AIFA: APPROXIMATE INFERENCE

18/03/2025

Koustav Rudra

Sampling

- Sampling from given distribution
  - Step 1: Get sample u from uniform distribution over [0, 1)
    - e.g. random() in python
  - 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

C	P(C)
red	0.6
green	0.1
blue	0.3

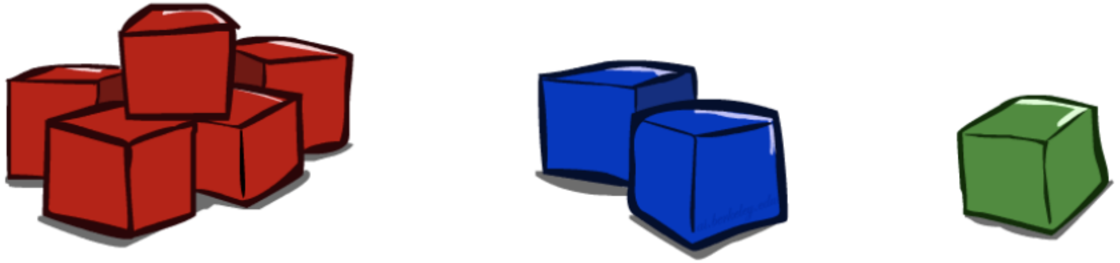
$0 \leq u < 0.6 \rightarrow C = red$

$0.6 \leq u < 0.7 \rightarrow C = blue$

$0.7 \leq u < 1 \rightarrow C = red$

If random() returns u=0.83
Our sample is C = blue

- E.g, after sampling 8 times:



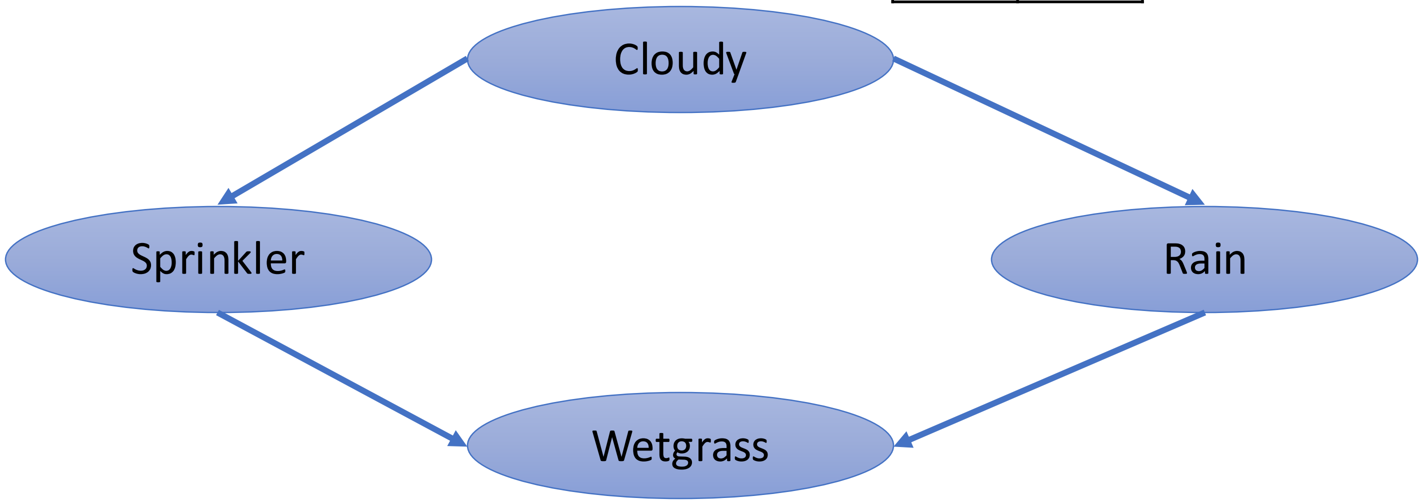
Sampling strategies

- Prior sampling
- Rejection sampling
- Likelihood weighting
- Gibbs sampling

Prior Sampling

P(C)	
+c	0.5

P(S|C)	
+c	0.1
-c	0.5



P(R|C)	
+c	0.8
-c	0.2

P(W|S,R)		
+s	+r	0.99
+s	-r	0.90
-s	+r	0.90
-s	-r	0.01

Samples:
+c, -s, +r, +w
-c, +s, -r, +w
…

Prior Sampling

- For i=1,2,…,n
    - Sample xi from P(Xi | Parents(Xi))
- Return (x1, x2, …, xn)

Prior Sampling

- This process generates samples with probability:

