COL333/671: Introduction to AI

Semester I, 2024-25

Miscellaneous Topics: Sampling

Rohan Paul

Acknowledgement

These slides are intended for teaching purposes only. Some material has been used/adapted from web sources and from slides by Doina Precup, Dorsa Sadigh, Percy Liang, Mausam, Parag, Emma Brunskill, Alexander Amini, Dan Klein, Anca Dragan, Nicholas Roy and others.

Answering Probabilistic Queries Fast: *Approximate Inference*

- **Exact Inference**
  - Inference by enumeration (Variable elimination)
  - Exact likelihood for probabilistic queries.
    - Exact Marginal likelihood P(Late = Yes)
    - Exact Conditional likelihood (posterior probability)
      - P(Late = True | Rain = Yes, Traffic = High) etc.
  - Problem:
    - In many practical applications variable elimination can be intractable. Variable elimination may need to create a large table.

- **Approximate Inference**
  - Compute an "approximate" posterior probability
  - Principle
    - Generate samples from the distribution.
    - Use the samples to construct an approximate estimate of the probabilistic query. P'(Late) or P'(Late| Rain, Traffic).
  - Advantage
    - Generating samples and constructing the approximate distribution is often faster.
    - Note that the estimate is approximate, not exact.

Methods

- Prior Sampling
- Rejection Sampling
- Gibbs Sampling

How to sample from a distribution?

- Sampling from given distribution
  - Step 1: Get sample *u* from uniform distribution over [0, 1)
  - Step 2: Convert this sample *u* into an outcome for the given distribution by having each outcome associated with a sub-interval of [0,1) with sub-interval size equal to probability of the outcome
- Utility? We should be able to sample from a CPT defining the probabilistic model.
- Next, we look at approaches for sampling from a Bayes Net.

Example:

C	P(C)
red	0.6
green	0.1
blue	0.3

$$0 \leq u < 0.6, \rightarrow C = red$$
$$0.6 \leq u < 0.7, \rightarrow C = green$$
$$0.7 \leq u < 1, \rightarrow C = blue$$

- If random() returns *u* = 0.83, then our sample is *C* = blue
- E.g, after sampling 8 times:

Prior Sampling

Sampling from an empty network (without evidence).
Called “prior sampling” or “ancestral sampling”

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

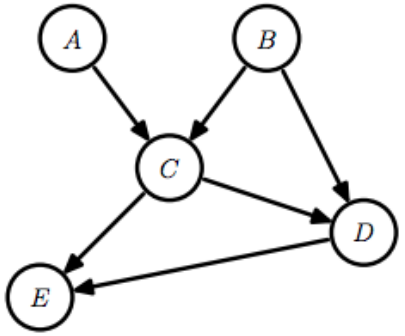
$\mathbf{x} \leftarrow$ an event with D elements
for $j = 1, \ldots, D$ **do**
  $\mathbf{x}[j] \leftarrow$ a random sample from $\mathbf{P}(X_j \mid$ values of $parents(X_j)$ in $\mathbf{x})$
return $\mathbf{x}$

- For i=1, 2, …, n
  - **Sample x_i from $P(X_i \mid Parents(X_i))$**
- Return (x_1, x_2, …, x_n)

Ancestral pass for directed graphical models:

