Lecture 24: 18 April, 2023

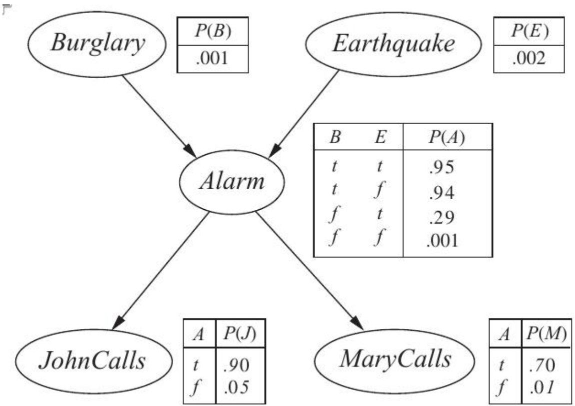
Madhavan Mukund

https://www.cmi.ac.in/~madhavan

Data Mining and Machine Learning
January–April 2023

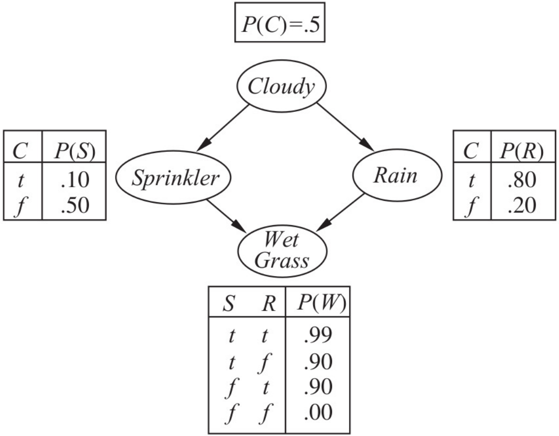
Approximate inference

- Exact inference is NP-complete
- Generate random samples, count to estimate probabilities
- Respect conditional probabilities — generate in topological order
- Suppose we are interested in $P(b \mid j, m)$
- Samples with $\neg j$ or $\neg m$ are useless
- Can we sample more efficiently?



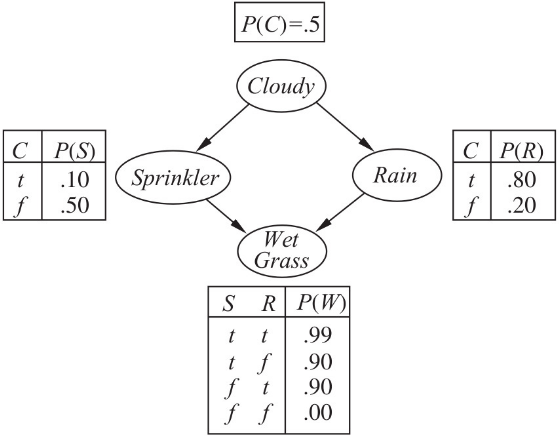
Rejection sampling

- $P(Rain \mid Cloudy, Wet\ Grass)$
- If we start with $\neg Cloudy$, sample is useless
- Immediately stop and reject this sample — rejection sampling
- General problem with low probability situation — many samples are rejected



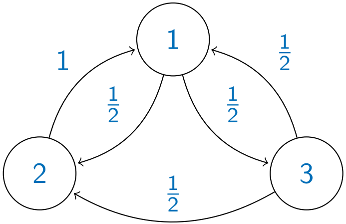
Likelihood weighted sampling

- $P(Rain \mid Cloudy, Wet\ Grass)$
- Fix evidence $Cloudy, Wet\ Grass$ true
- Then generate the other variables
- Compute likelihood of evidence
- Samples $s_1, s_2, \ldots, s_N$ with weights $w_1, w_2, \ldots w_N$
- $P(r \mid c, w) = \dfrac{\sum_{s_i\ has\ rain} w_i}{\sum_{1 \leq j \leq N} w_j}$



Markov chains

- Finite set of states, with transition probabilities between states
- For us, a state will be an assignment of values to variables
- A three state Markov Chain



