Intelligent Systems (AI-2)

Computer Science cpsc422, Lecture 12

Feb, 5, 2021

Slide credit: some slides adapted from Stuart Russell (Berkeley)

Lecture Overview

- **Recap of Forward and Rejection Sampling**
- **Likelihood Weighting**
- **Monte Carlo Markov Chain (MCMC) – Gibbs Sampling**
- **Application Requiring Approx. reasoning**

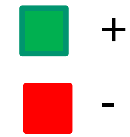
Sampling

The building block on any sampling algorithm **is the generation of samples from a known (or easy to compute, like in Gibbs) distribution**

We then **use these samples to derive estimates of probabilities hard-to-compute exactly**

And you want **consistent sampling methods... More samples... Closer to...**

Prior Sampling

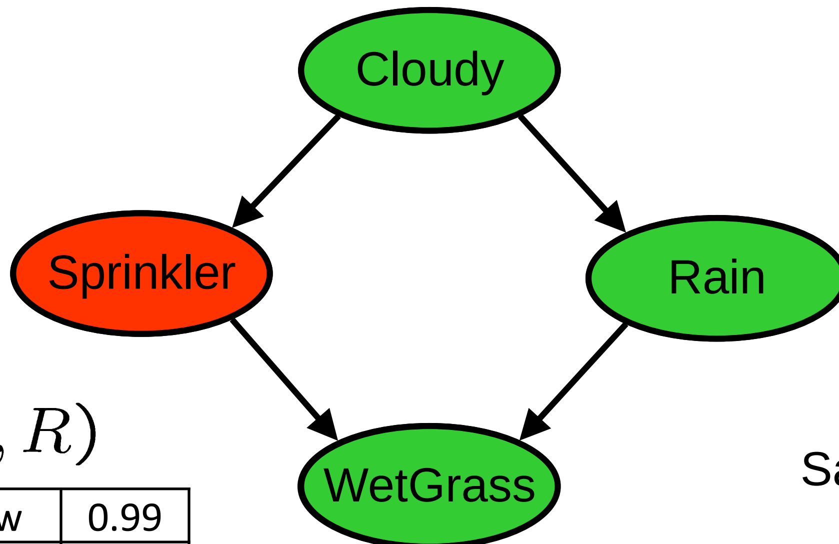


$P(C)$

+c	0.5
-c	0.5

$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5



$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S,R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

Samples:

+c, -s, +r, +w

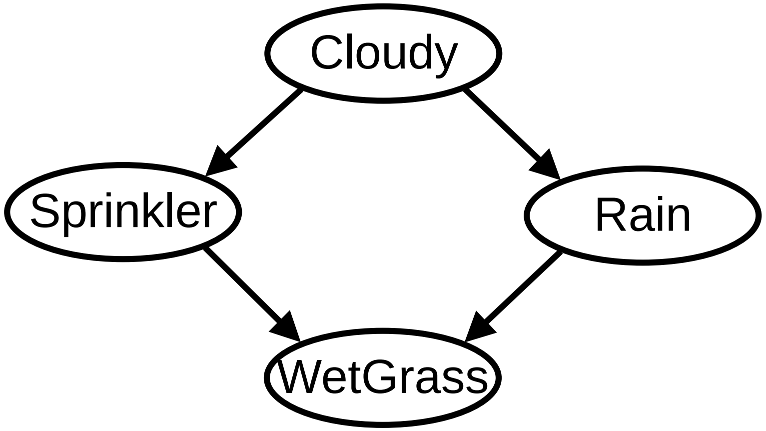
-c, +s, -r, +w

…

Example

We'll get a bunch of samples from the BN:

+c, -s, +r, +w
+c, +s, +r, +w
-c, +s, +r, -w
+c, -s, +r, +w
-c, -s, -r, +w



From these samples you can compute any distribution involving the four vars... P(W), P(S)…P(S,R)..