— sample each top level variable from its marginal

— sample each other node from its conditional once its parents have been sampled



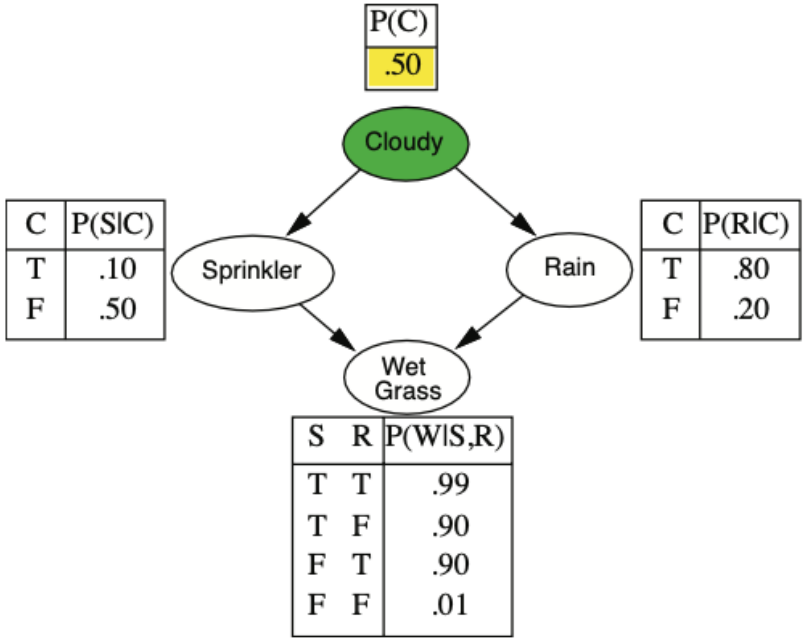
Sample:

$A \sim P(A)$
$B \sim P(B)$
$C \sim P(C|A,B)$
$D \sim P(D|B,C)$
$E \sim P(D|C,D)$

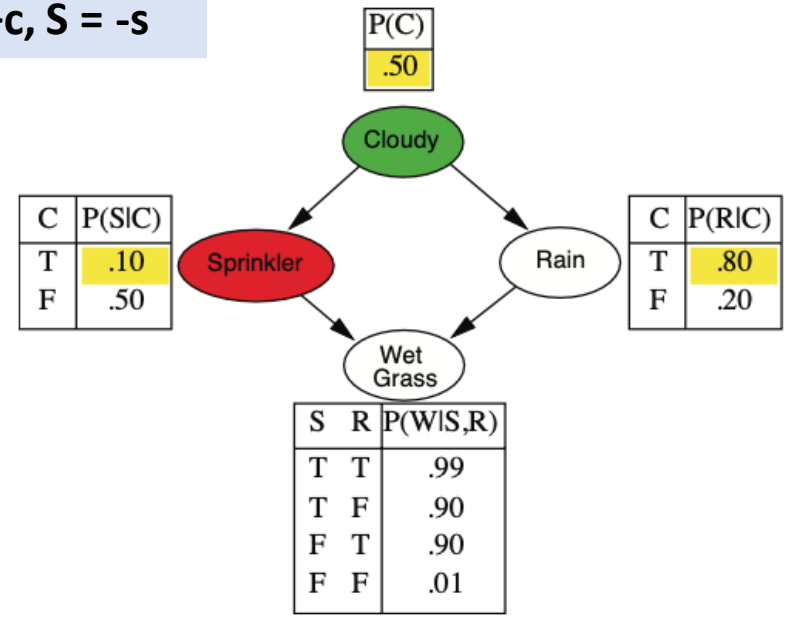
$$P(A,B,C,D,E) = P(A)\,P(B)\,P(C|A,B)\,P(D|B,C)\,P(E|C,D)$$

