time \mathbf{t} we will move to another state \mathbf{s}_j at time $\mathbf{t}+1$
- Markov property = next state depends only on one previous

$$S = \{s_1, s_2, s_3\}$$
$$P_0 = (0.5, 0.5, 0)$$
$$\mathbf{P} = \begin{pmatrix} 0 & 0.4 & 0.6 \\ 0.7 & 0 & 0.3 \\ 0 & 0 & 1 \end{pmatrix}$$



Typically circles are used to mark states
I used hexagons to distinguish from nodes in Bayesian networks
States are *values* of random variables

MCMC — Markov Chain



distribution over states

$$\begin{aligned} P(X^{t+1}) &= P(X^t = s_1)\mathbf{P}_{1\bullet} + P(X^t = s_2)\mathbf{P}_{2\bullet} + \dots + P(X^t = s_n)\mathbf{P}_{n\bullet} \\ &= P(X^t)\mathbf{P} \end{aligned}$$

where $P(X^t)$ is treated as a row vector

- Further: $P(X^t) = P(X^{t-1})\mathbf{P} = P(X^{t-2})\mathbf{P}^2 = \dots = P_0\mathbf{P}^t$

MCMC — Markov Chain

stationary distribution

Definition

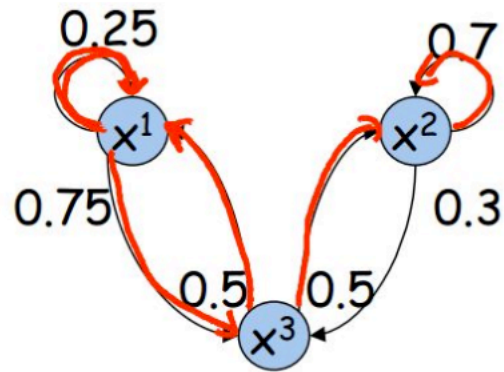
A **stationary distribution** of a Markov chain $\mathcal{M} = \langle S, P_0, \mathbf{P} \rangle$ is any distribution π over S satisfying

$$\pi \mathbf{P} = \pi$$

- A stationary distribution does not change over time
- Questions:
 - what are the conditions for existence of stationary distributions?
 - when does a Markov chain converge to a stationary distribution as $t \rightarrow \infty$?

MCMC — Markov Chain

stationary distribution example:



$$\pi(x^1) = 0.25\pi(x^1) + 0.5\pi(x^3)$$

$$\pi(x^2) = 0.7\pi(x^2) + 0.5\pi(x^3)$$

$$\pi(x^3) = 0.75\pi(x^1) + 0.3\pi(x^2)$$

$$\pi(x^1) + \pi(x^2) + \pi(x^3) = 1$$

$$\pi(x^1) = 0.2$$

$$\pi(x^2) = 0.5$$

$$\pi(x^3) = 0.3$$

Approximate inference using MCMC



“State” of network = current assignment to all variables.

Generate next state by sampling one variable given Markov blanket

Sample each variable in turn, keeping evidence fixed

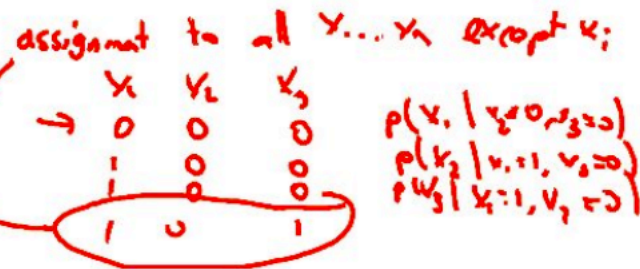
```
function MCMC-Ask( $X, \mathbf{e}, bn, N$ ) returns an estimate of  $P(X|\mathbf{e})$   
  local variables:  $\mathbf{N}[X]$ , a vector of counts over  $X$ , initially zero  
                     $\mathbf{Z}$ , the nonevidence variables in  $bn$   
                     $\mathbf{x}$ , the current state of the network, initially copied from  $\mathbf{e}$   
  
  initialize  $\mathbf{x}$  with random values for the variables in  $\mathbf{Y}$   
  for  $j = 1$  to  $N$  do  
    for each  $Z_i$  in  $\mathbf{Z}$  do  
      sample the value of  $Z_i$  in  $\mathbf{x}$  from  $\mathbf{P}(Z_i|mb(Z_i))$   
        given the values of  $MB(Z_i)$  in  $\mathbf{x}$   
       $\mathbf{N}[x] \leftarrow \mathbf{N}[x] + 1$  where  $x$  is the value of  $X$  in  $\mathbf{x}$   
  return NORMALIZE( $\mathbf{N}[X]$ )
```