P(W)

+w	4/5
-w	1/5

P(S)

+s	2/5
-s	3/5

P(S,R)

+s	+r	2/5
+s	-r	0
-s	+r	2/5
-s	-r	1/5

Example

Can estimate anything else from the samples, besides P(W), P(S) , etc:

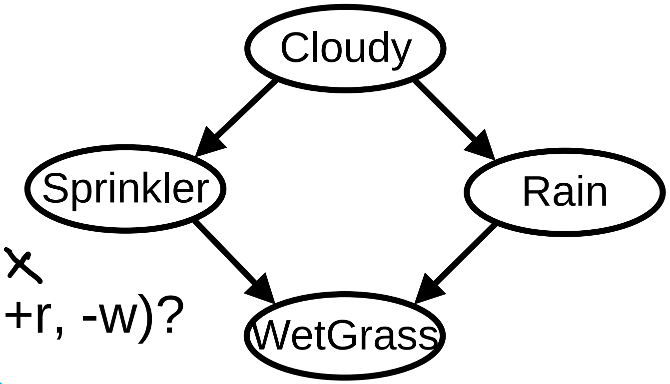
+c, -s, +r, +w

+c, +s, +r, +w

-c, +s, +r, -w

+c, -s, +r, +w

-c, -s, -r, +w



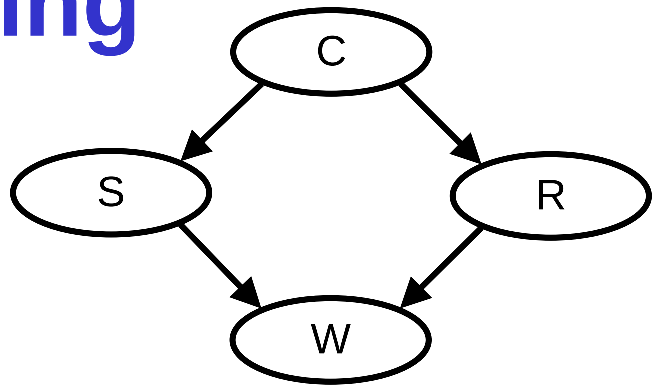
- What about P(C| +w)? P(C| +r, +w)? P(C| +r, -w)?

+c -c +c -c +c -c

3/4 1/4 1 0 0 1

Can use/generate fewer samples when we want to estimate a probability conditioned on evidence?

Rejection Sampling



Let's say we want P(W| +s)

- ignore (reject) samples which don't have S=+s
- This is called rejection sampling
- It is also consistent for conditional probabilities (i.e., correct in the limit)

+c, -s, +r, +w
+c, +s, +r, +w
-c, +s, +r, -w
+c, -s, +r, +w
-c, -s, -r, +w

But what happens if +s is rare?

And if the number of evidence vars grows...

iclicker

A. Less samples will be rejected

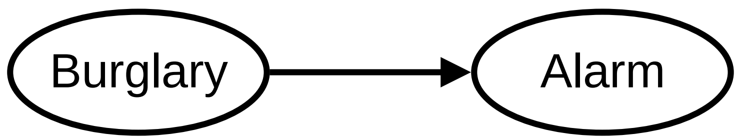
B. More samples will be rejected

C. The same number of samples will be rejected

Likelihood Weighting

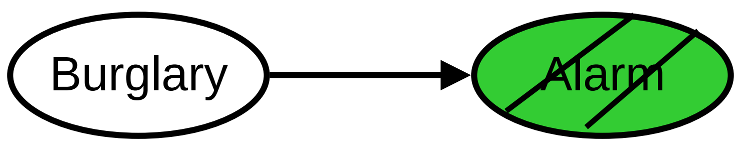
Problem with rejection sampling:

- If evidence is unlikely, you reject a lot of samples
- You don't exploit your evidence as you sample
- Consider P(B|+a)



-b, -a
-b, -a
-b, -a
-b, -a
+b, +a

Idea: fix evidence variables and sample the rest



-b +a
-b, +a
-b, +a
-b, +a
+b, +a

Problem?: sample distribution not consistent!

Solution: weight by probability of evidence given parents

Likelihood Weighting

+

-

+ evidence

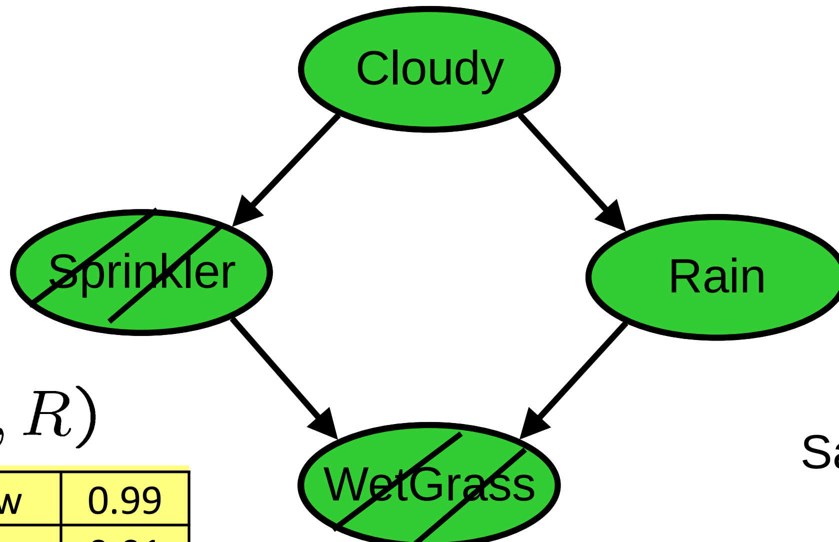
-

$P(C)$

+c	0.5
-c	0.5

$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5



$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S,R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

