UIT2504 Artificial Intelligence

Markov Chain Simulations in Bayesian Networks

C. Aravindan
<AravindanC@ssn.edu.in>

Professor of Information Technology
SSN College of Engineering

November 07, 2024

MCMC Algorithms

- *Monte Carlo Markov Chain (MCMC) algorithms* randomly modify the current state to generate a random next state; the initial state is fixed randomly

MCMC Algorithms

- *Monte Carlo Markov Chain (MCMC) algorithms* randomly modify the current state to generate a random next state; the initial state is fixed randomly
- Note that there is no need to sample the variables in topological order

MCMC Algorithms

- *Monte Carlo Markov Chain (MCMC) algorithms* randomly modify the current state to generate a random next state; the initial state is fixed randomly
- Note that there is no need to sample the variables in topological order
- Further, sampling of a variable need not depend upon its parents alone

MCMC Algorithms

- *Monte Carlo Markov Chain (MCMC) algorithms* randomly modify the current state to generate a random next state; the initial state is fixed randomly
- Note that there is no need to sample the variables in topological order
- Further, sampling of a variable need not depend upon its parents alone
- Since evidence variables are fixed, the state space is formed by nonevidence variables only; that means, one of the nonevidence variables is randomly sampled to generate the next state

MCMC Algorithms

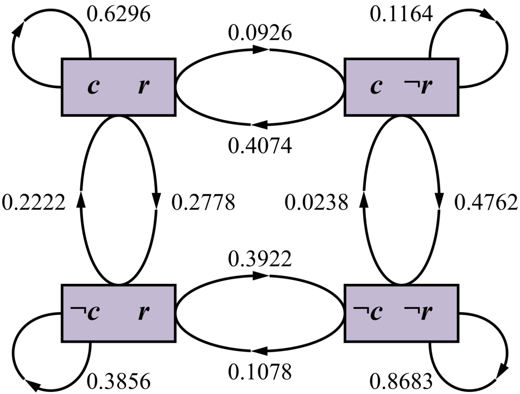
- *Monte Carlo Markov Chain (MCMC) algorithms* randomly modify the current state to generate a random next state; the initial state is fixed randomly
- Note that there is no need to sample the variables in topological order
- Further, sampling of a variable need not depend upon its parents alone
- Since evidence variables are fixed, the state space is formed by nonevidence variables only; that means, one of the nonevidence variables is randomly sampled to generate the next state
- In general, *Markov Chain* refers to a random process that generates a sequence of states

Markov Chain: Example

- In the sprinkler example, consider $Sprinkler = true$ and $WetGrass = true$ as evidences. A state space for Markov chain, where nonevidence variables $Cloudy$ and $Rain$ can be modified, is as shown below:

Markov Chain: Example

- In the sprinkler example, consider *Sprinkler* = *true* and *WetGrass* = *true* as evidences. A state space for Markov chain, where nonevidence variables *Cloudy* and *Rain* can be modified, is as shown below:



Gibbs Sampling

- Start with an arbitrary state (consistent with the evidence variables) and generate a next state by randomly sampling a value for one of the nonevidence variables X_i

Gibbs Sampling

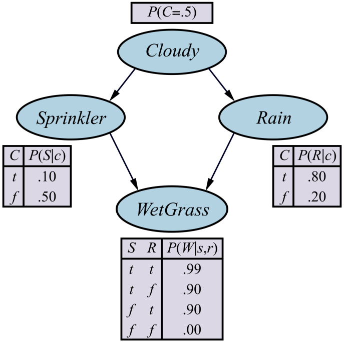
- Start with an arbitrary state (consistent with the evidence variables) and generate a next state by randomly sampling a value for one of the nonevidence variables X_i
- Gibbs sampling for X_i means sampling conditioned on the current values of the variables in its Markov blanket

