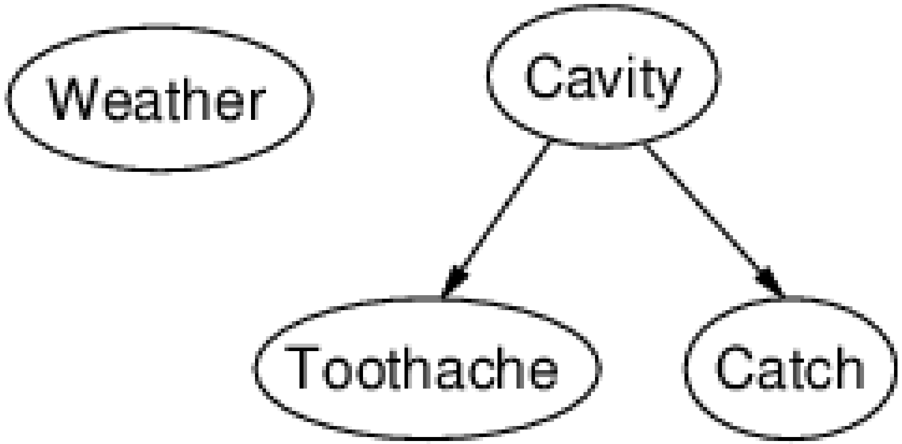
Bayesian Networks

Philipp Koehn

2 April 2024

Bayesian Network Example

- Topology of network encodes conditional independence assertions:



- $Weather$ is independent of the other variables

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

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:
    $\mathbf{P}(X_i|Parents(X_i))$

- 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

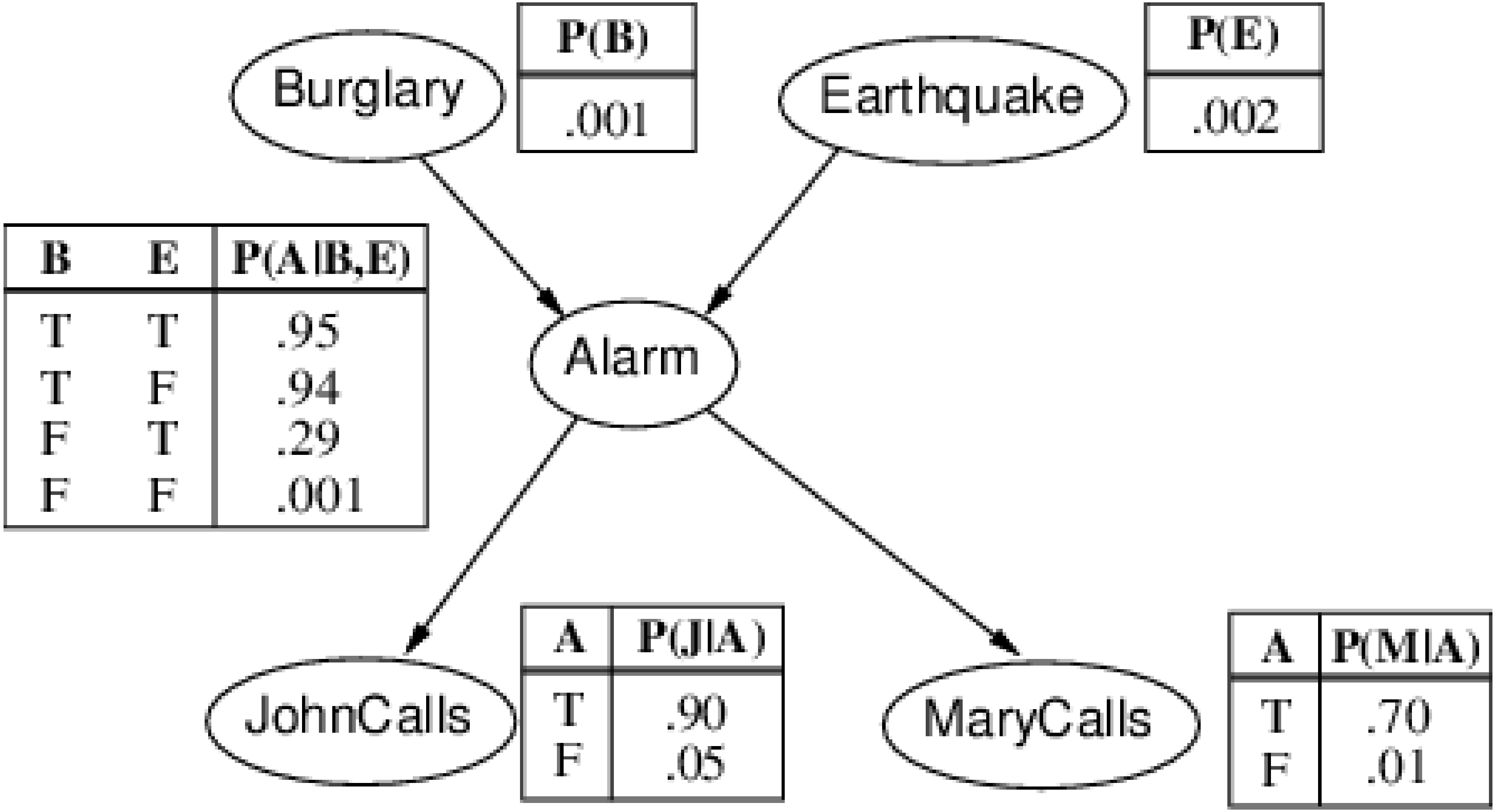
- *I'm at work, neighbor John calls to say my alarm is ringing, but neighbor Mary doesn't call. Sometimes it's set off by minor earthquakes.*

  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



P(B)
.001

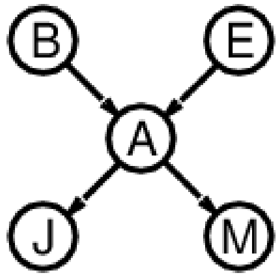
P(E)
.002

B	E	P(A|B,E)
T	T	.95
T	F	.94
F	T	.29
F	F	.001

A	P(J|A)
T	.90
F	.05

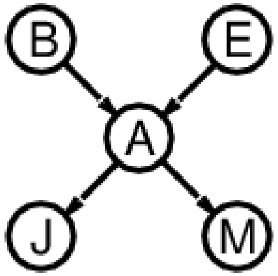
A	P(M|A)
T	.70
F	.01

Compactness



- A conditional probability table 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. $2^5 - 1 = 31$)