Samples:

+c +s +r +w

…

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

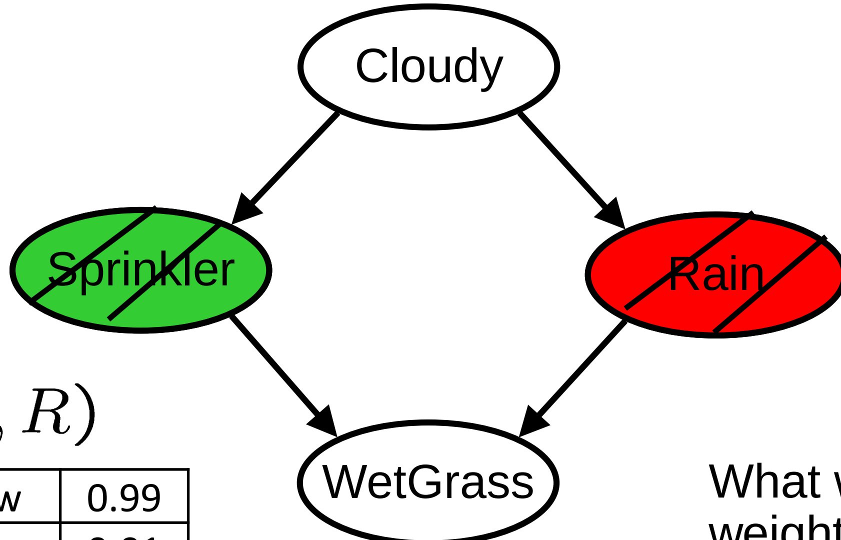
Likelihood Weighting

$P(C)$

+c	0.5
-c	0.5

$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5



$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S,R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

What would be the weight for this sample?

+c, +s, -r, +w

iclicker.

A 0.08

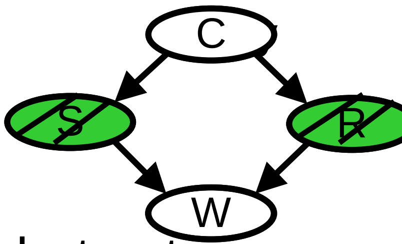
B 0.02

C. 0.005

Likelihood Weighting

Likelihood weighting is good

- We have taken evidence into account as we generate the sample
- All our samples will reflect the state of the world suggested by the evidence
- Uses all samples that it generates (much more efficient than rejection sampling)



Likelihood weighting doesn't solve all our problems

- Evidence influences the choice of downstream variables, but not upstream ones (*C isn't more likely to get a value matching the evidence*)
- Degradation in performance with large number of evidence vars -> each sample small weight

We would like to consider evidence when we sample *every* variable

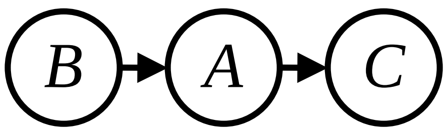
Lecture Overview

- **Recap of Forward and Rejection Sampling**
- **Likelihood Weighting**
- **Monte Carlo Markov Chain (MCMC) – Gibbs Sampling**
- **Application Requiring Approx. reasoning**

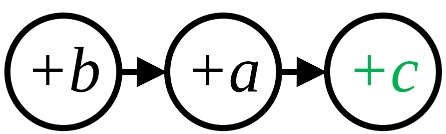
Markov Chain Monte Carlo

We would like to consider evidence when we sample *every* variable

Idea: instead of sampling from scratch, create samples that are each like the last one (only randomly change one var).