Source MLSS 2009. http://videolectures.net/mlss09uk_murray_mcmc/

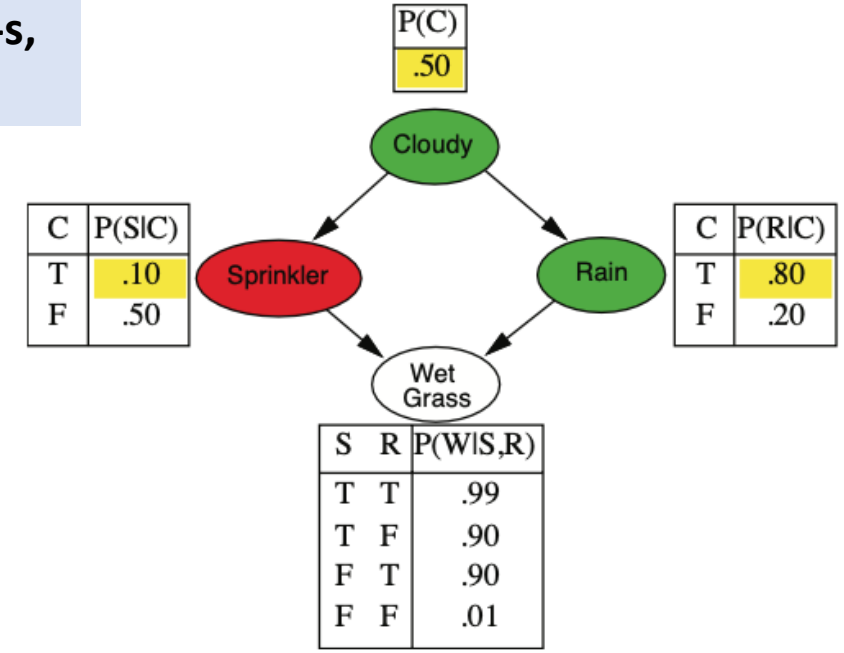
C = +c



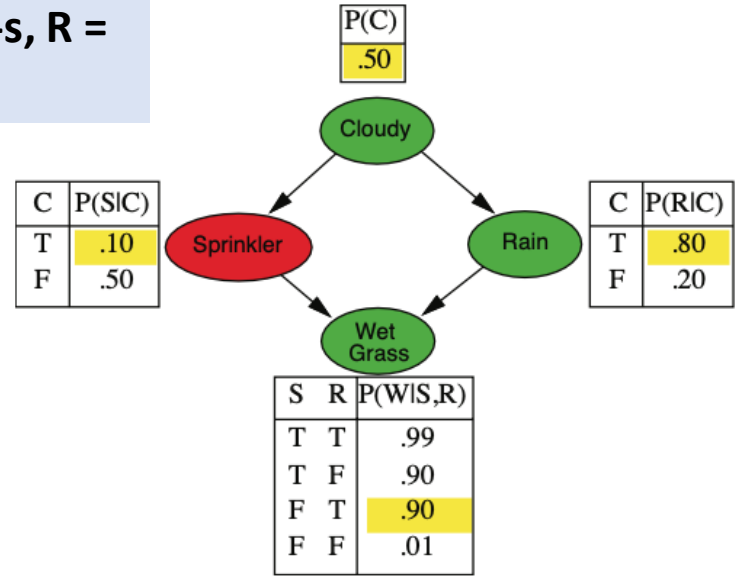
C = +c, S = -s



C = +c, S = -s, R = +r



C = +c, S = -s, R = +r, W = +w

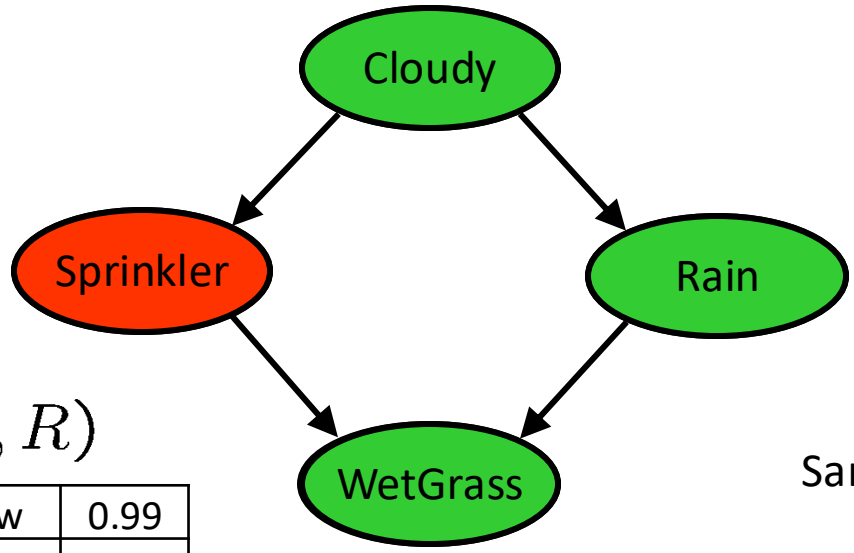


Prior Sampling

- For i=1, 2, …, n
  - **Sample x_i from $P(X_i \mid Parents(X_i))$**
- Return $(x_1, x_2, \ldots, x_n)$

