Inference in Bayesian networks

AIMA2e Chapter 14.4–5

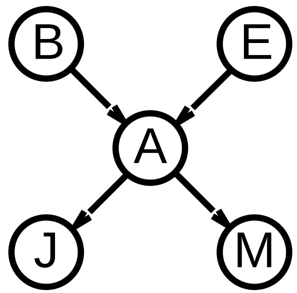
Outline

- ◊ Exact inference
- ◊ Approximate inference

Inference by enumeration

Simple query on the burglary network:

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

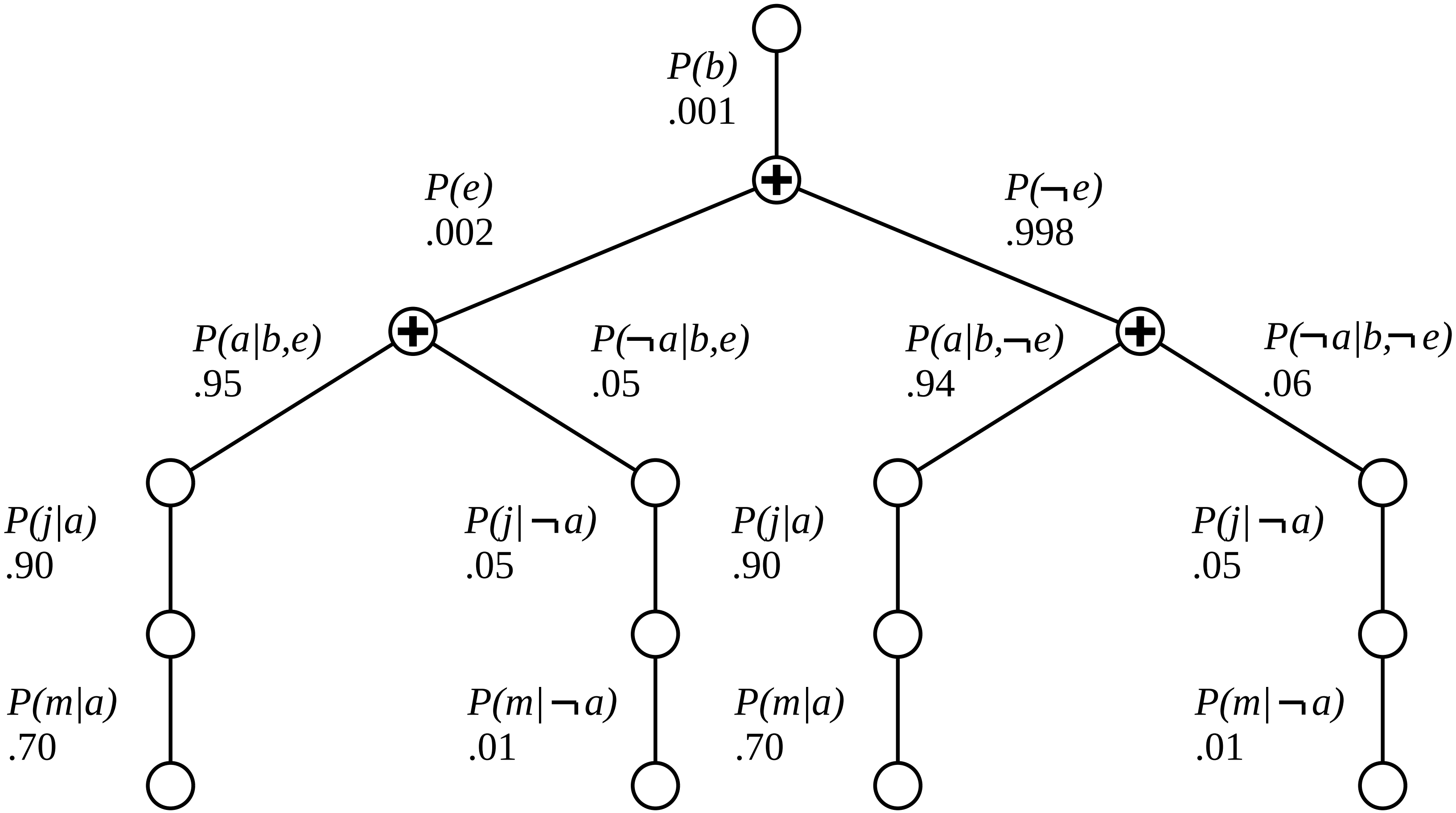


Rewrite full joint entries using product of CPT entries:

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

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

Evaluation tree



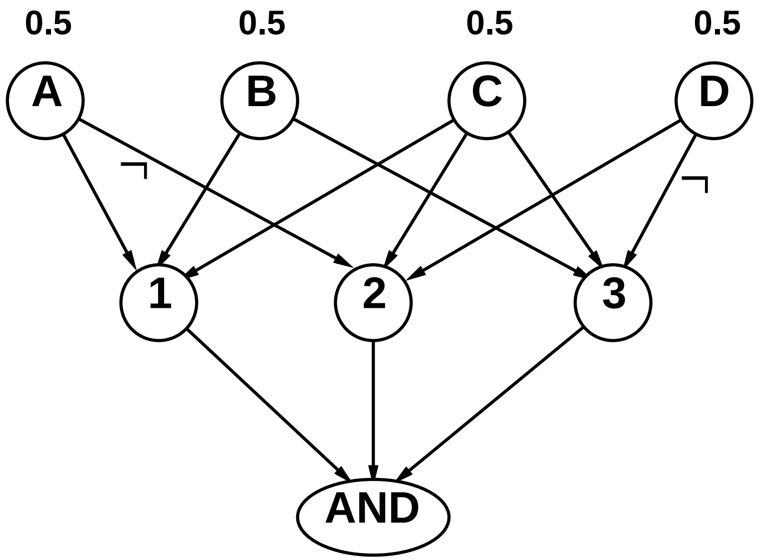
Inference by variable elimination

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

Time and space cost $O(d^k n)$ for singly connected networks (polytrees)

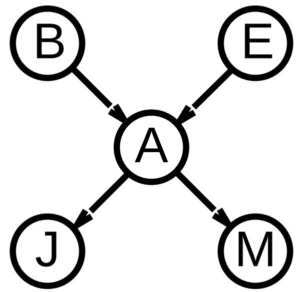
#P-hard (i.e. worse than NP hard) for multiply connected networks (equivalent to counting 3SAT models)

1. A v B v C
2. C v D v ¬A
3. B v C v ¬D



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 M is irrelevant

Inference by stochastic simulation

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

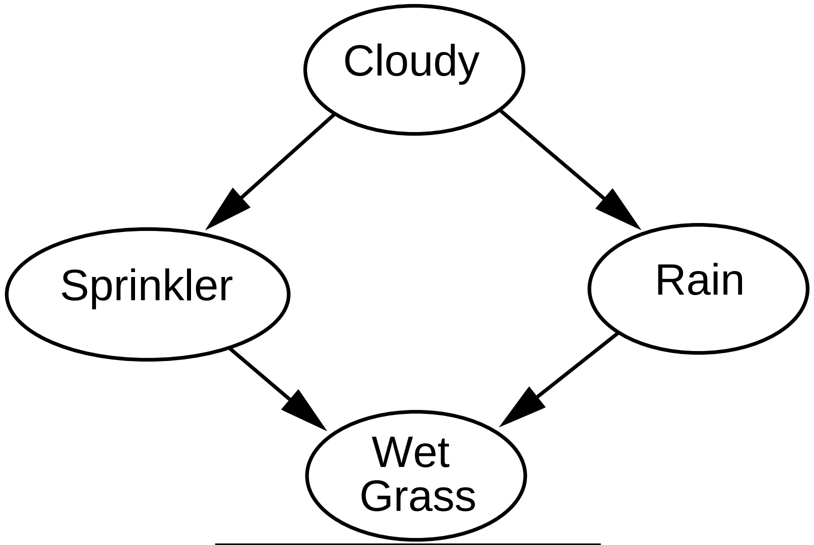
Sampling from an empty network

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

  $\mathbf{x} \leftarrow$ an event with n elements
  for $i = 1$ **to** n **do**
    $x_i \leftarrow$ a random sample from $\mathbf{P}(X_i \mid Parents(X_i))$
  return $\mathbf{x}$