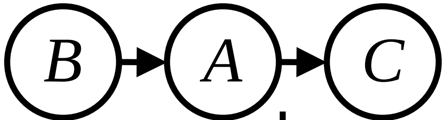
Initialize: assign all **non-evidence** variables randomly and fix **evidence** ones. E.g., for P(B|+c) assign A and B randomly and C to +c



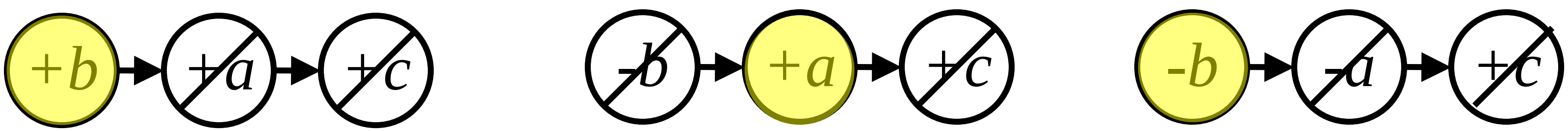
+b, +a, +c

Markov Chain Monte Carlo

Idea: instead of sampling from scratch, create samples that are each like the last one (only randomly change one var).



Procedure: resample one variable at a time, conditioned on all the rest, but keep evidence fixed. E.g., for P(B|+c):



+b, +a, +c

Sample b

- b, +a, +c

Sample a

- b, -a, +c

Sample b

- b, -a, +c

Sample a

- b, -a, +c

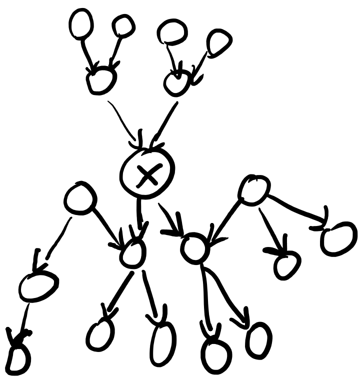
Sample b

+ b, -a, +c

Markov Chain Monte Carlo

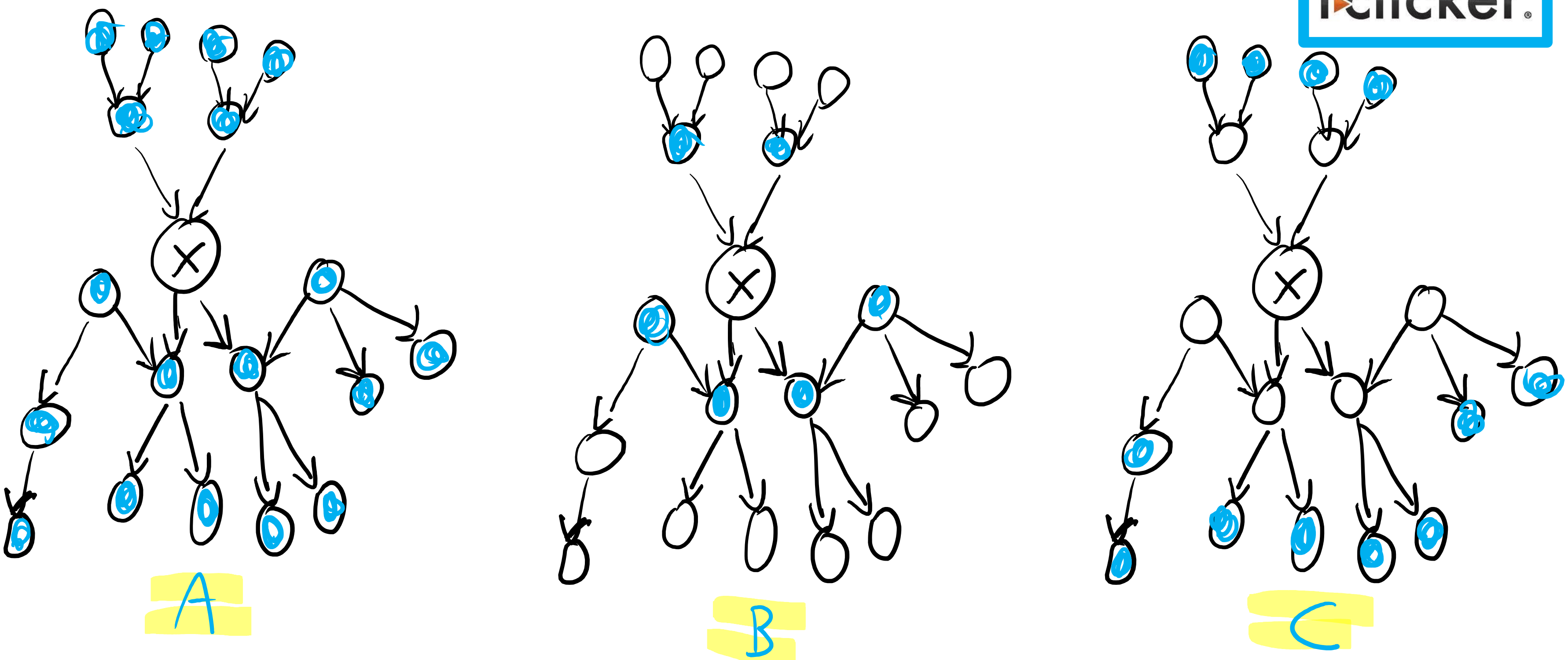
Properties: Now samples are not independent (in fact they're nearly identical), but sample averages are still consistent estimators! And can be computed efficiently