Can also choose a variable to sample at random each time

Approximate inference using MCMC

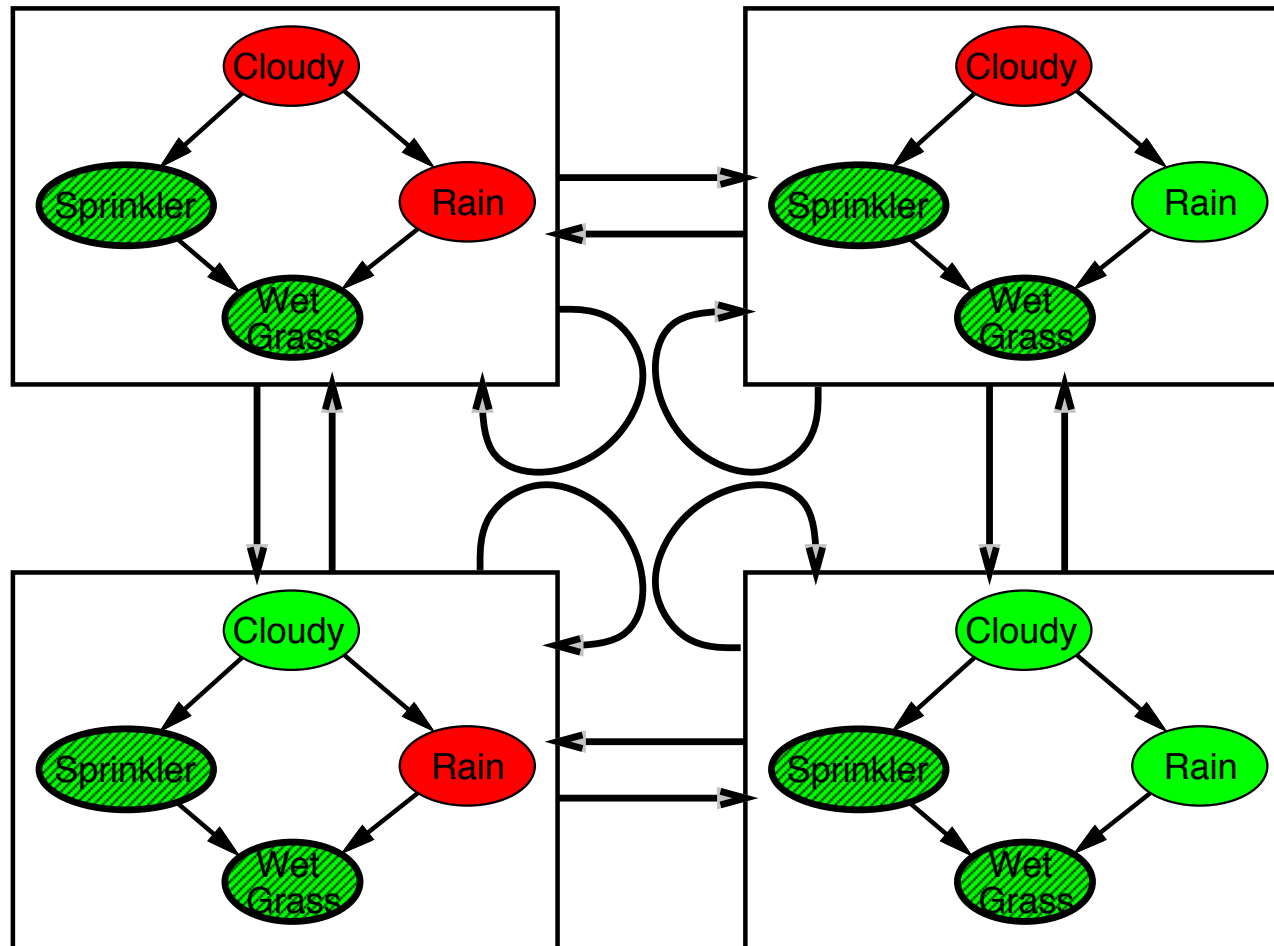
Gibbs sampling:

- Target distribution $P_{\Phi}(X_1, \dots, X_n)$
- Markov chain state space: complete assignments \mathbf{x} to $\mathbf{X} = \{X_1, \dots, X_n\}$
- Transition model given starting state \mathbf{x} :
 - For $i=1, \dots, n$
 - Sample $x_i \sim P_{\Phi}(X_i | \mathbf{x}_{-i})$
 - Set $\mathbf{x}' = \mathbf{x}$



The Markov chain

With *Sprinkler = true*, *WetGrass = true*, there are four states:



Wander about for a while, average what you see

MCMC example contd.