$P(C)$

+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:

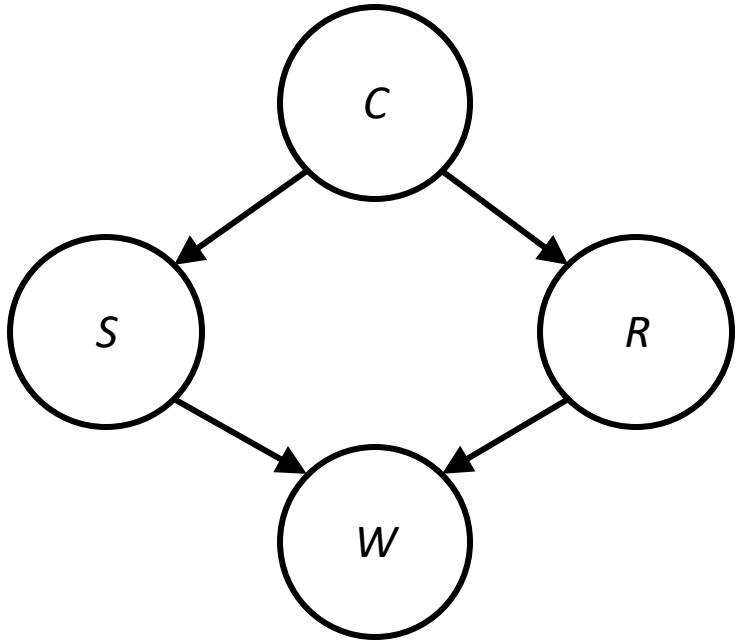
+c, -s, +r, +w

-c, +s, -r, +w

…

Approximate Probabilistic Queries with Samples

- Potential samples from the Bayes Net:
  - +c, -s, +r, +w
  - +c, +s, +r, +w
  - -c, +s, +r, -w
  - +c, -s, +r, +w
  - -c, -s, -r, +w
- What can we do with these samples?
  - Can empirically estimate the probabilistic queries
- Estimating P(W)
  - We have counts <+w:4, -w:1>
  - Normalize to get P(W) = <+w:0.8, -w:0.2>
- Can estimate other probabilistic queries as well:
  - P(C| +w)? P(C| +r, +w)? P(C| -r, -w)?
  - Note: if some evidence is not observed then we cannot estimate it

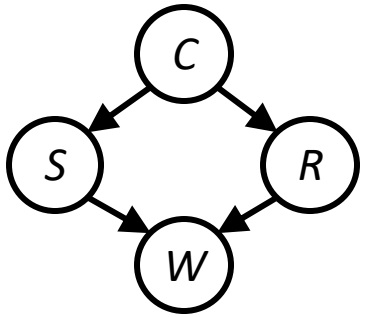


Problem: Prior sampling is unaware of the types of probabilistic queries that will be asked later?

Can we be more efficient if we knew the queries from the Bayes Net?

Rejection Sampling

- IN: evidence instantiation
- For i=1, 2, …, n
  - Sample x_i from $P(X_i \mid \text{Parents}(X_i))$
  - **If x_i not consistent with evidence**
    - **Reject: Return, and no sample is generated in this cycle**
- Return $(x_1, x_2, \ldots, x_n)$



Rejection Sampling

- Estimate P(C| +s)
- Tally the C outcomes, but reject samples which do not have S=+s.
- As you are sampling successively from the conditional probabilities, return if you find a sample inconsistent with the instantiated variables.