Example

P(C)
.50



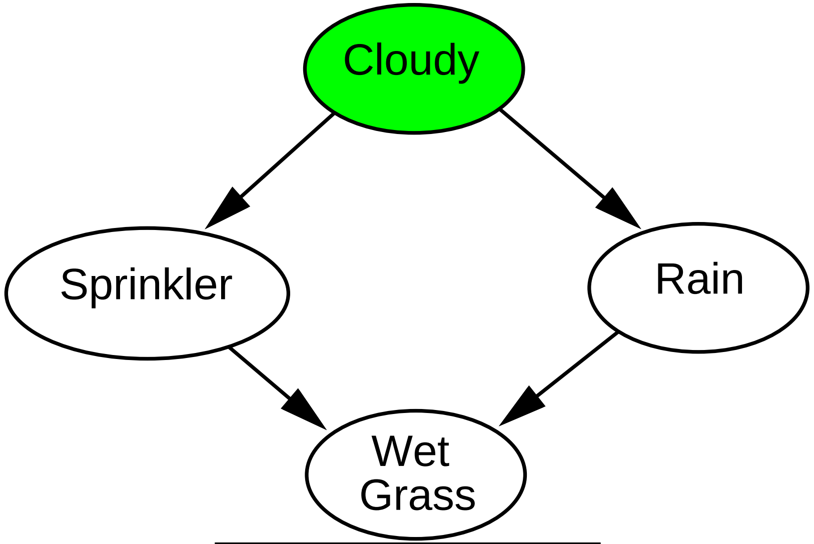
C	P(S|C)
T	.10
F	.50

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

P(C)
.50



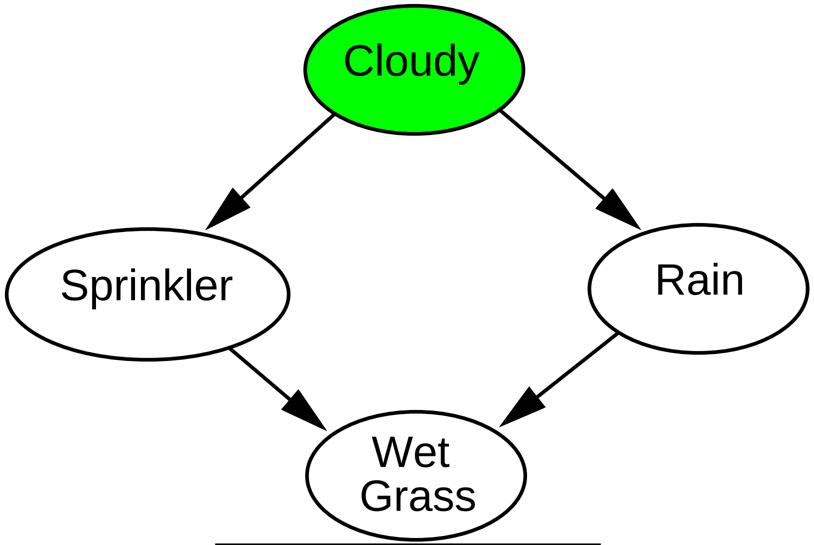
C	P(S|C)
T	.10
F	.50

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

P(C)
.50



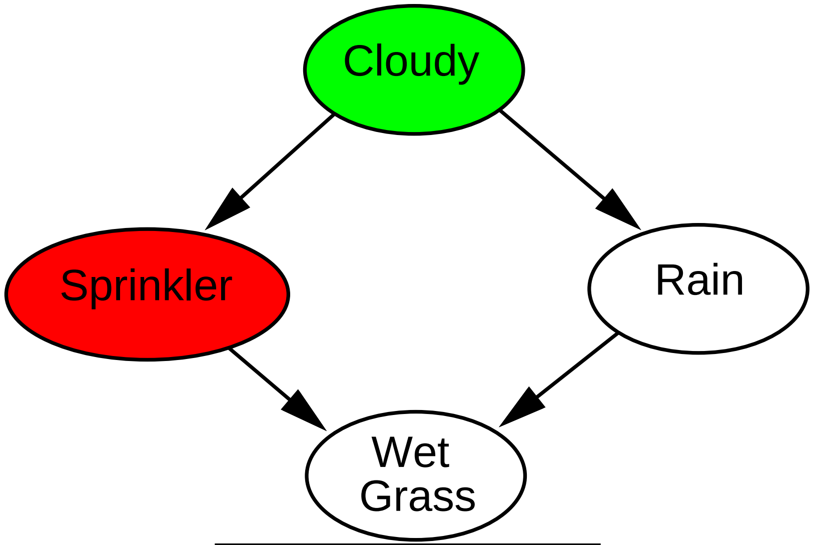
C	P(S|C)
T	.10
F	.50

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

P(C)
.50



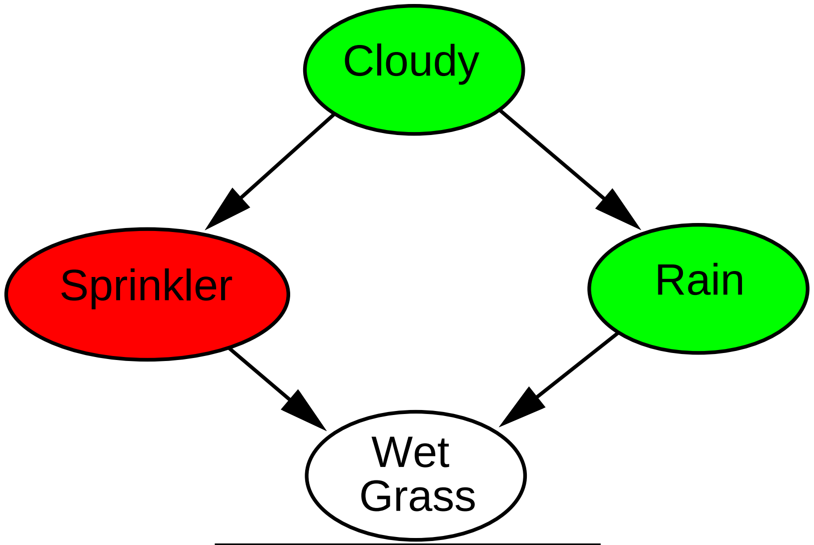
C	P(S|C)
T	.10
F	.50