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*

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

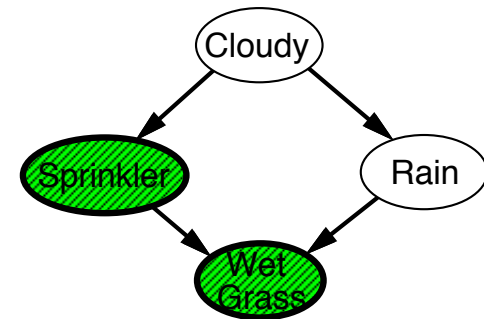
Theorem: chain approaches **stationary distribution**:

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:

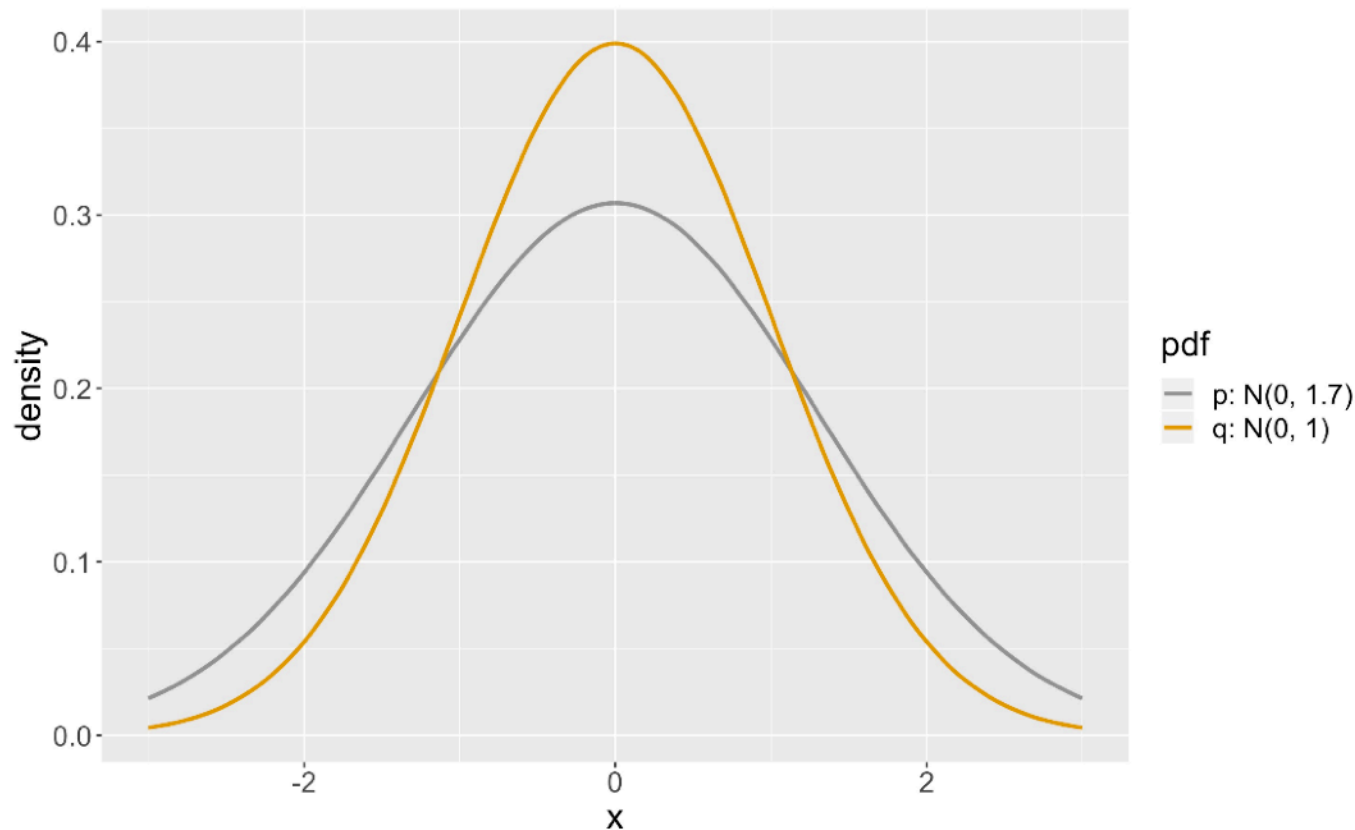
- 1) Difficult to tell if convergence has been achieved
- 2) Can be wasteful if Markov blanket is large:

$P(X_i | mb(X_i))$ won't change much (law of large numbers)

Variational Inference

- Variational inference turn the estimation of the posterior into an **optimization** problem (i.e. maximize or minimize)
- Main idea: find **another probability** function that is easier to work with than the posterior
- **Minimize** the difference between the new probability function an the posterior

Difference between density functions



- Let $p(x)$ and $q(x)$ be two probability densities
- A naive approach for a given x : $\frac{q(x)}{p(x)}$

Difference between density functions

- Let $p(x)$ and $q(x)$ be two probability densities
- A naive approach for a given x : $\frac{q(x)}{p(x)}$
- However, if we flip the terms we see that the absolute value of this measure also changes: e.g. $\frac{0.3}{0.1} = 3$, but $\frac{0.1}{0.3} = \frac{1}{3}$
- If we take the **logarithm**, the absolute value stays constant after flipping the probabilities, only the sign changes:

$$\log\left(\frac{q(x)}{p(x)}\right) = -\log\left(\frac{p(x)}{q(x)}\right)$$

KL Divergence

- Assume that we are interested mainly on $q(x)$
 - Use $q(x)$ as a weight for the difference measure: $q(x) \log\left(\frac{q(x)}{p(x)}\right)$
 - When $q(x)$ is low, the difference measure $\log\left(\frac{q(x)}{p(x)}\right)$ does not matter as much, as when $q(x)$ is high!
-
- As a final step, we integrate over all the possible values of x (or sum if the density is not continuous)