What's the point: when you sample a variable conditioned on all the rest, both upstream and downstream variables condition on evidence.



Open issue: what does it mean to sample a variable conditioned on all the rest ?

Sample for X is conditioned on all the rest

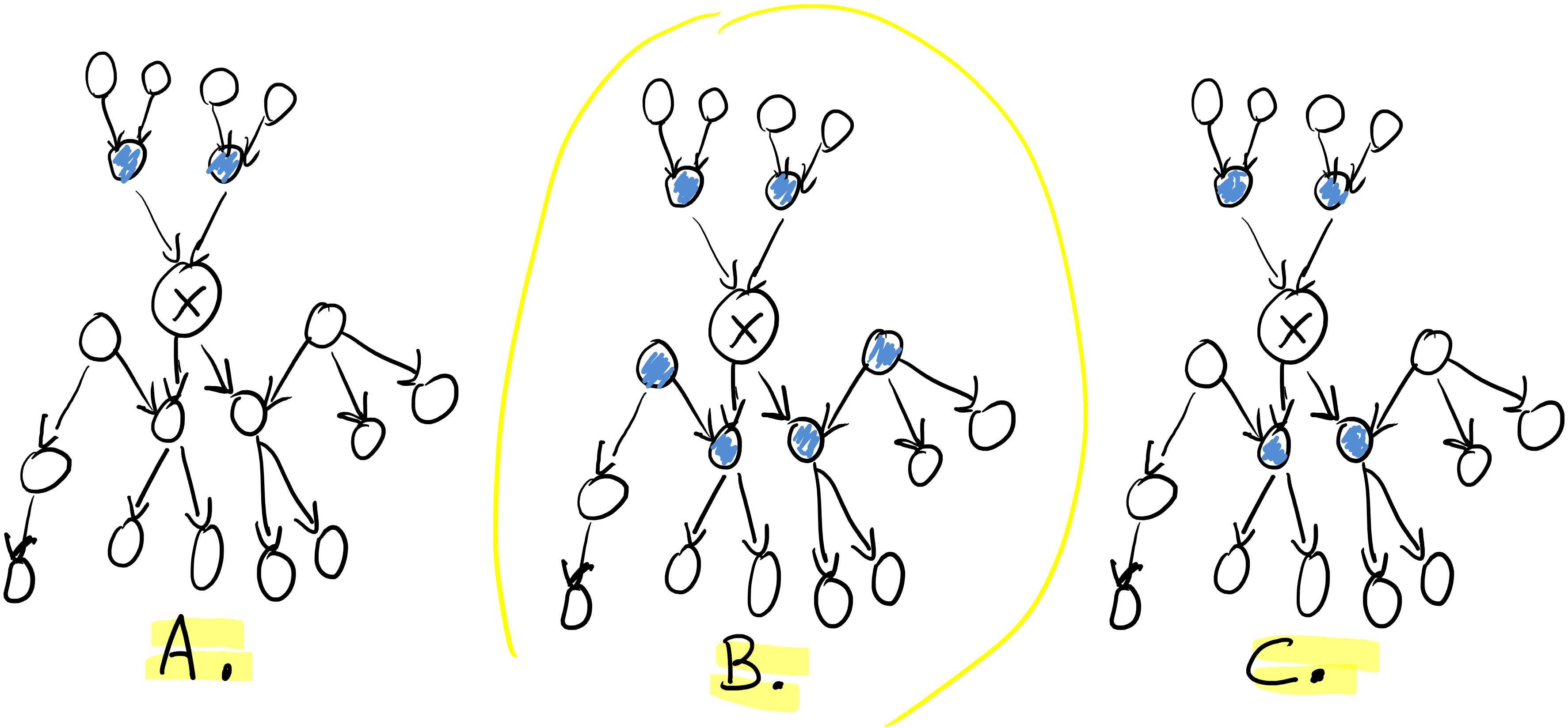


A. I need to consider all the other nodes

B. I only need to consider its Markov Blanket

C. I only need to consider all the nodes not in the Markov Blanket

Sample conditioned on all the rest



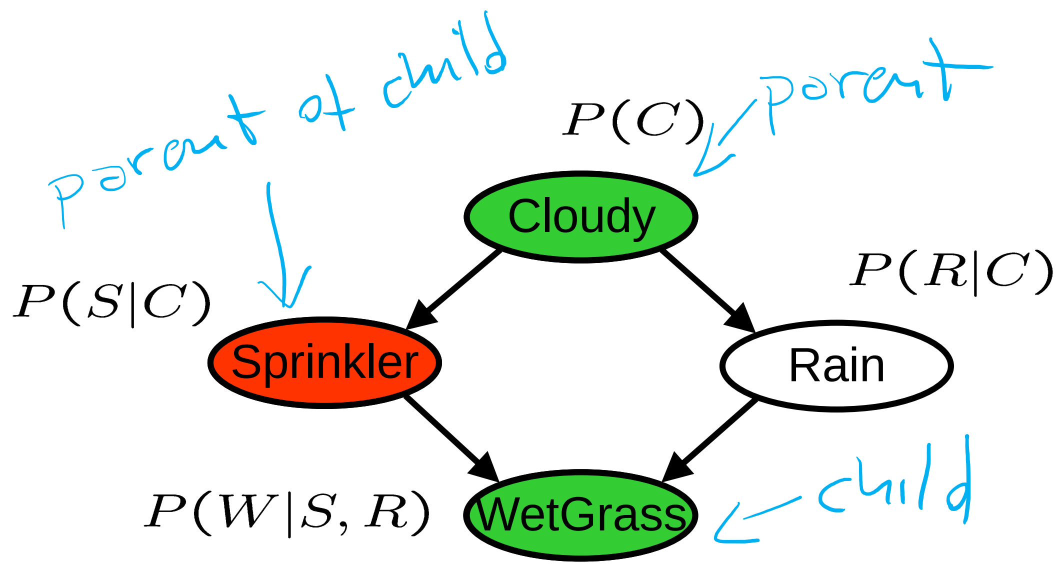
A node is conditionally independent from all the other nodes in the network, given its parents, children, and children's parents (i.e., its **Markov Blanket**) Configuration B