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

P(C)
.50



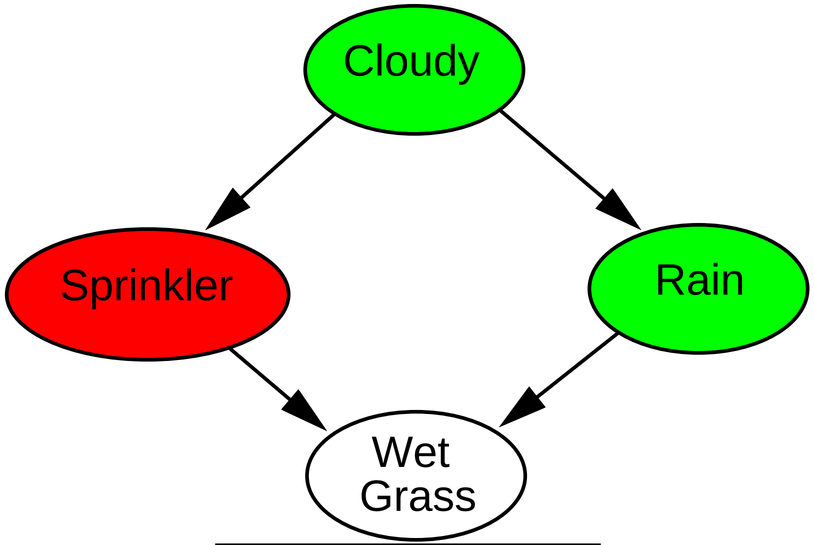
C	P(S|C)
T	.10
F	.50

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

P(C)
.50



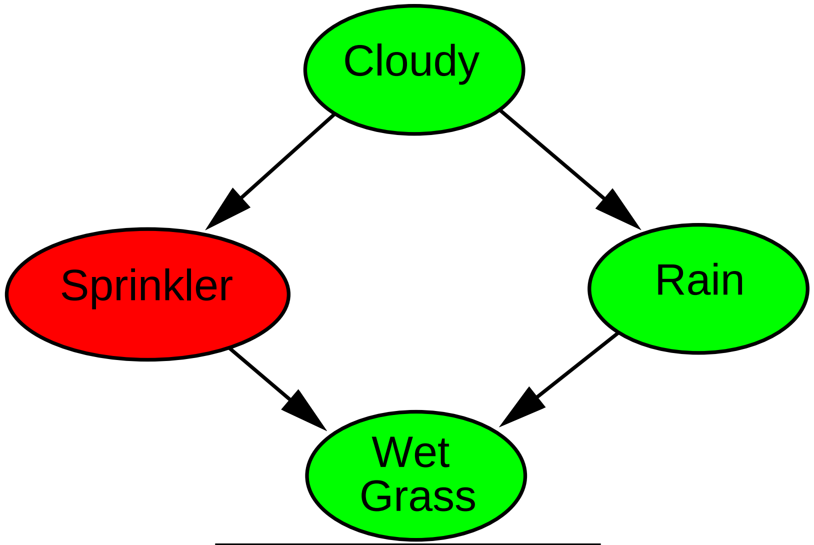
C	P(S|C)
T	.10
F	.50

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

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

Rejection sampling

$\hat{\mathbf{P}}(X|\mathbf{e})$ 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 $Rain=false$.

