Markov chain Monte Carlo
Lecture 9

David Sontag
New York University

Slides adapted from Eric Xing and Qirong Ho (CMU)

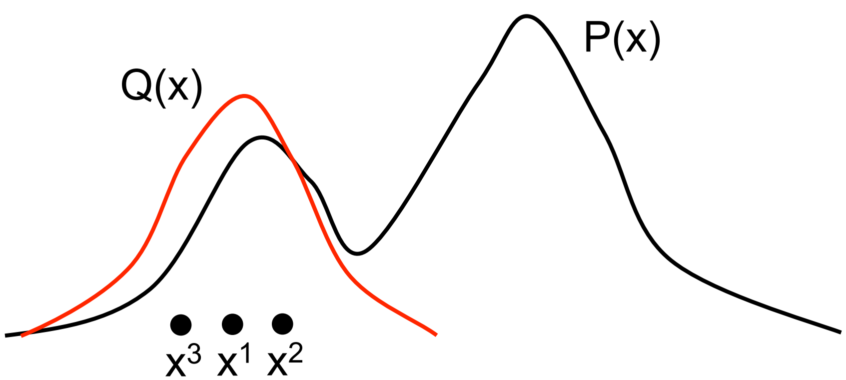
Limitations of Monte Carlo

- Direct (unconditional) sampling
  - Hard to get rare events in high-dimensional spaces
  - Infeasible for MRFs, unless we know the normalizer Z
- Rejection sampling, Importance sampling
  - Do not work well if the proposal Q(x) is very different from P(x)
  - Yet constructing a Q(x) similar to P(x) can be difficult
    - Making a good proposal usually requires knowledge of the analytic form of P(x) – but if we had that, we wouldn't even need to sample!
- Intuition: instead of a fixed proposal Q(x), what if we could use an adaptive proposal?

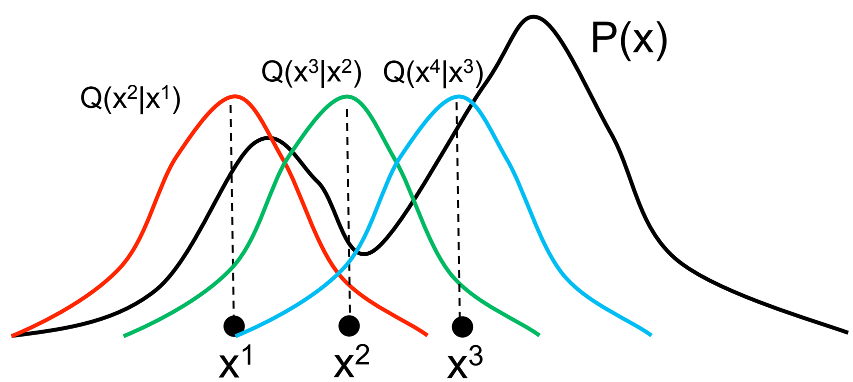
Markov Chain Monte Carlo

- MCMC algorithms feature adaptive proposals
  - Instead of Q(x'), they use Q(x'|x) where x' is the new state being sampled, and x is the previous sample
  - As x changes, Q(x'|x) can also change (as a function of x')

Importance sampling with a (bad) proposal Q(x)



MCMC with adaptive proposal Q(x'|x)



Metropolis-Hastings

- Let's see how MCMC works in practice
  - Later, we'll look at the theoretical aspects
- Metropolis-Hastings algorithm
  - Draws a sample x' from Q(x'|x), where x is the previous sample
  - The new sample x' is **accepted** or **rejected** with some probability A(x'|x)
    - This acceptance probability is $A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$
    - A(x'|x) is like a ratio of importance sampling weights
      - P(x')/Q(x'|x) is the importance weight for x', P(x)/Q(x|x') is the importance weight for x
      - We divide the importance weight for x' by that of x
      - Notice that we only need to compute P(x')/P(x) rather than P(x') or P(x) separately
    - A(x'|x) ensures that, after sufficiently many draws, our samples will come from the true distribution P(x) – we shall learn why later in this lecture

The MH Algorithm

1. Initialize starting state $x^{(0)}$, set t =0
2. Burn-in: while samples have "not converged"
   - x=$x^{(t)}$
   - t = t +1,
   - sample $x^* \sim Q(x^*|x)$ // draw from proposal
   - sample $u \sim$ Uniform(0,1) // draw acceptance threshold
   -
     - if $u < A(x^* \mid x) = \min\left(1, \dfrac{P(x^*)Q(x \mid x^*)}{P(x)Q(x^* \mid x)}\right)$
       - $x^{(t)} = x^*$ // transition
     - else
       - $x^{(t)} = x$ // stay in current state

Function
Draw sample (x(t))

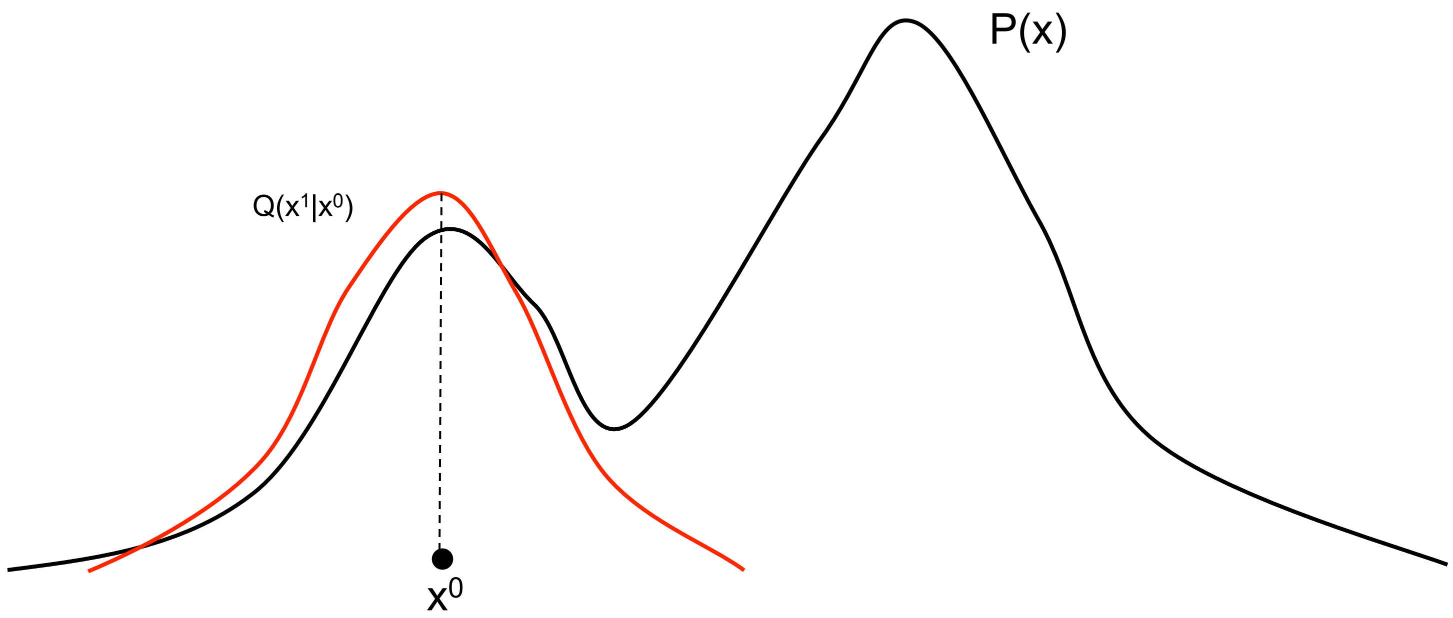
- Take samples from P(x): Reset t=0, for t =1:N
  - x(t+1) ← Draw sample (x(t))