Gibbs Sampling

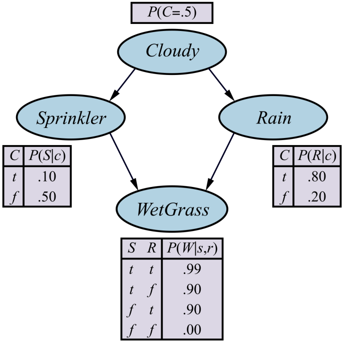
- Start with an arbitrary state (consistent with the evidence variables) and generate a next state by randomly sampling a value for one of the nonevidence variables X_i
- Gibbs sampling for X_i means sampling conditioned on the current values of the variables in its Markov blanket
- The algorithm wanders randomly around the state space flipping one variable at a time (but keeping the evidence variables fixed)

Gibbs Sampling: Example

- Consider the query $\mathbf{P}(R|s, w)$

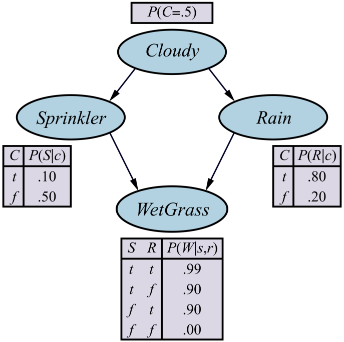


Gibbs Sampling: Example



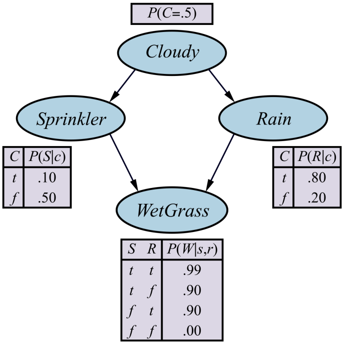
- Consider the query $\mathbf{P}(R|s,w)$
- Nonevidence variables C and R are randomly initialized to, say, *true* and *false*, giving the state $[true, \mathbf{true}, false, \mathbf{true}]$

Gibbs Sampling: Example



- Consider the query $\mathbf{P}(R|s, w)$
- Nonevidence variables C and R are randomly initialized to, say, *true* and *false*, giving the state $[true, \mathbf{true}, false, \mathbf{true}]$
- Choose a nonevidence variable randomly, say C. Markov blanket of C is S, R. So, sample from $\mathbf{P}(C|s, \neg r)$. Suppose we get $Cloudy = false$, the new state is $[false, \mathbf{true}, false, \mathbf{true}]$

Gibbs Sampling: Example



- Consider the query $\mathbf{P}(R|s, w)$
- Nonevidence variables C and R are randomly initialized to, say, *true* and *false*, giving the state $[true, \mathbf{true}, false, \mathbf{true}]$
- Choose a nonevidence variable randomly, say C. Markov blanket of C is S, R. So, sample from $\mathbf{P}(C|s, \neg r)$. Suppose we get $Cloudy = false$, the new state is $[false, \mathbf{true}, false, \mathbf{true}]$
- Suppose R is chosen next. Markov blanket of R is C, S, W. So, sample from $\mathbf{P}(R|\neg c, s, w)$. Suppose this yields $Rain = true$, the next state is

Gibbs Sampling Algorithm

function GIBBS-ASK($X, \mathbf{e}, bn, N$) **returns** an estimate of $\mathbf{P}(X \mid \mathbf{e})$
  local variables: $\mathbf{C}$, a vector of counts for each value of X, initially zero
    $\mathbf{Z}$, the nonevidence variables in bn
    $\mathbf{x}$, the current state of the network, initialized from $\mathbf{e}$

  initialize $\mathbf{x}$ with random values for the variables in $\mathbf{Z}$
  for $k = 1$ **to** N **do**
    choose any variable Z_i from $\mathbf{Z}$ according to any distribution $\rho(i)$
    set the value of Z_i in $\mathbf{x}$ by sampling from $\mathbf{P}(Z_i \mid mb(Z_i))$
    $\mathbf{C}[j] \leftarrow \mathbf{C}[j] + 1$ where x_j is the value of X in $\mathbf{x}$
  return NORMALIZE($\mathbf{C}$)

Gibbs Sampling

- How do we get the Markov blanket distribution, $\mathbf{P}(X_i|mb(X_i))$, of a nonevidence variable X_i?

