# Gibbs Sampling

Frederik Mallmann-Trenn
6CCS3AIN

# Gibbs sampling

- A rather different approach to sampling.
- Part of the Markov Chain Monte Carlo (MCMC) family of algorithms.
- Don’t generate each sample from scratch.
- Generate samples by making a random change to the previous sample.

# Gibbs sampling

- Gibbs sampling for Bayesian networks starts with an arbitrary state.
- So pick state with evidence variables fixed at observed values.
  (If we know $Cloudy = true$, we pick that.)
- Generate next state by randomly sampling from a non-evidence variable.
- Do this sampling conditional on the current values of the Markov blanket.
- "The algorithm therefore wanders randomly around the state space . . . flipping one variable at a time, but keeping the evidence variables fixed".

# Gibbs sampling

- Consider the query:

$$\mathbf{P}(Cloudy|Sprinker = true, WetGrass = true)$$

- The evidence variables are fixed to their observed values.
- The non-evidence variables are initialised randomly.

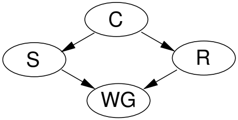
$$Cloudy = true$$
$$Rain = false$$

- State is thus:

$$[Cloudy = true, \color{blue}{Sprinkler = true},$$
$$Rain = false, \color{blue}{WetGrass = true}].$$

# Gibbs sampling

- First we sample $Cloudy$ given the current state of its Markov blanket.
- Markov blanket is $Sprinkler$ and $Rain$.
- So, sample from:



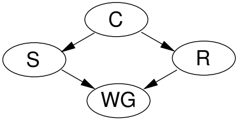

$$\mathbf{P}(Cloudy|Sprinkler = true, Rain = false)$$

- Suppose we get $Cloudy = false$, then new state is:

$$[Cloudy = false, Sprinkler = true,$$
$$Rain = false, WetGrass = true].$$

# Gibbs sampling

- Next we sample $Rain$ given the current state of its Markov blanket.
- Markov blanket is $Cloudy$, $Sprinkler$ and $WetGrass$.
- So, sample from:




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

- Suppose we get $Rain = true$, then new state is:

$$[Cloudy = false, Sprinkler = true, Rain = true, WetGrass = true].$$

# Gibbs sampling

- Each state visited during this process contributes to our estimate for:

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

- Say the process visits 80 states.
- In 20, $Cloudy = true$
- In 60, $Cloudy = false$
- Then

$$\mathbf{P}(Cloudy|Sprinkler = true, WetGrass = true) = \alpha \begin{pmatrix} 20 \\ 60 \end{pmatrix} = \begin{pmatrix} 0.25 \\ 0.75 \end{pmatrix}$$

# Gibbs sampling

**function** GIBBS-ASK($X, \mathbf{e}, bn, N$) **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 $\mathbf{Z}$
  **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

- All of this begs the question:

  "How do we sample a variable given the state of its Markov blanket?"
- For a value $x$ of a variable $X$:

$$\mathbf{P}(X|mb(X)) = \alpha\mathbf{P}(X|parents(X)) \prod_{Y \in Children(X)} \mathbf{P}(Y|parents(Y))$$

  where $mb(X)$ is the Markov blanket of $X$.
- Given $\mathbf{P}(X|mb(X))$, we can sample from it just as we have before.

# To summarise

- Bayesian networks exploit conditional independence to create a more compact set of information.
- Reasonably efficient computation for some problems.
- Five approaches to inference in Bayesian networks.
  - Exact: Inference by enumeration.
  - Approximate: Prior sampling
  - Approximate: Rejection sampling
  - Approximate: Importance sampling/likelihood weighting
  - Approximate: Gibbs sampling
- Can answer a simple query for any BN.