$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

The MH Algorithm

- Example:
  - Let Q(x'|x) be a Gaussian centered on x
  - We're trying to sample from a bimodal distribution P(x)

Initialize $x^{(0)}$
…

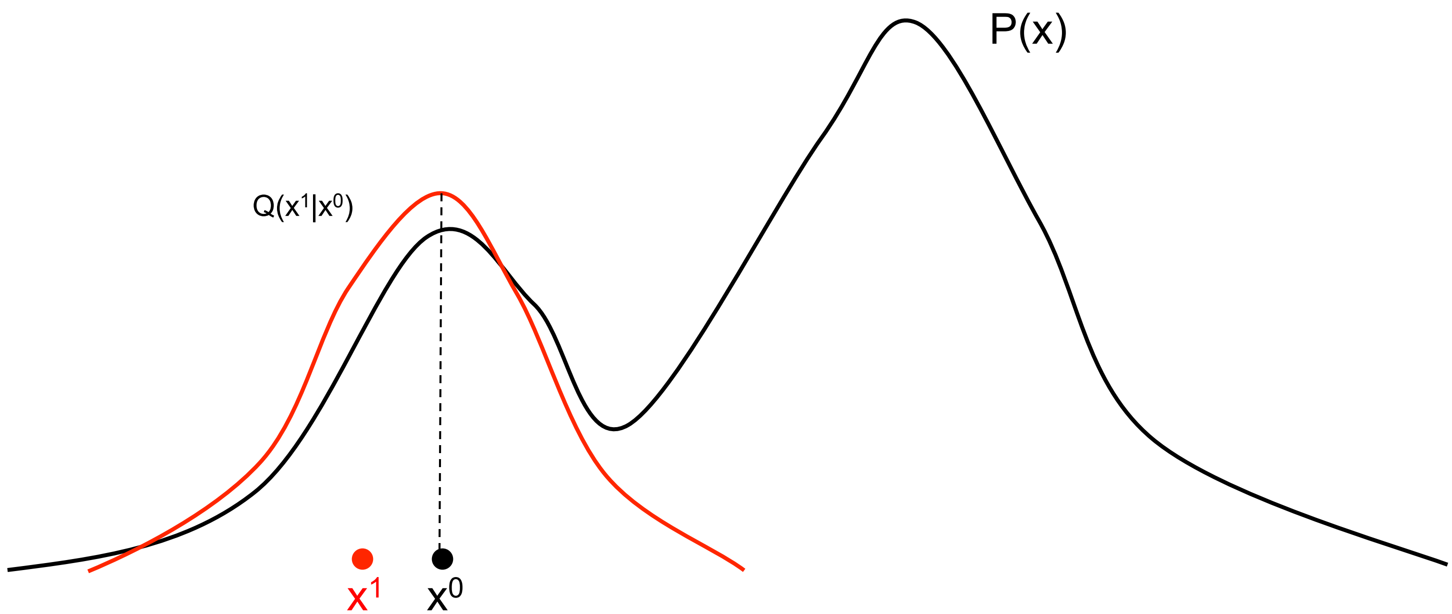


$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

The MH Algorithm

- Example:
  - Let Q(x'|x) be a Gaussian centered on x
  - We're trying to sample from a bimodal distribution P(x)

Initialize $x^{(0)}$
Draw, accept x^1

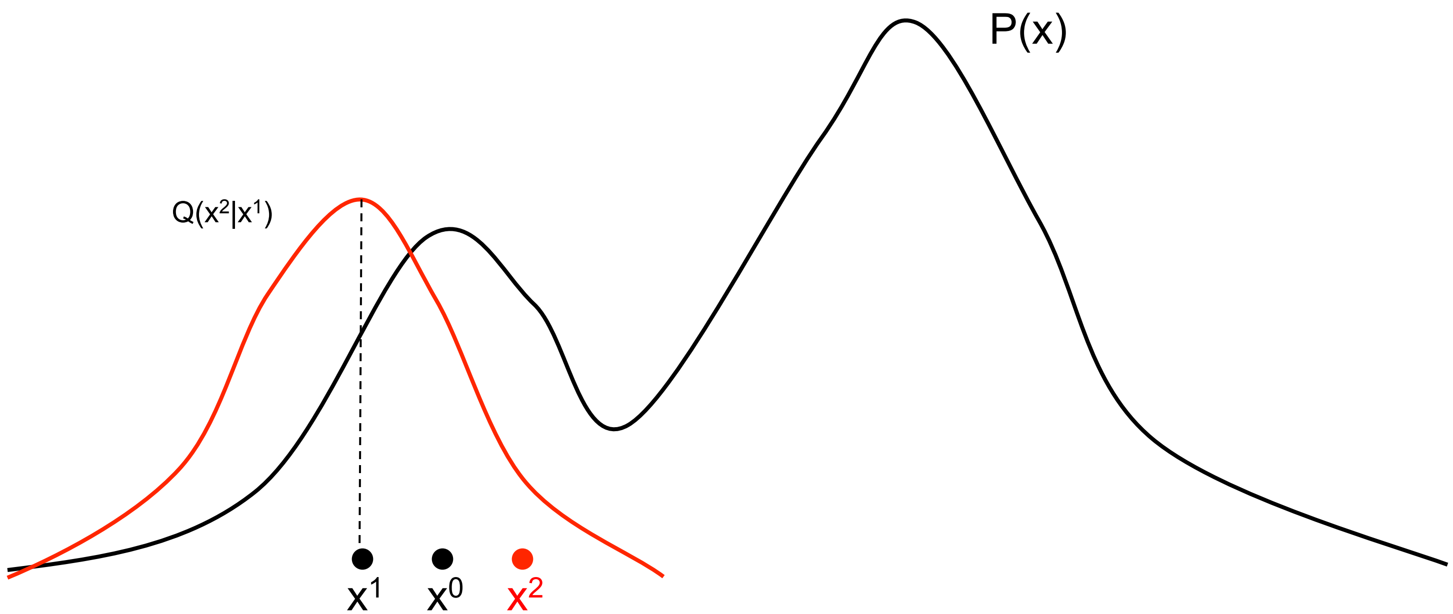


The MH Algorithm

$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

- Example:
  - Let Q(x'|x) be a Gaussian centered on x
  - We're trying to sample from a bimodal distribution P(x)