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

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

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, ..., X_n
2.  For i = 1 to n
       add X_i to the network
       select parents from X_1, ..., X_{i-1} such that
          P(X_i|Parents(X_i)) = P(X_i|X_1, ..., X_{i-1})
```

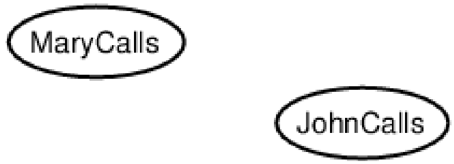
- This choice of parents guarantees the global semantics:

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

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

Example

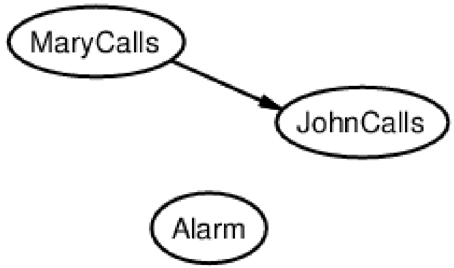
- Suppose we choose the ordering M, J, A, B, E



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

Example

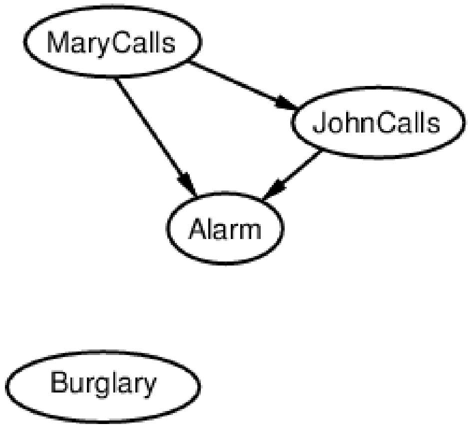
- Suppose we choose the ordering M, J, A, B, E



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

Example

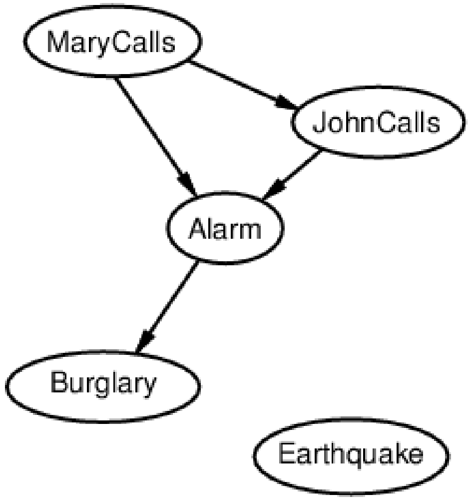
- Suppose we choose the ordering M, J, A, B, E



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

Example

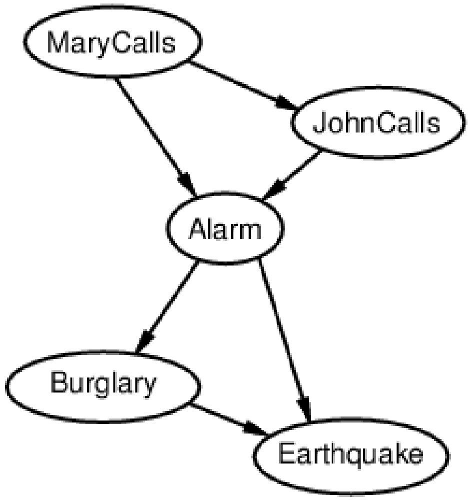
- Suppose we choose the ordering M, J, A, B, E



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

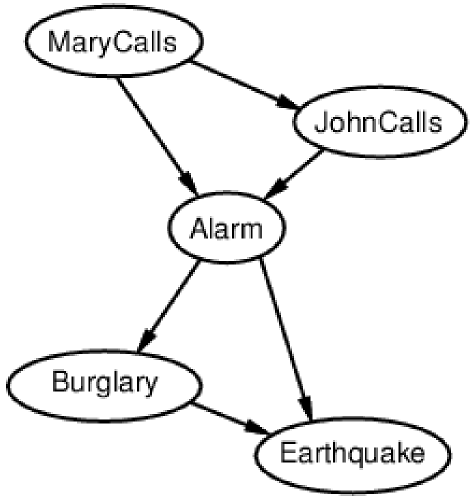
Example

- 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



- Deciding conditional independence is hard in noncausal directions

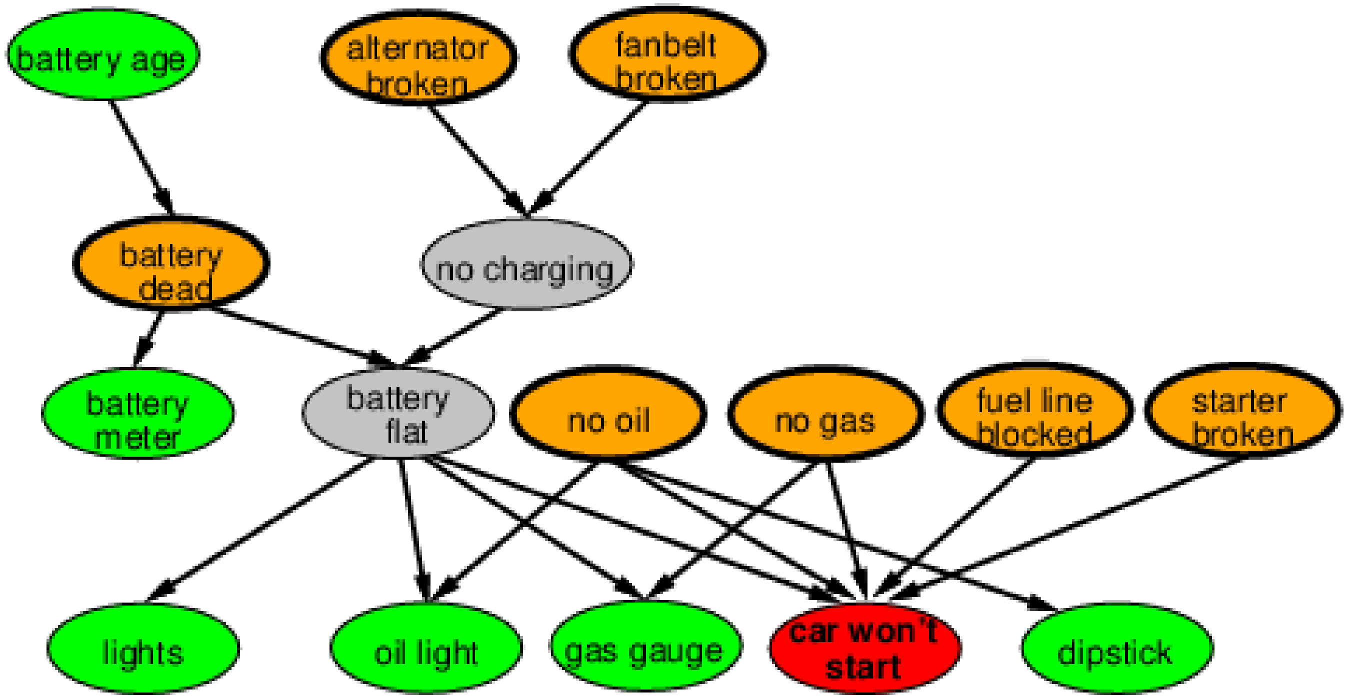
  (Causal models and conditional independence seem hardwired for humans!)

- Assessing conditional probabilities is hard in noncausal directions

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

Example: Car Diagnosis

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



inference

Inference Tasks

- 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?