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

Analysis of rejection sampling

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

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

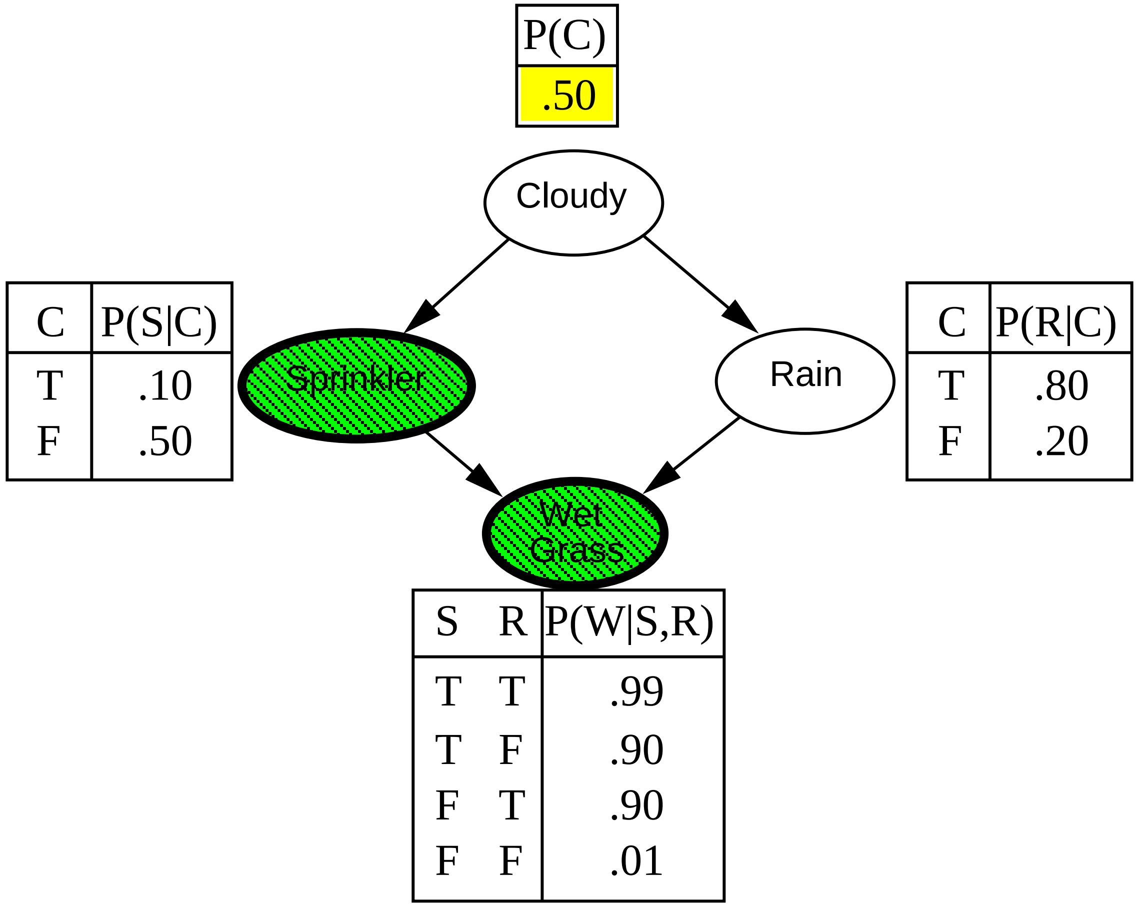
```
function LIKELIHOOD-WEIGHTING(X, e, bn, N) returns an estimate of P(X|e)
   local variables: W, a vector of weighted counts over X, initially zero

   for j = 1 to N do
       x, w ← WEIGHTED-SAMPLE(bn)
       W[x] ← W[x] + w where x is the value of X in x
   return NORMALIZE(W[X])
```

```
function WEIGHTED-SAMPLE(bn, e) returns an event and a weight

   x ← an event with n elements; w ← 1
   for i = 1 to n do
       if X_i has a value x_i in e
           then w ← w × P(X_i = x_i | Parents(X_i))
           else x_i ← a random sample from P(X_i | Parents(X_i))
   return x, w
```