$$D_{KL} = \int_{\mathcal{X}} q(x) \log\left(\frac{q(x)}{p(x)}\right) dx$$

- D_{KL} is called the **Kullback-Leibler divergence** of p from q

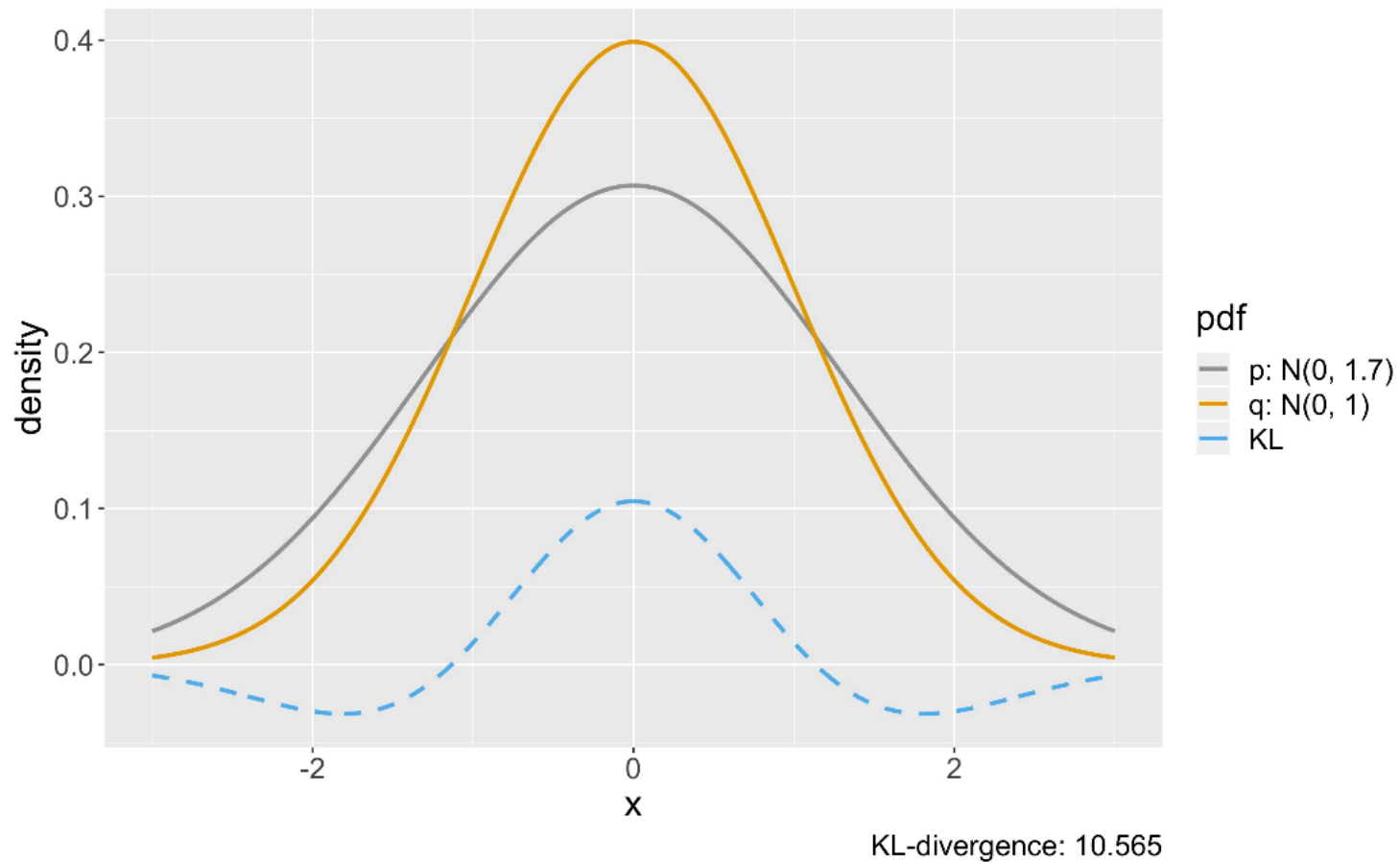
KL Divergence

$$D_{KL}(q \parallel p) = \int_{\mathcal{X}} q(x) \log\left(\frac{q(x)}{p(x)}\right) dx$$

- Common measure for the divergence between two probability densities
- The Kullback-Leibler divergence is **not symmetric**, and thus is cannot be called a "distance": $D_{KL}(q \parallel p) \neq D_{KL}(p \parallel q)$

KL Divergence

Examples of KL-Divergence: $D_{KL}(Q \parallel P)$



Variational approximation of the posterior

- Find a $q^*(\theta)$ which minimizes the KL divergence between $q(\theta)$ and the posterior $p(\theta|y)$:

$$D_{KL}(q \parallel p) = \int_{\Theta} q(\theta) \log\left(\frac{q(\theta)}{p(\theta|y)}\right)$$

- However, in order to calculate the KL divergence we would have to know the posterior $p(\theta|y)$ which is **intractable**.

Find an alternative way to optimize the divergence!

ELBO: evidence lower bound

- It can be shown that:

$$\log p(y) = \underbrace{\int_{\Theta} q(\theta) \log \left(\frac{p(y, \theta)}{q(\theta)} \right)}_{\mathcal{L}} + \underbrace{\int_{\Theta} q(\theta) \log \left(\frac{q(\theta)}{p(\theta|y)} \right)}_{D_{KL}(q||p)}$$

- $D_{KL}(q || p)$ is the Kullback-Leibler divergence
- \mathcal{L} is the evidence lower bound
- The Kullback-Leibler divergence is intractable, because it contains the posterior
- On the other hand, it is possible to compute all the terms in \mathcal{L} , as $p(y, \theta) = p(\theta)p(y|\theta)$ is known

ELBO: evidence lower bound

$$\begin{aligned}\log p(y) &= \log p(y) \cdot 1 = \log p(y) \int_{\Theta} q(\theta) d\theta \\ &= \int_{\Theta} q(\theta) \log p(y) d\theta = \int_{\Theta} q(\theta) \log \left(\frac{p(y, \theta)}{p(\theta|y)} \right) d\theta \\ &= \int_{\Theta} q(\theta) \log \left(\frac{p(y, \theta)}{p(\theta|y)} \cdot \frac{q(\theta)}{q(\theta)} \right) d\theta \\ &= \int_{\Theta} q(\theta) \log \left(\frac{p(y, \theta)}{q(\theta)} \right) + \int_{\Theta} q(\theta) \log \left(\frac{q(\theta)}{p(\theta|y)} \right)\end{aligned}$$

Which can be written as (Ormerod & Wand, 2010):

$$\log p(y) = \mathcal{L} + D_{KL}(q \parallel p)$$

- By rearranging we get:

$$\mathcal{L} = \log p(y) - D_{KL}(q \parallel p)$$

- As $D_{KL} \geq 0 \Rightarrow \log p(y) \geq \mathcal{L}$
- Therefore, \mathcal{L} is the lower bound of the logarithm of the evidence $p(y)$
- Hence the name evidence lower bound or ELBO for \mathcal{L}

ELBO: evidence lower bound

$$\mathcal{L} = \log p(y) - D_{KL}(q \parallel p)$$

- **Key idea:** Maximizing the ELBO is equivalent to minimizing the Kullback-Leibler divergence
- The idea of maximizing the ELBO is the basis of most variational inference approaches

Summary

Exact inference by variable elimination:

- polytime on polytrees, NP-hard on general graphs
- space = time, very sensitive to topology

Approximate inference by LW, MCMC:

- LW does poorly when there is 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