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

- …i.e. the BN's joint probability

- Let the number of samples of an event be $N_{PS}(x_1, x_2, \ldots, x_n)$

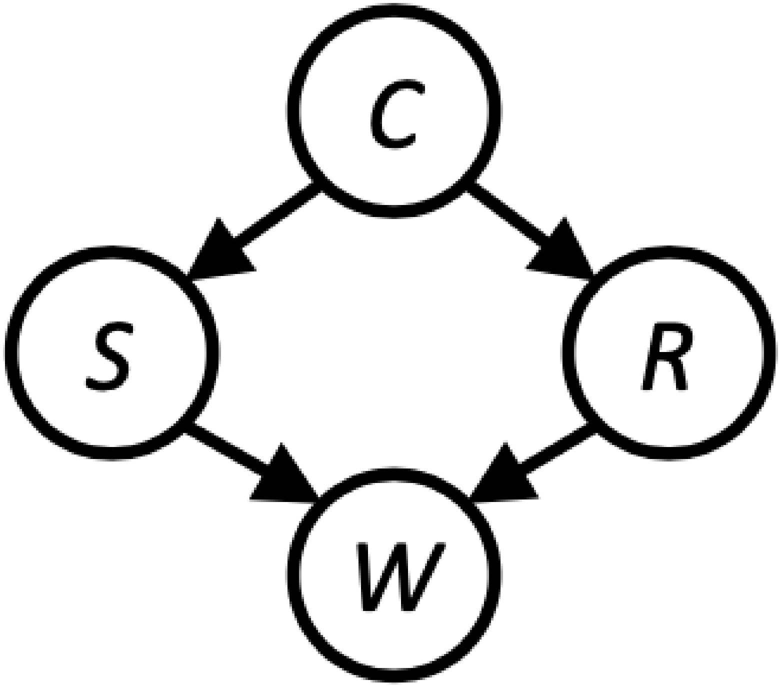
- **Then**

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

- the sampling procedure is consistent

Prior Sampling

- 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



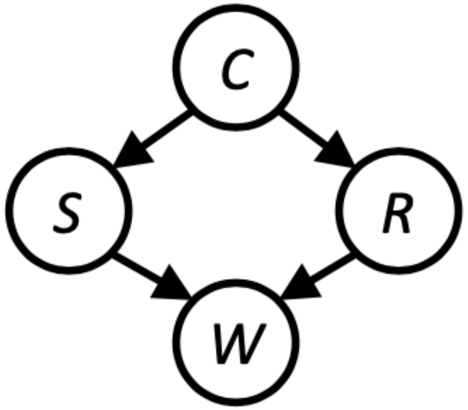
- If we want to know P(W)
- We have counts <+w:4, -w:1>
- Normalize to get P(W) = <+w:0.8, -w:0.2>
- This will get closer to the true distribution with more samples
- Can estimate anything else, too
- What about P(C| +w)? P(C| +r, +w)? P(C| -r, -w)?
- Fast: can use fewer samples if less time (what’s the drawback?)

Rejection Sampling

- Let’s say we want P(C)
  - No point keeping all samples around
  - Just tally counts of C as we go

- Let’s say we want P(C| +s)
  - Same thing: tally C outcomes, but 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)



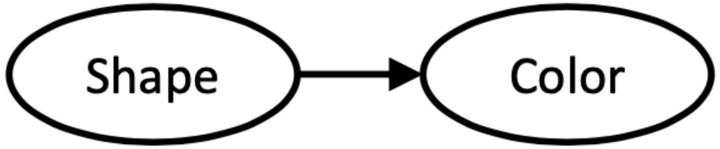
+c, -s, +r, +w
+c, +s, +r, +w
-c, +s, +r, -w
+c, -s, +r, +w
-c, -s, -r, +w

Rejection Sampling

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

Likelihood Weighting

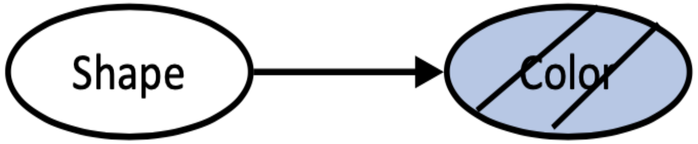
- Problem with rejection sampling:
  - If evidence is unlikely, rejects lots of samples
  - Evidence not exploited as you sample
  - Consider P(Shape|blue)



