Lecture 22: 04 April, 2024

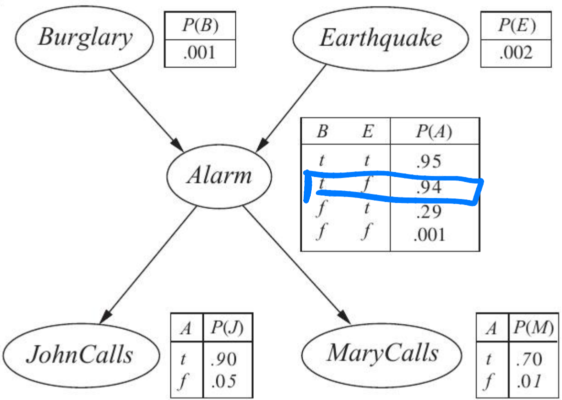
Madhavan Mukund

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

Data Mining and Machine Learning
January–April 2024

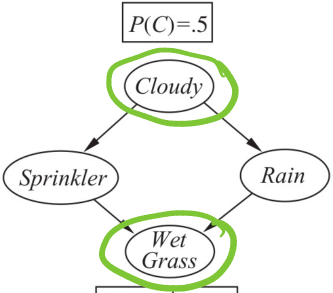
Approximate inference

- 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



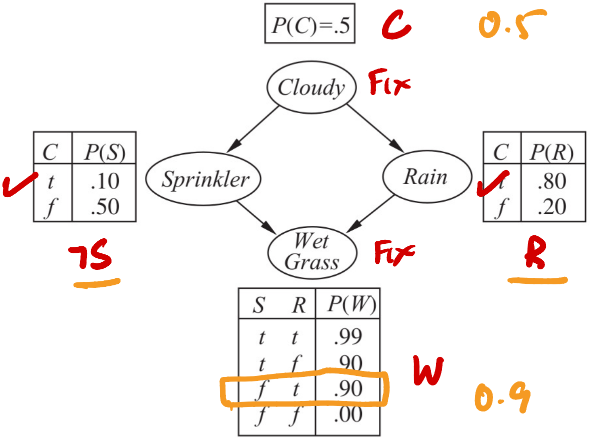
C	P(S)
t	.10
f	.50

C	P(R)
t	.80
f	.20

S	R	P(W)
t	t	.99
t	f	.90
f	t	.90
f	f	.00

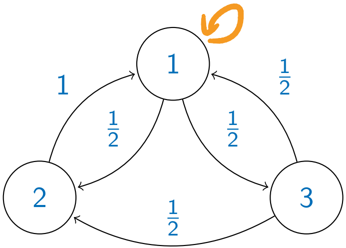
Likelihood weighted sampling

- $P(Rain \mid Cloudy, Wet\ Grass)$
- 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 $w_1, w_2, \ldots w_N$
- $P(r \mid c, w) = \dfrac{\sum_{s_i \text{ has rain}} w_i}{\sum_{1 \le j \le N} w_j}$



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

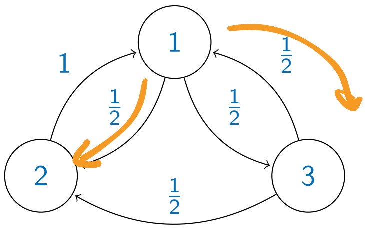
row sums are 1

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

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

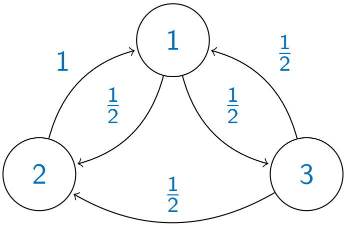
Ergodicity

- 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)



Approximate inference using Markov chains

- 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^* (which we shall write as π)

Reversible Markov chains

- Ergodic Markov chain with stationary distribution π^* (which we shall write as π)
- 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 π^* (which we shall write as π)
- Transition matrix A, write p_{jk} for $A[j][k]$
  - Probability of transition from state j to state k
- Reversibility : $\pi_j \cdot p_{jk} = \pi_k \cdot p_{kj}$, for all j,k (balance equations)
  - In steady state, probability of being in state j and then moving to k same as probability of being in state k and then moving to j