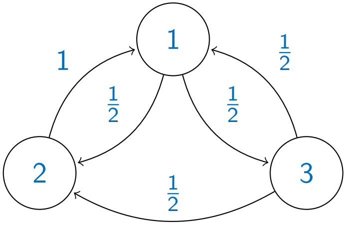
- Represent using a transition matrix — stochastic

$$A = \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix}$$

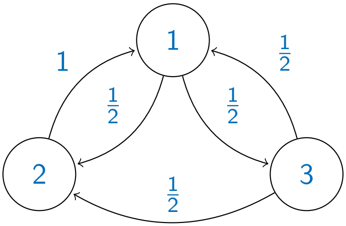
- $P[j]$ is probability of being in state j

Ergodicity

- Markov chain A is ergodic if there is some t_0 such that for every P, for all $t > t_0$, for every j, $(P^\top A^t)[j] > 0$.
- Ergodic Markov chain has a stationary distribution π^*, $(\pi^*)^\top A = \pi^*$
- For *any* starting distribution P, $\lim_{t \to \infty} P^\top A^t = \pi^*$
- Stationary distribution represents fraction of visits to each state in a long enough execution
- Sufficient conditions for ergodicity
  - Irreducible (strong connected)
  - Aperiodic (paths of all lengths between states)



- Bayesian network has variables $V_1, V_2, \ldots, V_n$
- Each assignment of values to the variables is a state
- Set up a Markov chain based on these states
- Stationary distribution should assign to state s the probability $P(s)$ in the Bayesian network
- How to reverse engineer the transition probabilities to achieve this?



Reversible Markov chains

- Ergodic Markov chain with stationary distribution π^*

Reversible Markov chains

- Ergodic Markov chain with stationary distribution $\pi*$
- Transition matrix A, write p_{jk} for $A[j][k]$
  - Probability of transition from state j to state k

Reversible Markov chains

- Ergodic Markov chain with stationary distribution $\pi*$
- Transition matrix A, write p_{jk} for $A[j][k]$
  - Probability of transition from state j to state k
- Reversibility — in steady state, probability of going from j to k should equal probability of going from k to j

Reversible Markov chains

- Ergodic Markov chain with stationary distribution $\pi*$
- Transition matrix A, write p_{jk} for $A[j][k]$
  - Probability of transition from state j to state k
- Reversibility — in steady state, probability of going from j to k should equal probability of going from k to j

$A[j][k] = p_{jk}$

- Given an evolution $x_1x_2\ldots$, for large n, $P[x_n = k \mid x_{n-1} = j] = P[x_{n-1} = j \mid x_n = k]$

$j \rightarrow k$ $\qquad$ $j \leftarrow k$

Reversible Markov chains

- Ergodic Markov chain with stationary distribution $\pi*$
- Transition matrix A, write p_{jk} for $A[j][k]$
  - Probability of transition from state j to state k
- Reversibility — in steady state, probability of going from j to k should equal probability of going from k to j
- Given an evolution $x_1 x_2 \ldots$, for large n, $P[x_n = k \mid x_{n-1} = j] = P[x_{n-1} = j \mid x_n = k]$
- $P[x_{n-1} = j \mid x_n = k] = P[x_n = k \mid x_{n-1} = j] \cdot \dfrac{P[x_{n-1} = j]}{P[x_n = k]}$

$\pi^*_j = \pi_j$

π_k

Reversible Markov chains

- Ergodic Markov chain with stationary distribution $\pi*$
- Transition matrix A, write p_{jk} for $A[j][k]$
  - Probability of transition from state j to state k
- Reversibility — in steady state, probability of going from j to k should equal probability of going from k to j
- Given an evolution $x_1 x_2 \ldots$, for large n, $P[x_n = k \mid \underbrace{x_{n-1} = j]}_{p_{jk}} = P[x_{n-1} = j \mid x_n = k]$
- $P[x_{n-1} = j \mid x_n = k] = P[x_n = k \mid x_{n-1} = j] \cdot \dfrac{\pi_j}{\pi_k}$, in steady state

π^k vs π

Reversible Markov chains

- Ergodic Markov chain with stationary distribution $\pi*$
- Transition matrix A, write p_{jk} for $A[j][k]$
  - Probability of transition from state j to state k
- Reversibility — in steady state, probability of going from j to k should equal probability of going from k to j
- Given an evolution $x_1 x_2 \ldots$, for large n, $P[x_n = k \mid x_{n-1} = j] = P[x_{n-1} = j \mid x_n = k]$
- $P[x_{n-1} = j \mid x_n = k] = P[x_n = k \mid x_{n-1} = j] \cdot \dfrac{\pi_j}{\pi_k}$, in steady state
- $p_{kj} = p_{jk} \dfrac{\pi_j}{\pi_k}$