~~pyramid, green~~
~~pyramid, red~~
sphere, blue
~~cube, red~~
~~sphere, green~~

Idea: fix evidence variables and sample the rest

- Problem: sample distribution not consistent!
- Solution: weight by probability of evidence given parents



pyramid, blue
pyramid, blue
sphere, blue
cube, blue
sphere, blue

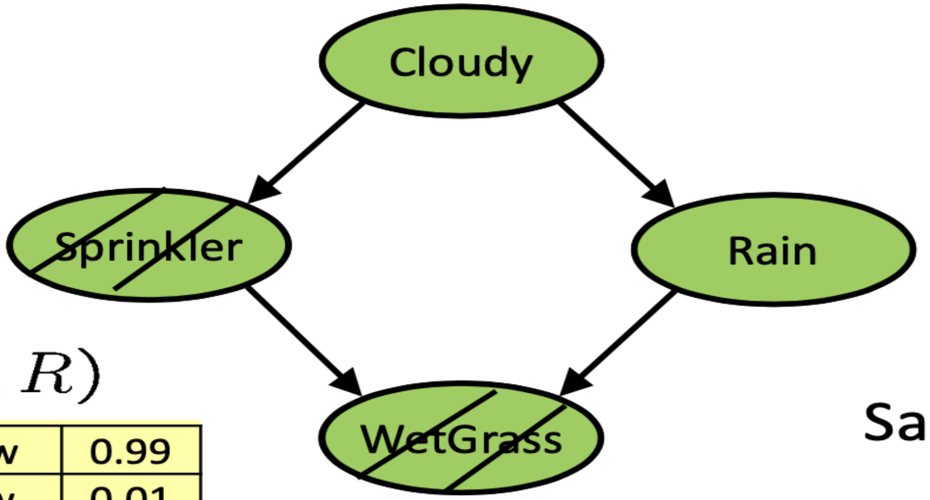
Likelihood Weighting

$P(C)$

+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

Samples:

+c, +s, +r, +w

…

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

P(Rain|Sprinkler=True, WetGrass=True)

Likelihood Weighting

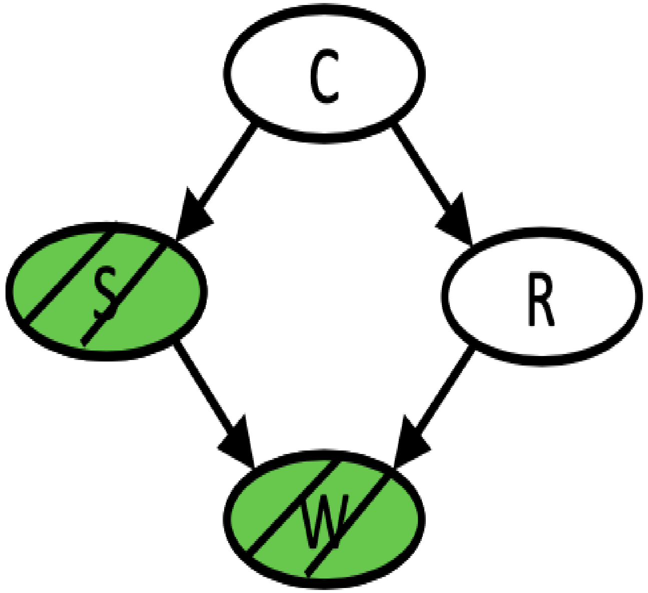
- IN: evidence instantiation
- w = 1.0
- for i=1, 2, …, n
  - if X_i is an evidence variable
    - X_i = observation x_i for X_i
    - Set w = w * $P(x_i \mid Parents(X_i))$
  - else
    - Sample x_i from $P(X_i \mid Parents(X_i))$
- return $(x_1, x_2, \ldots, x_n)$, w

Likelihood Weighting

- Sampling distribution if z sampled and e fixed evidence
  - $S_{WS}(z,e) = \prod_{i=1}^{l} P(z_i|Parents(z_i))$

- Now, samples have weights
  - $w(z,e) = \prod_{i=1}^{m} P(e_i|Parents(e_i))$

- Together, weighted sampling distribution is consistent
  - $S_{WS}(z,e)w(z,e) = \prod_{i=1}^{l} P(z_i|Parents(z_i)) \prod_{i=1}^{m} P(e_i|Parents(e_i))$
  - $S_{WS}(z,e)w(z,e) = P(z,e)$



Likelihood Weighting

- Likelihood weighting is good
  - We have taken evidence into account as we generate the sample
  - E.g. here, W's value will get picked based on the evidence values of S, R
  - More of our samples will reflect the state of the world suggested by the evidence

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)
- We would like to consider evidence when we sample every variable
  - Gibbs sampling