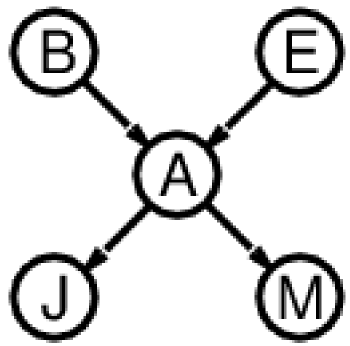
- Sensitivity analysis: which probability values are most critical?

- Explanation: why do I need a new starter motor?

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



- 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

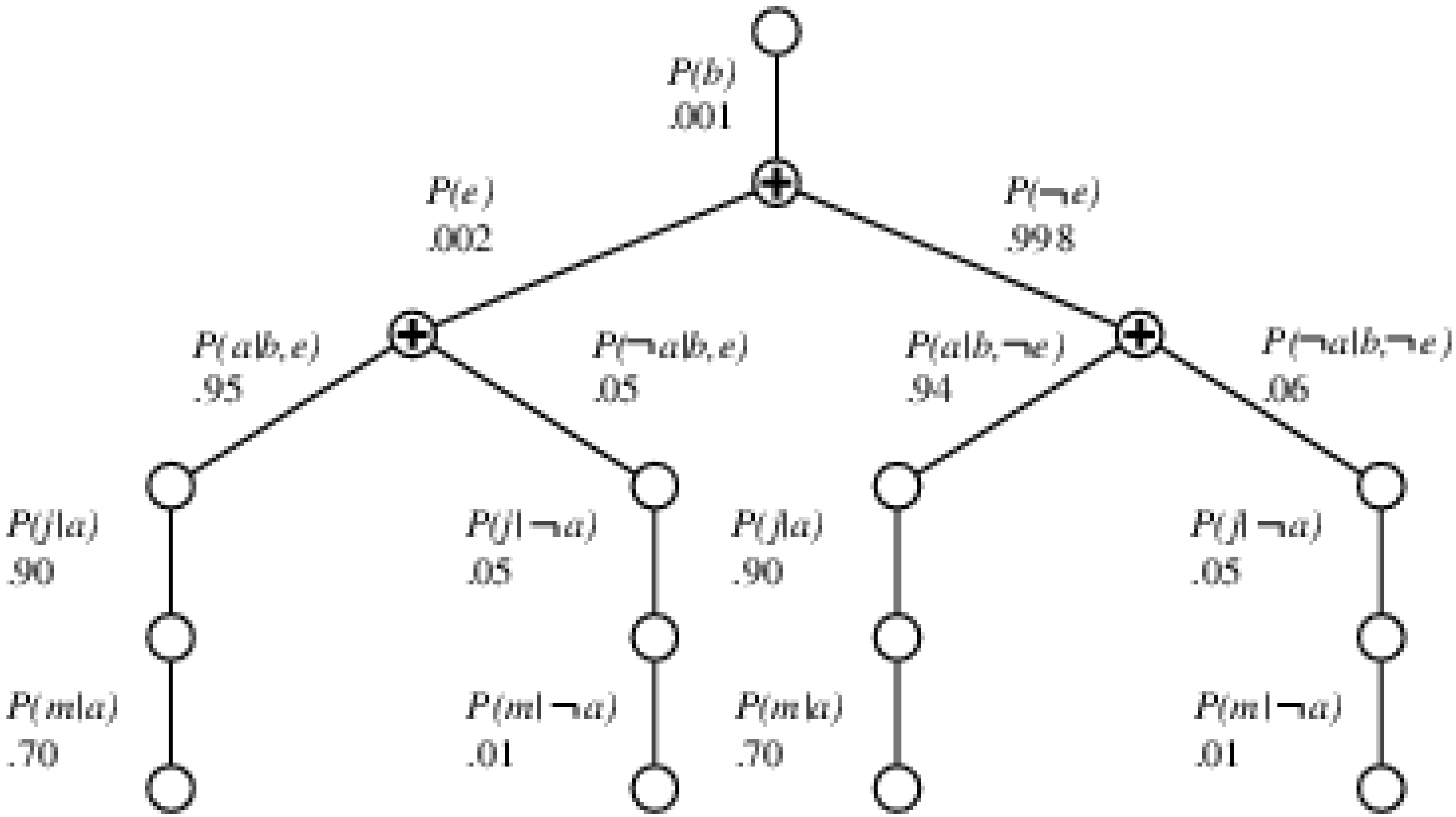
function ENUMERATION-ASK(*X*, **e**, *bn*) **returns** a distribution over *X*
  inputs: *X*, the query variable
    e, observed values for variables **E**
    bn, a Bayesian network with variables $\{X\}$ ∪ **E** ∪ **Y**

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

---

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

Evaluation Tree



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

Inference by Variable Elimination

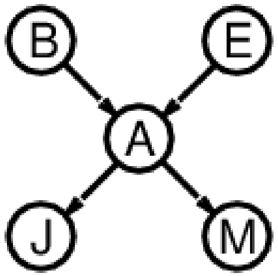
- Variable elimination: carry out summations right-to-left, storing intermediate results (factors) to avoid recomputation

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

Variable Elimination 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 P(X_1, ..., X_n)
   factors ← [ ]; vars ← REVERSE(VARS[bn])
   for each var in vars do
      factors ← [MAKE-FACTOR(var, e)|factors]
      if var is a hidden variable then factors ← SUM-OUT(var, factors)
   return NORMALIZE(POINTWISE-PRODUCT(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 1: Y is irrelevant unless $Y \in Ancestors(\{X\} \cup \mathbf{E})$

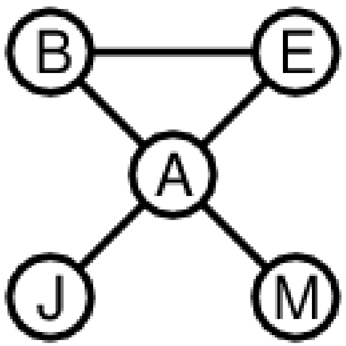
- Here
  - $X = JohnCalls$, $\mathbf{E} = \{Burglary\}$
  - $Ancestors(\{X\} \cup \mathbf{E}) = \{Alarm, Earthquake\}$

$\Rightarrow$ $MaryCalls$ is irrelevant

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

Irrelevant Variables

- Definition: moral graph of Bayes net: marry all parents and drop arrows