Reversible Markov chains

- Ergodic Markov chain with stationary distribution $\pi*$
- Transition matrix A, write p_{jk} for $A[j][k]$
  - Probability of transition from state j to state k
- Reversibility — in steady state, probability of going from j to k should equal probability of going from k to j
- Given an evolution $x_1x_2\ldots$, for large n, $P[x_n = k \mid x_{n-1} = j] = P[x_{n-1} = j \mid x_n = k]$
- $P[x_{n-1} = j \mid x_n = k] = P[x_n = k \mid x_{n-1} = j] \cdot \dfrac{\pi_j}{\pi_k}$, in steady state
- $p_{kj} = p_{jk}\dfrac{\pi_j}{\pi_k}$
- $\pi_j \cdot p_{jk} = \pi_k \cdot p_{kj}$ — Treat this as the defn we want to use

Alleged explan

Reversible Markov chains

- Ergodic Markov chain

Reversible Markov chains

- Ergodic Markov chain
- Suppose $a^T = (a_1, a_2, \ldots, a_n)$ satifies reversibility condition for all j,k
  - $a_j \cdot p_{jk} = a_k \cdot p_{kj}$

$\pi_j \, p_{jk} = \pi_k \, p_{kj}$

Reversible Markov chains

- Ergodic Markov chain
- Suppose $a^T = (a_1, a_2, \ldots, a_n)$ satifies reversibility condition for all j, k
  - $a_j \cdot p_{jk} = a_k \cdot p_{kj}$
- $\displaystyle\sum_k a_j \cdot p_{jk} = \sum_k a_k \cdot p_{kj}$

Reversible Markov chains

- Ergodic Markov chain
- Suppose $a^T = (a_1, a_2, \ldots, a_n)$ satifies reversibility condition for all j,k
  - $a_j \cdot p_{jk} = a_k \cdot p_{kj}$ — for every j & k
- $\sum_k a_j \cdot p_{jk} = \sum_k a_k \cdot p_{kj}$
- $a_j \underbrace{\sum_k p_{jk}}_{\text{row } j \text{ in } A} = \sum_k a_k \cdot p_{kj}$

$a_{j1} \cdot p_{j1} = a_1 p_{1j}$

$a_{j2} \cdot p_{j2}$

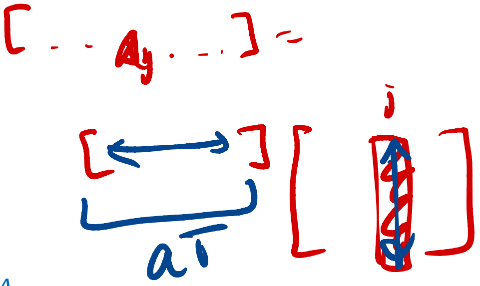
Reversible Markov chains

- Ergodic Markov chain
- Suppose $a^T = (a_1, a_2, \ldots, a_n)$ satifies reversibility condition for all j,k
  - $a_j \cdot p_{jk} = a_k \cdot p_{kj}$
- $\displaystyle\sum_k a_j \cdot p_{jk} = \sum_k a_k \cdot p_{kj}$
- $\displaystyle a_j \sum_k p_{jk} = \sum_k a_k \cdot p_{kj}$
- $\displaystyle a_j \cdot 1 = \sum_k a_k \cdot p_{kj}$

Reversible Markov chains

$a_j = a_1 p_{1j} + a_2 p_{2j} + \cdots \; a_n p_{nj}$

- Ergodic Markov chain
- Suppose $a^T = (a_1, a_2, \ldots, a_n)$ satifies reversibility condition for all j, k
  - $a_j \cdot p_{jk} = a_k \cdot p_{kj}$
- $\displaystyle\sum_k a_j \cdot p_{jk} = \sum_k a_k \cdot p_{kj}$
- $\displaystyle a_j \sum_k p_{jk} = \sum_k a_k \cdot p_{kj}$
- $\displaystyle a_j \cdot 1 = \sum_k a_k \cdot p_{kj}$
- $a^\top = a^\top A$, so $a^\top$ is the stationary distribution of A



Gibbs sampling

- State of a Bayesian network is a valuation of variables $(V_1, V_2, \ldots, V_n)$

