CMU 15-781

Lecture 9: Bayesian Networks II

Teacher:
Gianni A. Di Caro

Probabilistic Inference Tasks

- Simple queries: $\boldsymbol{P}(X_i \mid \boldsymbol{E} = \mathbf{e})$
- Conjunctive queries: $\boldsymbol{P}(X_i, X_j \mid \mathbf{E} = \mathbf{e}) =$
$$\boldsymbol{P}(X_i \mid \mathbf{E} = \mathbf{e})\ \boldsymbol{P}(X_j \mid X_i, \mathbf{E} = \mathbf{e})$$
- Optimal decisions: decision networks include utility information; probabilistic inference is required for P(outcome | action, evidence)
- Value of information: which evidence to seek next?
- Sensitivity analysis: which variables are most critical?

General inference procedure

Let's partition the set of random variables in the model in:

- Evidence variables **E**, and be **e** the list of observed values from them
- Remaining unobserved / hidden variables **Y**
- Query variables X (let's consider single query for simplicity)

An **inference query is P(*X* | e)?** and can be evaluated as:

$$\boldsymbol{P}(X \mid \boldsymbol{e}) = \boldsymbol{P}(X, \boldsymbol{e})/P(\boldsymbol{e}) = \alpha \boldsymbol{P}(X, \boldsymbol{e}) = \alpha \sum_{y} \boldsymbol{P}(X, \boldsymbol{e}, \boldsymbol{y})$$

(from prob recall: marginal of a subset of variables + normalization constants)

$\rightarrow$ Given the full joint distribution, any probabilistic query can then be answered

INFERENCE WITH BNs

An inference query $\mathrm{P}(X \mid \mathbf{e})$ can be evaluated as:

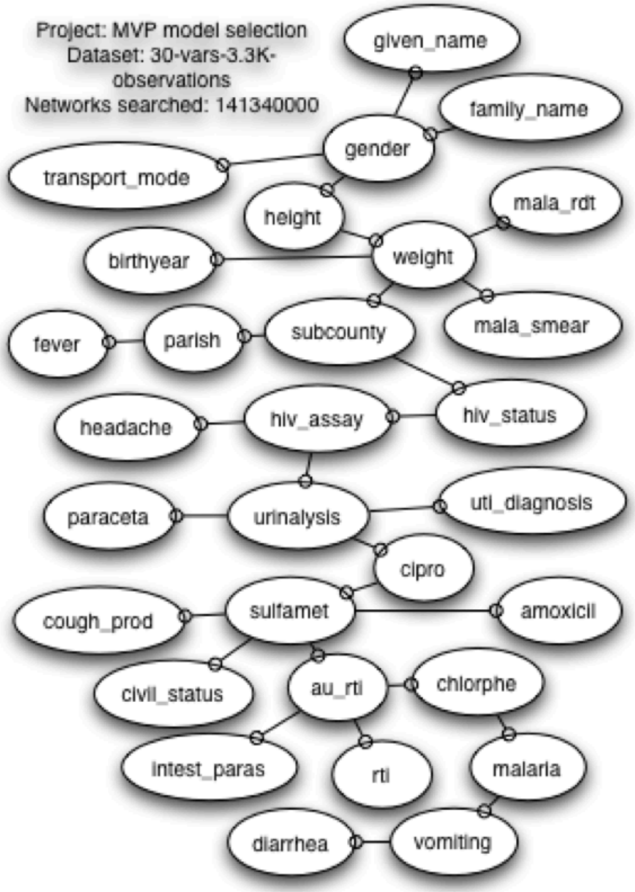
$$\boldsymbol{P}(X \mid \boldsymbol{e}) = \boldsymbol{P}(X, \boldsymbol{e})/P(\boldsymbol{e}) = \alpha \boldsymbol{P}(X, \boldsymbol{e}) = \alpha \sum_{y} \boldsymbol{P}(X, \boldsymbol{e}, \boldsymbol{y})$$

$\rightarrow$ Given the full joint distribution, any probabilistic query can then be answered

A BN is a compact way to represent a joint distribution, where the terms in the joint distribution are written as products of conditional probabilities from the network

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

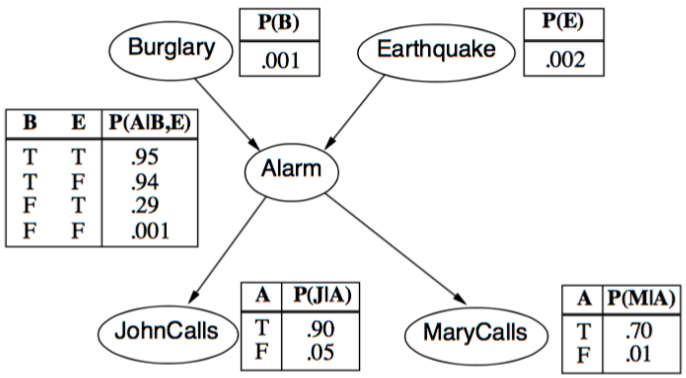
(Exact) Inference by enumeration?



Exact inference by enumeration

$$\boldsymbol{P}(X \mid \boldsymbol{e}) = \alpha \boldsymbol{P}(X, \boldsymbol{e}) = \alpha \sum_{y} \boldsymbol{P}(X, \boldsymbol{e}, \boldsymbol{y})$$

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



$\boldsymbol{P}$(B, | J=T, M=T)

Evidence: J, M

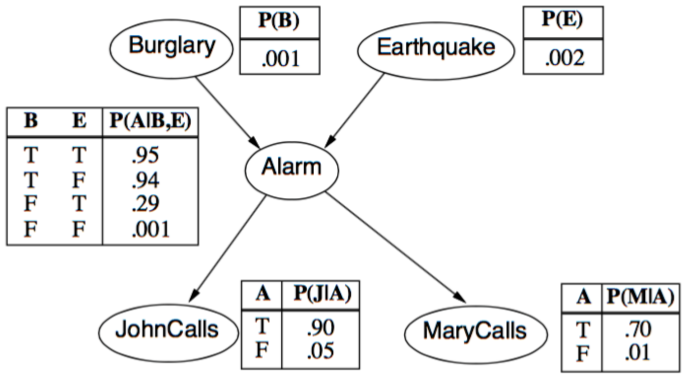
Hidden: E, A

Query: B

$$\boldsymbol{P}(B \mid j, m) = \alpha \boldsymbol{P}(B, j, m) = \alpha \sum_{e} \sum_{a} \boldsymbol{P}(B, e, a, j, m)$$

$$\boldsymbol{P}(B \mid j, m) = \alpha \sum_{e} \sum_{a} \boldsymbol{P}(B) P(e) \boldsymbol{P}(a \mid b, e) P(j \mid a) P(m \mid a)$$