Gibbs Sampling

- Procedure:
  - keep track of a full instantiation x1, x2, …, xn
  - 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

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

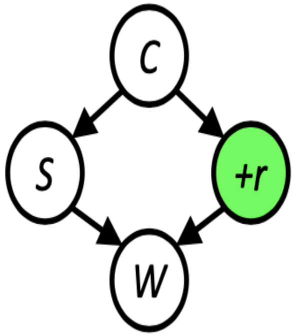
Gibbs Sampling

- Rationale:
  - Both upstream and downstream variables condition on evidence

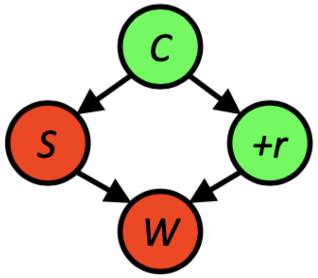
- In contrast:
  - likelihood weighting only conditions on upstream evidence, and hence weights obtained in likelihood weighting can sometimes be very small
  - Sum of weights over all samples is indicative of how many “effective” samples were obtained, so want high weight

Gibbs Sampling: P(s|+r)

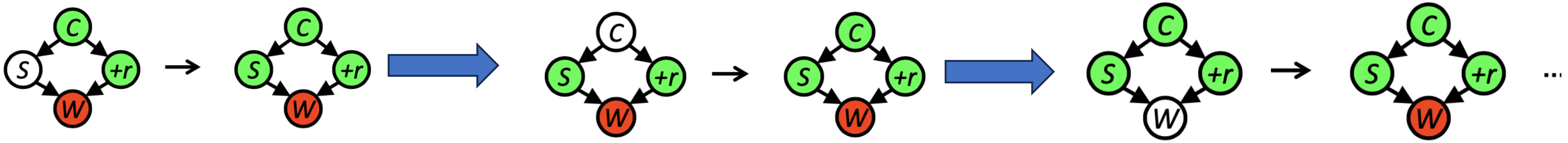
- Step 1: Fix evidence
  - R = +r



- Step 2: Initialize other variables
  - Randomly



- Steps 3: Repeat
  - Choose a non-evidence variable X
  - Resample X from P(X | all other variables)



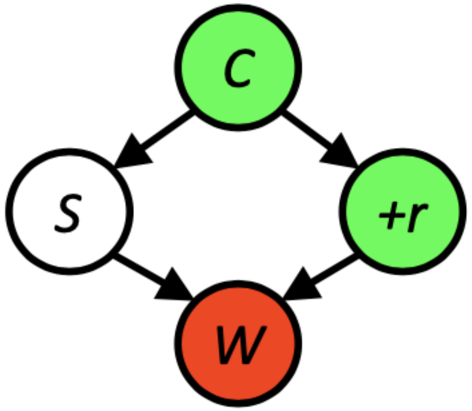
Sample from $P(S|+c,-w,+r)$

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

Sample from $P(W|+s,+c,+r)$

Efficient Resampling of One Variable

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



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

- Many things cancel out – only CPTs with S remain!
- More generally: only CPTs that have resampled variable need to be considered, and joined together

Gibbs Sampling

- Gibbs sampling produces sample from the query distribution P(Q | e) in limit of re-sampling infinitely often

- Gibbs sampling is a special case of more general methods called Markov chain Monte Carlo (MCMC) methods
    - Metropolis-Hastings is one of the more famous MCMC methods (in fact, Gibbs sampling is a special case of Metropolis-Hastings)

Approximate Inference

- Basic idea
  - If we had access to a set of examples from the joint distribution, we could just count
    - $E[f(x)] \approx \frac{1}{N} \sum_{i=1}^{N} f(x^{(i)})$

  - For inference, we generate instances from the joint and count
  - How do we generate instances?

Generating Instances

- Sampling from the Bayesian Network
  - Conditional probabilities i.e., P(X|E)
  - Only generate instances that are consistent with E

- Problems?
  - How many samples? [Law of large numbers]
  - What if the evidence EE is a very low probability event?