Reversible Markov chains

- Ergodic Markov chain with stationary distribution π^* (which we shall write as π)
- Transition matrix A, write p_{jk} for $A[j][k]$
  - Probability of transition from state j to state k
- Reversibility : $\pi_j \cdot p_{jk} = \pi_k \cdot p_{kj}$, for all j,k (balance equations)
  - In steady state, probability of being in state j and then moving to k same as probability of being in state k and then moving to j
- Derivation of balance equations

Reversible Markov chains

- Ergodic Markov chain with stationary distribution π^* (which we shall write as π)
- Transition matrix A, write p_{jk} for $A[j][k]$
  - Probability of transition from state j to state k
- Reversibility : $\pi_j \cdot p_{jk} = \pi_k \cdot p_{kj}$, for all j,k (balance equations)
  - In steady state, probability of being in state j and then moving to k same as probability of being in state k and then moving to j
- Derivation of balance equations
  - Given an evolution $x_1 x_2 \ldots$, for large n, $P[x_n = j \mid x_{n-1} = k] = P[x_{n-1} = j \mid x_n = k]$

Reversible Markov chains

- Ergodic Markov chain with stationary distribution π^* (which we shall write as π)
- Transition matrix A, write p_{jk} for $A[j][k]$
  - Probability of transition from state j to state k
- Reversibility : $\pi_j \cdot p_{jk} = \pi_k \cdot p_{kj}$, for all j, k (balance equations)
  - In steady state, probability of being in state j and then moving to k same as probability of being in state k and then moving to j
- Derivation of balance equations
  - Given an evolution $x_1 x_2 \ldots$, for large n, $P[x_n = j \mid x_{n-1} = k] = P[x_{n-1} = j \mid x_n = k]$
  - $P[x_{n-1} = j \mid x_n = k] = P[x_n = k \mid x_{n-1} = j] \cdot \dfrac{P[x_{n-1} = j]}{P[x_n = k]}$ π_j / π_k

Reversible Markov chains

- Ergodic Markov chain with stationary distribution π^* (which we shall write as π)
- Transition matrix A, write p_{jk} for $A[j][k]$
  - Probability of transition from state j to state k
- Reversibility : $\pi_j \cdot p_{jk} = \pi_k \cdot p_{kj}$, for all j,k (balance equations)
  - In steady state, probability of being in state j and then moving to k same as probability of being in state k and then moving to j
- Derivation of balance equations
  - Given an evolution $x_1 x_2 \ldots$, for large n, $P[x_n = j \mid x_{n-1} = k] = P[x_{n-1} = j \mid x_n = k]$
  - $P[x_{n-1} = j \mid x_n = k] = P[x_n = k \mid x_{n-1} = j] \cdot \dfrac{\pi_j}{\pi_k}$, in steady state

Reversible Markov chains

- Ergodic Markov chain with stationary distribution π^* (which we shall write as π)
- Transition matrix A, write p_{jk} for $A[j][k]$
  - Probability of transition from state j to state k
- Reversibility : $\pi_j \cdot p_{jk} = \pi_k \cdot p_{kj}$, for all j, k (balance equations)
  - In steady state, probability of being in state j and then moving to k same as probability of being in state k and then moving to j
- Derivation of balance equations
  - Given an evolution $x_1 x_2 \ldots$, for large n, $P[x_n = j \mid x_{n-1} = k] = P[x_{n-1} = j \mid x_n = k]$
  - $P[x_{n-1} = j \mid x_n = k] = P[x_n = k \mid x_{n-1} = j] \cdot \dfrac{\pi_j}{\pi_k}$, in steady state
  - $p_{kj} = p_{jk} \dfrac{\pi_j}{\pi_k}$, so $\pi_j \cdot p_{jk} = \pi_k \cdot p_{kj}$