Inference by enumeration



$$\boldsymbol{P}(B \mid j, m) = \alpha \sum_{e} \sum_{a} \boldsymbol{P}(B) P(e) \boldsymbol{P}(a \mid b, e) P(j \mid a) P(m \mid a)$$

What is the complexity of this calculation? $O(n2^n)$

INFERENCE BY ENUMERATION

- P(B=b) is a constant and can be moved outside the sums
- P(e) can be moved outside the summation over a

$$\mathbf{P}(B \mid j, m) = \alpha \mathbf{P}(B) \sum_{e} P(e) \sum_{a} \mathbf{P}(a \mid B, e) P(j \mid a) P(m \mid a)$$

$$\mathbf{P}(b \mid j, m) = \alpha \mathbf{P}(b) \textstyle\sum_{e} P(e) \sum_{a} \mathbf{P}(a \mid b, e) P(j \mid a) P(m \mid a)$$

$$= 0.001\alpha \textstyle\sum_{e} P(e) \Big[P(\neg a|b,e) P(j|\neg a) P(m|\neg a) + P(a|b,e) P(j|a) P(m|a) \Big]$$

$$= 0.001\alpha \textstyle\sum_{e} P(e) [0.598525]$$

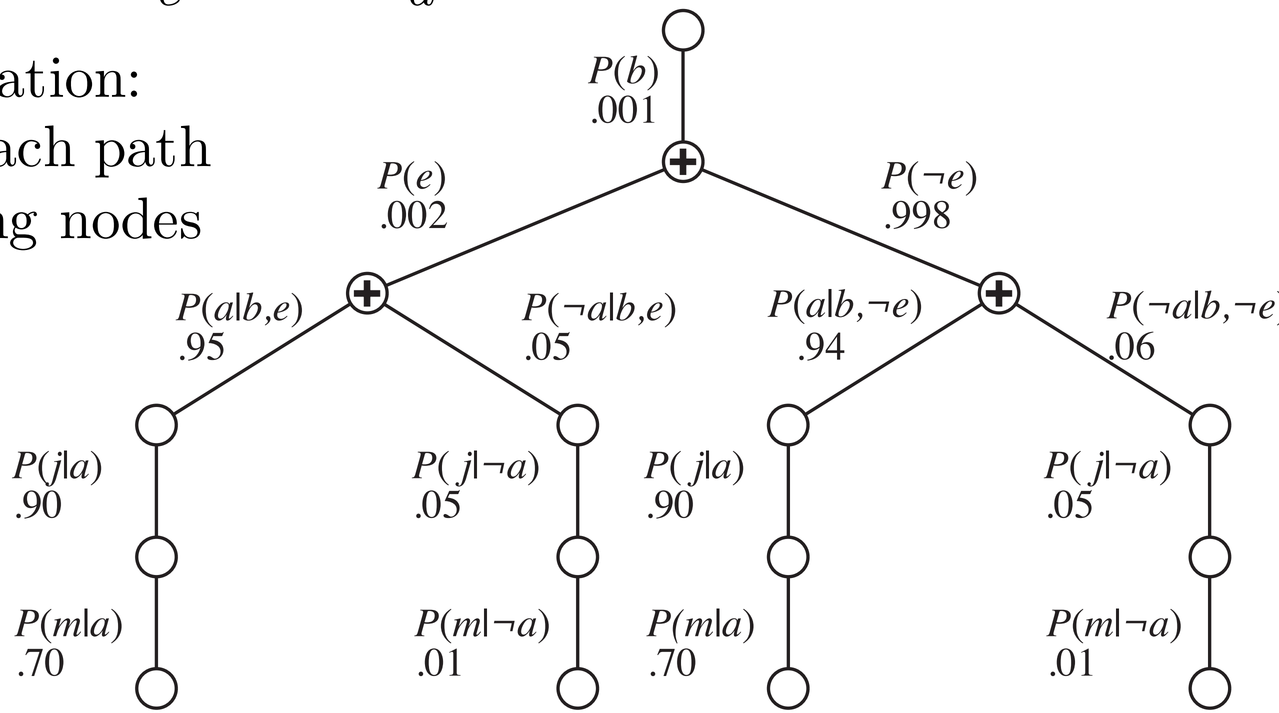
$$= 0.001\alpha (0.02 \cdot 0.598525) + (0.998 \cdot 0.598525) = 0.000602\alpha$$

Inference by enumeration

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

Top-down DF evaluation:

- $\times$ Values along each path
- $+$ at the branching nodes



Worst Case: NP-Hard

- Consider the 3-SAT clause:

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee \neg x_4) \wedge (x_2 \vee \neg x_2 \vee x_4) \wedge (\neg x_3 \vee \neg x_4 \vee \neg x_5) \wedge (x_2 \vee x_5 \vee x_7) \wedge (x_4 \vee x_5 \vee x_6) \wedge (\neg x_5 \vee x_6 \vee \neg x_7) \wedge (\neg x_5 \vee \neg x_6 \vee x_7)$$

which can be encoded by the following Bayes' net:

$P(X_i = 0) = P(X_i = 1) = 0.5$

$Y_1 = X_1 \vee X_2 \vee \neg X_3$

...