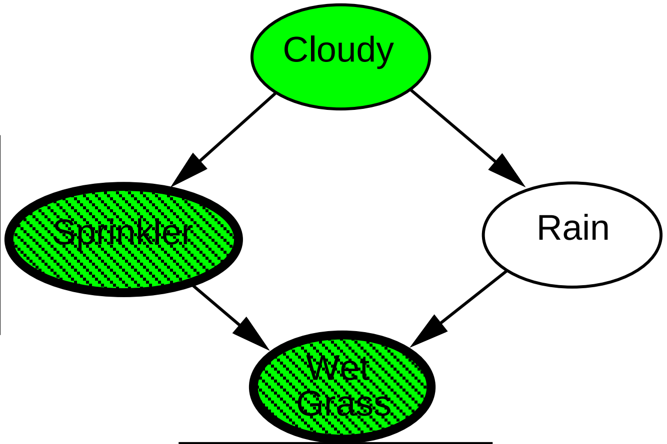
Likelihood weighting example



$w = 1.0$

Likelihood weighting example

P(C)
.50



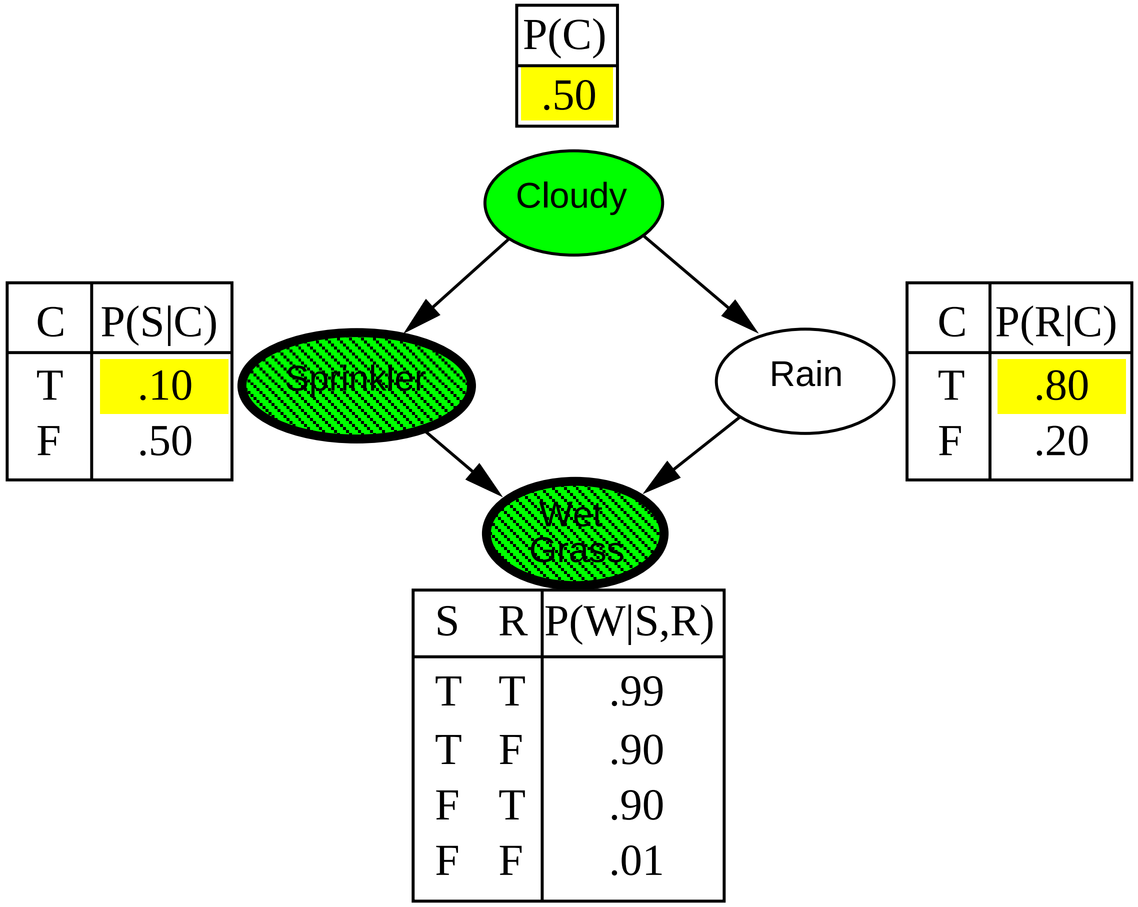
C	P(S|C)
T	.10
F	.50

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

$w = 1.0$

Likelihood weighting example



P(C)
.50

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

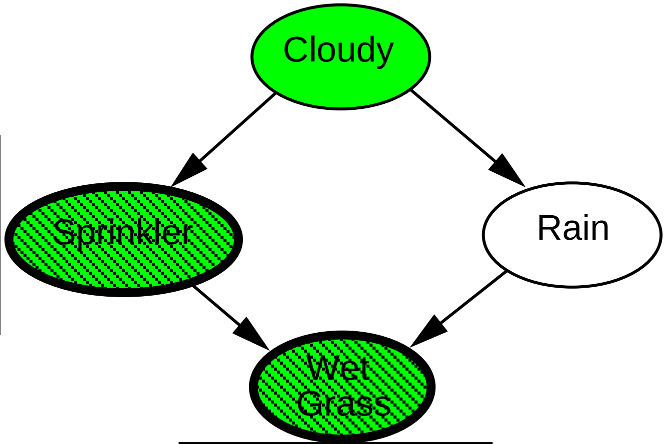
C	P(R|C)
T	.80
F	.20

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

$w = 1.0$

Likelihood weighting example

P(C)
.50



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

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

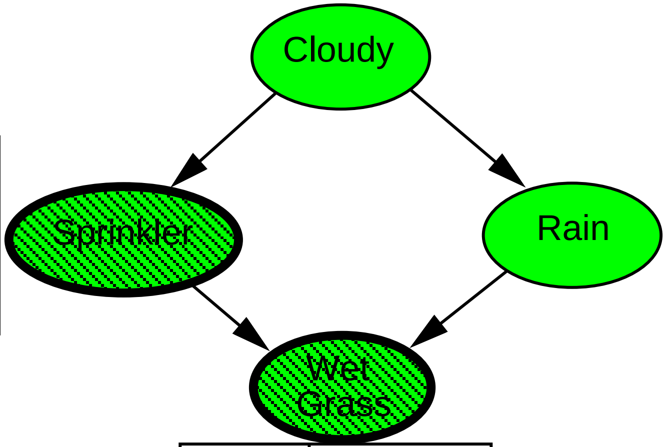
S	R	P(W|S,R)
T	T	.99
T	F	.90
F	T	.90
F	F	.01