Gibbs sampling

- State of a Bayesian network is a valuation of variables $(V_1, V_2, \ldots, V_n$
- Move probabilistically from $s_j = (x_1, x_2, \ldots, x_n)$ to $s_k = (y_1, y_2, \ldots, y_n)$

Gibbs sampling

- State of a Bayesian network is a valuation of variables $(V_1, V_2, \ldots, V_n$
- Move probabilistically from $s_j = (x_1, x_2, \ldots, x_n)$ to $s_k = (y_1, y_2, \ldots, y_n)$
- Allow such a move only when s_j, s_k differ at exactly one position
  - $s_j = (x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$
  - $s_k = (x_1, x_2, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)$

Gibbs sampling

- State of a Bayesian network is a valuation of variables $(V_1, V_2, \ldots, V_n$
- Move probabilistically from $s_j = (x_1, x_2, \ldots, x_n)$ to $s_k = (y_1, y_2, \ldots, y_n)$
- Allow such a move only when s_j, s_k differ at exactly one position
  - $s_j = (x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$
  - $s_k = (x_1, x_2, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)$
- Sampling algorithm
  - Current state is $s_j = (x_1, x_2, \ldots, x_n)$
  - Choose i uniformly in $[1, n]$
  - Resample x_i given current values $(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$

Gibbs sampling

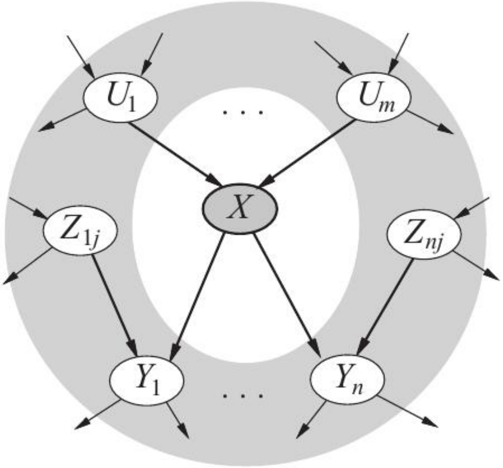
- State of a Bayesian network is a valuation of variables $(V_1, V_2, \ldots, V_n$
- Move probabilistically from $s_j = (x_1, x_2, \ldots, x_n)$ to $s_k = (y_1, y_2, \ldots, y_n)$
- Allow such a move only when s_j, s_k differ at exactly one position
  - $s_j = (x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$
  - $s_k = (x_1, x_2, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)$

could be x_i

- Sampling algorithm
  - Current state is $s_j = (x_1, x_2, \ldots, x_n)$
  - Choose i uniformly in $[1, n]$
  - Resample x_i given current values $(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$
- Need to compute $P[y_i \mid x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$

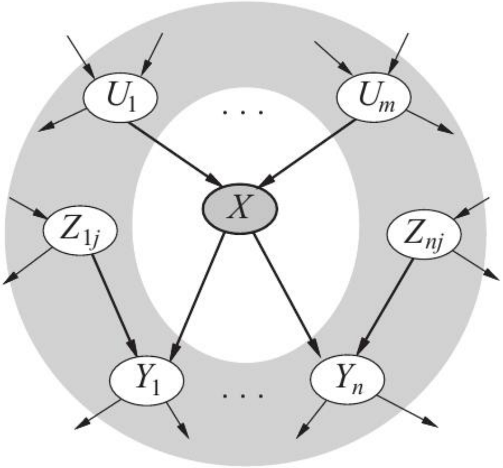
Markov blanket

- Recall $MB(X)$ — Markov blanket of X
  - $Parents(X)$
  - $Children(X)$
  - $Parents\ of\ Children(X)$
- $X \perp \neg MB(X) \mid MB(X)$