$Y_8 = \neg X_5 \vee X_6 \vee X_7$

$Y_{1,2} = Y_1 \wedge Y_2$

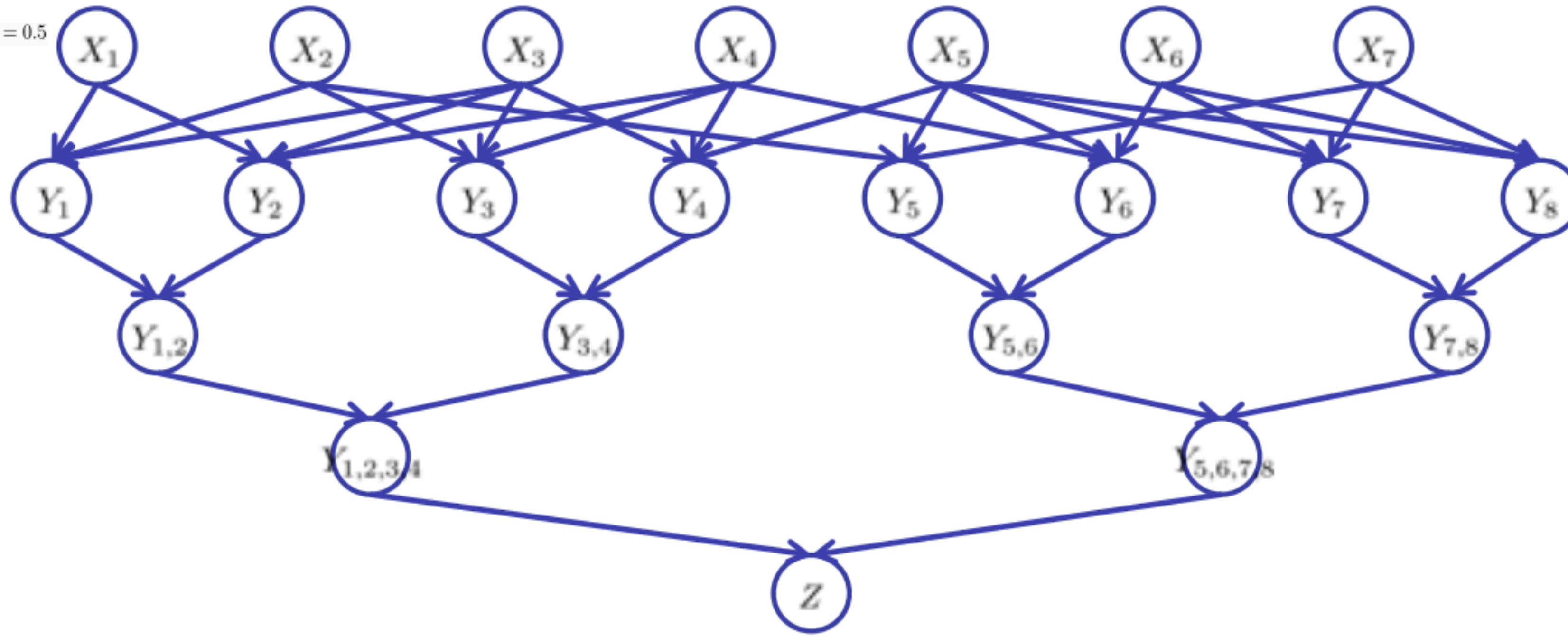
...

$Y_{7,8} = Y_7 \wedge Y_8$

$Y_{1,2,3,4} = Y_{1,2} \wedge Y_{3,4}$

$Y_{5,6,7,8} = Y_{5,6} \wedge Y_{7,8}$

$Z = Y_{1,2,3,4} \wedge Y_{5,6,7,8}$

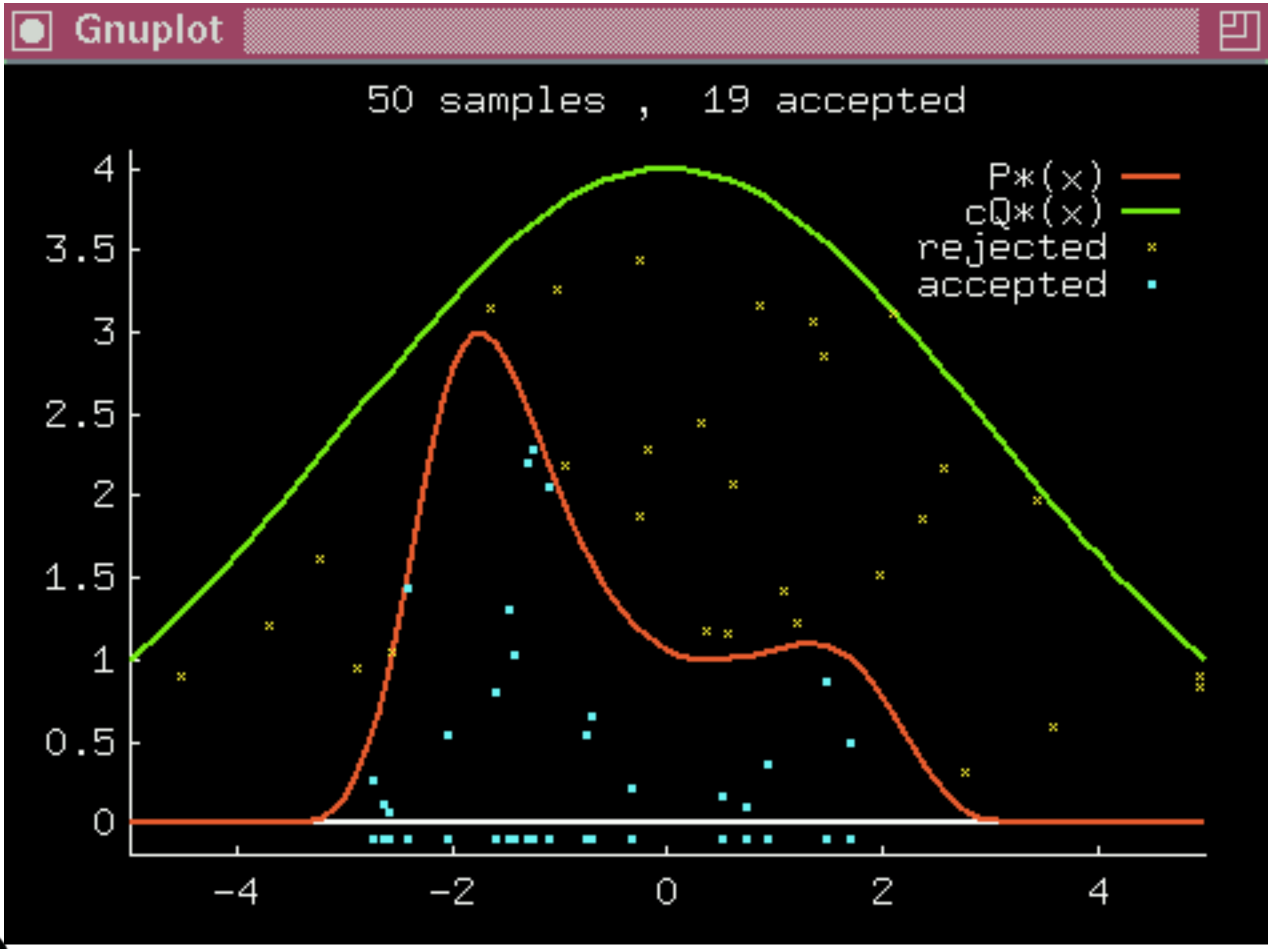


If we can answer P(z) equal to zero or not, we answered whether the 3-SAT problem has a solution.