$w = 1.0 \times 0.1$

Likelihood weighting example

P(C)
.50

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

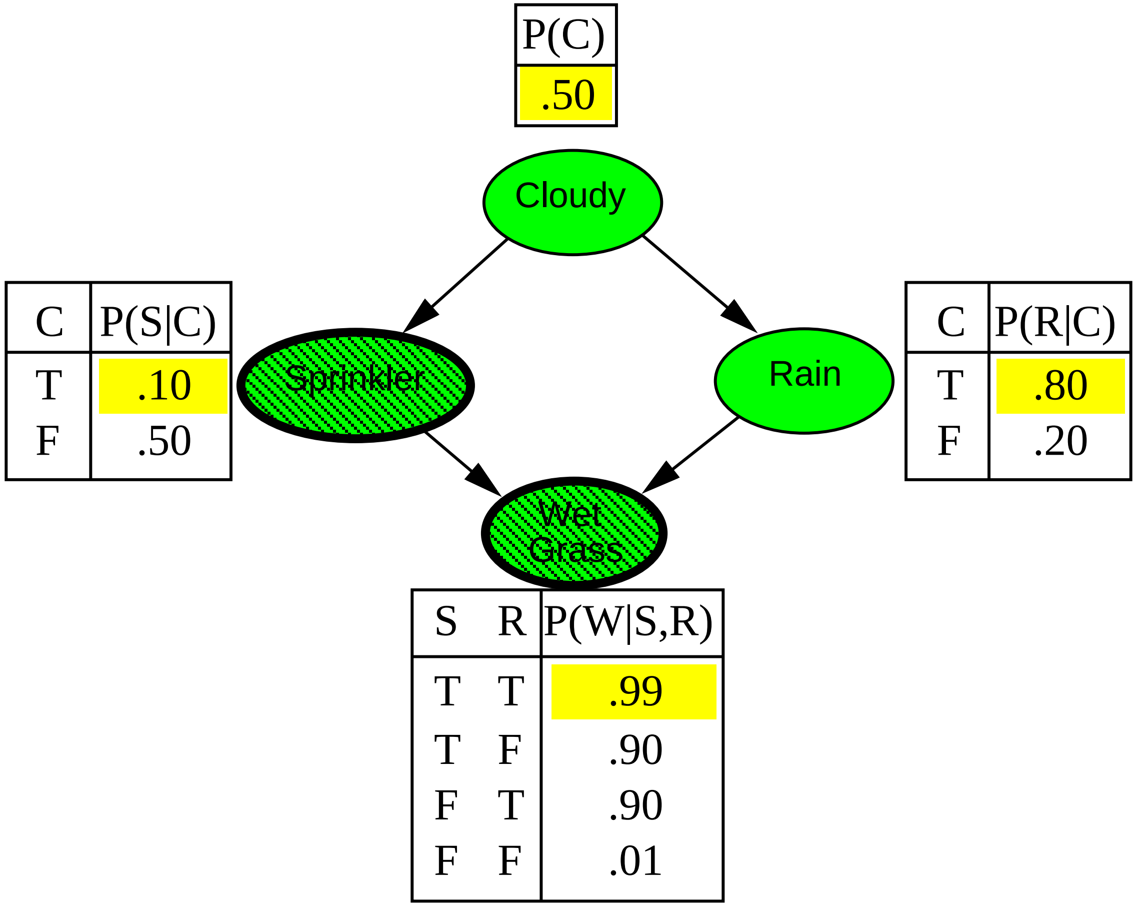


C	P(R|C)
T	.80
F	.20

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

$w = 1.0 \times 0.1$

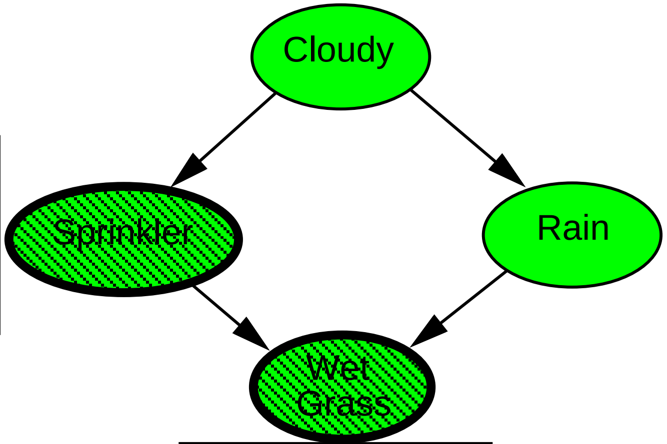
Likelihood weighting example



$w = 1.0 \times 0.1$

Likelihood weighting example

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 analysis

Sampling probability for WeightedSample is

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

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

Approximate inference using MCMC

"State" of network = current assignment to all variables.