Markov Chain Monte Carlo

- Our goal: To sample from $P(X|e)$

- Overall idea:
  - The next sample is a function of the current sample
  - The samples can be thought of as coming from a Markov Chain whose stationary distribution is the distribution we want

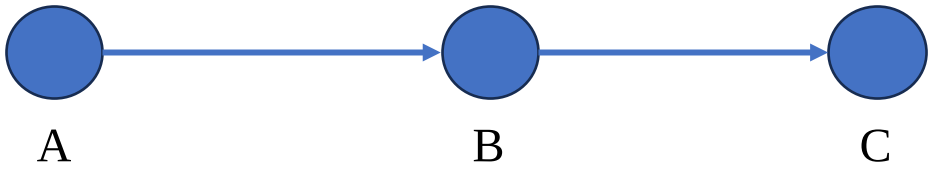
- Can approximate any distribution

Gibbs Sampling

- Algorithm:
  - Initialize X randomly
  - Iterate:
    - Pick a variable Xi uniformly at random
    - Sample $x_i^{(t+1)}$ from $P(x_i | x_1^{(t)}, \ldots, x_{i-1}^{(t)} x_{i+1}^{(t)}, \ldots, x_n^{(t)}, e)$
    - $X_k^{(t+1)} = x_k^{(t+1)}$ for all other k
    - This is the next sample

- Using the samples, we approximate the posterior by counting.

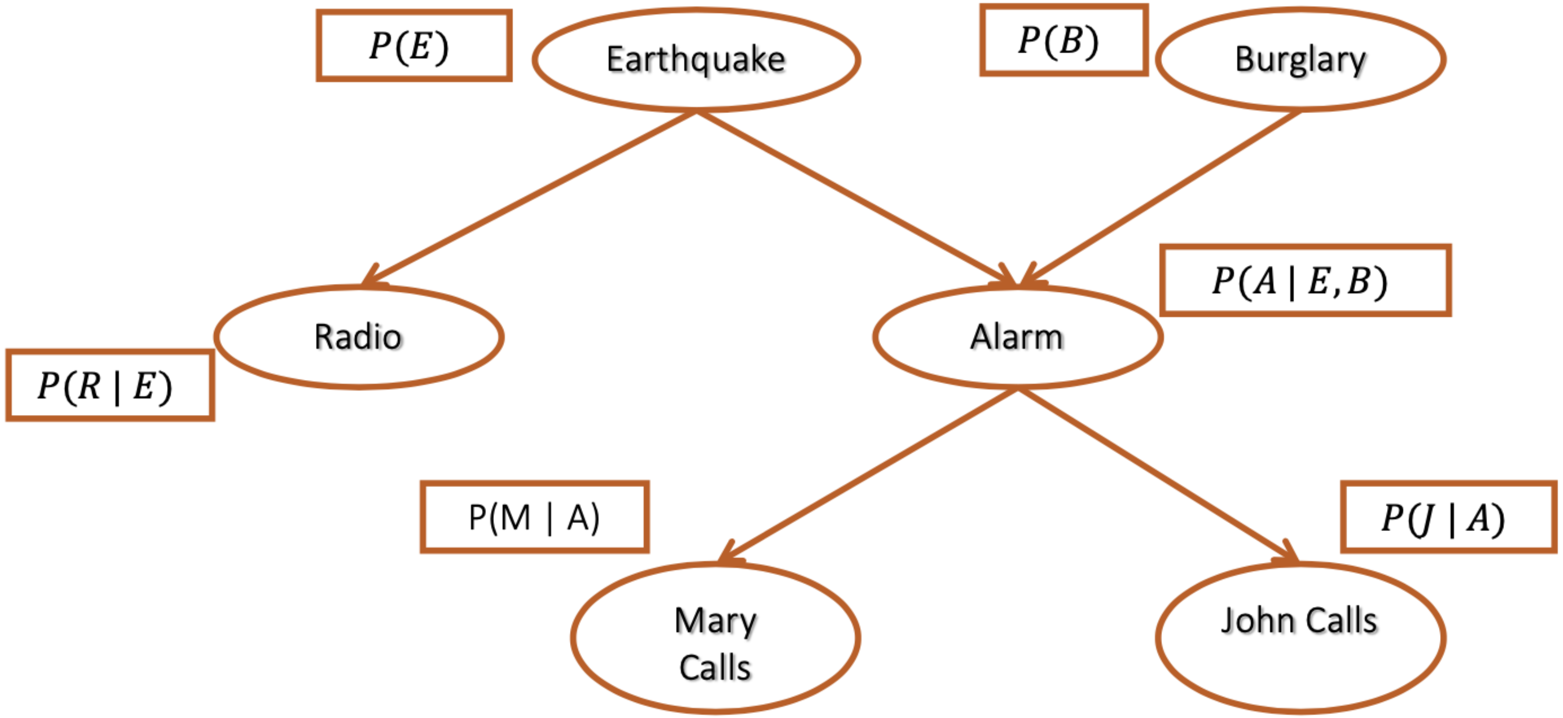
Gibbs Sampling: Example 1



- We want to compute $P(C)$:
  - Suppose, after burn in, the Markov Chain is at A=true, B = false, C= false