Initialize $x^{(0)}$
Draw, accept x^1
Draw, accept x^2

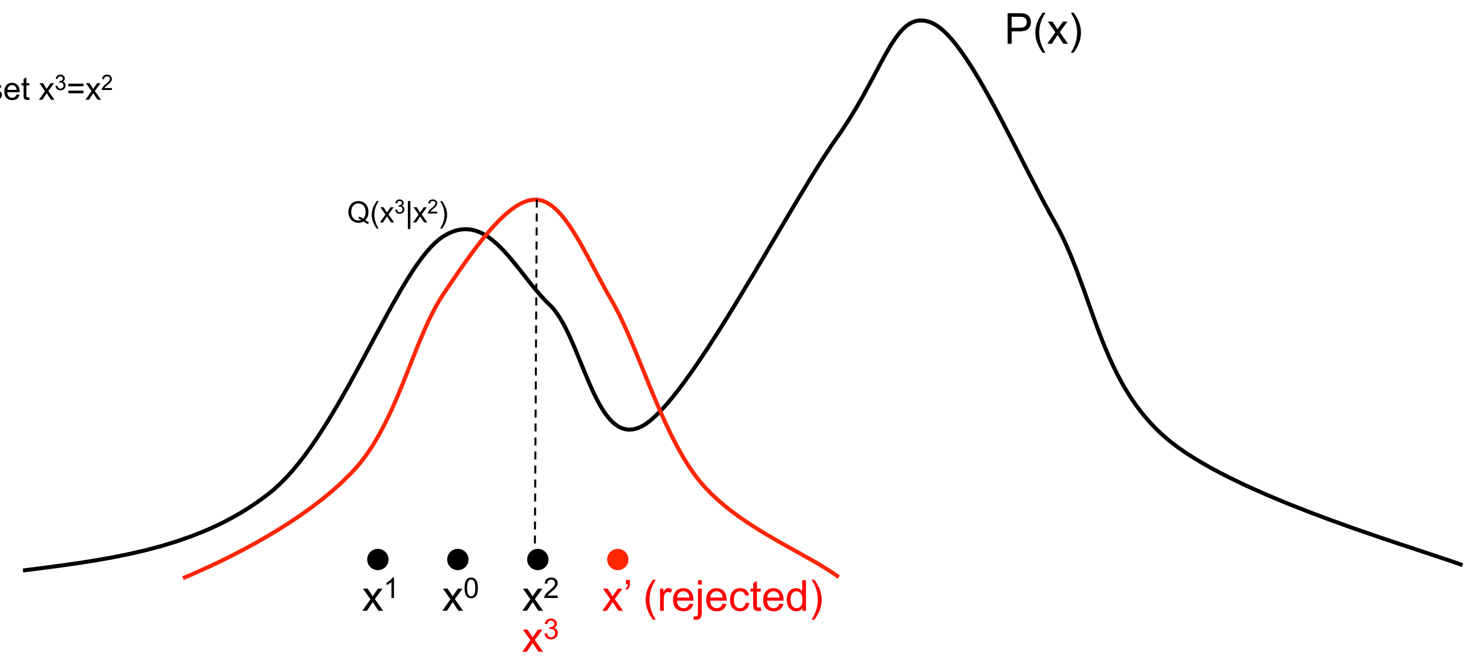


The MH Algorithm

$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

- Example:
  - Let Q(x'|x) be a Gaussian centered on x
  - We're trying to sample from a bimodal distribution P(x)

Initialize $x^{(0)}$
Draw, accept x^1
Draw, accept x^2
Draw but reject; set $x^3=x^2$



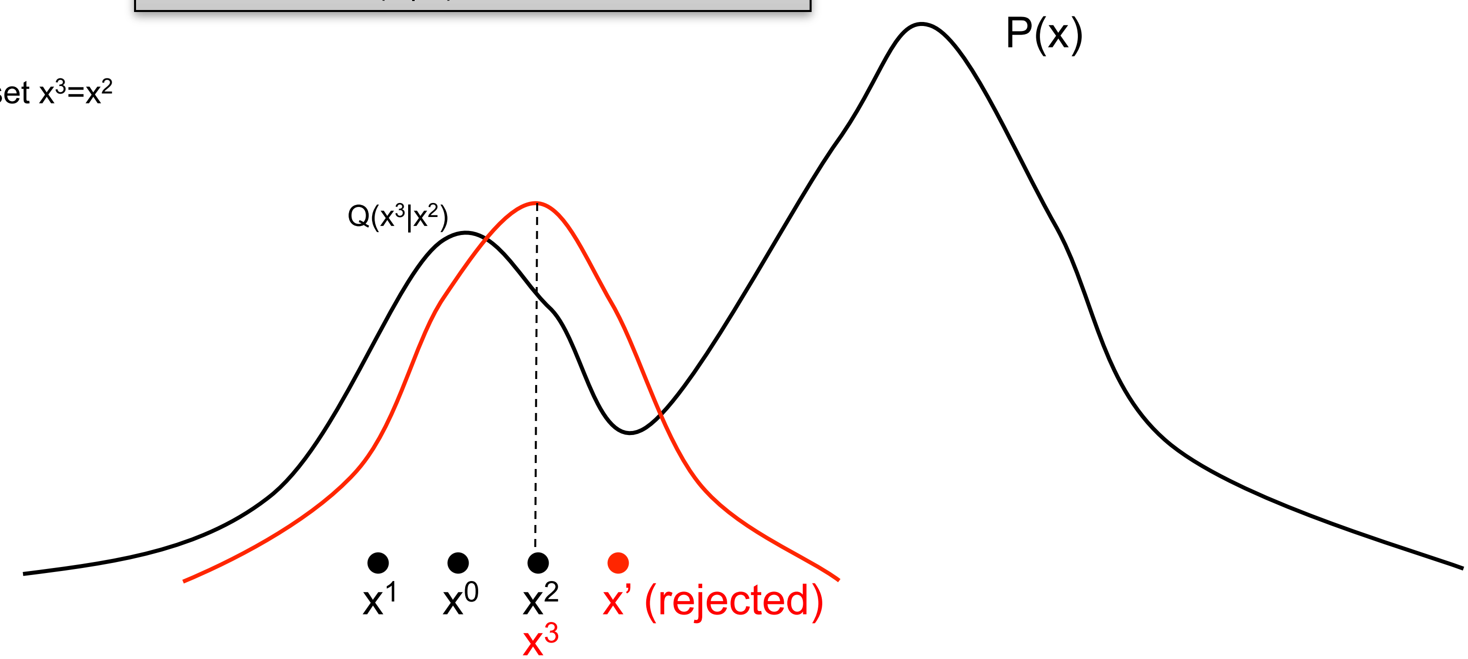
The MH Algorithm

$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

- Example:
  - Let Q(x'|x) be a Gaussian centered on x
  - We're trying to sample from a bimodal distribution P(x)

Initialize $x^{(0)}$
Draw, accept x^1
Draw, accept x^2
Draw but reject; set $x^3=x^2$

We reject because $P(x')/P(x^2)$ is very small, hence $A(x'|x^2)$ is close to zero!