- Definition: **A** is m-separated from **B** by **C** iff separated by **C** in the moral graph
- Theorem 2: Y is irrelevant if m-separated from X by **E**

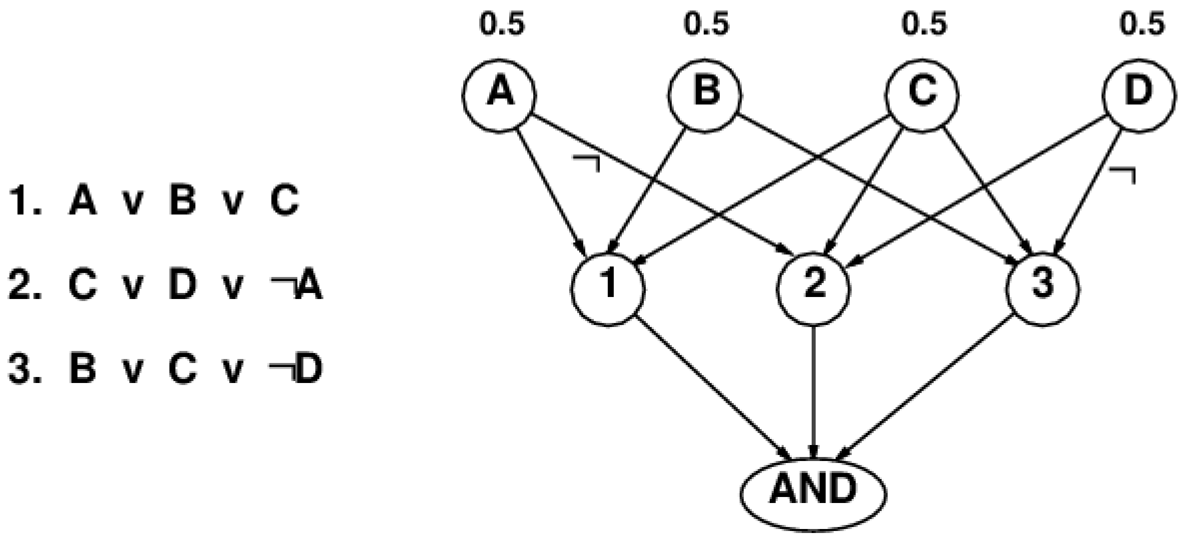


- For $P(JohnCalls|Alarm\!=\!true)$, both $Burglary$ and $Earthquake$ are irrelevant
  (**A** = $JohnCalls$, **B** = $Burglary$ and $Earthquake$, **C** = $Alarm$)

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 $\Longrightarrow$ NP-hard



approximate inference

Inference by Stochastic Simulation

- Basic idea
  - Draw N samples from a sampling distribution S
  - Compute an approximate posterior probability $\hat{P}$
  - Show this converges to the true probability P

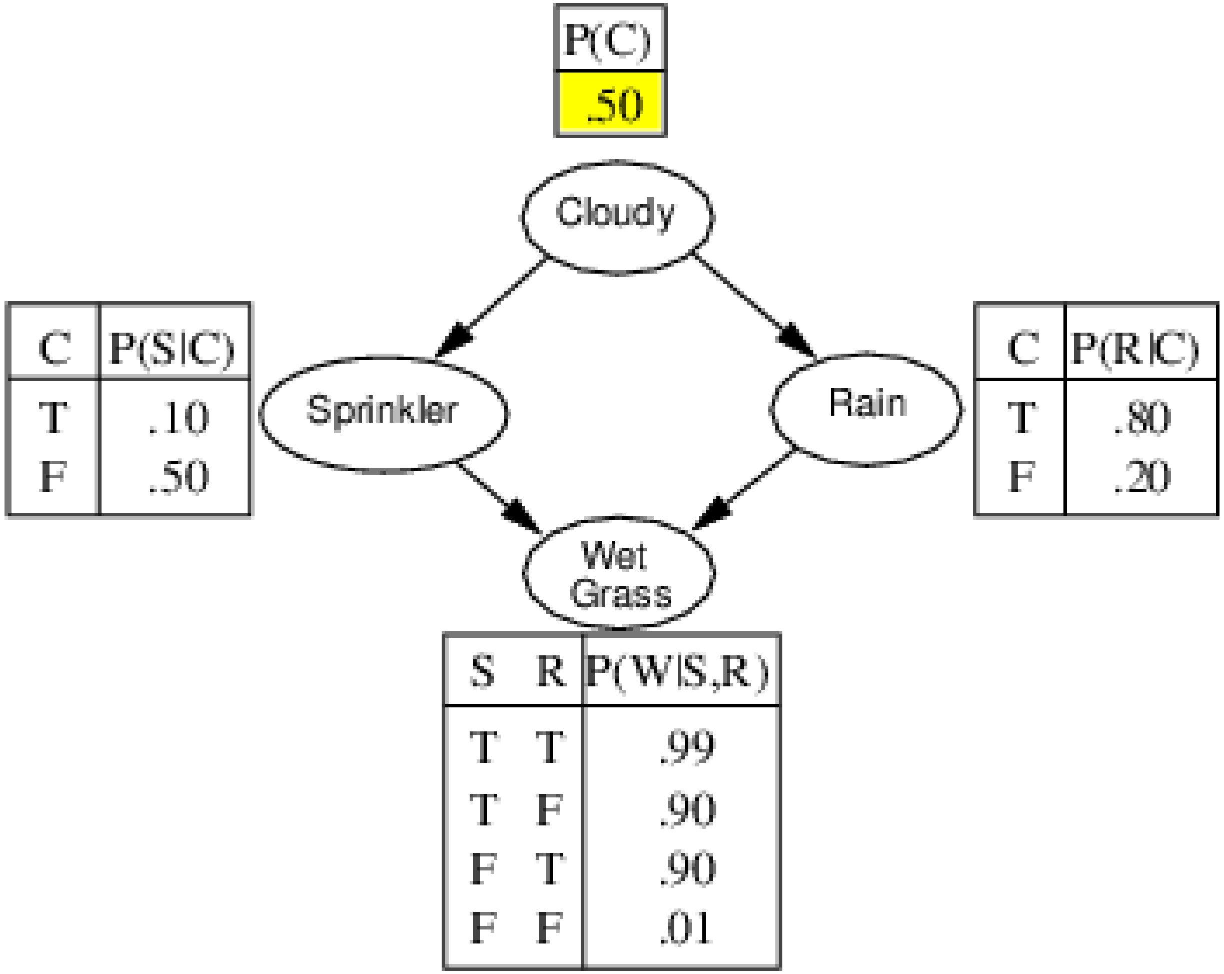
- Outline
  - Sampling from an empty network
  - Rejection sampling: reject samples disagreeing with evidence
  - Likelihood weighting: use evidence to weight samples
  - Markov chain Monte Carlo (MCMC): sample from a stochastic process whose stationary distribution is the true posterior

Sampling from an Empty Network

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

  x ← an event with n elements
  for $i = 1$ **to** n **do**
    x_i ← a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
      given the values of $Parents(X_i)$ in **x**
  return x

Example



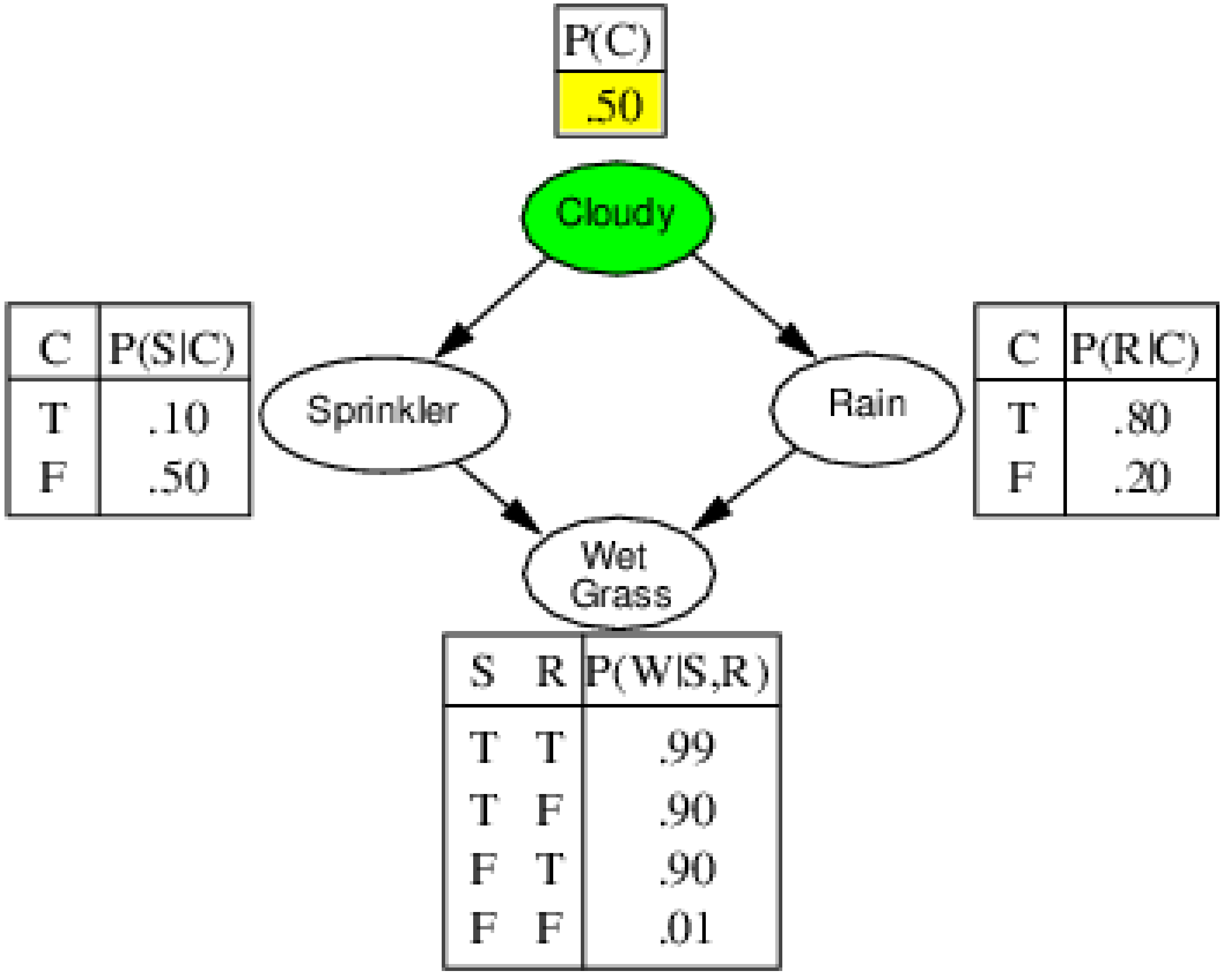
P(C)
.50

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

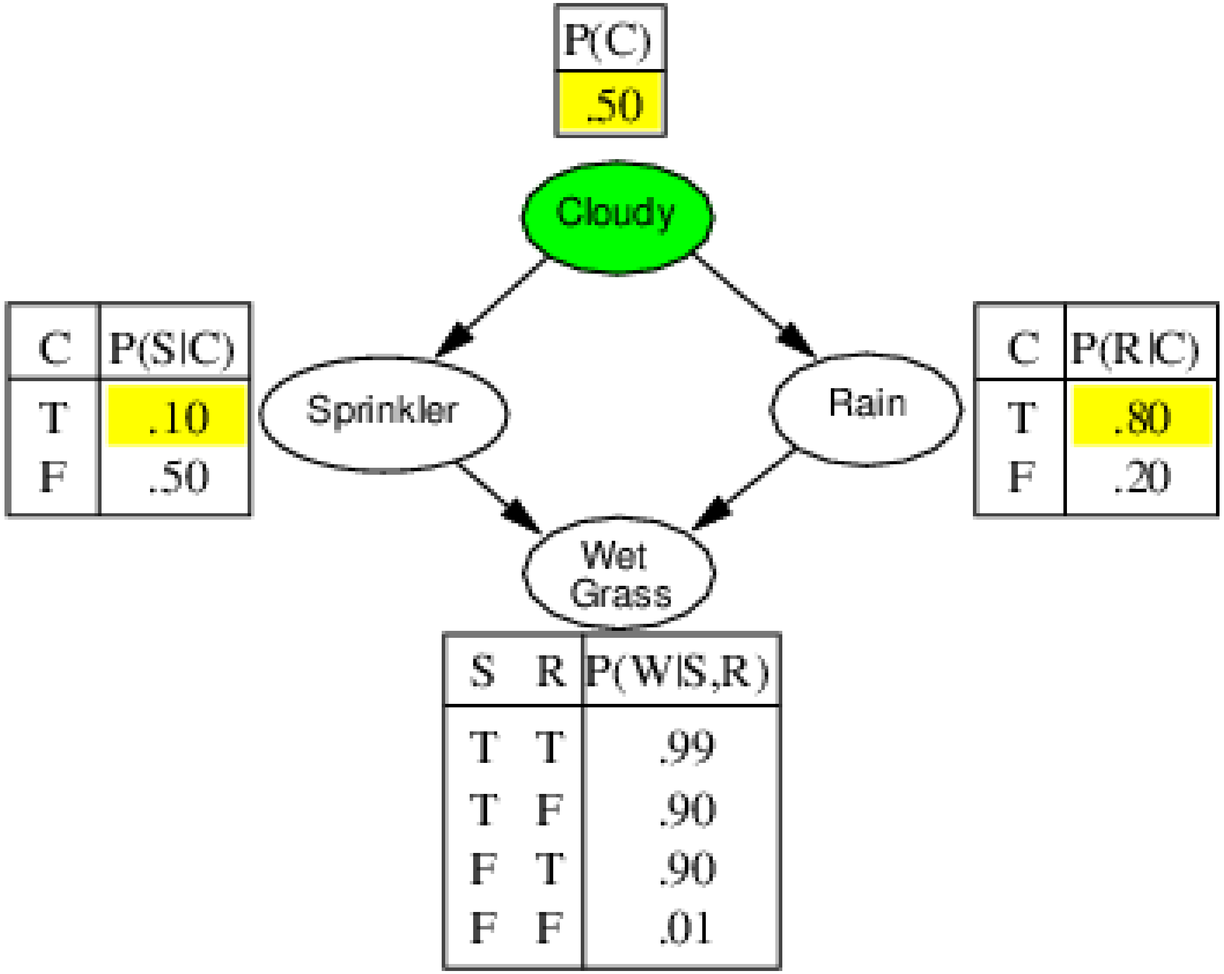
C	P(R|C)
T	.80
F	.20

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

Example


P(C)
.50
Cloudy
C P(S|C)
T .10
F .50
Sprinkler
Rain
C P(R|C)
T .80
F .20
Wet Grass
S R P(W|S,R)
T T .99
T F .90
F T .90
F F .01

Example



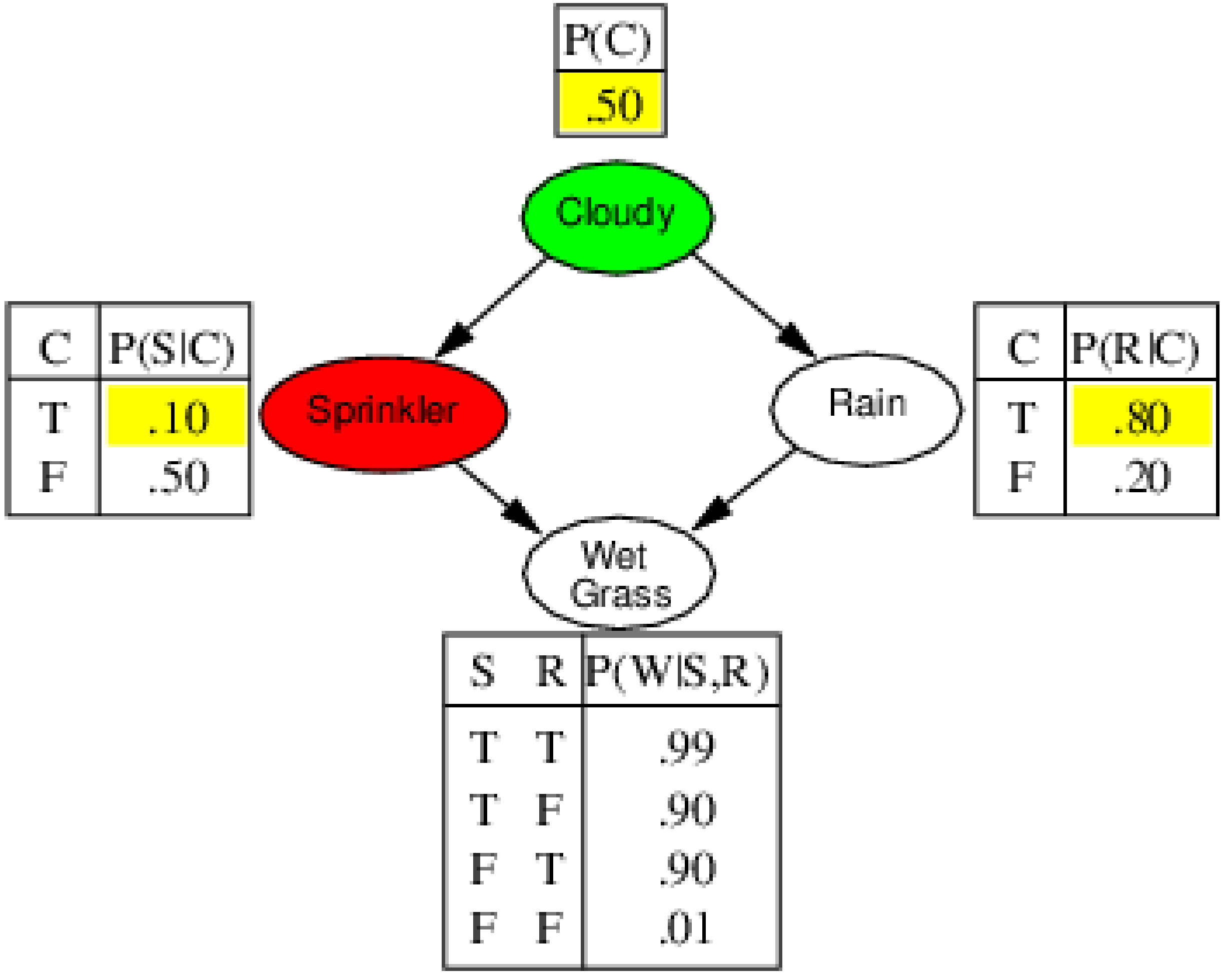
P(C)
.50

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