Approximate inference in BN

- Given the general intractability of exact inference in large, multiply connected networks, it is essential to consider approximate inference methods
- **Monte Carlo algorithms**: randomized sampling methods … we have already seen one example of them!
- **Basic idea:** we express the quantity we want to know as the expected value of a random variable $\boldsymbol{X}$, such as $\boldsymbol{\mu}=\mathrm{E}(\boldsymbol{X})$. Then we generate values $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ independently and randomly from the distribution of X and take their sample average as the estimate of μ ($\rightarrow$ *law of large numbers*)
- **Problems:** it might be difficult to sample from $\boldsymbol{X}$'s distribution; a large number of samples might be needed

Monte Carlo sampling



Sampling for inference

- Have some method for generating samples given a known probability distribution (e.g., uniform in [0,1])
- A sample is an assignment of values to each variable in the network
- **Use samples to approximately compute posterior probabilities**
- Queries can be issued after finish sampling

P

X_1	X_2	X_3	X_4	X_5
T	T	F	T	T
F	F	F	F	T
T	T	T	F	F
T	F	F	F	T
T	T	T	T	T

Prob (T,T,F,F,T)?
#(T,T,F,F,T) / #Samples

DIRECT SAMPLING METHODS

- Generate events from a network with no evidence
- Each variable is sampled in turn, in *topological order*
- The probability distribution from which the value of a variable is sampled is conditioned on the values already assigned to the variable's parents

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

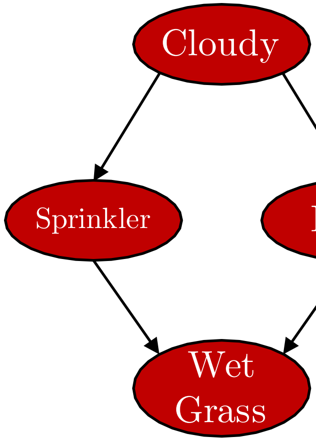
    $\mathbf{x} \leftarrow$ an event with n elements
    for $i = 1$ **to** n **do**
        $x_i \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
            given the values of $Parents(X_i)$ in $\mathbf{x}$
    return $\mathbf{x}$

Direct Sampling

- Sample $\Pr[C]=(.5,.5)$
  $\Rightarrow$ true

Order: C, S, R, W

+c	0.5
-c	0.5



+c	+s	.1
+c	-s	.9
-c	+s	.5
-c	-s	.5

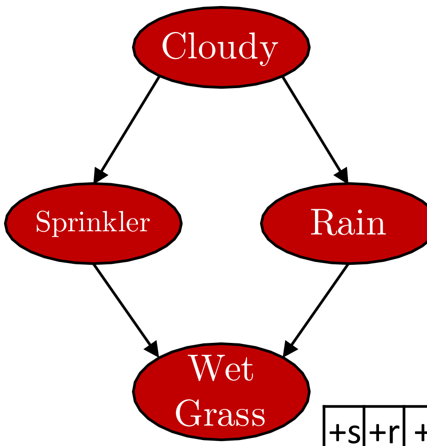
+c	+r	.8
+c	-r	.2
-c	+r	.2
-c	-r	.8

+s	+r	+w	.99
+s	+r	-w	.01
+s	-r	+w	.90
+s	-r	-w	.10
-s	+r	+w	.90
-s	+r	-w	.10
-s	-r	+w	0
-s	-r	-w	1.0

Direct Sampling

- Sample $\Pr[C]=(.5,.5)$
  $\Rightarrow$ true
- Sample $\Pr[S|C=t]=(.1,.9)$
  $\Rightarrow$ false

+c	0.5
-c	0.5



+c	+s	.1
+c	-s	.9
-c	+s	.5
-c	-s	.5

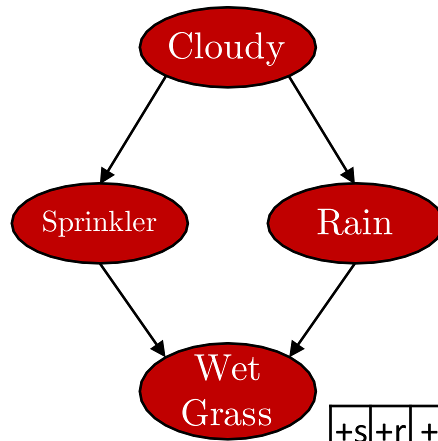
+c	+r	.8
+c	-r	.2
-c	+r	.2
-c	-r	.8

+s	+r	+w	.99
+s	+r	-w	.01
+s	-r	+w	.90
+s	-r	-w	.10
-s	+r	+w	.90
-s	+r	-w	.10
-s	-r	+w	0
-s	-r	-w	1.0

DIRECT SAMPLING

- Sample $\Pr[C]=(.5,.5)$
  $\Rightarrow$ true
- Sample $\Pr[S|C=t]=(.1,.9)$
  $\Rightarrow$ false
- Sample $\Pr[R|C=t]=(.8,.2)$
  $\Rightarrow$ true

+c	0.5
-c	0.5



+c	+s	.1
+c	-s	.9
-c	+s	.5
-c	-s	.5

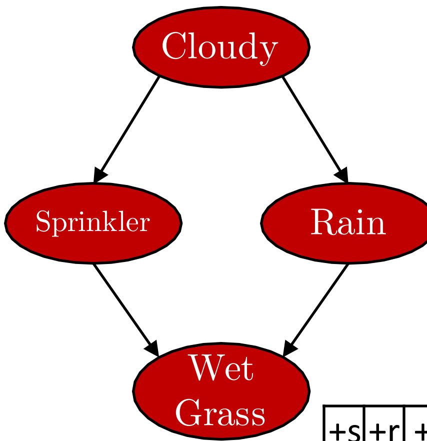
+c	+r	.8
+c	-r	.2
-c	+r	.2
-c	-r	.8

+s	+r	+w	.99
+s	+r	-w	.01
+s	-r	+w	.90
+s	-r	-w	.10
-s	+r	+w	.90
-s	+r	-w	.10
-s	-r	+w	0
-s	-r	-w	1.0