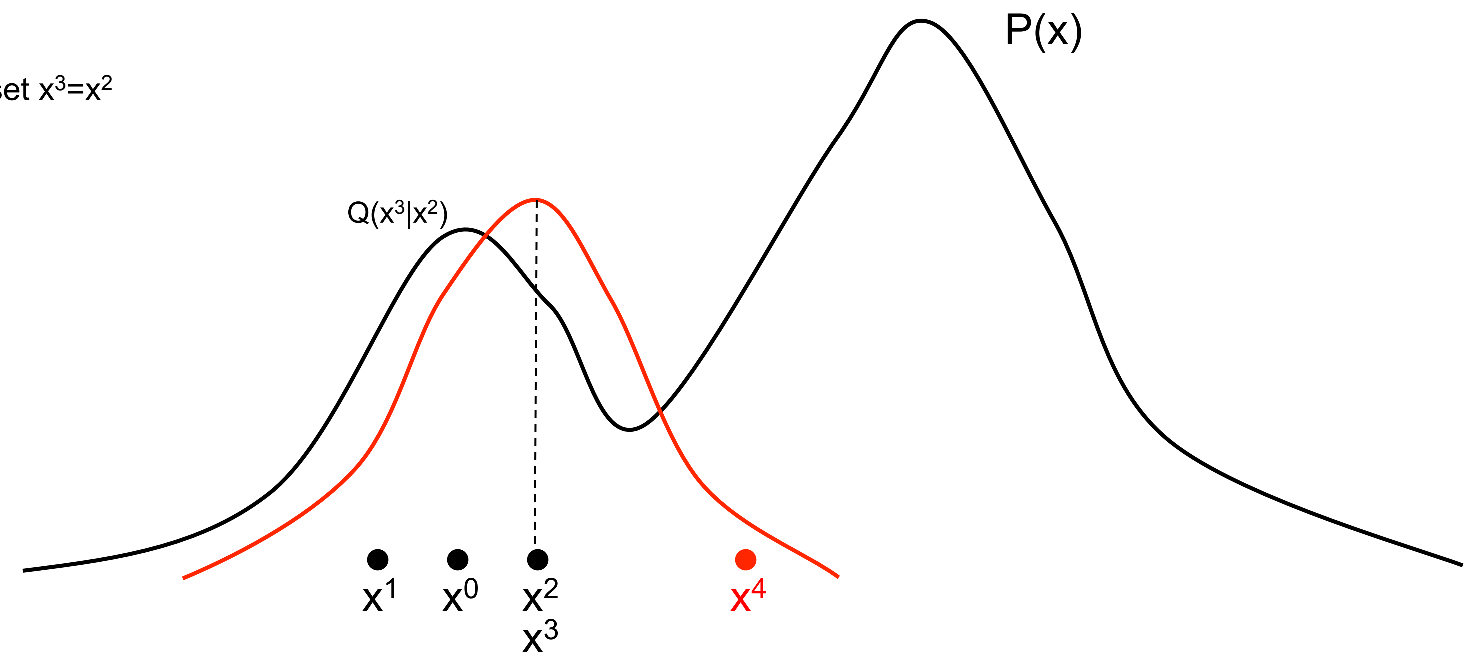


$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

The MH Algorithm

- Example:
  - Let Q(x'|x) be a Gaussian centered on x
  - We're trying to sample from a bimodal distribution P(x)

Initialize $x^{(0)}$
Draw, accept x^1
Draw, accept x^2
Draw but reject; set $x^3=x^2$
Draw, accept x^4

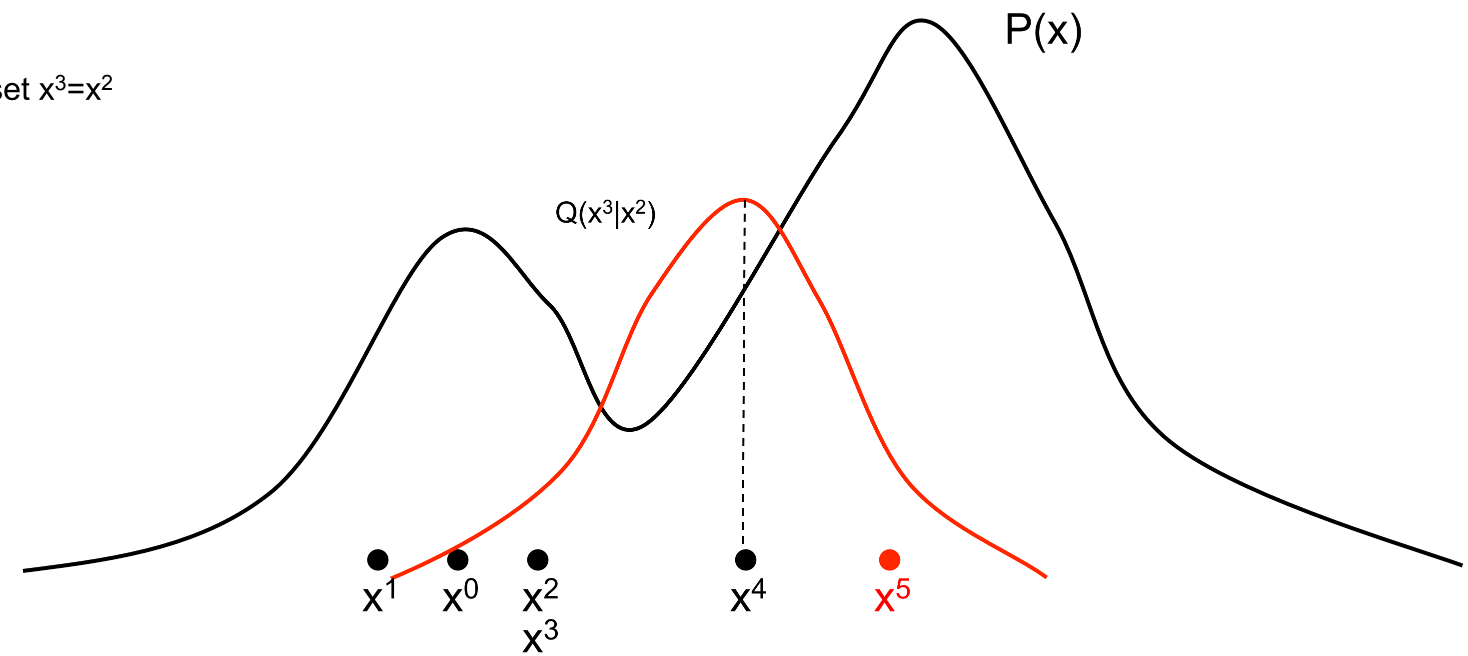


The MH Algorithm

$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

- Example:
  - Let Q(x'|x) be a Gaussian centered on x
  - We're trying to sample from a bimodal distribution P(x)

Initialize $x^{(0)}$
Draw, accept x^1
Draw, accept x^2
Draw but reject; set $x^3=x^2$
Draw, accept x^4
Draw, accept x^5