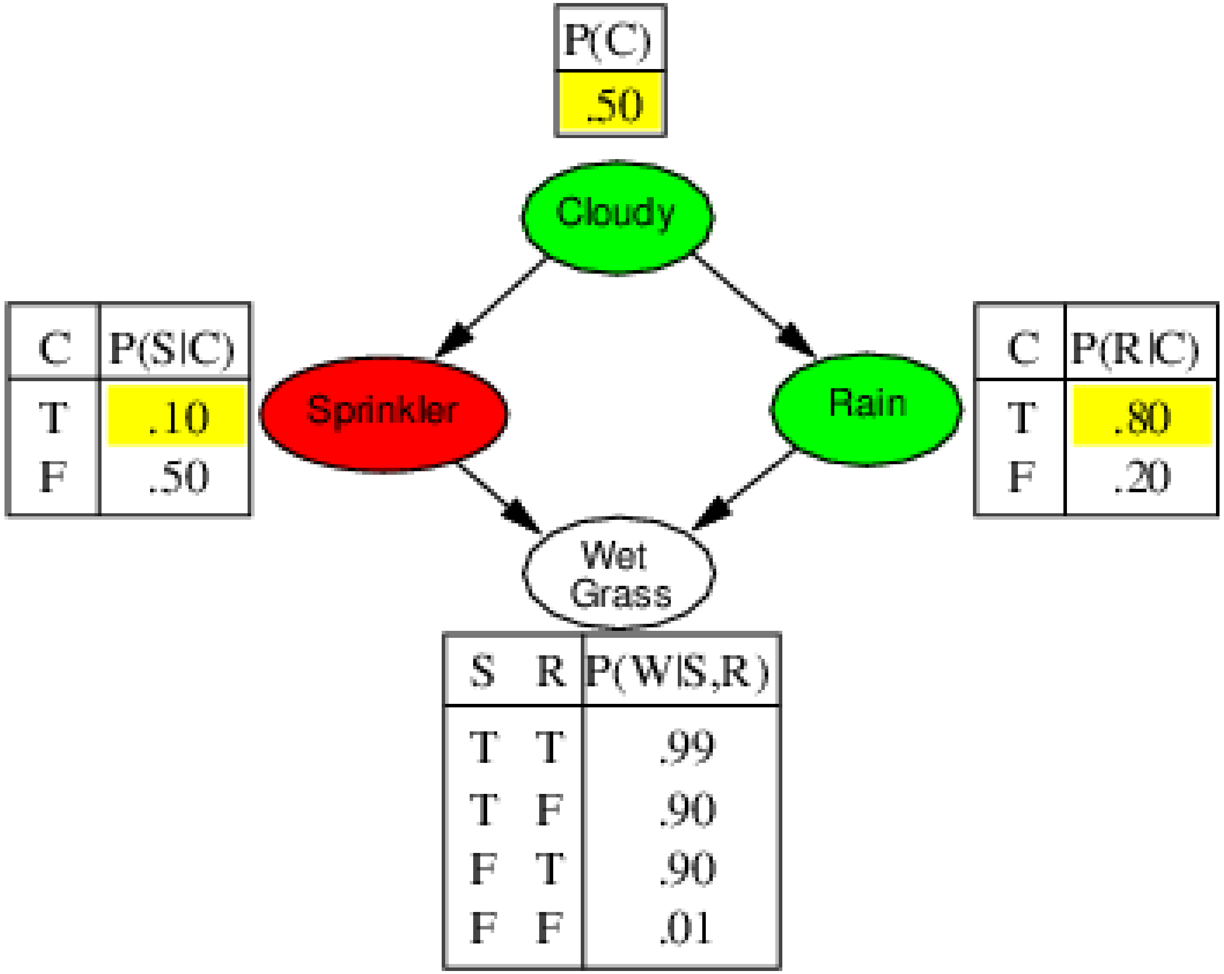
C	P(R|C)
T	.80
F	.20

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

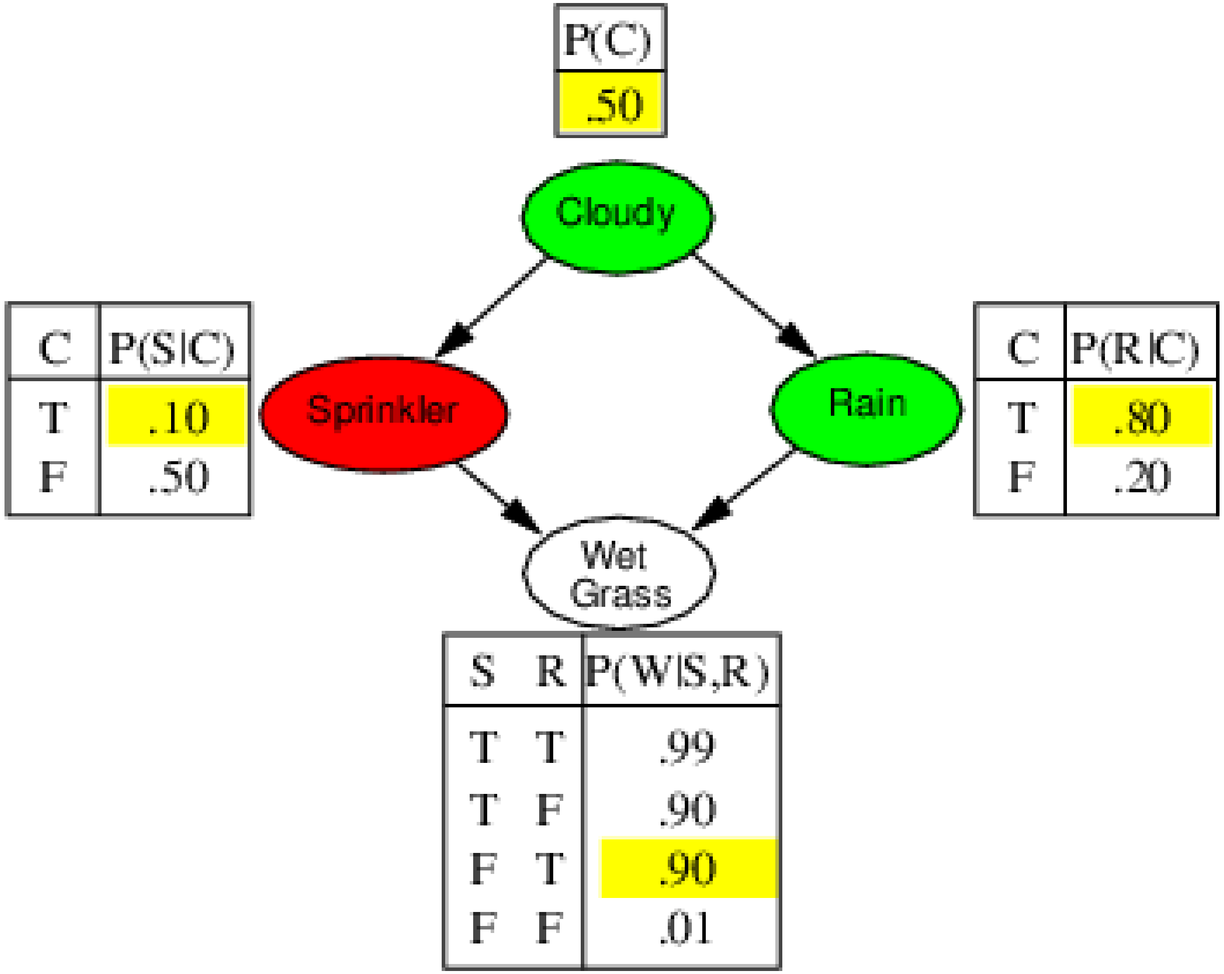
Example


P(C)
.50
Cloudy
C P(S|C)
T .10
F .50
Sprinkler
Rain
C P(R|C)
T .80
F .20
Wet Grass
S R P(W|S,R)
T T .99
T F .90
F T .90
F F .01

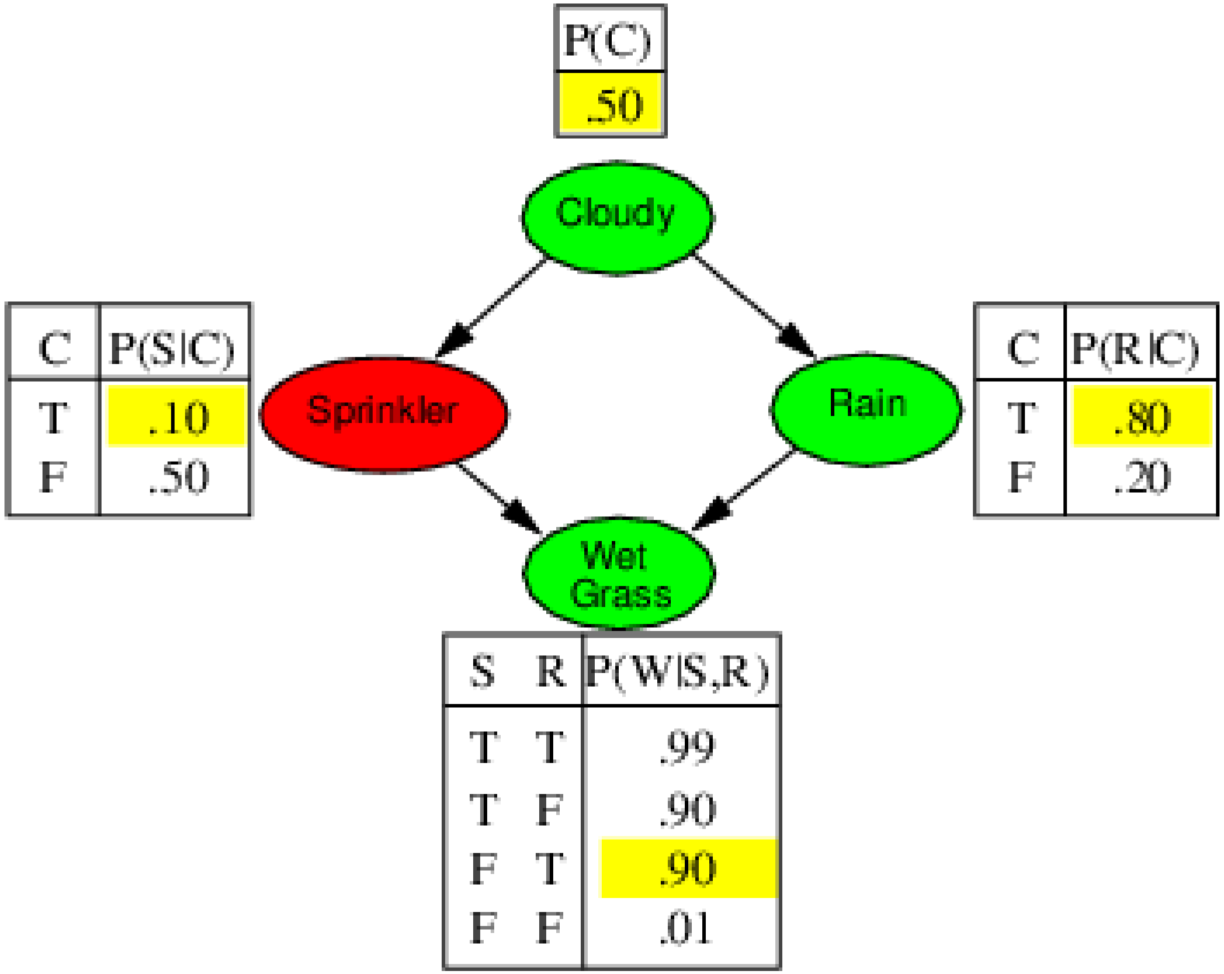
Example


P(C)
.50
Cloudy
C P(S|C)
T .10
F .50
Sprinkler
Rain
C P(R|C)
T .80
F .20
Wet Grass
S R P(W|S,R)
T T .99
T F .90
F T .90
F F .01

Example


P(C)
.50
Cloudy
C P(S|C)
T .10
F .50
Sprinkler
Rain
C P(R|C)
T .80
F .20
Wet Grass
S R P(W|S,R)
T T .99
T F .90
F T .90
F F .01

Example



P(C)
.50
Cloudy
C P(S|C)
T .10
F .50
Sprinkler
Rain
C P(R|C)
T .80
F .20
Wet Grass
S R P(W|S,R)
T T .99
T F .90
F T .90
F F .01

Sampling from an Empty Network

- Probability that PRIORSAMPLE generates a particular event
  $S_{PS}(x_1 \ldots x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i)) = P(x_1 \ldots x_n)$
  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 $N_{PS}(x_1 \ldots x_n)$ be the number of samples generated for event $x_1, \ldots, x_n$

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

- That is, estimates derived from PRIORSAMPLE are consistent

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

Rejection Sampling

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

function REJECTION-SAMPLING(X, **e**, *bn*, N) **returns** an estimate of $P(X|\mathbf{e})$
  local variables: **N**, a vector of counts over X, initially zero

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

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

- 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!

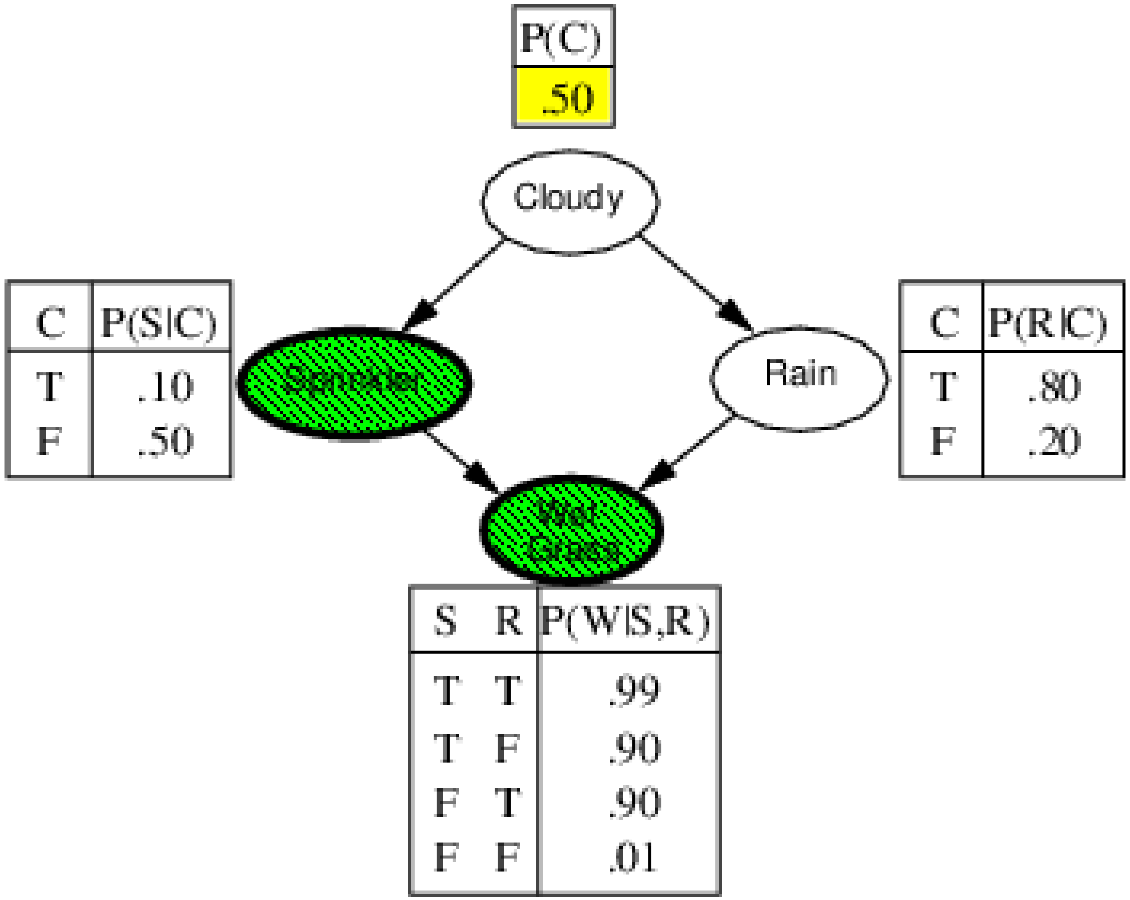
Likelihood Weighting

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