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

Rejection Sampling

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

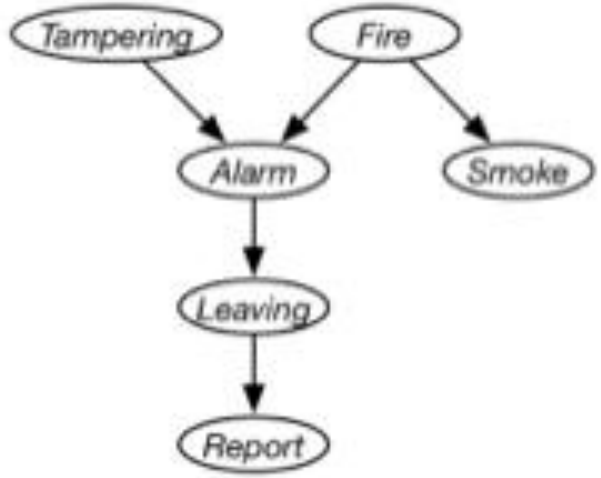
function REJECTION-SAMPLING($X, \mathbf{e}, bn, N$) **returns** an estimate of $\mathbf{P}(X|\mathbf{e})$
  local variables: $\mathbf{N}$, a vector of counts for each value of X, initially zero

  for $i = 1$ 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($\mathbf{N}$)

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 $Rain = false$.

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

We can reject earlier, say, if there are 1000 variables and at the third variable we detect inconsistency with evidence, we can reject the entire sample.



In this P (fire) = 0.01, P (smoke | fire) = 0.9 *and* P (smoke | ¬fire) = 0.01. *Suppose* Smoke = true *is observed, and another descendant of* Fire *is queried.*

Starting with 1000 samples, approximately 10 will have Fire = true, *and the other 990 samples will have* Fire = false.

In rejection sampling, of the 990 with Fire = false, *1%, which is approximately 10, will have* Smoke = true *and so will not be rejected. The remaining 980 samples will be rejected. Of the 10 with* Fire = true, *about 9 will not be rejected.*

Thus about 98% of the samples are rejected.

Rejection sampling may be rejecting a large number of samples!

Gibbs Sampling – Ensuring that each sample gets used

- Procedure:
  - Track of a full instantiation $x_1, x_2, \ldots, x_n$.
  - Start with an arbitrary instantiation consistent with the evidence.
  - Sample one variable at a time, conditioned on all the rest, but keep evidence fixed.
  - Keep repeating this for a long time.
  - After repeating you get *one* sample from the distribution.
  - To get more samples: start again.
  - *Note: this is like local search.*

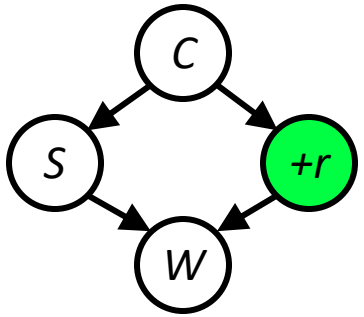
- Property: in the limit of repeating this infinitely many times the resulting sample is coming from the correct distribution.

Gibbs Sampling: Example

Estimating P(S | +r)