Probability given the Markov blanket is calculated as follows:

$$P(x_i'|mb(X_i)) = \alpha P(x_i'|parents(X_i))\Pi_{Z_j \in Children(X_i)} P(z_j|parents(Z_j))$$

We want to sample Rain

Rain's Markov Blanket is ?



$$P(r|c^+, s^-, w^+) = \alpha P(r|c^+)\, P(w^+|r, s^-)$$

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

Note: need to sample a different prob. Distribution for each configuration of the markov blanket

$$P(r \mid c^+, s^-, w^+) = \alpha P(r \mid c^+)\, P(w^+ \mid r, s^-)$$

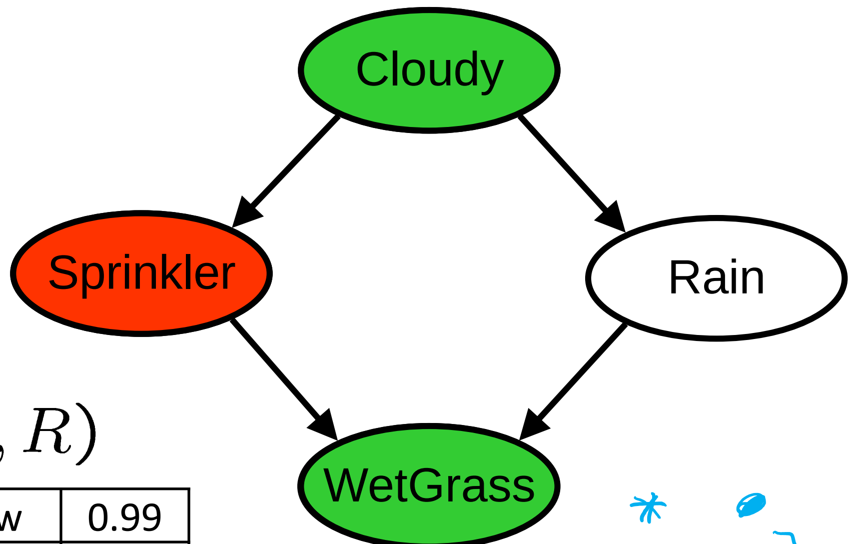
We want to sample Rain

$P(C)$

+c	0.5
-c	0.5

$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5



$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S, R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