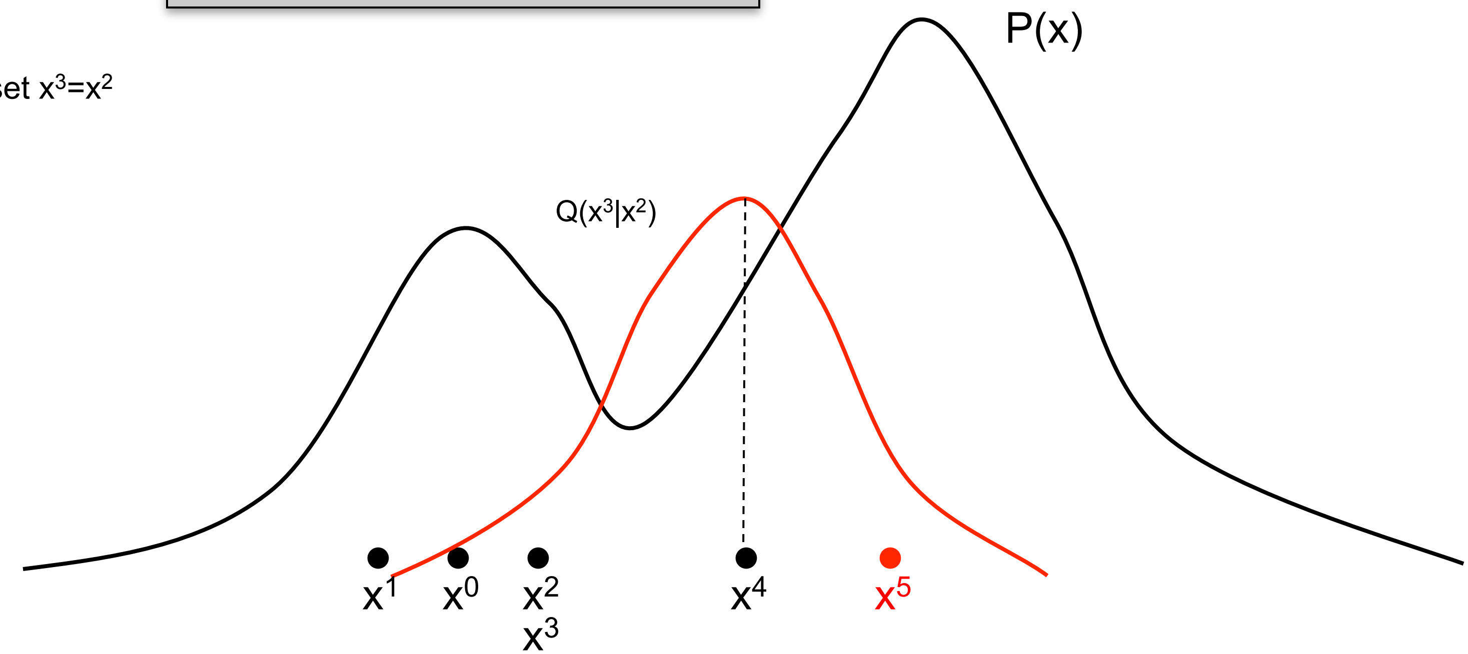
$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

The MH Algorithm

- Example:
  - Let Q(x'|x) be a Gaussian centered on x
  - We're trying to sample from a bimodal distribution P(x)

Initialize $x^{(0)}$
Draw, accept x^1
Draw, accept x^2
Draw but reject; set $x^3=x^2$
Draw, accept x^4
Draw, accept x^5

The adaptive proposal Q(x'|x) allows us to sample both modes of P(x)!



Theoretical aspects of MCMC

- The MH algorithm has a "burn-in" period
  - Why do we throw away samples from burn-in?

- Why are the MH samples guaranteed to be from P(x)?
  - The proposal Q(x'|x) keeps changing with the value of x; how do we know the samples will eventually come from P(x)?
  - Has to do with the connection between Markov chains & MCMC
  - We will return to this later

- What are good, general-purpose, proposal distributions?

Gibbs Sampling

- Gibbs Sampling is an MCMC algorithm that samples each random variable of a graphical model, one at a time
  - GS is a special case of the MH algorithm

- GS algorithms…
  - Are fairly easy to derive for many graphical models (e.g. mixture models, Latent Dirichlet allocation)
  - Have reasonable computation and memory requirements, because they sample one r.v. at a time
  - Can be Rao-Blackwellized (integrate out some r.v.s) to decrease the sampling variance – what we call **collapsed Gibbs sampling**

Gibbs Sampling

- The GS algorithm:
  1. Suppose the graphical model contains variables $x_1,\ldots,x_n$
  2. Initialize starting values for $x_1,\ldots,x_n$
  3. Do until convergence:
     1. Pick an ordering of the n variables (can be fixed or random)
     2. For each variable x_i in order:
        1. Sample $x \sim P(x_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$, i.e. the conditional distribution of x_i given the current values of all other variables
        2. Update $x_i \leftarrow x$

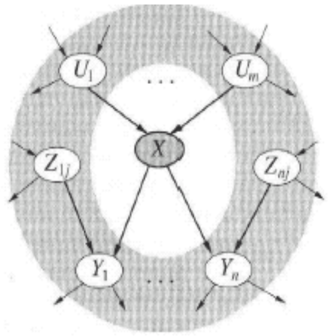
- When we update x_i, we <u>immediately</u> use its new value for sampling other variables x_j

Markov Blankets

- The conditional $P(x_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ looks intimidating, but recall Markov Blankets:
  - Let $MB(x_i)$ be the Markov Blanket of x_i, then

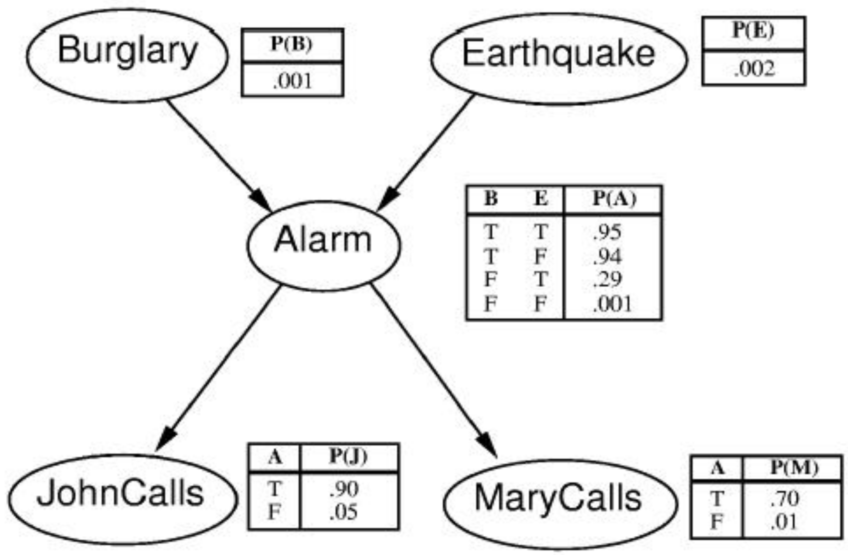
$$P(x_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i \mid MB(x_i))$$

- For a BN, the Markov Blanket of x_i is the set containing its parents, children, and co-parents



- For an MRF, the Markov Blanket of x_i is its immediate neighbors

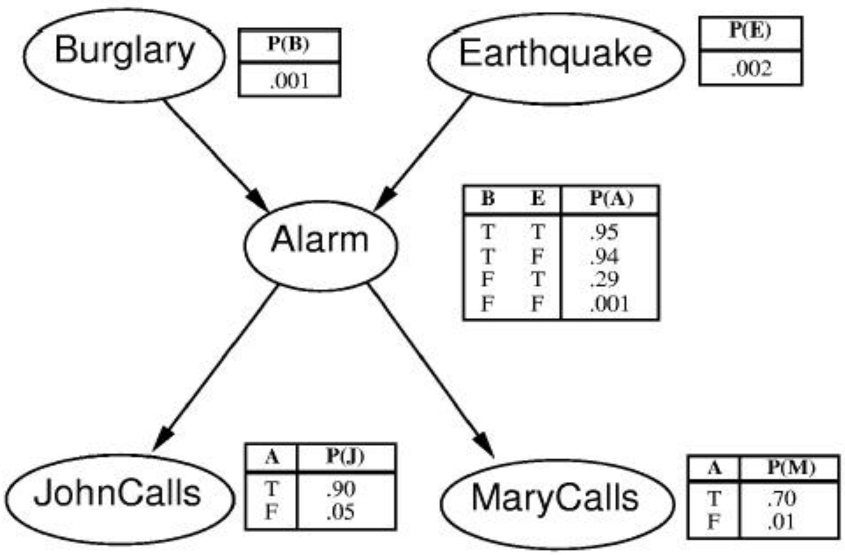
Gibbs Sampling: An Example



t	B	E	A	J	M
0	F	F	F	F	F
1					
2					
3					
4					

- Consider the alarm network
  - Assume we sample variables in the order B,E,A,J,M
  - Initialize all variables at t = 0 to False