Gibbs Sampling

- How do we get the Markov blanket distribution, $\mathbf{P}(X_i|mb(X_i))$, of a nonevidence variable X_i?

$$P(x_i|mb(X_i)) = \alpha P(x_i|parents(X_i)) \prod_{Y_j \in Children(X_i)} P(y_j|parents(Y_j))$$

Gibbs Sampling

- How do we get the Markov blanket distribution, $\mathbf{P}(X_i|mb(X_i))$, of a nonevidence variable X_i?

$$P(x_i|mb(X_i)) = \alpha P(x_i|parents(X_i)) \prod_{Y_j \in Children(X_i)} P(y_j|parents(Y_j))$$

- In our sprinkler example, We can calculate $\mathbf{P}(C|s, \neg r)$ as follows

$$P(c|s, \neg r) = \alpha P(c)P(s|c)P(\neg r|c) = \alpha \times 0.5 \times 0.1 \times 0.2$$
$$P(\neg c|s, \neg r) = \alpha P(\neg c)P(s|\neg c)P(\neg r|\neg c) = \alpha \times 0.5 \times 0.5 \times 0.8$$

Gibbs Sampling

- How do we get the Markov blanket distribution, $\mathbf{P}(X_i|mb(X_i))$, of a nonevidence variable X_i?

$$P(x_i|mb(X_i)) = \alpha P(x_i|parents(X_i)) \prod_{Y_j \in Children(X_i)} P(y_j|parents(Y_j))$$

- In our sprinkler example, We can calculate $\mathbf{P}(C|s,\neg r)$ as follows

$$P(c|s,\neg r) = \alpha P(c)P(s|c)P(\neg r|c) = \alpha \times 0.5 \times 0.1 \times 0.2$$
$$P(\neg c|s,\neg r) = \alpha P(\neg c)P(s|\neg c)P(\neg r|\neg c) = \alpha \times 0.5 \times 0.5 \times 0.8$$

- So, the sampling distribution is $\alpha\langle 0.001, 0.020\rangle \approx \langle 0.048, 0.952\rangle$

Does Gibbs sampling work?

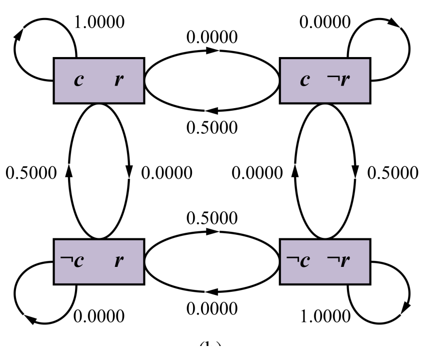
- Gibbs sampling work as intended (probability estimates converge to true probability values) when the chain is *ergodic* — that is, every state must be reachable from every other and there are no periodic cycles

Does Gibbs sampling work?

- Gibbs sampling work as intended (probability estimates converge to true probability values) when the chain is *ergodic* — that is, every state must be reachable from every other and there are no periodic cycles
- It does not work for cases as follows (where *Rain* is constrained to have the same value as *Cloudy*)

Does Gibbs sampling work?

- Gibbs sampling work as intended (probability estimates converge to true probability values) when the chain is *ergodic* — that is, every state must be reachable from every other and there are no periodic cycles
- It does not work for cases as follows (where *Rain* is constrained to have the same value as *Cloudy*)



Metropolis–Hastings Sampling

- We saw that Gibbs sampling may get stuck to a part of the Markov Chain
- This may be solved by occassional random "restarts" where next state is generated from scratch
- For example, next state generation may be as follows:
    - With probability 0.95, perform a Gibbs sampling step to generate the next state x'
    - Otherwise, generate x' by running likelihood-weighted sampling algorithm
- This is an example of a *Metropolis–Hastings* algorithm, where the next state x' is generated from the current state x from a poposal distribution (an example is shown above)
- The proposed next state may be accepted or rejected according to the acceptance probability (depends on the ratio $\frac{\pi(x')}{\pi(x)}$ which can be computed by $\frac{P(x',e)}{P(x,e)}$

Summary

- We have discussed how knowledge under uncertainty can be represented using Bayesian Networks

What Next?

- Read chapter 13 of the text book!