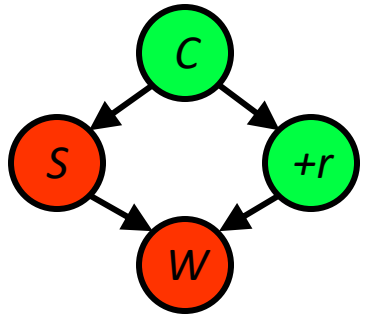
Step 1: Fix evidence

- R = +r



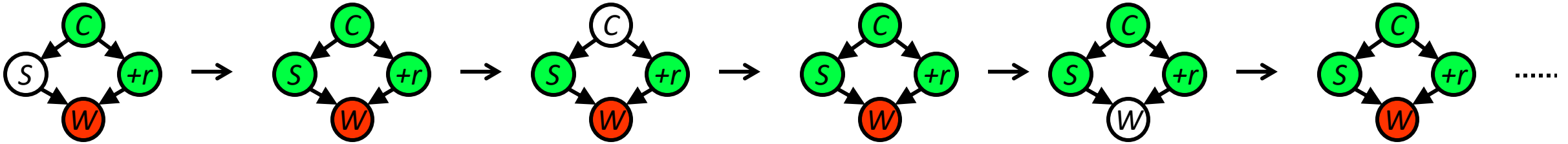
Step 2: Initialize other variables

- Randomly



Steps 3: Repeat

- Randomly select a non-evidence variable X
- Resample X from P(X | all other variables)

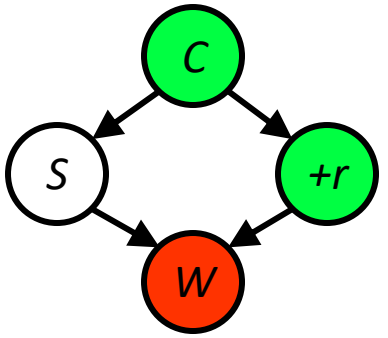


Sample from $P(S|+c,-w,+r)$ Sample from $P(C|+s,-w,+r)$ Sample from $P(W|+s,+c,+r)$

Sampling from the conditional

- Sample from P(S | +c, +r, -w)

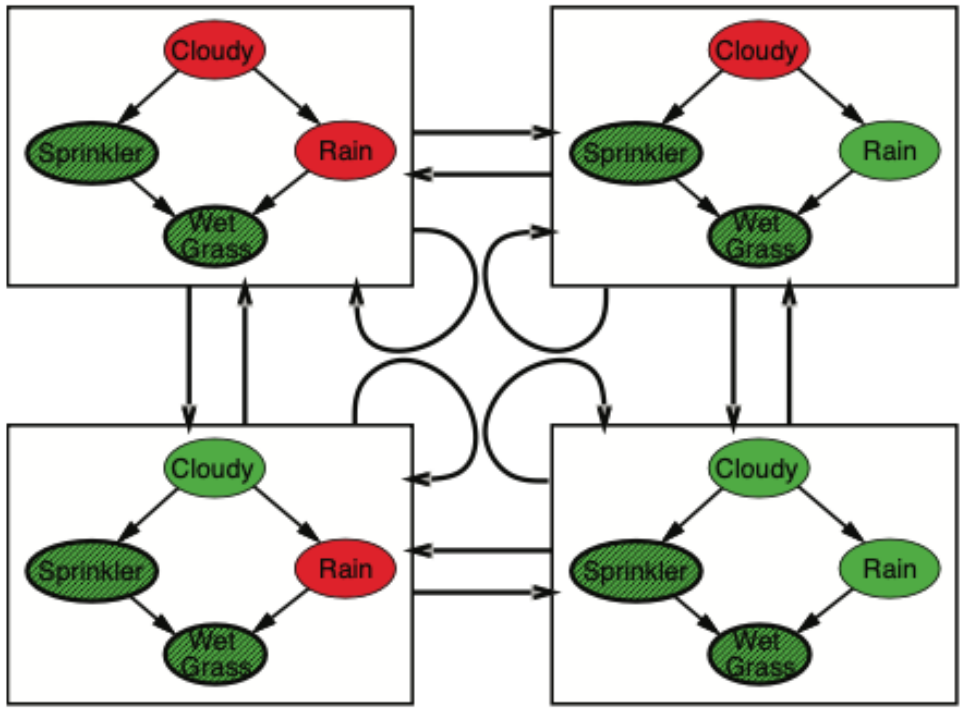
$$
\begin{aligned}
P(S|+c,+r,-w) &= \frac{P(S,+c,+r,-w)}{P(+c,+r,-w)} \\
&= \frac{P(S,+c,+r,-w)}{\sum_s P(s,+c,+r,-w)} \\
&= \frac{P(+c)P(S|+c)P(+r|+c)P(-w|S,+r)}{\sum_s P(+c)P(s|+c)P(+r|+c)P(-w|s,+r)} \\
&= \frac{P(+c)P(S|+c)P(+r|+c)P(-w|S,+r)}{P(+c)P(+r|+c)\sum_s P(s|+c)P(-w|s,+r)} \\
&= \frac{P(S|+c)P(-w|S,+r)}{\sum_s P(s|+c)P(-w|s,+r)}
\end{aligned}
$$