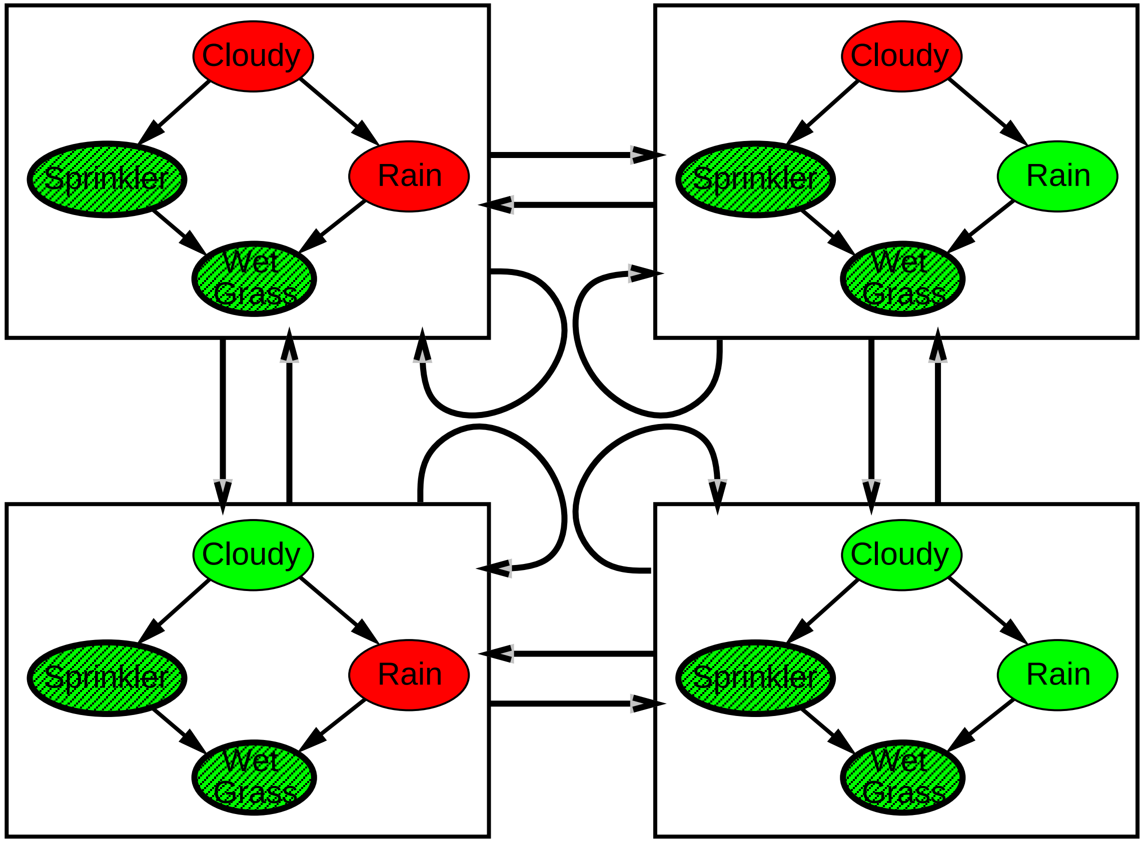
Generate next state by sampling one variable given Markov blanket
Sample each variable in turn, keeping evidence fixed

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 $\mathbf{Z}$
    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 in $\mathbf{x}$
            $\mathbf{N}[x] \leftarrow \mathbf{N}[x] + 1$ where x is the value of X in $\mathbf{x}$
    return Normalize($\mathbf{N}[X]$)

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.

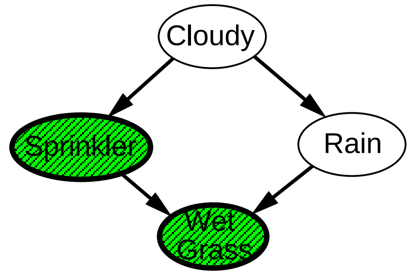
E.g., visit 100 states
31 have $Rain = true$, 69 have $Rain = false$

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

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:

$$P(x_i'|MB(X_i)) = \alpha \times P(x_i'|Parents(X_i)) \prod_{Z_j \in Children(X_i)} P(z_j|Parents(Z_j))$$

Main computational problems:
1) Difficult to tell if convergence has been achieved
2) Can be slow if Markov blanket is large

Summary

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

Approximate inference by LW, MCMC:
– LW does poorly when there is lots of 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