```
function LIKELIHOOD-WEIGHTING(X, 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])
```

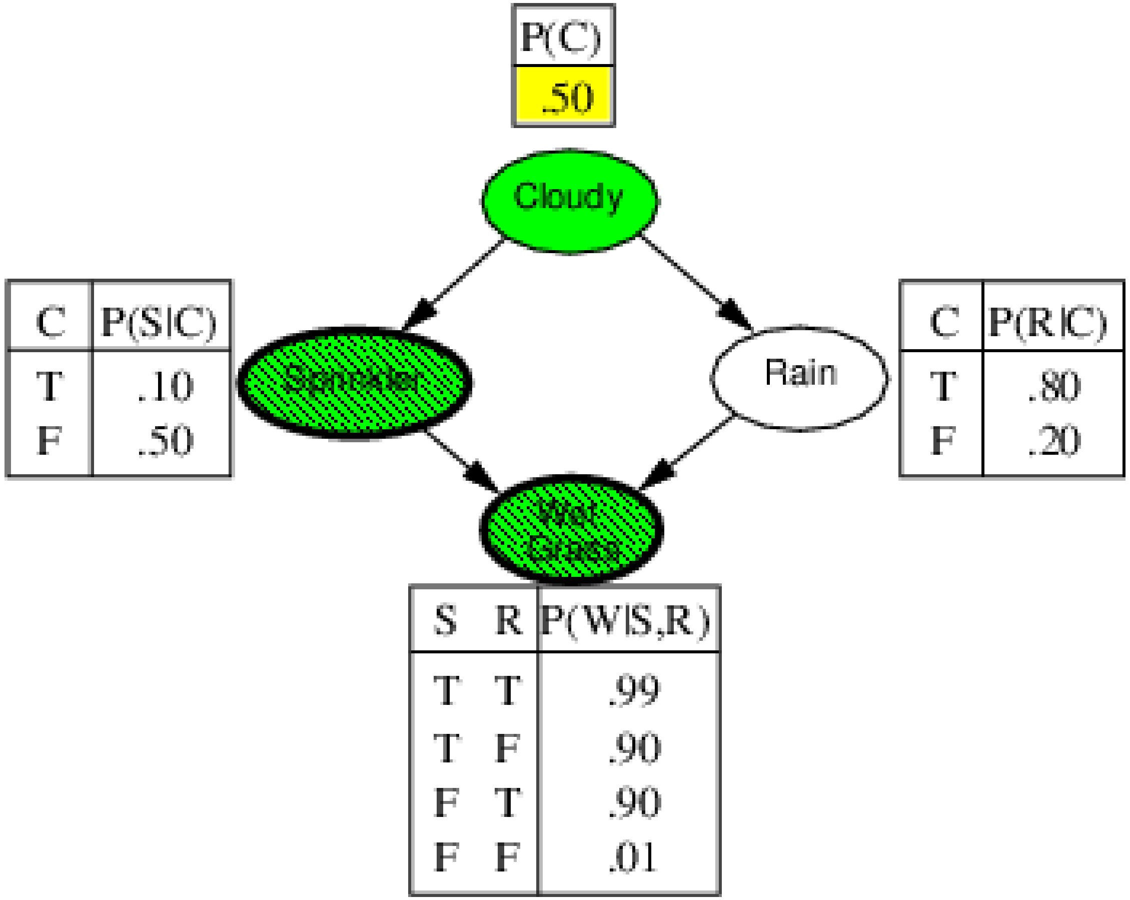
```
function WEIGHTED-SAMPLE(bn, e) returns an event and a weight