Direct Sampling

- Sample Pr[C]=(.5,.5)
  $\Rightarrow$ true
- Sample Pr[S|C=t]=(.1,.9)
  $\Rightarrow$ false
- Sample Pr[R|C=t]=(.8,.2)
  $\Rightarrow$ true
- Sample Pr[W|S=f,R=t]=(.9,.1)
  $\Rightarrow$ true
- Sampled [t,f,t,t]

+c	0.5
-c	0.5



+c	+s	.1
+c	-s	.9
-c	+s	.5
-c	-s	.5

+c	+r	.8
+c	-r	.2
-c	+r	.2
-c	-r	.8

+s	+r	+w	.99
+s	+r	-w	.01
+s	-r	+w	.90
+s	-r	-w	.10
-s	+r	+w	.90
-s	+r	-w	.10
-s	-r	+w	0
-s	-r	-w	1.0

Direct Sampling

Probability that PriorSample generates a particular event

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

i.e., the true prior probability

For large N, 32.4% of events (t,f,f,t) are expected

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

- What about P(X|**e**), i.e., when we have **evidence**?

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**
    $\mathbf{x} \leftarrow$ PRIOR-SAMPLE(bn)
    if x is consistent with **e then**
      $\mathbf{N}[x] \leftarrow \mathbf{N}[x]+1$ where x is the value of X in **x**
  return NORMALIZE($\mathbf{N}[X]$)

Similar to estimation of conditional Probabilities directly from the real world

Problem: try to estimate Pr[Rain | RedSkyAtNight=t]!

Rejection sampling

- Want to estimate Pr[Rain=t | Sprinkler=t]
- 100 direct samples (no evidence included) are generated
- 73 have S=f, of which 12 have R=t
- 27 have S=t, of which 8 have R=t

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

- Error goes as $1/\sqrt{n}$, n = useful samples
- The estimate is consistent
- Too many samples thrown away! (because they are generated with direct sampling)

SOLUTION: LIKELIHOOD WEIGHTING

- Generate only samples that **agree with evidence**
- Fix the evidence vars and sample the nonevidence only
- $\rightarrow$ Each generated event is consistent with evidence
- **Weight each generated event according to likelihood that the event accords to evidence**
- **The likelihood is measured as the product of the conditional probabilities for each evidence variable given its parents**
- Event unlikely according to current evidence should weight less

Likelihood weighting

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($\mathbf{W}[X]$)

---

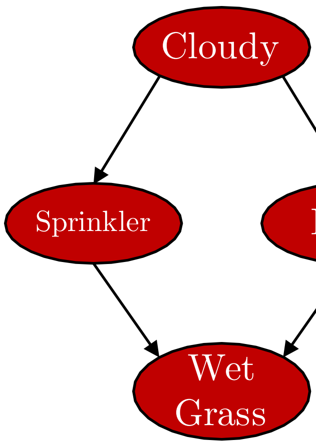
function WEIGHTED-SAMPLE(bn, $\mathbf{e}$) **returns** an event and a weight

  $\mathbf{x} \leftarrow$ 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 $w \leftarrow w \times P(X_i = x_i \mid parents(X_i))$
      else $x_i \leftarrow$ a random sample from $\mathbf{P}(X_i \mid parents(X_i))$
  return $\mathbf{x}$, w

Likelihood weighting

- Evidence: C=t,W=t
- C is evidence var
  $\Rightarrow$ w = 1·Pr[C=t] = 0.5

+c	0.5
-c	0.5



+c	+s	.1
+c	-s	.9
-c	+s	.5
-c	-s	.5

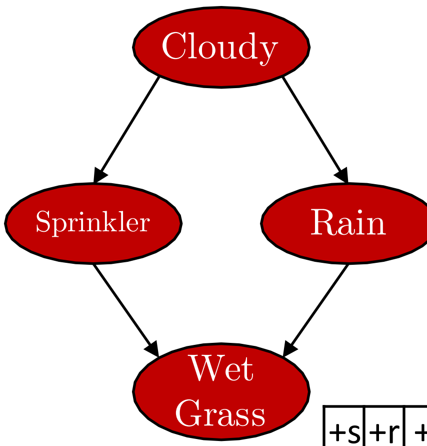
+c	+r	.8
+c	-r	.2
-c	+r	.2
-c	-r	.8

+s	+r	+w	.99
+s	+r	-w	.01
+s	-r	+w	.90
+s	-r	-w	.10
-s	+r	+w	.90
-s	+r	-w	.10
-s	-r	+w	0
-s	-r	-w	1.0

Likelihood weighting

- Evidence: C=t,W=t
- C is evidence var
  $\Rightarrow$ w = 1·Pr[C=t] = 0.5
- Sample Pr[S|C=t]=(.1,.9)
  $\Rightarrow$ false

+c	0.5
-c	0.5



+c	+s	.1
+c	-s	.9
-c	+s	.5
-c	-s	.5

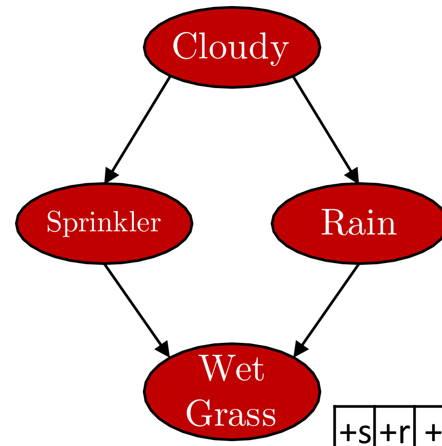
+c	+r	.8
+c	-r	.2
-c	+r	.2
-c	-r	.8

+s	+r	+w	.99
+s	+r	-w	.01
+s	-r	+w	.90
+s	-r	-w	.10
-s	+r	+w	.90
-s	+r	-w	.10
-s	-r	+w	0
-s	-r	-w	1.0

Likelihood weighting

- Evidence: C=t,W=t
- C is evidence var
  $\Rightarrow$ w = 1·Pr[C=t] = 0.5
- Sample Pr[S|C=t]=(.1,.9)
  $\Rightarrow$ false
- Sample Pr[R|C=t]=(.8,.2)
  $\Rightarrow$ true