$$= \alpha [.8, .2] \cdot [.9, .01]$$

$$= \alpha [.72, .002] = [.997, .003]$$

sample this

MCMC Example

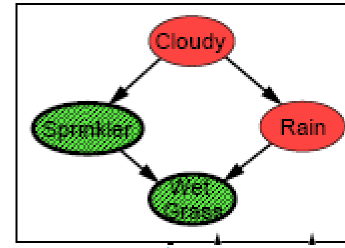
Estimate $\mathbf{P}(Rain|Sprinkler = true, WetGrass = true)$

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

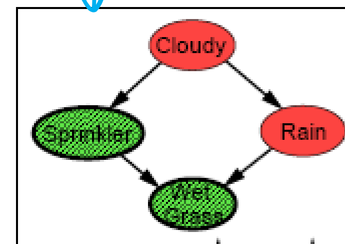
E.g., Do it 100 times
31 have $Rain = true$, 69 have $Rain = false$

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

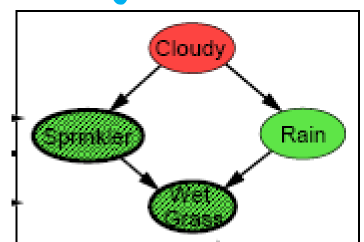
Full samples
$-c\ +r$
$+c\ +r$
$-c\ -r$
⋮