1. Pick a $variable \rightarrow B$
2. Draw the new value of B from
   1. P(B|A=true, C=false) = P(B|A=true)
   2. Suppose $B^{new} = true$
3. Our new sample is A = true, B=true, C=false
4. Repeat

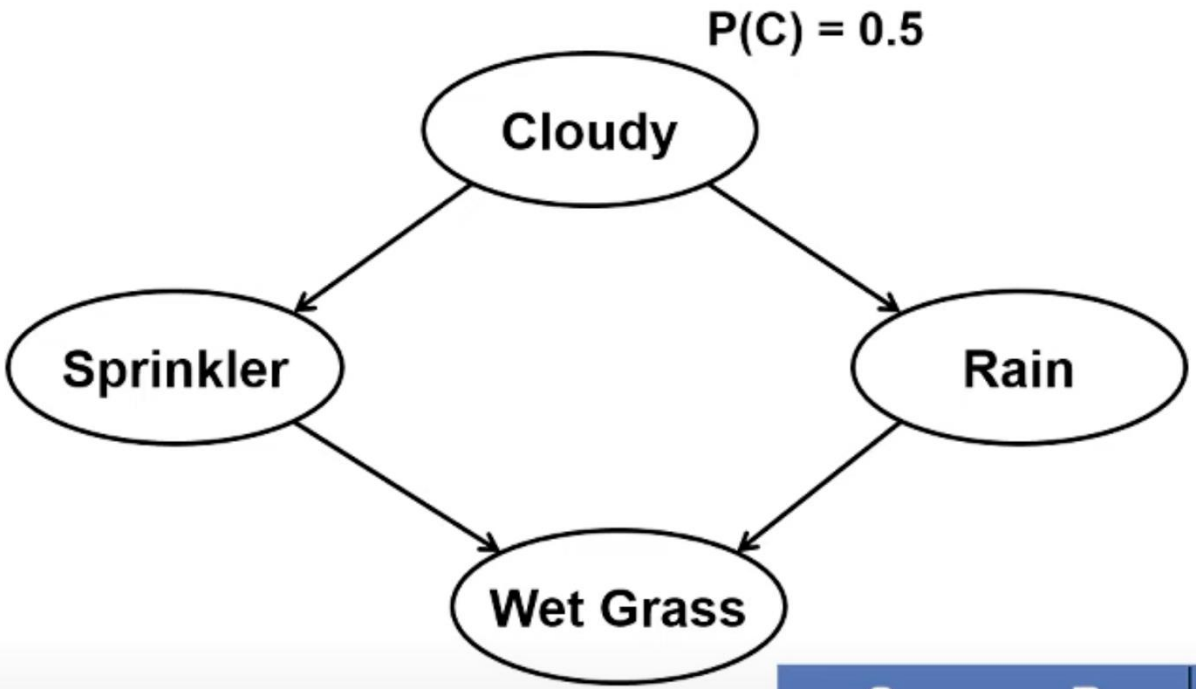
Gibbs Sampling: Example 2



Exercise: $P(M,J|B)$?