+c	0.5
-c	0.5



+c	+s	.1
+c	-s	.9
-c	+s	.5
-c	-s	.5

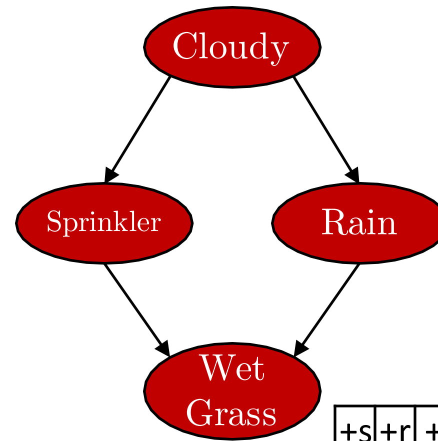
+c	+r	.8
+c	-r	.2
-c	+r	.2
-c	-r	.8

+s	+r	+w	.99
+s	+r	-w	.01
+s	-r	+w	.90
+s	-r	-w	.10
-s	+r	+w	.90
-s	+r	-w	.10
-s	-r	+w	0
-s	-r	-w	1.0

Likelihood weighting

- Evidence: C=t,W=t
- C is evidence var
  $\Rightarrow$ w $=$ 1·Pr[C=t] $=$ 0.5
- Sample Pr[S|C=t]=(.1,.9)
  $\Rightarrow$ false
- Sample Pr[R|C=t]=(.8,.2)
  $\Rightarrow$ true
- W is evidence var
  $\Rightarrow$ w $=$ 0.5·Pr[W=t | S=f,R=t] $=$ .45
- Sampled [t,f,t,t] with weight .45, tallied under R=t

+c	0.5
-c	0.5



+c	+s	.1
+c	-s	.9
-c	+s	.5
-c	-s	.5

+c	+r	.8
+c	-r	.2
-c	+r	.2
-c	-r	.8

+s	+r	+w	.99
+s	+r	-w	.01
+s	-r	+w	.90
+s	-r	-w	.10
-s	+r	+w	.90
-s	+r	-w	.10
-s	-r	+w	0
-s	-r	-w	1.0

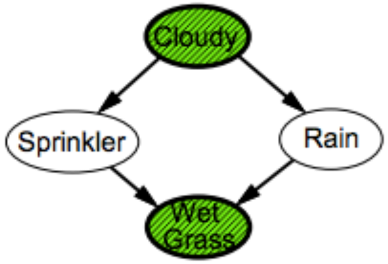
Discussion

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

Discussion

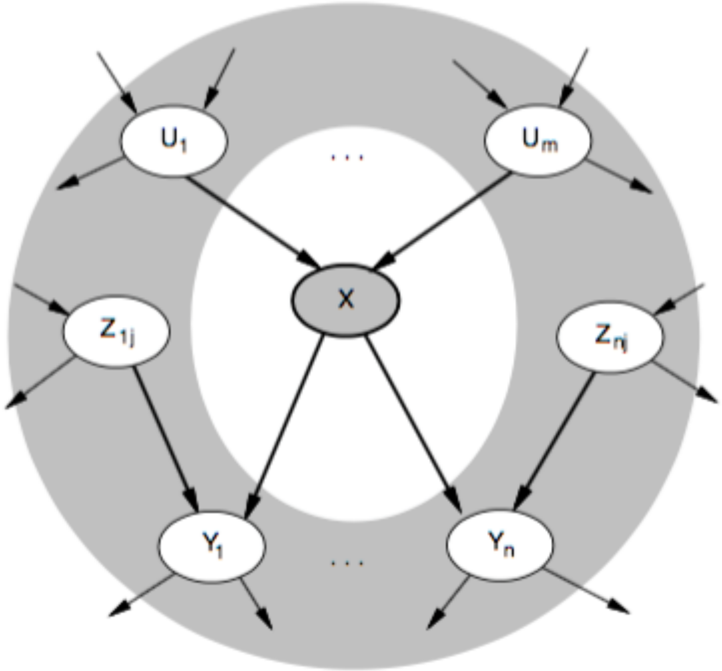
- Because of topological order, when sampling S and R the evidence W=t is ignored ⇒ samples with S=f and R=f although evidence rules this out
- Weight makes up for this difference: above weight would be 0
- If we have 100 samples with R=t and total weight 1, and 400 samples with R=f and total weight 2, what is estimate of R=t?
- Problem: bad if evidence variables occur later in ordering

Gibbs sampling

- **Markov Chain Monte Carlo (MCMC):** each sample is generated by making a random change to the preceding sample
- MCMC are algorithms with a *state,* the next state is generated from the current one
- Specific MCMC: Gibbs sampling, the sampling process settles into a "dynamic equilibrium" in which the long-run fraction of time spent in each state is exactly proportional to its posterior probability
- The states are generated given the *Markov blanket*: state transitions are defined by the conditional distribution