Gibbs Sampling: An Example



t	B	E	A	J	M
0	F	F	F	F	F
1	F				
2					
3					
4					

- Sampling P(B|A,E) at t = 1: Using Bayes Rule,

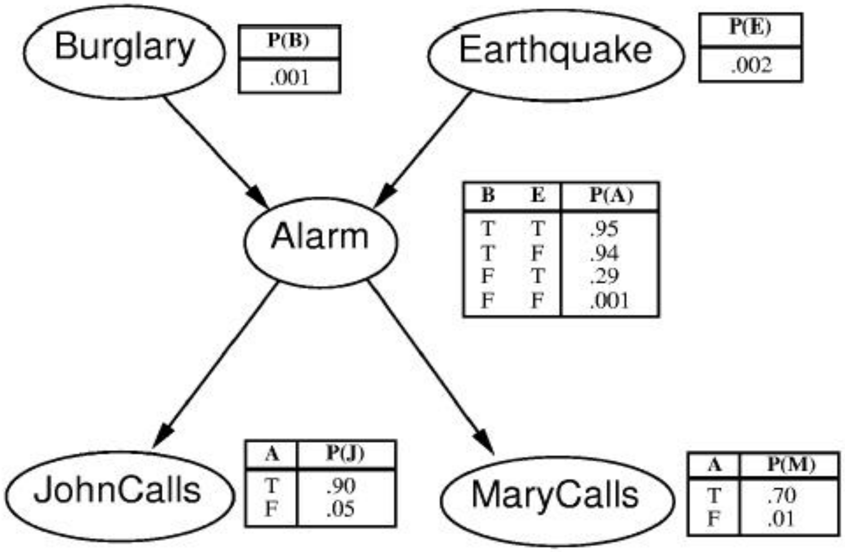
$$P(B \mid A, E) \propto P(A \mid B, E) P(B)$$

- A=false, E=false, so we compute:

$$P(B = T \mid A = F, E = F) \propto (0.06)(0.01) = 0.0006$$

$$P(B = F \mid A = F, E = F) \propto (0.999)(0.999) = 0.9980$$

Gibbs Sampling: An Example



t	B	E	A	J	M
0	F	F	F	F	F
1	F	T			
2					
3					
4					

- Sampling P(E|A,B): Using Bayes Rule,

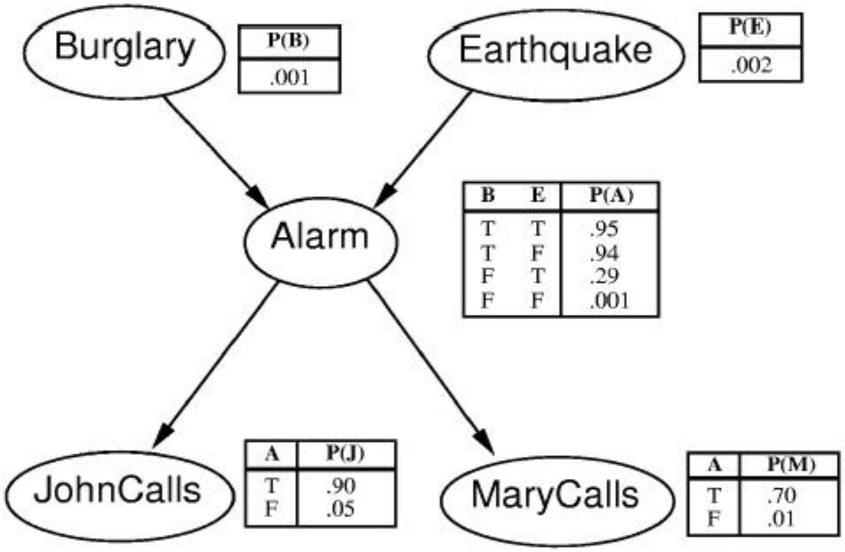
$$P(E \mid A, B) \propto P(A \mid B, E) P(E)$$

- (A,B) = (F,F), so we compute the following,

$$P(E = T \mid A = F, B = F) \propto (0.71)(0.02) = 0.0142$$

$$P(E = F \mid A = F, B = F) \propto (0.999)(0.998) = 0.9970$$

Gibbs Sampling: An Example



t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F		
2					
3					
4					

- Sampling P(A|B,E,J,M): Using Bayes Rule,

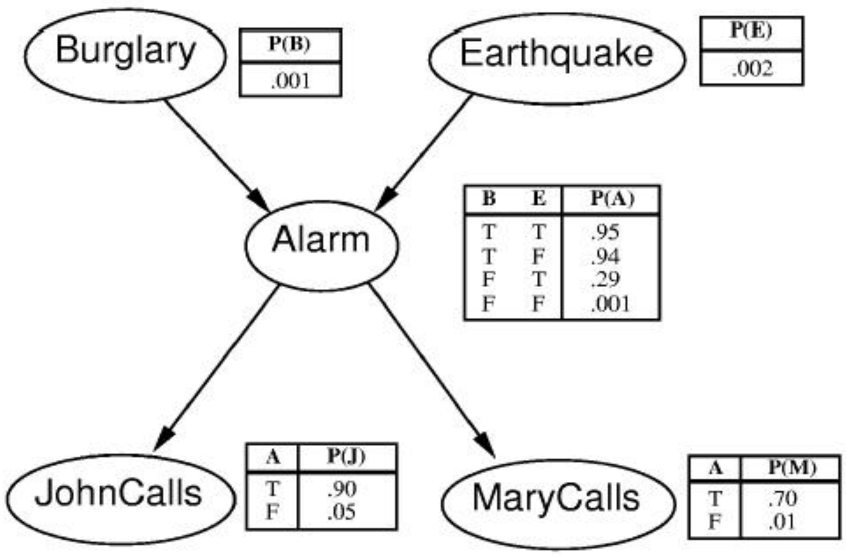
$$P(A \mid B,E,J,M) \propto P(J \mid A)P(M \mid A)P(A \mid B,E)$$

- (B,E,J,M) = (F,T,F,F), so we compute:

$$P(A=T \mid B=F,E=T,J=F,M=F) \propto (0.1)(0.3)(0.29) = 0.0087$$

$$P(A=F \mid B=F,E=T,J=F,M=F) \propto (0.95)(0.99)(0.71) = 0.6678$$

Gibbs Sampling: An Example



t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F	T	
2					
3					
4					

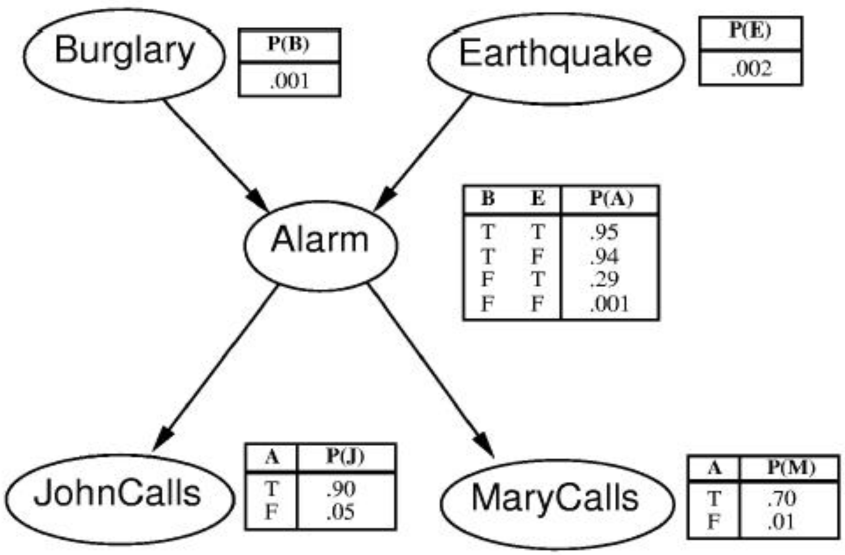
- Sampling P(J|A): No need to apply Bayes Rule