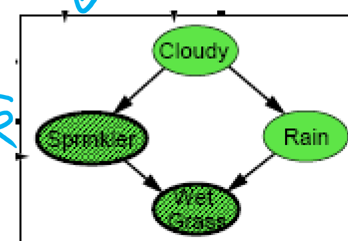
sample C $-c$



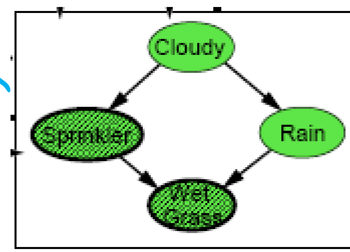
sample R $+r$



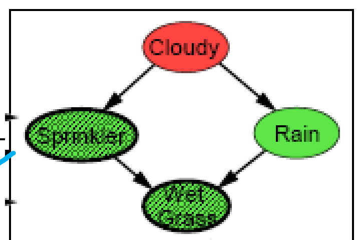
sample C $+c$



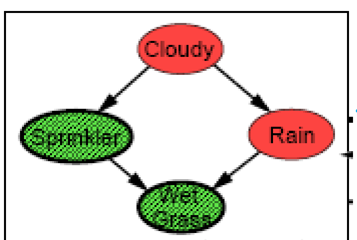
sample R $+r$



sample C $-c$



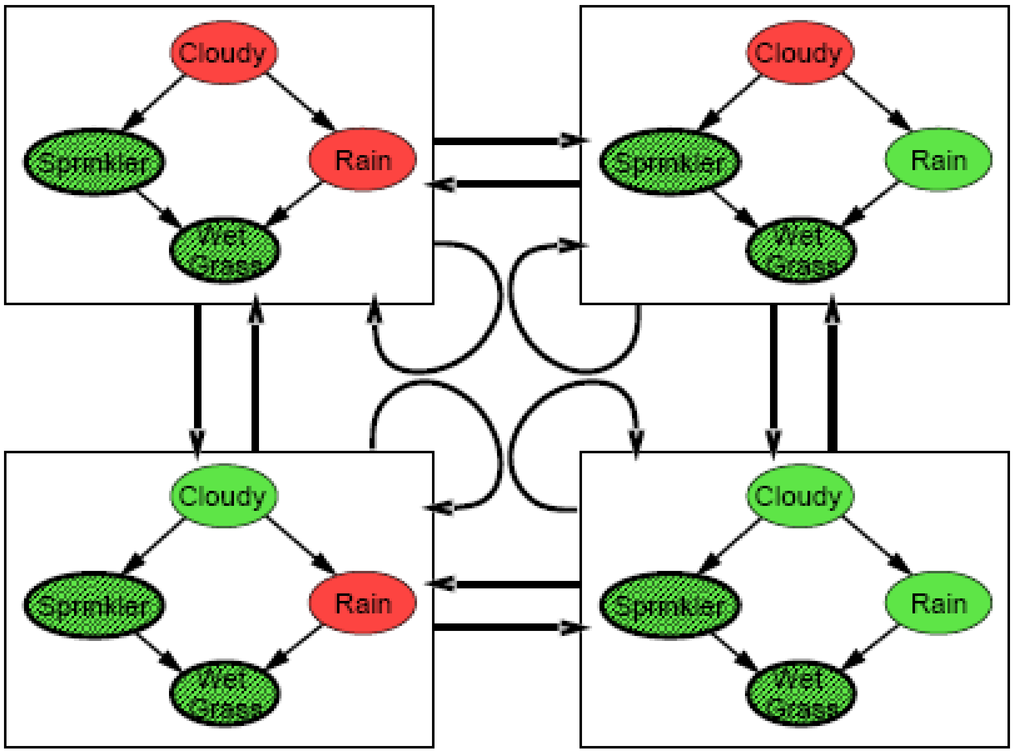
sample R $-r$



...

Why it is called Markov Chain MC

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



States of the chain are possible samples (fully instantiated Bnet)

Wander about for a while, average what you see

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

Hoeffding's inequality

- Suppose *p* is the true probability and *s* is the sample average from *n* independent samples.

$$P(|s-p|>\varepsilon) \le 2e^{-2n\varepsilon^2} < \delta$$

- *p* above can be the probability of any event for random variable $X = \{X_1, \ldots X_n\}$ described by a Bayesian network

- If you want an infinitely small probability of having an error greater than ε, you need infinitely many samples
- But if you settle on something less than infinitely small, let's say δ, then you just need to set

$$2e^{-2n\varepsilon^2} < \delta$$

- So you pick
  - the error ε you can tolerate,
  - the frequency δ with which you can tolerate it
- And solve for *n*, i.e., the number of samples that can ensure this performance

$$n > \frac{-\ln\frac{\delta}{2}}{2\varepsilon^2} \qquad (1)$$

Hoeffding's inequality

- Examples:
  - You can tolerate an error greater than 0.1 only in 5% of your cases
  - Set ε =0.1, δ = 0.05
  - Equation (1) gives you n > 184

$$n > \frac{-\ln \frac{\delta}{2}}{2\varepsilon^2} \qquad (1)$$

can rewrite as

$$n > \frac{\ln \frac{2}{\delta}}{2\varepsilon^2}$$

- If you can tolerate the same error (0.1) only in 1% of the cases, then you need 265 samples

- If you want an error greater than 0.01 in no more than 5% of the cases, you need 18,445 samples

so it should be clear that

$\varepsilon \downarrow$ $\quad n \uparrow$

$\delta \downarrow$

$\downarrow$ goes down

$\uparrow$ goes up

Learning Goals for today's class

➢ **You can:**

- Describe and justify the Likelihood Weighting sampling method
- Describe and justify Markov Chain Monte Carlo sampling method
- Describe and Apply Hoeffding's inequality

TODO for Mon

Next research paper: Using Bayesian Networks to Manage Uncertainty in Student Modeling. *Journal of User Modeling and User-Adapted Interaction* **2002** Dynamic BN *(required only up to page 400, do not have to answer the question "How was the system evaluated?)*

Very influential paper 500+ citations

•**Follow instructions on course WebPage**

<Readings>

- Start working on assignment-2 (due on Mon, Mar 1)

Not Required for 422

Proof for the formula computing the probability of a node given its markov blanket (which is the one you sample in Gibbs)

a. There are several ways to prove this. Probably the simplest is to work directly from the global semantics. First, we rewrite the required probability in terms of the full joint:

$$P(x_i|x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n) = \frac{P(x_1,\ldots,x_n)}{P(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n)}$$

$$= \frac{P(x_1,\ldots,x_n)}{\sum_{x_i} P(x_1,\ldots,x_n)}$$

$$= \frac{\prod_{j=1}^{n} P(x_j|parentsX_j)}{\sum_{x_i}\prod_{j=1}^{n} P(x_j|parentsX_j)}$$

Now, all terms in the product in the denominator that do not contain x_i can be moved outside the summation, and then cancel with the corresponding terms in the numerator. This just leaves us with the terms that do mention x_i, i.e., those in which X_i is a child or a parent. Hence, $P(x_i|x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n)$ is equal to

$$\frac{P(x_i|parentsX_i)\prod_{Y_j\in Children(X_i)} P(y_j|parents(Y_j))}{\sum_{x_i} P(x_i|parentsX_i)\prod_{Y_j\in Children(X_i)} P(y_j|parents(Y_j))}$$

Now, by reversing the argument in part (b), we obtain the desired result.