Reversible Markov chains

- Ergodic Markov chain

Reversible Markov chains

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

$\pi_j \cdot P_{jk} = \pi_k \cdot P_{kj}$

Reversible Markov chains

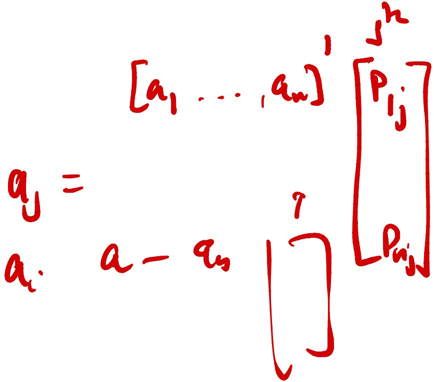
- Ergodic Markov chain
- Suppose $a^\top = (a_1, a_2, \ldots, a_n)$ satifies reversibility balance equations for all j, k
  - $a_j \cdot p_{jk} = a_k \cdot p_{kj}$
- $\sum_k a_j \cdot p_{jk} = \sum_k a_k \cdot p_{kj}$

Reversible Markov chains

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

Reversible Markov chains

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



Reversible Markov chains

- Ergodic Markov chain
- Suppose $a^\top = (a_1, a_2, \ldots, a_n)$ satifies reversibility balance equations for all j,k
  - $a_j \cdot p_{jk} = a_k \cdot p_{kj}$
- $\sum_k a_j \cdot p_{jk} = \sum_k a_k \cdot p_{kj}$
- $a_j \sum_k p_{jk} = \sum_k a_k \cdot p_{kj}$
- $a_j \cdot 1 = \sum_k a_k \cdot p_{kj}$
- $a^\top = a^\top A$, 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 $(V_1, V_2, \ldots, V_n)$ Boolean $\Rightarrow 2^n$ states
- 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
  - $s_j = (x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n)$
  - $s_k = (x_1, x_2, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)$

n neighbours if boolean

Gibbs sampling

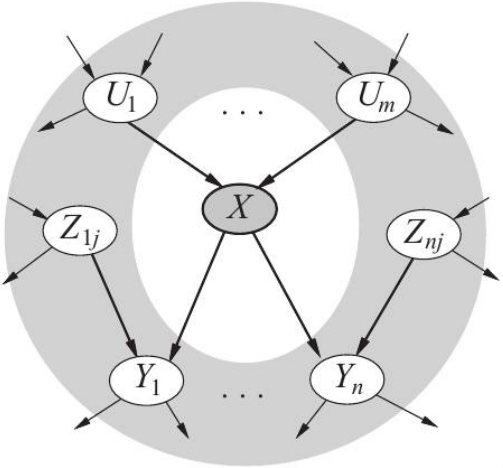
- State of a Bayesian network is a valuation of variables $(V_1, V_2, \ldots, V_n)$
- Move probabilistically from $s_j = (x_1, x_2, \ldots, x_n)$ to $s_k = (y_1, y_2, \ldots, y_n)$
- Allow such a move only when s_j, s_k differ at exactly one position
  - $s_j = (x_1, x_2, \ldots, x_{i-1}, x_i, x_{i+1},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

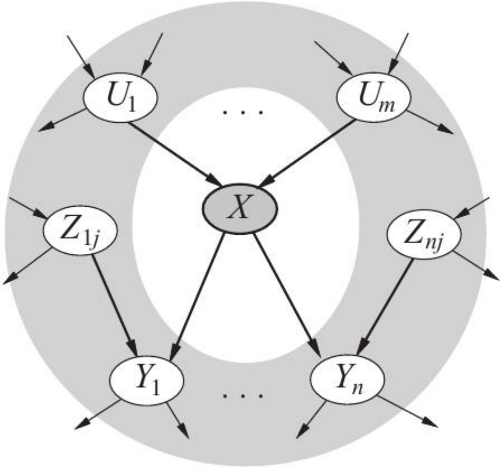
Markov blanket

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

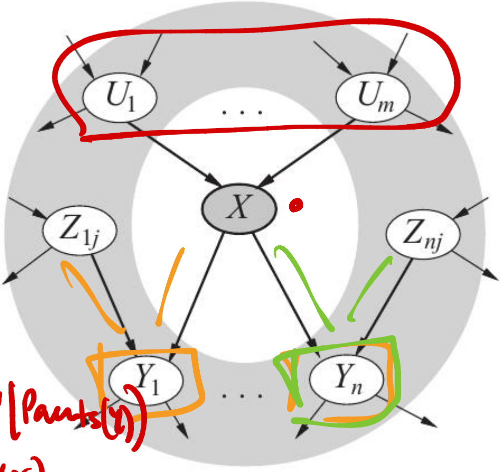


Markov blanket

- 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
  $P[y_i \mid x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$