- A = F, so we compute the following, and sample

$$P(J = T \mid A = F) \propto 0.05$$

$$P(J = F \mid A = F) \propto 0.95$$

Gibbs Sampling: An Example



t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F	T	F
2					
3					
4					

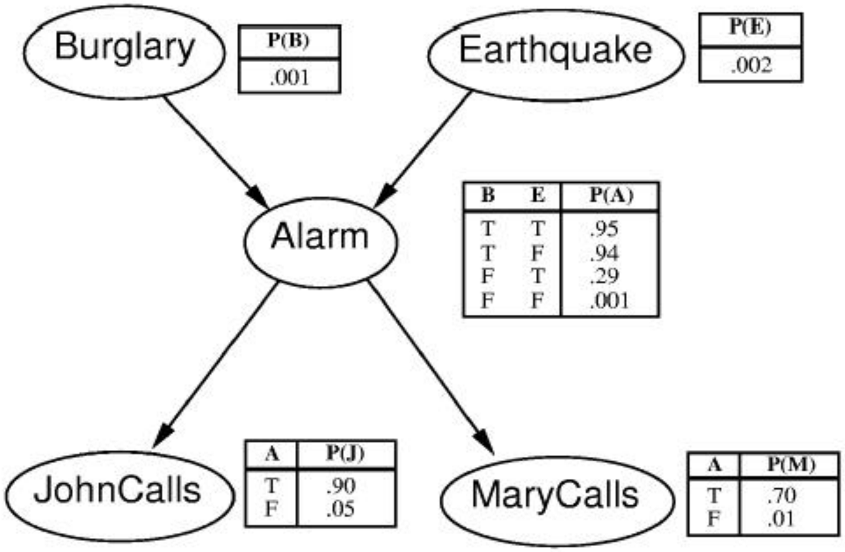
- Sampling P(M|A): No need to apply Bayes Rule

- A = F, so we compute the following, and sample

$$P(M = T \mid A = F) \propto 0.01$$

$$P(M = F \mid A = F) \propto 0.99$$

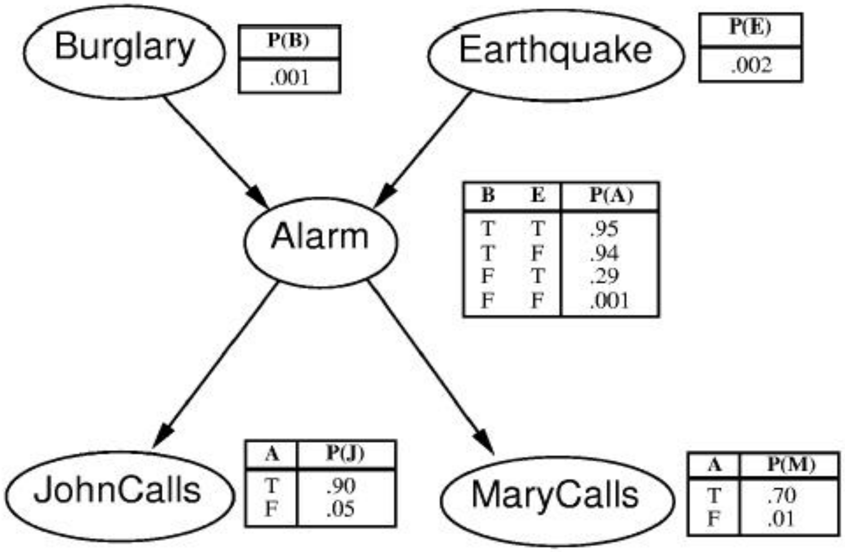
Gibbs Sampling: An Example



t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F	T	F
2	F	T	T	T	T
3					
4					

- Now t = 2, and we repeat the procedure to sample new values of B,E,A,J,M …

Gibbs Sampling: An Example



t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F	T	F
2	F	T	T	T	T
3	T	F	T	F	T
4	T	F	T	F	F

- Now t = 2, and we repeat the procedure to sample new values of B,E,A,J,M …

- And similarly for t = 3, 4, etc.

Gibbs Sampling is a special case of MH

- The GS proposal distribution is

$$Q(x_i', \mathbf{x}_{-i} \mid x_i, \mathbf{x}_{-i}) = P(x_i' \mid \mathbf{x}_{-i})$$

($\mathbf{x}_{-i}$ denotes all variables except x_i)

- Applying Metropolis-Hastings with this proposal, we obtain:

$$A(x_i', \mathbf{x}_{-i} \mid x_i, \mathbf{x}_{-i}) = \min\left(1, \frac{P(x_i', \mathbf{x}_{-i}) Q(x_i, \mathbf{x}_{-i} \mid x_i', \mathbf{x}_{-i})}{P(x_i, \mathbf{x}_{-i}) Q(x_i', \mathbf{x}_{-i} \mid x_i, \mathbf{x}_{-i})}\right)$$

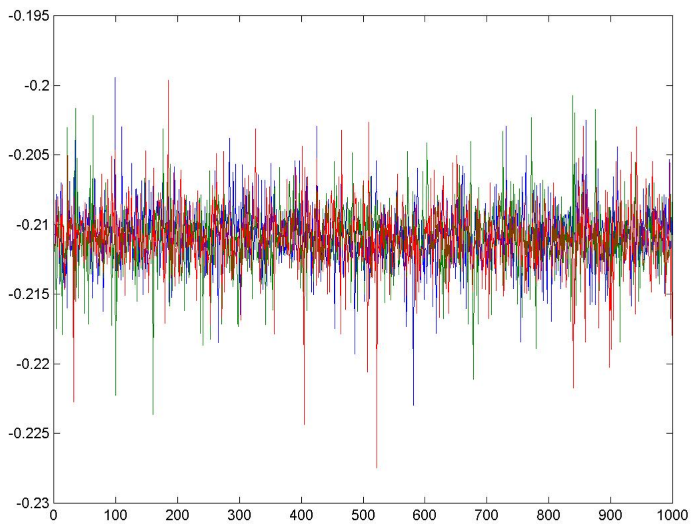
$$= \min\left(1, \frac{P(x_i', \mathbf{x}_{-i}) P(x_i \mid \mathbf{x}_{-i})}{P(x_i, \mathbf{x}_{-i}) P(x_i' \mid \mathbf{x}_{-i})}\right) = \min\left(1, \frac{P(x_i' \mid \mathbf{x}_{-i}) P(\mathbf{x}_{-i}) P(x_i \mid \mathbf{x}_{-i})}{P(x_i \mid \mathbf{x}_{-i}) P(\mathbf{x}_{-i}) P(x_i' \mid \mathbf{x}_{-i})}\right)$$