   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



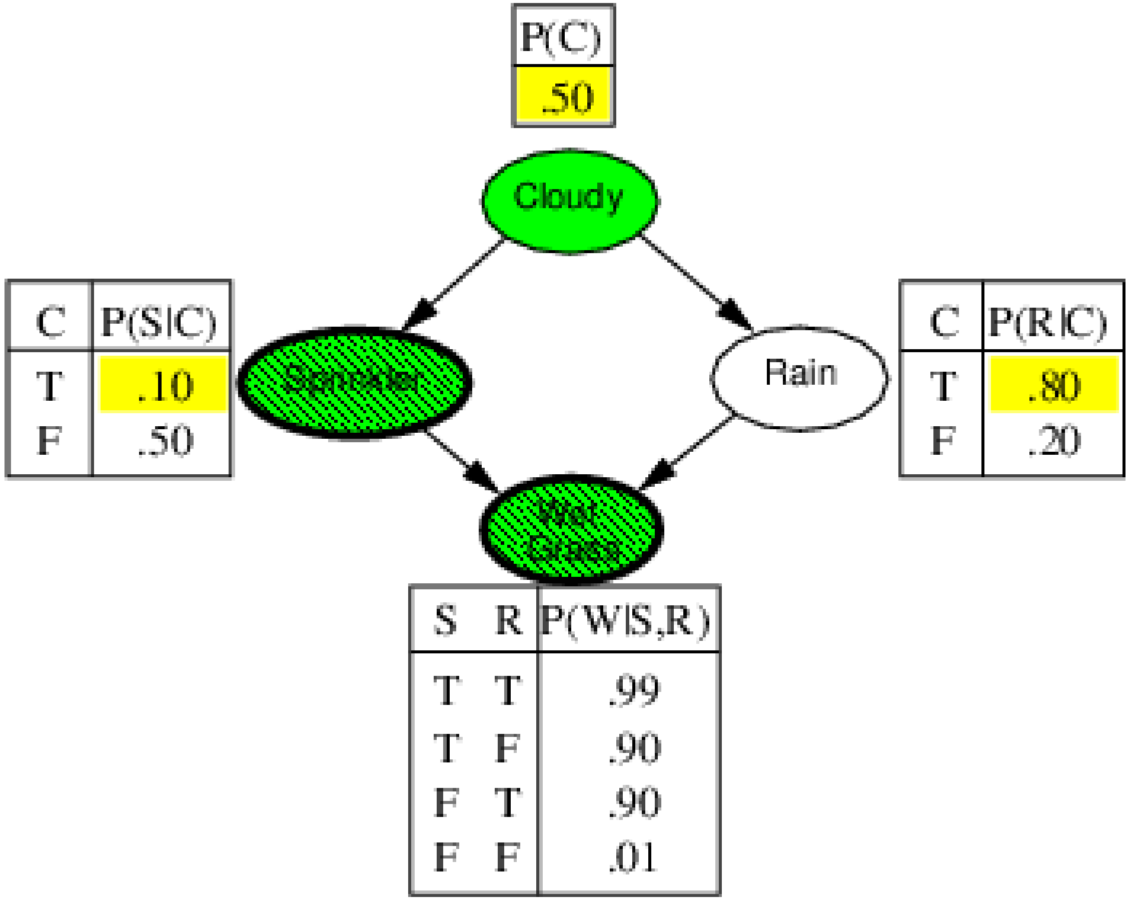
$$w = 1.0$$

Likelihood Weighting Example



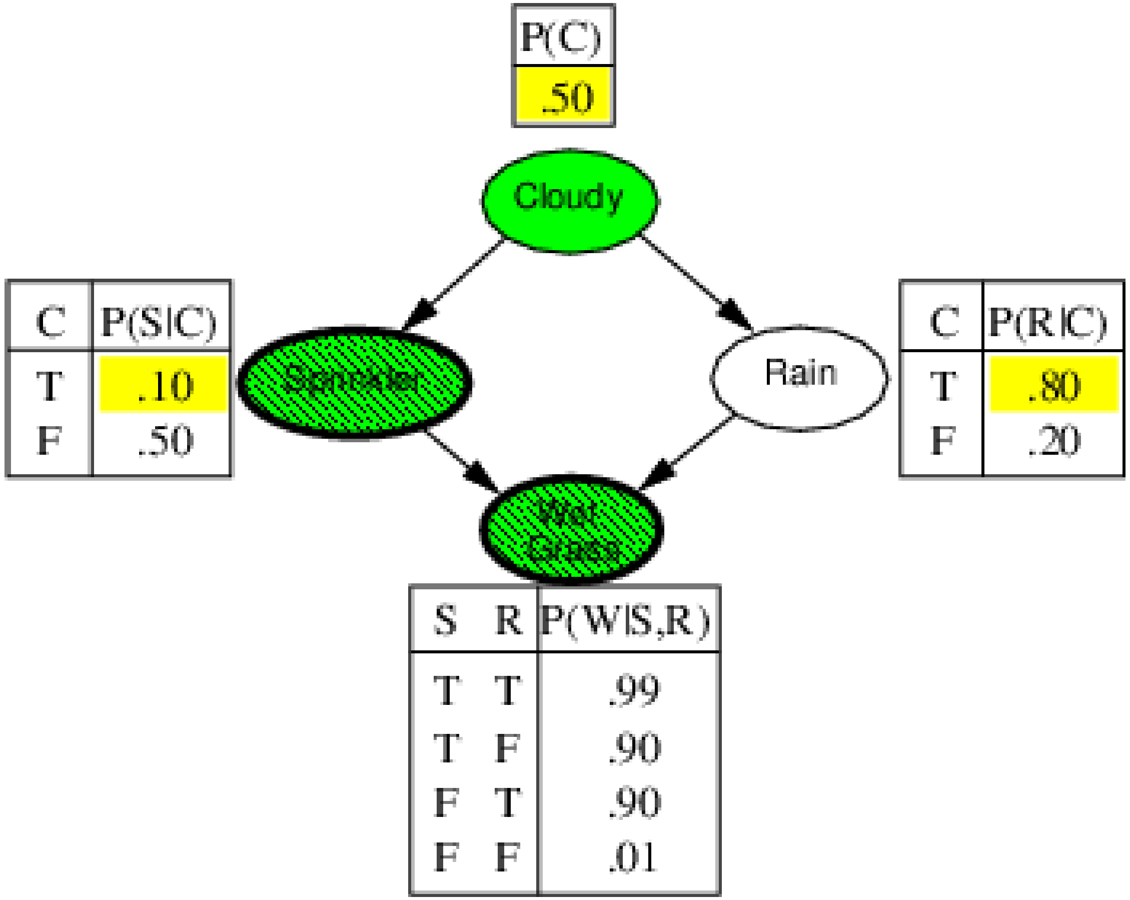
$$w = 1.0$$

Likelihood Weighting Example



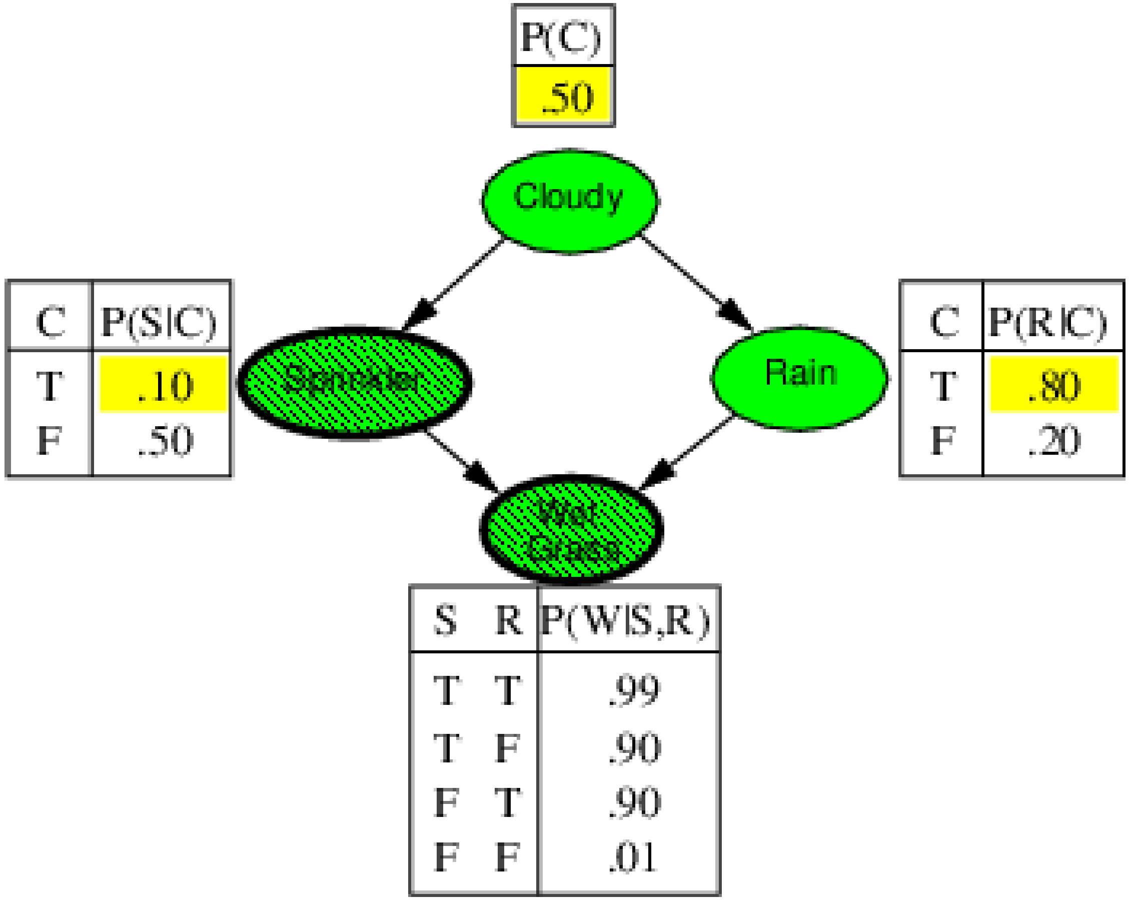
$w = 1.0$

Likelihood Weighting Example



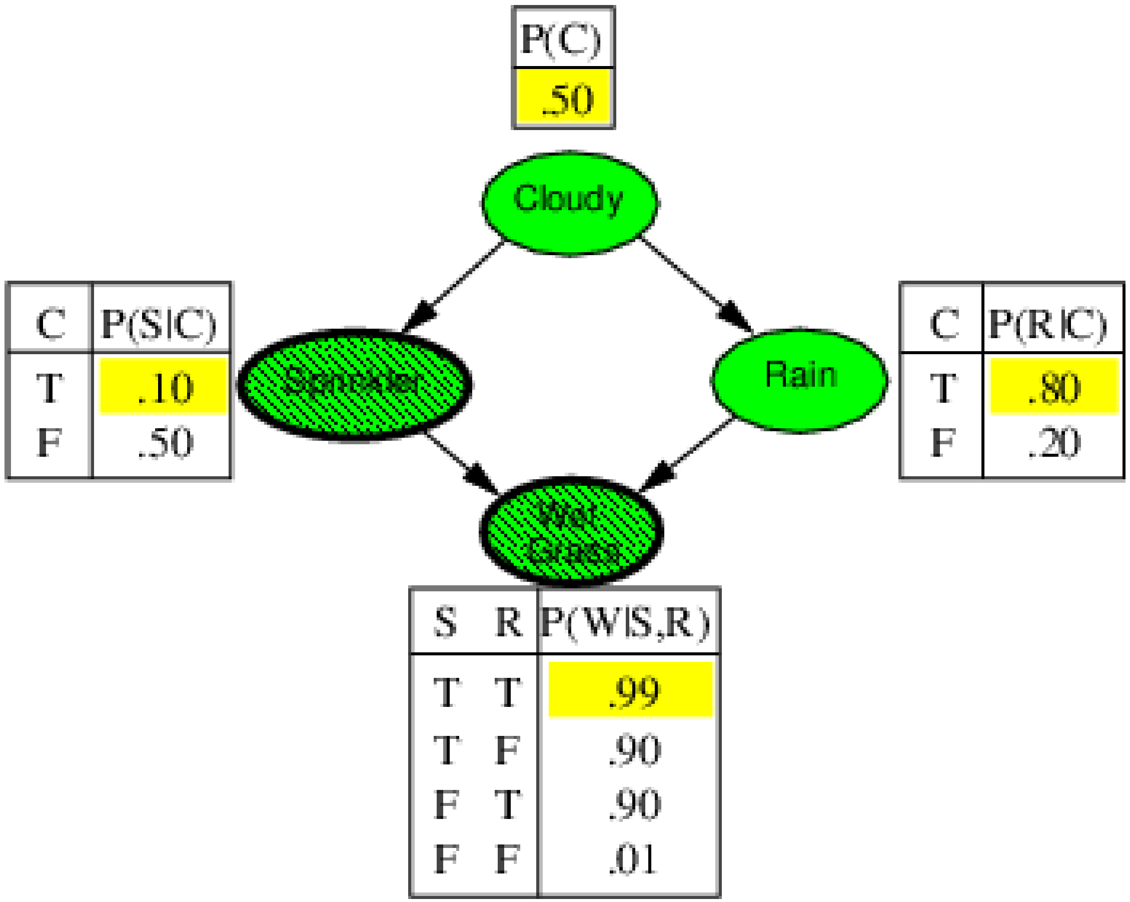
$$w = 1.0 \times 0.1$$

Likelihood Weighting Example



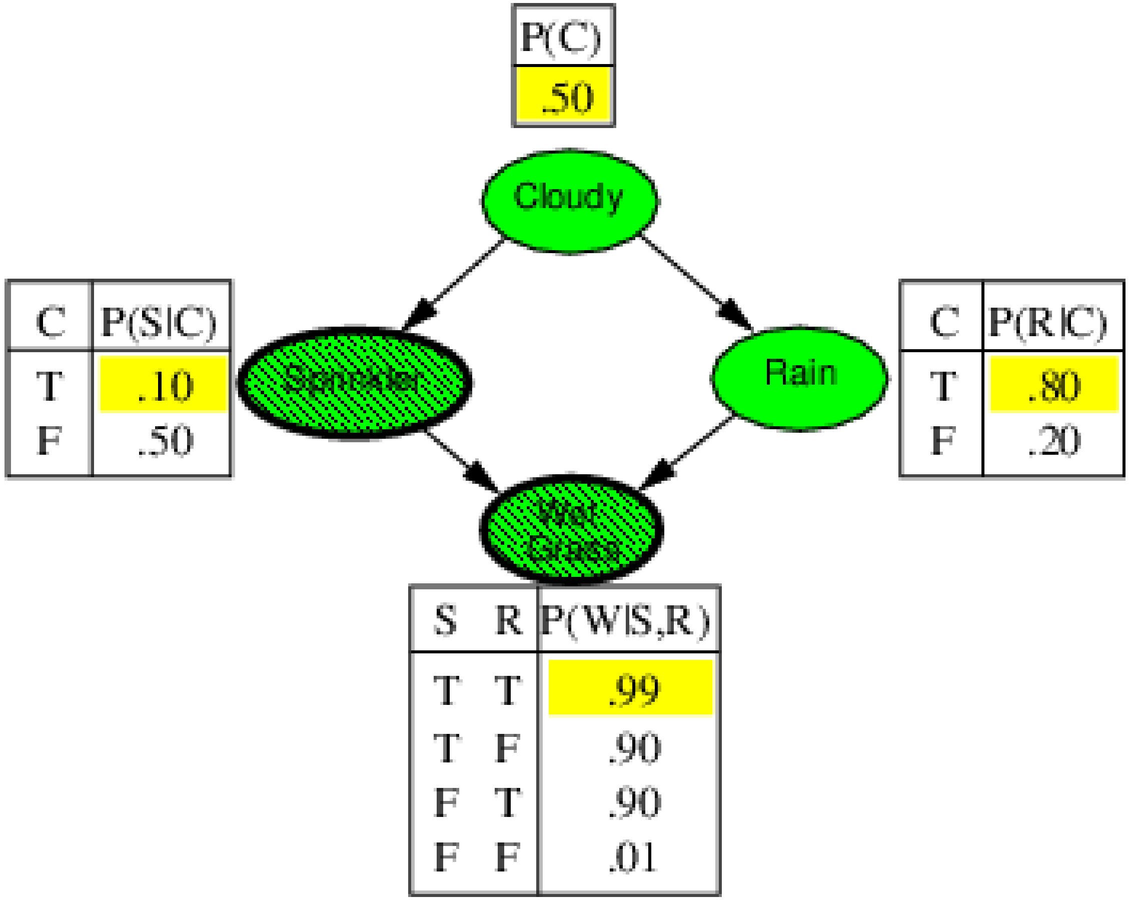
$$w = 1.0 \times 0.1$$

Likelihood Weighting Example



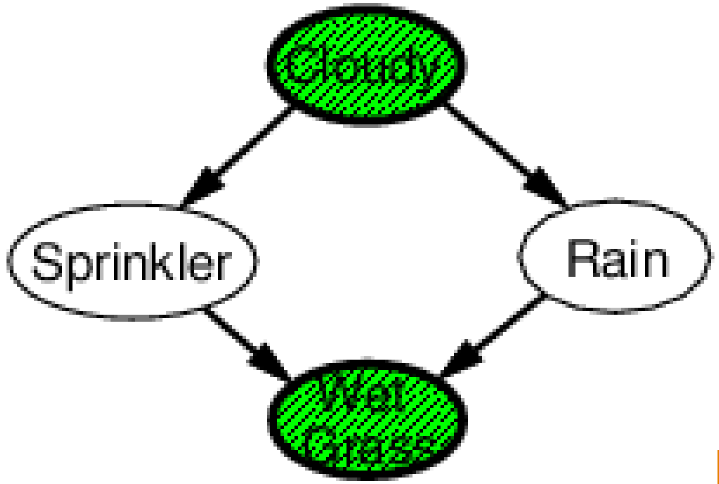
$$w = 1.0 \times 0.1$$

Likelihood Weighting Example



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

Likelihood Weighting Analysis



- Sampling probability for WEIGHTEDSAMPLE is
  $S_{WS}(\mathbf{z},\mathbf{e}) = \prod_{i=1}^{l} P(z_i|parents(Z_i))$

- 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)