Challenges in Inference

Inference in multiply connected Belief Networks



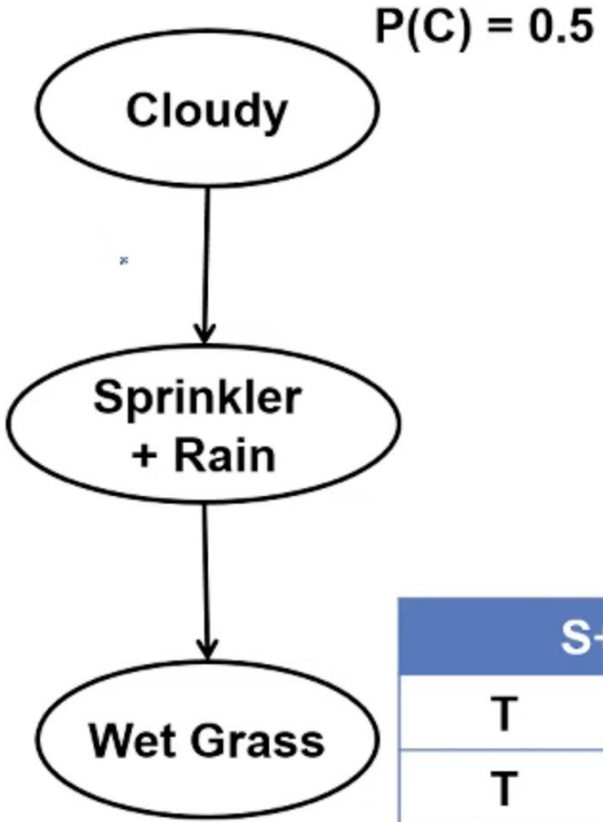
C	P(S)
T	0.10
F	0.50

C	P(R)
T	0.80
F	0.20

S	R	P(W)
T	T	0.99
T	F	0.90
F	T	0.90
F	F	0.00

Clustering Methods

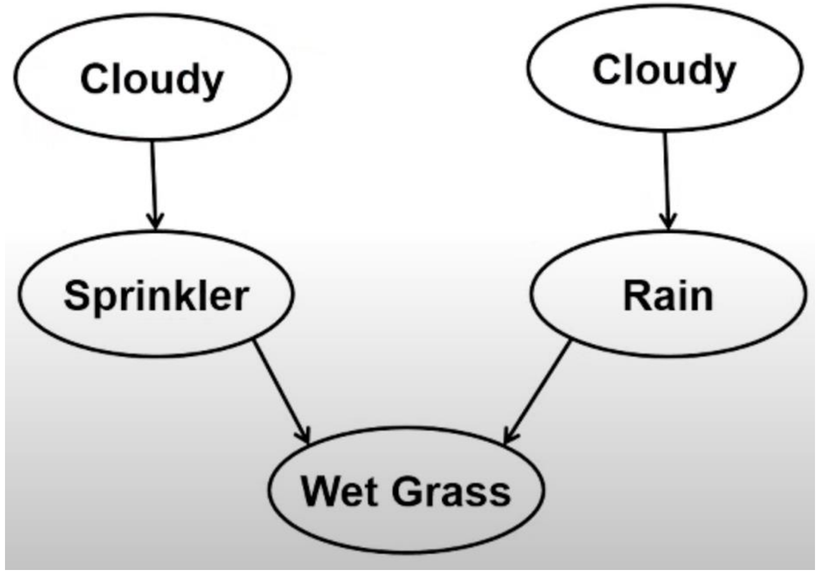
C	P(S+R = x) TT	TF	FT	FF
T	0.08	0.02	0.72	0.18
F	0.40	0.10	0.40	0.10