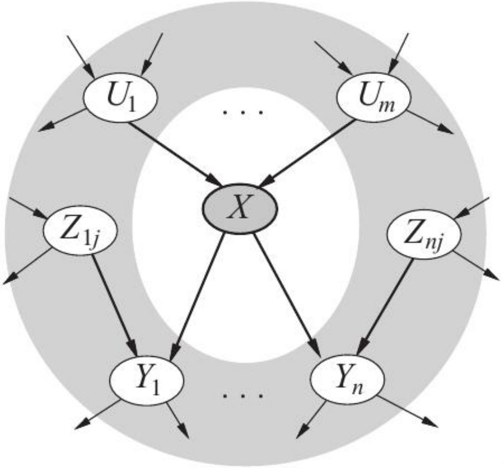
Markov blanket

- Recall $MB(X)$ — Markov blanket of X
  - $Parents(X)$
  - $Children(X)$
  - $Parents\ of\ Children(X)$
- $X \perp \neg MB(X) \mid MB(X)$
- Need to compute
  $P[y_i \mid x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$



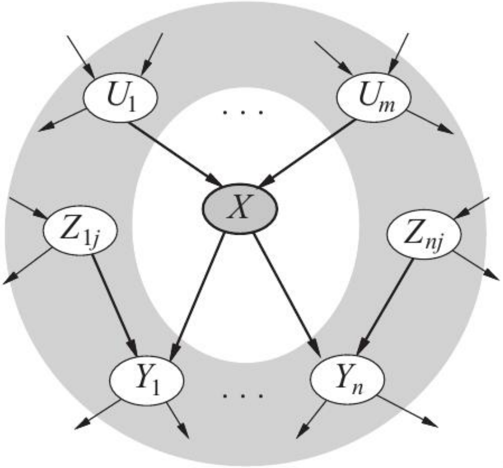
Markov blanket

- Recall $MB(X)$ — Markov blanket of X
  - $Parents(X)$
  - $Children(X)$
  - $Parents\ of\ Children(X)$
- $X \perp \neg MB(X) \mid MB(X)$
- Need to compute
  $P[y_i \mid x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$
- $x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$ fixes $MB(V_i)$



Markov blanket

- Recall $MB(X)$ — Markov blanket of X
  - $Parents(X)$
  - $Children(X)$
  - $Parents\ of\ \ Children(X)$
- $X \perp \neg MB(X) \mid MB(X)$
- Need to compute $P[y_i \mid x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$
- $x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$ fixes $MB(V_i)$
- Can compute $P[y_i \mid x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$ given conditional probability tables in the network



Gibbs sampling

- Move from $s_j = (x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$ to $s_k = (x_1, x_2, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)$

Gibbs sampling

- Move from $s_j = (x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$ to $s_k = (x_1, x_2, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)$
- Let $\bar{x} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$

Gibbs sampling

- Move from $s_j = (x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$ to $s_k = (x_1, x_2, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)$
- Let $\bar{x} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$
- $P_{jk} = \frac{1}{n} P[y_i \mid \bar{x}]$

Pick $i \in [1, n]$

Gibbs sampling

- Move from $s_j = (x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$ to $s_k = (x_1, x_2, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)$
- Let $\bar{x} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$
- $P_{jk} = \frac{1}{n} P[y_i \mid \bar{x}] \; = \frac{1}{n} \frac{P(s_k)}{P(\bar{x})}$

Gibbs sampling

- Move from $s_j = (x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$ to $s_k = (x_1, x_2, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)$
- Let $\bar{x} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$
- $P_{jk} = \frac{1}{n} P[y_i \mid \bar{x}] \; = \frac{1}{n} \frac{P(s_k)}{P(\bar{x})} \quad = \frac{1}{n} \frac{\pi_k}{P(\bar{x})}$

Gibbs sampling

- Move from $s_j = (x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$ to $s_k = (x_1, x_2, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)$
- Let $\bar{x} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$
- $P_{jk} = \frac{1}{n} P[y_i \mid \bar{x}] = \frac{1}{n} \frac{P(s_k)}{P(\bar{x})} = \frac{1}{n} \frac{\pi_k}{P(\bar{x})}$
- Likewise $P_{kj} = \frac{1}{n} P[x_i \mid \bar{x}] = \frac{1}{n} \frac{P(s_j)}{P(\bar{x})} = \frac{1}{n} \frac{\pi_j}{P(\bar{x})}$

Gibbs sampling

- Move from $s_j = (x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$ to $s_k = (x_1, x_2, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)$
- Let $\bar{x} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$
- $P_{jk} = \frac{1}{n} P[y_i \mid \bar{x}] = \frac{1}{n}\frac{P(s_k)}{P(\bar{x})} \cancel{= \frac{1}{n}\frac{\pi_k}{P(\bar{x})}}$
- Likewise $P_{kj} = \frac{1}{n} P[x_i \mid \bar{x}] = \frac{1}{n}\frac{P(s_j)}{P(\bar{x})} \cancel{= \frac{1}{n}\frac{\pi_j}{P(\bar{x})}}$
- Therefore, $\frac{P_{jk}}{P_{kj}} = \cancel{\frac{\pi_k}{\pi_j}} \quad \frac{P(s_k)}{P(s_j)}$