Markov Blanket

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



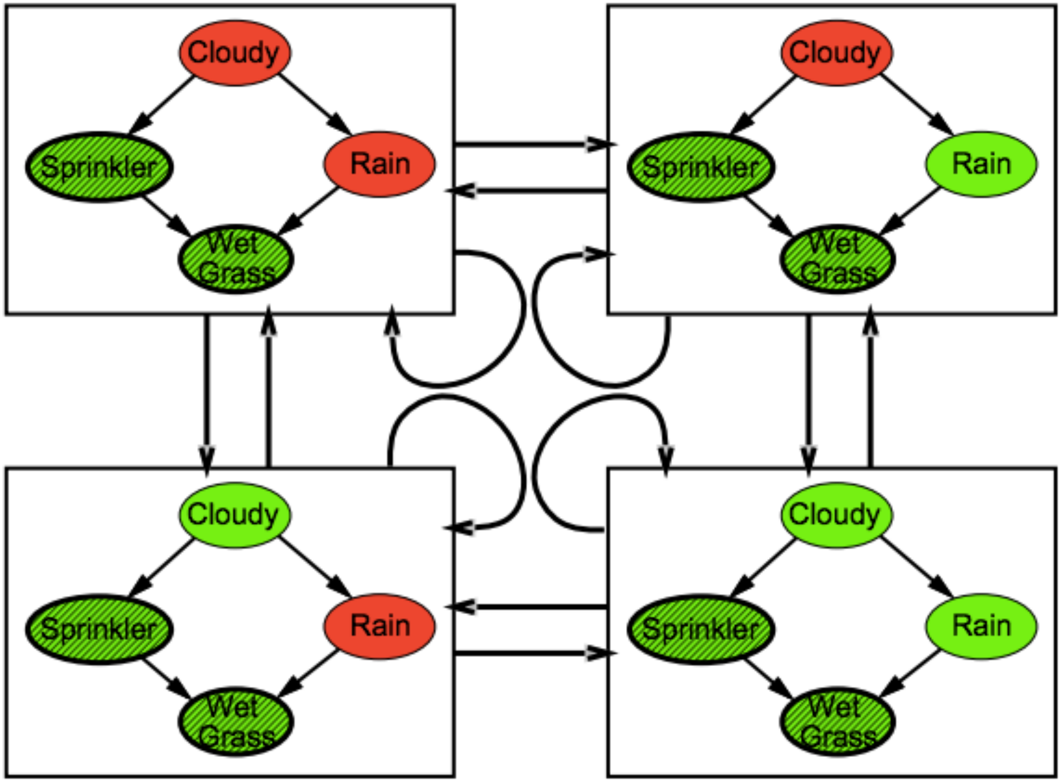
Gibbs sampling

function GIBBS-SAMPLING(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($\mathbf{N}[X]$)

Gibbs sampling

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



Wander about for a while, average what you see

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.

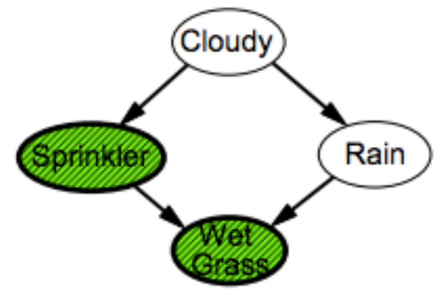
E.g., visit 100 states
31 have $Rain = true$, 69 have $Rain = false$

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

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

MB 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:

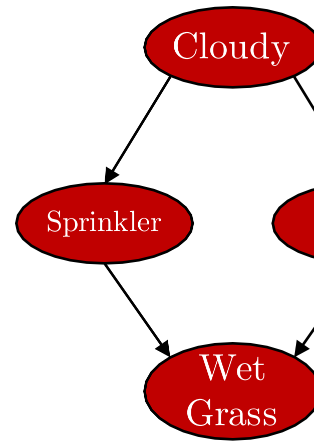
1) Difficult to tell if convergence has been achieved
2) Can be wasteful if Markov blanket is large:
   $P(X_i|mb(X_i))$ won't change much (law of large numbers)

From G. Mori

Gibbs Sampling Example

- Want Pr(R|S=t,W=t)
- Non-evidence variables are C & R
- Initialize randomly: C= t and R=f
- Initial state (C,S,R,W)= [t,t,f,t]
- Sample C given current values of its Markov Blanket

+c	0.5
-c	0.5



+c	+s	.1
+c	-s	.9
-c	+s	.5
-c	-s	.5

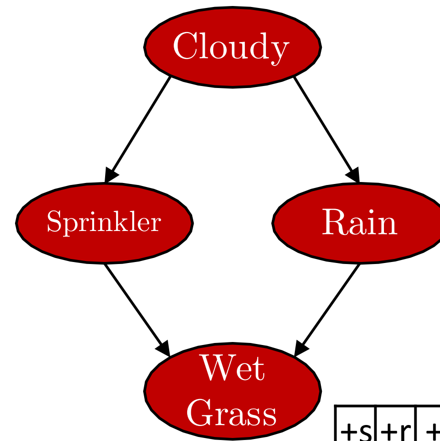
+c	+r	.8
+c	-r	.2
-c	+r	.2
-c	-r	.8

+s	+r	+w	.99
+s	+r	-w	.01
+s	-r	+w	.90
+s	-r	-w	.10
-s	+r	+w	.90
-s	+r	-w	.10
-s	-r	+w	0
-s	-r	-w	1.0

Gibbs Sampling Example

- Want Pr(R|S=t,W=t)
- Non-evidence variables are C & R
- Initialize randomly: C= t and R=f
- Initial state (C,S,R,W)= [t,t,f,t]
- Sample C given current values of its Markov Blanket
- Markov blanket is parents, children and children's parents: for C=S & R
- Sample C given P(C|S=t,R=f)
- First have to compute P(C|S=t,R=f)
- Use exact inference to do this

+c	0.5
-c	0.5



+c	+s	.1
+c	-s	.9
-c	+s	.5
-c	-s	.5

+c	+r	.8
+c	-r	.2
-c	+r	.2
-c	-r	.8

+s	+r	+w	.99
+s	+r	-w	.01
+s	-r	+w	.90
+s	-r	-w	.10
-s	+r	+w	.90
-s	+r	-w	.10
-s	-r	+w	0
-s	-r	-w	1.0

How do we compute P(C | S=t, R=f)? (recall of exact inference)

- $P(C|S=t,R=f)$
- What is the probability $P(C=t \mid S=t, R=f)$?

$= P(C=t, S=t, R=f) \;/\; (P(S=t,R=f))$

Proportional to $P(C=t, S=t, R=f)$

Use normalization trick, & compute the above for C=t and C=f

$P(C=t, S=t, R=f) = P(C=t)\, P(S=t|C=t)\, P(R=f \mid C=t, S=t)$ product rule

$= P(C=t)\, P(S=t|C=t)\, P(R=f \mid C=t)$ (BN independencies)