Sampling from the conditional distribution is needed as a sub-routine for Gibbs sampling. It is typically easier to sample from. The expression is simpler due to instantiated variables, can even construct the probability table if needed.

The Markov Chain

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



Gibbs sampling iteratively moves in the state space.

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 $Rain = false$

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

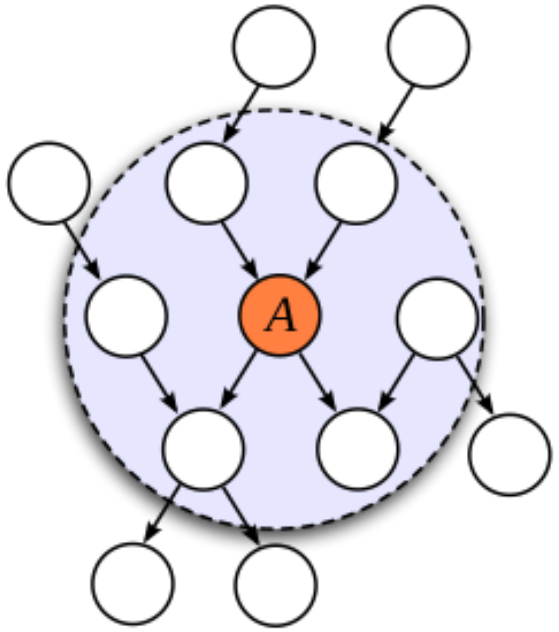
Collect samples from the sequence (say after every k-steps).

Then use the samples to form an estimate.

Demo: https://www.cs.cmu.edu/~./15281/demos/gibbsDemo/

Markov Blanket

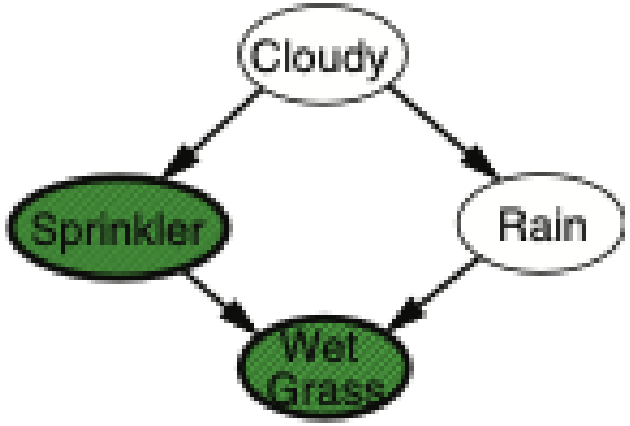
The Markov boundary of a node A in a Bayesian Network is the set of nodes composed of A's parents, A's children, and A's children's other parents.



Gibbs Sampling- it is enough to sample the conditional distribution from the Markov blanket variables.

Example:

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

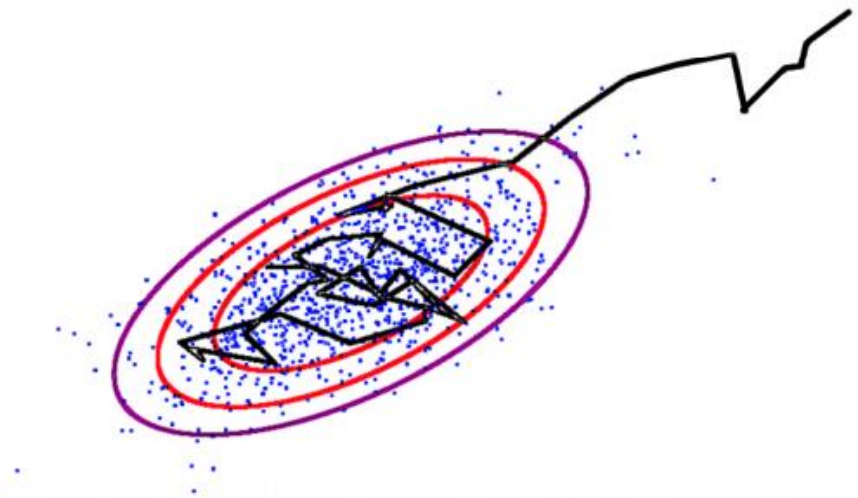


Markov Chain Monte Carlo – General Idea

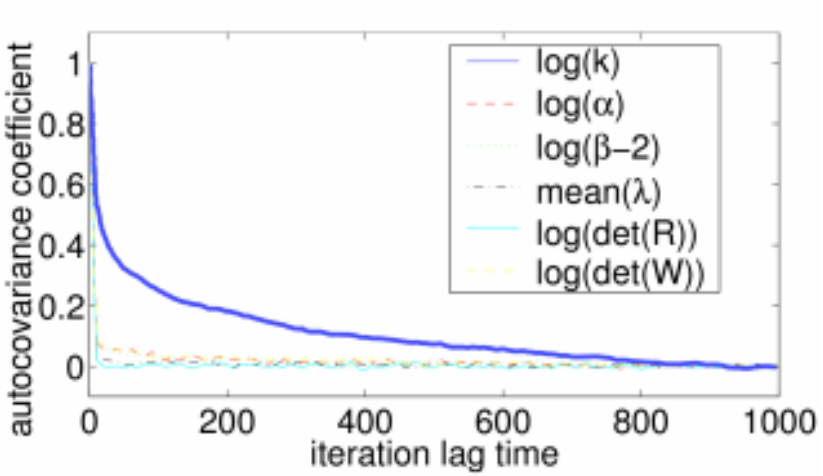
MCMC is a general technique for obtaining samples from distributions (also applied to continuous distributions).

Construct a biased random walk that explores target dist $P^{\star}(x)$

Markov steps, $x_t \sim T(x_t \leftarrow x_{t-1})$



MCMC gives approximate, correlated samples from $P^{\star}(x)$



Usually, there is a burn in period after which one accepts the samples.

Demo: https://chi-feng.github.io/mcmc-demo/app.html

Gibbs Sampling with Markov Blanket Sampling

Markov blanket property: $P(X_j \mid \text{all other variables}) = P(X_j \mid mb(X_j))$
so generate next state by sampling a variable given its Markov blanket

function GIBBS-ASK(X, **e**, bn, N) **returns** an estimate of $\mathbf{P}(X|\mathbf{e})$
  local variables: $\mathbf{N}$, a vector of counts for each value of X, initially zero
    $\mathbf{Z}$, the nonevidence variables in bn
    $\mathbf{z}$, the current state of variables $\mathbf{Z}$, initially random
  for $i = 1$ to N **do**
    choose Z_j in $\mathbf{Z}$ uniformly at random
    set the value of Z_j in $\mathbf{z}$ by sampling from $\mathbf{P}(Z_j|mb(Z_j))$
    $\mathbf{N}[x] \leftarrow \mathbf{N}[x] + 1$ where x is the value of X in $\mathbf{z}$
  return NORMALIZE($\mathbf{N}$)

Probability given the Markov blanket is calculated as follows:

$$P(x'_j|mb(X_j)) = \alpha P(x'_j|parents(X_j))\Pi_{Z_\ell \in Children(X_j)} P(z_\ell|parents(Z_\ell))$$

Note: during Gibbs sampling, one can condition on the values of a smaller set of variables (mb).