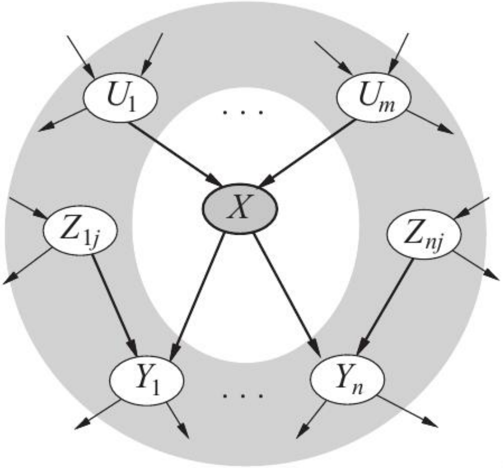
- Recall $MB(X)$ — Markov blanket of X
  - $Parents(X)$
  - $Children(X)$
  - $Parents\ of\ Children(X)$
- $X \perp \neg MB(X) \mid MB(X)$
- Need to compute $P[y_i \mid x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$
- $x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ fix $MB(V_i)$



$P(y_i \mid Parents(X)) \cdot \prod_{Y \in Child(X)} P(Y \mid Parents(Y))$

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 $P[y_i \mid x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n]$
- $x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$ fix $MB(V_i)$
- 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 $\bar{x} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$
- $p_{jk} = \frac{1}{n} P[y_i \mid \bar{x}]$

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 $\bar{x} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$
- $p_{jk} = \frac{1}{n} P[y_i \mid \bar{x}] \ = \frac{1}{n} \frac{P(s_k)}{P(\bar{x})}$

$x_1 - x_{i-1}, x_{i+1}, \cdot\ x_n$

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

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 $\bar{x} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$
- $p_{jk} = \frac{1}{n} P[y_i \mid \bar{x}] = \frac{1}{n} \frac{P(s_k)}{P(\bar{x})}$
- Likewise $p_{kj} = \frac{1}{n} P[x_i \mid \bar{x}] = \frac{1}{n} \frac{P(s_j)}{P(\bar{x})}$
- Therefore, $\frac{p_{jk}}{p_{kj}} = \frac{P(s_k)}{P(s_j)}$, so $P(s_j) \cdot p_{jk} = P(s_k) \cdot p_{kj}$ and this chain is reversible

a_j a_k

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 $\bar{x} = (x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$
- $p_{jk} = \frac{1}{n} P[y_i \mid \bar{x}] = \frac{1}{n} \frac{P(s_k)}{P(\bar{x})}$
- Likewise $p_{kj} = \frac{1}{n} P[x_i \mid \bar{x}] = \frac{1}{n} \frac{P(s_j)}{P(\bar{x})}$
- Therefore, $\frac{p_{jk}}{p_{kj}} = \frac{P(s_k)}{P(s_j)}$, so $P(s_j) \cdot p_{jk} = P(s_k) \cdot p_{kj}$ and this chain is reversible
- By our previous observation about any vector $a^\top$ satisfying balance equations, we must have $(P(s_1), P(s_2), \ldots, P(s_n)) = (\pi_1, \pi_2, \ldots, \pi_n)$ for the current Markov chain

Gibbs sampling

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

P[j]

- $\pi_j \cdot p_{jk} = \pi_k \cdot p_{kj}$
- 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

Markov Chain Monte Carlo

[MCMC]

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 $y_1, x_3, \ldots, x_n$
  - $\ldots$
  - Then generate y_n, given $y_1, y_2, \ldots, y_{n-1}$
- Standard Gibbs sampler — again a reversible Markov chain

Approximate inference using Markov chains

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