$= 0.5 * 0.1 * 0.2 = 0.01$

$P(C=f, S=t, R=f) = P(C=f)\, P(S=t|C=f)\, P(R=f|C=f)$

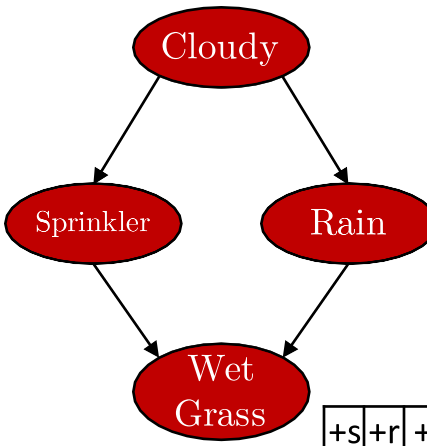
$= 0.5 * 0.5 * 0.8 = 0.2$

$(P(S=t,R=f))$ use sum rule $= P(C=f, S=t, R=f) + P(C=t, S=t, R=f)$

$P(C = t \mid S=t, R= f) = 0.21$

$P(C=t \mid S=t, R = f) = 0.01 / 0.21 \sim 0.0476$

+c	0.5
-c	0.5



+c	+s	.1
+c	-s	.9
-c	+s	.5
-c	-s	.5

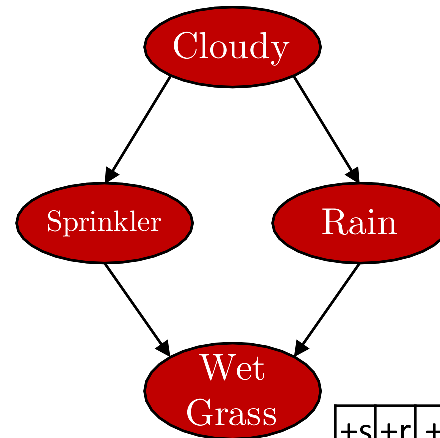
+c	+r	.8
+c	-r	.2
-c	+r	.2
-c	-r	.8

+s	+r	+w	.99
+s	+r	-w	.01
+s	-r	+w	.90
+s	-r	-w	.10
-s	+r	+w	.90
-s	+r	-w	.10
-s	-r	+w	0
-s	-r	-w	1.0

Gibbs Sampling Example

- Want Pr(R|S=t,W=t)
- Non-evidence variables are C & R
- Initialize randomly: C= t and R=f
- Initial state (C,S,R,W)= [t,t,f,t]
- Sample C given current values of its Markov Blanket
- Markov blanket is parents, children and children's parents: for C=S & R
- Exactly compute P(C|S=t,R=f)
- Sample C given P(C|S=t,R=f)
- Get C = f
- New state (f,t,f,t)

+c	0.5
-c	0.5



+c	+s	.1
+c	-s	.9
-c	+s	.5
-c	-s	.5

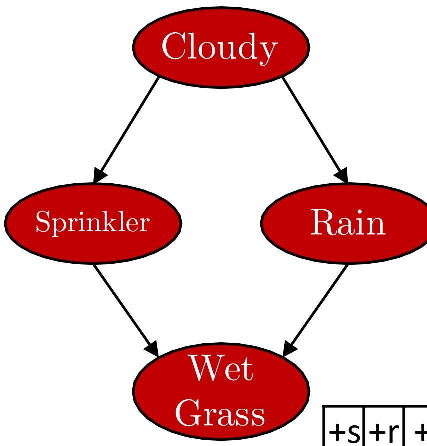
+c	+r	.8
+c	-r	.2
-c	+r	.2
-c	-r	.8

+s	+r	+w	.99
+s	+r	-w	.01
+s	-r	+w	.90
+s	-r	-w	.10
-s	+r	+w	.90
-s	+r	-w	.10
-s	-r	+w	0
-s	-r	-w	1.0

Gibbs Sampling Example

- Want Pr(R|S=t,W=t)
- Initialize non-evidence variables (C and R) randomly to t and f
- Initial state (C,S,R,W)= [t,t,f,t]
- Sample C given current values of its Markov Blanket, p(C|S=t,R=f)
- Suppose result is C=f
- New state (f,t,f,t)
- Sample Rain given its MB
- What is its Markov blanket?

+c	0.5
-c	0.5



+c	+s	.1
+c	-s	.9
-c	+s	.5
-c	-s	.5

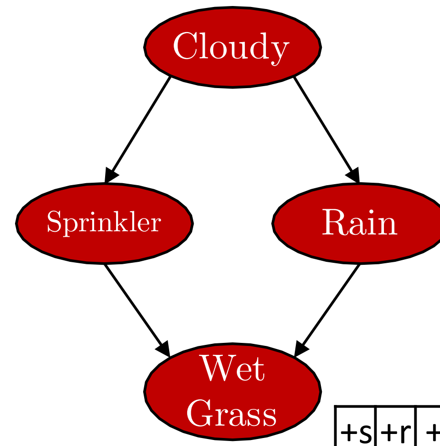
+c	+r	.8
+c	-r	.2
-c	+r	.2
-c	-r	.8

+s	+r	+w	.99
+s	+r	-w	.01
+s	-r	+w	.90
+s	-r	-w	.10
-s	+r	+w	.90
-s	+r	-w	.10
-s	-r	+w	0
-s	-r	-w	1.0

Gibbs Sampling Example

- Want Pr(R|S=t,W=t)
- Initialize non-evidence variables (C and R) randomly to t and f
- Initial state (C,S,R,W)= [t,t,f,t]
- Sample C given current values of its Markov Blanket, p(C|S=t,R=f)
- Suppose result is C=f
- New state (f,t,f,t)
- Sample Rain given its MB, p(R|C=f,S=t,W=t)
- Suppose result is R=t
- New state (f,t,t,t)

+c	0.5
-c	0.5



+c	+s	.1
+c	-s	.9
-c	+s	.5
-c	-s	.5

+c	+r	.8
+c	-r	.2
-c	+r	.2
-c	-r	.8

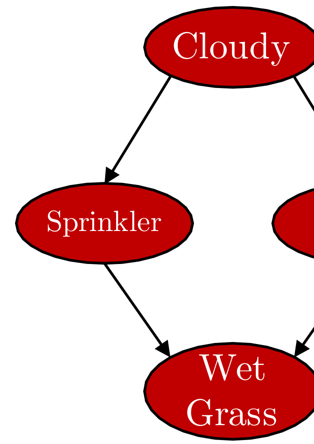
+s	+r	+w	.99
+s	+r	-w	.01
+s	-r	+w	.90
+s	-r	-w	.10
-s	+r	+w	.90
-s	+r	-w	.10
-s	-r	+w	0
-s	-r	-w	1.0

Poll: Gibbs Sampling Ex.

- Want Pr(R|S=t,W=t)
- Initialize non-evidence variables (C and R) randomly to t and f
- Initial state (C,S,R,W)= [t,t,f,t]
- Current state (f,t,t,t)

- What is **not** a possible next state
- 1. (f,t,t,t)
- 2. (t,t,t,t)
- 3. (f,t,f,t)
- 4. (f,f,t,t)
- 5. Not sure

+c	0.5
-c	0.5



+c	+s	.1
+c	-s	.9
-c	+s	.5
-c	-s	.5

+c	+r	.8
+c	-r	.2
-c	+r	.2
-c	-r	.8

+s	+r	+w	.99
+s	+r	-w	.01
+s	-r	+w	.90
+s	-r	-w	.10
-s	+r	+w	.90
-s	+r	-w	.10
-s	-r	+w	0
-s	-r	-w	1.0