$\forall j, k$

$$\Rightarrow \underbrace{P(s_j)}_{\Downarrow\, \pi_j} \cdot P_{jk} = \underbrace{P(s_k)}_{\Downarrow\, \pi_k} P_{kj}$$

Gibbs sampling

- Move from $s_j = (x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$ to $s_k = (x_1, x_2, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)$
- Let $\bar{x} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$
- $P_{jk} = \frac{1}{n} P[y_i \mid \bar{x}] \;= \frac{1}{n} \frac{P(s_k)}{P(\bar{x})} \;= \frac{1}{n} \frac{\pi_k}{P(\bar{x})}$
- Likewise $P_{kj} = \frac{1}{n} P[x_i \mid \bar{x}] = \frac{1}{n} \frac{P(s_j)}{P(\bar{x})} = \frac{1}{n} \frac{\pi_j}{P(\bar{x})}$
- Therefore, $\frac{P_{jk}}{P_{kj}} = \frac{\pi_k}{\pi_j}$
- Hence, $\pi_j \cdot P_{jk} = \pi_k \cdot P_{kj}$

Gibbs sampling

- Move from $s_j = (x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$ to $s_k = (x_1, x_2, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)$
- $\pi_j \cdot P_{jk} = \pi_k \cdot P_{kj}$
- We have created a reversible Markov chain whose stationary distribution provides the true probabilities of the original Bayesian network!

Gibbs sampling

- Move from $s_j = (x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$ to $s_k = (x_1, x_2, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)$
- $\pi_j \cdot P_{jk} = \pi_k \cdot P_{kj}$
- We have created a reversible Markov chain whose stationary distribution provides the true probabilities of the original Bayesian network!
- Gibbs sampling is a special case of the more general Metropolis-Hastings algorithm

Gibbs sampling

- Since we are dealing with steady state probabilities, it is not necessary to change just one variable at a time

Gibbs sampling

- Since we are dealing with steady state probabilities, it is not necessary to change just one variable at a time
    - Generate an entirely new sample state $(y_1, y_2, \ldots, y_n)$

Gibbs sampling

- Since we are dealing with steady state probabilities, it is not necessary to change just one variable at a time
  - Generate an entirely new sample state $(y_1, y_2, \ldots, y_n)$
  - First generate y_1, given $x_2, x_3, \ldots, x_n$

Gibbs sampling

- Since we are dealing with steady state probabilities, it is not necessary to change just one variable at a time
  - Generate an entirely new sample state $(y_1, y_2, \ldots, y_n)$
  - First generate y_1, given $x_2, x_3, \ldots, x_n$
  - Then generate y_2, given $y_1, x_3, \ldots, x_n$

Gibbs sampling

- Since we are dealing with steady state probabilities, it is not necessary to change just one variable at a time
  - Generate an entirely new sample state $(y_1, y_2, \ldots, y_n)$
  - First generate y_1, given $x_2, x_3, \ldots, x_n$
  - Then generate y_2, given $y_1, x_3, \ldots, x_n$
  - ...
  - Then generate y_n, given $y_1, y_2, \ldots, y_{n-1}$

Gibbs sampling

- Since we are dealing with steady state probabilities, it is not necessary to change just one variable at a time
  - Generate an entirely new sample state $(y_1, y_2, \ldots, y_n)$
  - First generate y_1, given $x_2, x_3, \ldots, x_n$
  - Then generate y_2, given $y_1, x_3, \ldots, x_n$
  - ...
  - Then generate y_n, given $y_1, y_2, \ldots, y_{n-1}$
- Standard Gibbs sampler — again a reversible Markov chain

MCMC Markov Chain Monte Carlo

Approximate inference using Markov chains

- Bayesian network has variables $V_1, V_2, \ldots, V_n$
- Use Gibbs sampling to set up a a reversible Markov chain
- Stationary distribution will assign to state s the probability $P(s)$ in the Bayesian network