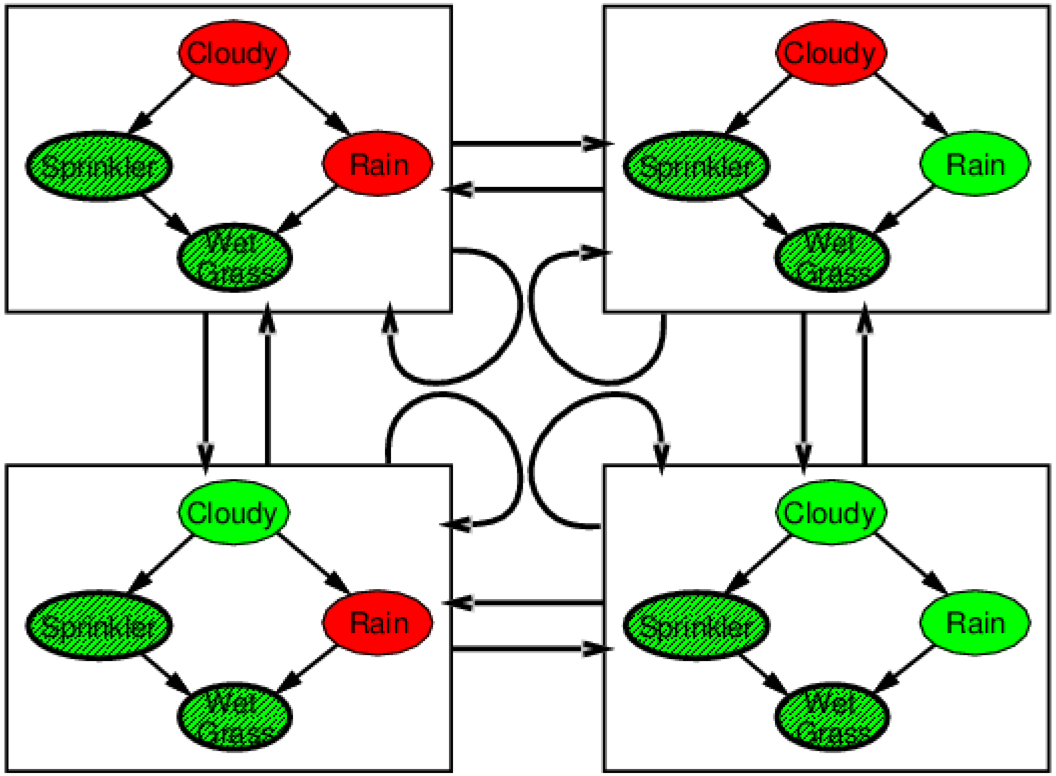
- Hence likelihood weighting returns consistent estimates
  but performance still degrades with many evidence variables
  because a few samples have nearly all the total weight

Approximate Inference using MCMC

- "State" of network = current assignment to all variables
- Generate next state by sampling one variable
  Sample each variable in turn, keeping evidence fixed
- Can also choose a variable to sample at random each time

The Markov Chain

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



- Wander about for a while, average what you see

MCMC Example

- 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

Summary

- Bayes nets provide a natural representation for (causally induced) conditional independence
- Generally easy for (non)experts to construct
- 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