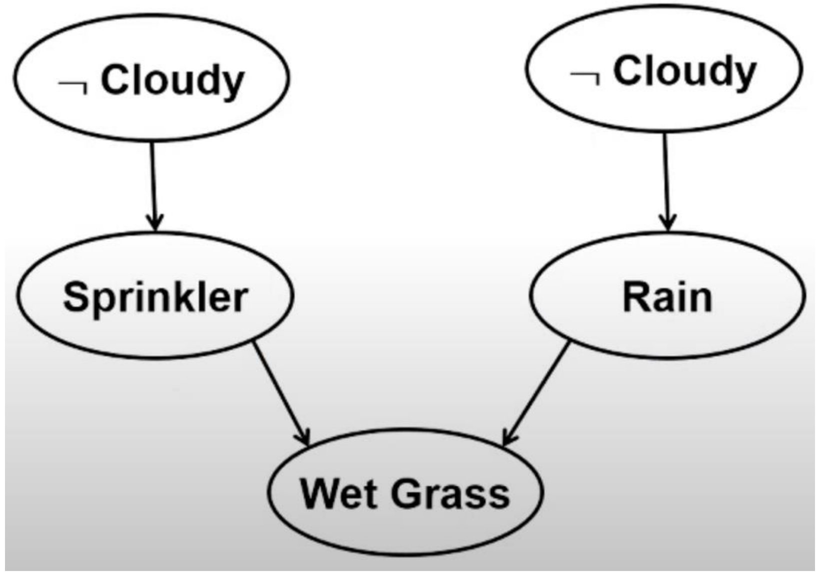


S+R		P(W)
T	T	0.99
T	F	0.90
F	T	0.90
F	F	0.00

Cutset Conditioning Method

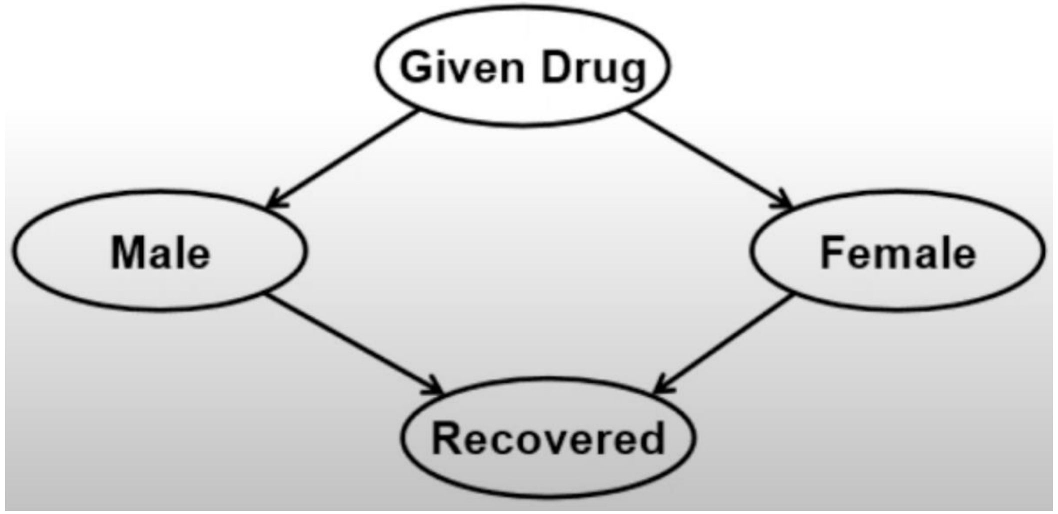
- A set of variables that can be instantiated to yield a poly-tree is called a cutset
- Instantiate the cutset variables to definite values
  - Then evaluate a poly-tree for each possible instantiation





Stochastic Simulation Methods

- Use the network to generate a large number of concrete models of the domain that are consistent with the network distribution
- They give an approximation of the exact evaluation
- Statistical bias can lead to misleading results – Simpson’s Paradox



Simpson's Paradox

Males	Recovered	Not Recovered	Rec. Rate
Given drug	18	12	60%
Not given drug	7	3	70%

Females	Recovered	Not Recovered	Rec. Rate
Given drug	2	8	20%
Not given drug	9	21	30%

Combined	Recovered	Not Recovered	Rec. Rate
Given drug	20	20	50%
Not given drug	16	24	40%

- Should the drug be administered or not?

Drug is administered on too few females

Simpson’s Paradox

Males	Recovered	Not Recovered	Rec. Rate
Given drug	18	12	60%
Not given drug	7	3	70%

Females	Recovered	Not Recovered	Rec. Rate
Given drug	2	8	20%
Not given drug	9	21	30%

Combined	Recovered	Not Recovered	Rec. Rate
Given drug	20	20	50%
Not given drug	16	24	40%

$$P(recovery|male \wedge given\_drug) = 0.6$$

$$P(recovery|given\_drug) = P(recovery|male \wedge given\_drug)P(given\_drug|male) + P(recovery|female \wedge given\_drug)P(given\_drug|female)$$

$$P(recovery|given\_drug) = \left(0.6 \times \frac{30}{40}\right) + \left(0.20 \times \frac{10}{40}\right) = 0.5$$

- Should the drug be administered or not?

Default Reasoning

- Some conclusions are made by default unless a counter-evidence is obtained
  - Non-monotonic reasoning

- Points to think:
  - What is the semantic status of default rules?
  - What happens when the evidence matches the premises of two default rules with conflicting conclusions?
  - If a belief is retracted later, how can a system keep track of which conclusions need to be retracted as a consequence?

Issues in Rule-based methods for Uncertain Reasoning

- Locality
  - In logical reasoning systems, if we have A=>B, then we can conclude B given evidence A, without worrying about any other rules
  - In probabilistic systems, we have to consider all available evidence
- Detachment
  - Once a logical proof is found for proposition B, we can use it regardless of how it is derived (it can be detached from its justification)
  - In probabilistic reasoning, the source of the evidence is important for subsequent reasoning

Issues in Rule-based methods for Uncertain Reasoning

- Truth functionality
  - In logic, the truth of the complex sentences can be computed from the truth of the components
  - Probability combination does not work this way, except under strong independence assumptions

- A famous example of a truth functional system for uncertain reasoning is the certainly factors model, developed for Mycin medical diagnostic problem

Dempster-Shafer Theory

- Designed to deal with the distinction between uncertainty and ignorance
- We use a belief function Bel(X) – probability that the evidence supports the proposition
- When we do not have any evidence about X, we assign Bel(X)=0 as well as Bel(~X)=0

- For example, if we do not know whether a coin is fair, then:
  - Bel(heads) = Bel(~heads) = 0
- If we are given that coin is fair with 90% certainty, then:
  - Bel(heads) = 0.9x0.5 = 0.45
  - Bel(~heads) = 0.9x0.5 = 0.45
  - We still have a gap of 0.10 that is not accounted for by the evidence

Thank You