$$= \min(1,1) = 1$$

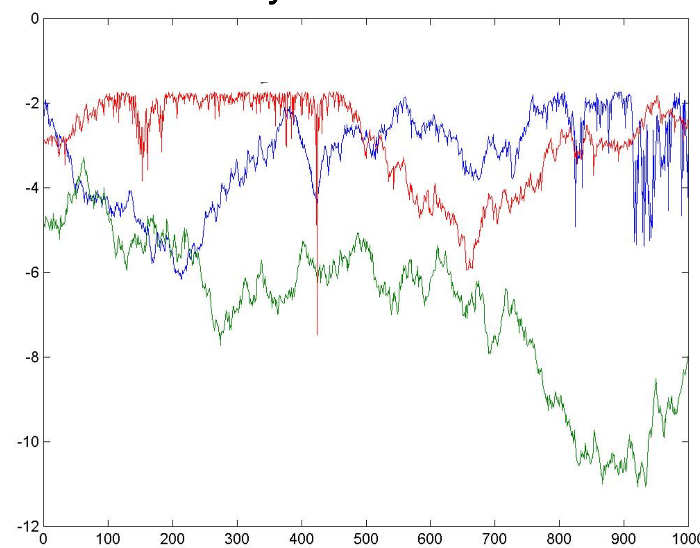
GS is simply MH with a proposal that is always accepted!

Sample Values vs Time

Well-mixed chains



Poorly-mixed chains



- Monitor convergence by plotting samples (of r.v.s) from multiple MH runs (chains)
  - If the chains are well-mixed (left), they are probably converged
  - If the chains are poorly-mixed (right), we should continue burn-in

Markov Chains

- A Markov Chain is a sequence of random variables $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$ with the Markov Property

$$P(x^{(n)} = x \mid x^{(1)}, \ldots, x^{(n-1)}) = P(x^{(n)} = x \mid x^{(n-1)})$$

  - $P(x^{(n)} = x \mid x^{(n-1)})$ is known as the transition kernel
  - The next state depends only on the preceding state – recall HMMs!
  - Note: the r.v.s $x^{(i)}$ can be vectors
    - We define $x^{(t)}$ to be the t-th sample of all variables in a graphical model
    - $X^{(t)}$ represents the entire state of the graphical model at time t

- We study homogeneous Markov Chains, in which the transition kernel $P(x^{(t)} = x \mid x^{(t-1)})$ is fixed with time
  - To emphasize this, we will call the kernel $T(x' \mid x)$, where x is the previous state and x' is the next state

Markov Chain Concepts

- To understand MCs, we need to define a few concepts:
  - Probability distributions over states: $\pi^{(t)}(x)$ is a distribution over the state of the system x, at time t
    - When dealing with MCs, we don't think of the system as being in one state, but as having a distribution over states
    - For graphical models, remember that x represents <u>all</u> variables
  - Transitions: recall that states transition from $x^{(t)}$ to $x^{(t+1)}$ according to the transition kernel $T(x' \mid x)$. We can also transition entire distributions:

$$\pi^{(t+1)}(x') = \sum_x \pi^{(t)}(x) T(x' \mid x)$$

    - At time t, state x has probability mass $\pi^{(t)}(x)$. The transition probability redistributes this mass to other states x'.
  - Stationary distributions: $\pi(x)$ is stationary if it does not change under the transition kernel:

$$\pi(x') = \sum_x \pi(x) T(x' \mid x) \qquad \text{for all } x'$$

Markov Chain Concepts

- Stationary distributions are of great importance in MCMC. To understand them, we need to define some notions:
  - Irreducible: an MC is irreducible if you can get from any state x to any other state x' with probability > 0 in a finite number of steps
    - i.e. there are no unreachable parts of the state space
    - This is a function of the transition kernel!
  - Aperiodic: an MC is aperiodic if you can return to any state x at any time
    - Periodic MCs have states that need ≥2 time steps to return to (cycles)
  - Ergodic (or **regular**): an MC is ergodic if it is irreducible and aperiodic
- Ergodicity is important: it implies you can reach the stationary distribution $\pi_{st}(x)$, no matter the initial distribution $\pi^{(0)}(x)$
  - All good MCMC algorithms must satisfy ergodicity, so that you can't initialize in a way that will never converge

Markov Chain Concepts

- Reversible (detailed balance): an MC is reversible if there exists a distribution $\pi(x)$ such that the detailed balance condition is satisfied:

$$\pi(x')T(x \mid x') = \pi(x)T(x' \mid x)$$

  - Probability of x'→x is the same as x→x'

- $\pi(x)$ is a stationary distribution of the MC! Proof:

$$\pi(x')T(x \mid x') = \pi(x)T(x' \mid x)$$
$$\sum_x \pi(x')T(x \mid x') = \sum_x \pi(x)T(x' \mid x)$$
$$\pi(x')\sum_x T(x \mid x') = \sum_x \pi(x)T(x' \mid x)$$
$$\pi(x') = \sum_x \pi(x)T(x' \mid x)$$

  - The last line is the definition of a stationary distribution!

Why does Metropolis-Hastings work?

- Recall that we draw a sample x' according to Q(x'|x), and then accept/reject according to A(x'|x).
  - In other words, the transition kernel is

$$T(x' \mid x) = Q(x' \mid x) A(x' \mid x)$$

- We can prove that MH is reversible:
  - Recall that

$$A(x' \mid x) = \min\left(1, \frac{P(x')Q(x \mid x')}{P(x)Q(x' \mid x)}\right)$$

  - Notice this implies the following:

if $A(x' \mid x) < 1$ then $\dfrac{P(x)Q(x' \mid x)}{P(x')Q(x \mid x')} > 1$ and thus $A(x \mid x') = 1$

Why does Metropolis-Hastings work?

if $A(x'|x) < 1$ then $\frac{\pi(x)Q(x'|x)}{\pi(x')Q(x|x')} > 1$ and thus $A(x|x') = 1$

- Now suppose A(x'|x) < 1 and A(x|x') = 1. We have

$$A(x'|x) = \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}$$
$$P(x)Q(x'|x)A(x'|x) = P(x')Q(x|x')$$
$$P(x)Q(x'|x)A(x'|x) = P(x')Q(x|x')A(x|x')$$
$$P(x)T(x'|x) = P(x')T(x|x')$$

- The last line is exactly the **detailed balance condition**
  - In other words, the MH algorithm leads to a stationary distribution P(x)
  - Recall we defined P(x) to be the true distribution of x
  - Thus, the MH algorithm eventually converges to the true distribution!

Why does Metropolis-Hastings work?

- Theorem: If a Markov chain is **regular** and satisfies **detailed balance** with respect to p(x), then p(x) is its unique stationary distribution

- Easy to verify that Gibbs sampling satisfies aperiodicity and is irreducible, and thus is regular

- The *mixing time*, or how long it takes to **reach** something close the stationary distribution, can be very long

Summary

- Markov Chain Monte Carlo methods use adaptive proposals $Q(x'|x)$ to sample from the true distribution $P(x)$
- Metropolis-Hastings allows you to specify any proposal $Q(x'|x)$
  - But choosing a good $Q(x'|x)$ requires care
- Gibbs sampling sets the proposal $Q(x'|x)$ to the conditional distribution $P(x'|x)$
  - Acceptance rate always 1!
  - But remember that high acceptance usually entails slow exploration
  - In fact, there are better MCMC algorithms for certain models
- Knowing when to halt burn-in is an art