Probability, Approximate Inference, and Sampling

Shichang Zhang

UCLA

Slides adapted from Rob Hall, Eric Xing, Qirong Ho (CMU), Stefano Ermon, Yumeng Zhang (Stanford), and David Sontag (MIT)

Agenda

- Quick Recap
- Markov Chain Monte Carlo (MCMC)
  - Theoretical Aspects of MCMC
- Gibbs Sampling and Practical MCMC

Recap

- Last time we talked about sampling methods. Most importantly the following two concepts.
- Monte Carlo estimation
  - Write any probability query we care about as an expectation. Then use the sample mean as an unbiased estimator.
- Importance sampling
  - The idea is to sample the nonevidence variables directly.
  - We first find a proposal distribution Q over the nonevidence variables Z. Then we compute the importance weight P/Q for estimation.

Generate samples from Q and estimate $P(E = e)$ using the following Monte Carlo estimate:

$$\hat{P}(E = e) = \frac{1}{T} \sum_{t=1}^{T} \frac{P(Z = z^t, E = e)}{Q(Z = z^t)} = \frac{1}{T} \sum_{t=1}^{T} w(z^t)$$

where $(z^1, \dots, z^T)$ are sampled from Q.

Recap

- Error bound of importance sampling
  - μ (think of it as proposal distribution Q) and v (think of it as true distribution P) are two probability measures on a set X, v is absolutely continuous with respect to μ (i.e. μ(A) = 0 implies v(A) = 0). ρ is the probability density of v with respect to μ (ρ = dv/dμ, which is roughly the probability ratio)
  - Our target, the expectation we want to estimate $I(f) := \int_{\mathcal{X}} f(y) d\nu(y)$
  - Our estimation, result of the importance sampling $I_n(f) := \frac{1}{n} \sum_{i=1}^{n} f(X_i)\rho(X_i)$

Theorem 1.1. *Let $\mathcal{X}$, μ, ν, ρ, f, $I(f)$ and $I_n(f)$ be as above. Let Y be an $\mathcal{X}$-valued random variable with law ν. Let $L = D(\nu||\mu)$ be the Kullback–Leibler divergence of μ from ν, that is,*

$$L = D(\nu||\mu) = \int_{\mathcal{X}} \rho(x) \log \rho(x) d\mu(x) = \int_{\mathcal{X}} \log \rho(y) d\nu(y) = \mathbb{E}(\log \rho(Y)) .$$

Let $\|f\|_{L^2(\nu)} := (\mathbb{E}(f(Y)^2))^{1/2}$. If $n = \exp(L + t)$ for some $t \geq 0$, then

$$\mathbb{E}|I_n(f) - I(f)| \leq \|f\|_{L^2(\nu)} \left(e^{-t/4} + 2\sqrt{\mathbb{P}(\log \rho(Y) > L + t/2)} \right) .$$

Recap

- We want the proposal distribution Q to be close to the actual distribution P

that under a certain condition that often holds in practice, the sample size n required for $|I_n(f) - I(f)|$ to be close to zero with high probability is roughly $\exp(D(\nu \parallel \mu))$ where $D(\nu \parallel \mu)$ is the Kullback–Leibler divergence of μ from ν. More precisely, it says that if s is the typical order of fluctuations of $\log \rho(Y)$ around its expected value, then a sample of size $\exp(D(\nu \parallel \mu) + O(s))$ is sufficient and a sample of size $\exp(D(\nu \parallel \mu) - O(s))$ is necessary for $|I_n(f) - I(f)|$ to be close to zero with high probability. The necessity is proved by considering the worst possible f, which as it turns out, is the function that is identically equal to 1.

Theorem 1.1. *Let $\mathcal{X}$, μ, ν, ρ, f, $I(f)$ and $I_n(f)$ be as above. Let Y be an $\mathcal{X}$-valued random variable with law ν. Let $L = D(\nu||\mu)$ be the Kullback–Leibler divergence of μ from ν, that is,*

$$L = D(\nu||\mu) = \int_{\mathcal{X}} \rho(x) \log \rho(x) d\mu(x) = \int_{\mathcal{X}} \log \rho(y) d\nu(y) = \mathbb{E}(\log \rho(Y)) .$$

Let $\|f\|_{L^2(\nu)} := (\mathbb{E}(f(Y)^2))^{1/2}$. If $n = \exp(L + t)$ for some $t \geq 0$, then

$$\mathbb{E}|I_n(f) - I(f)| \leq \|f\|_{L^2(\nu)} \big(e^{-t/4} + 2\sqrt{\mathbb{P}(\log \rho(Y) > L + t/2)}\big) .$$

Agenda

- Quick Recap
- Markov Chain Monte Carlo (MCMC)
  - Theoretical Aspects of MCMC
- Gibbs Sampling and Practical MCMC

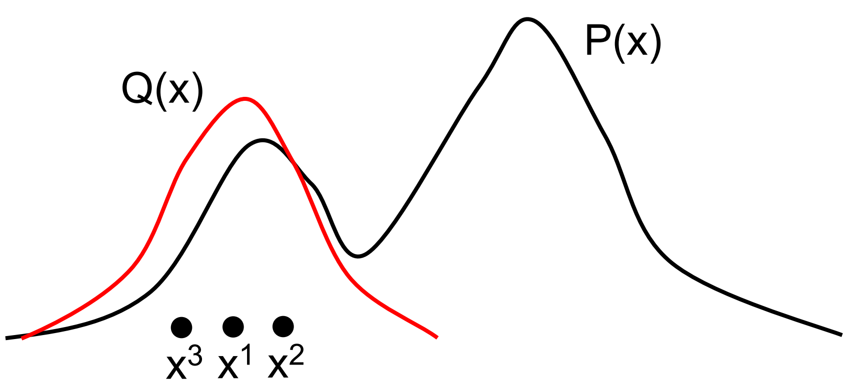
Limitations of IS

- Does 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?

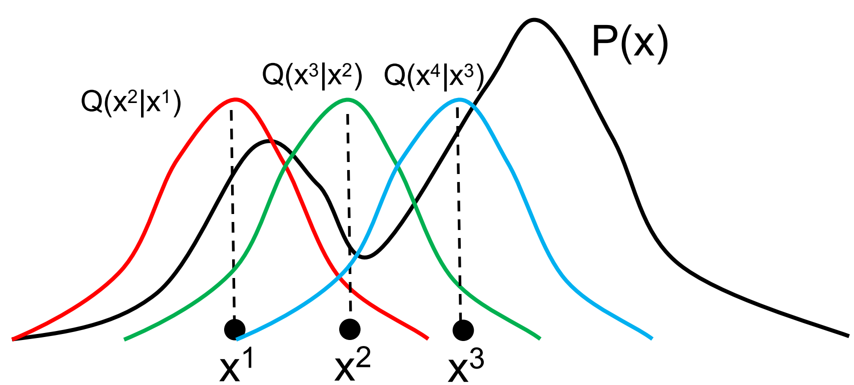
Markov Chain Monte Carlo

- 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 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)

Metropolis-Hastings 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

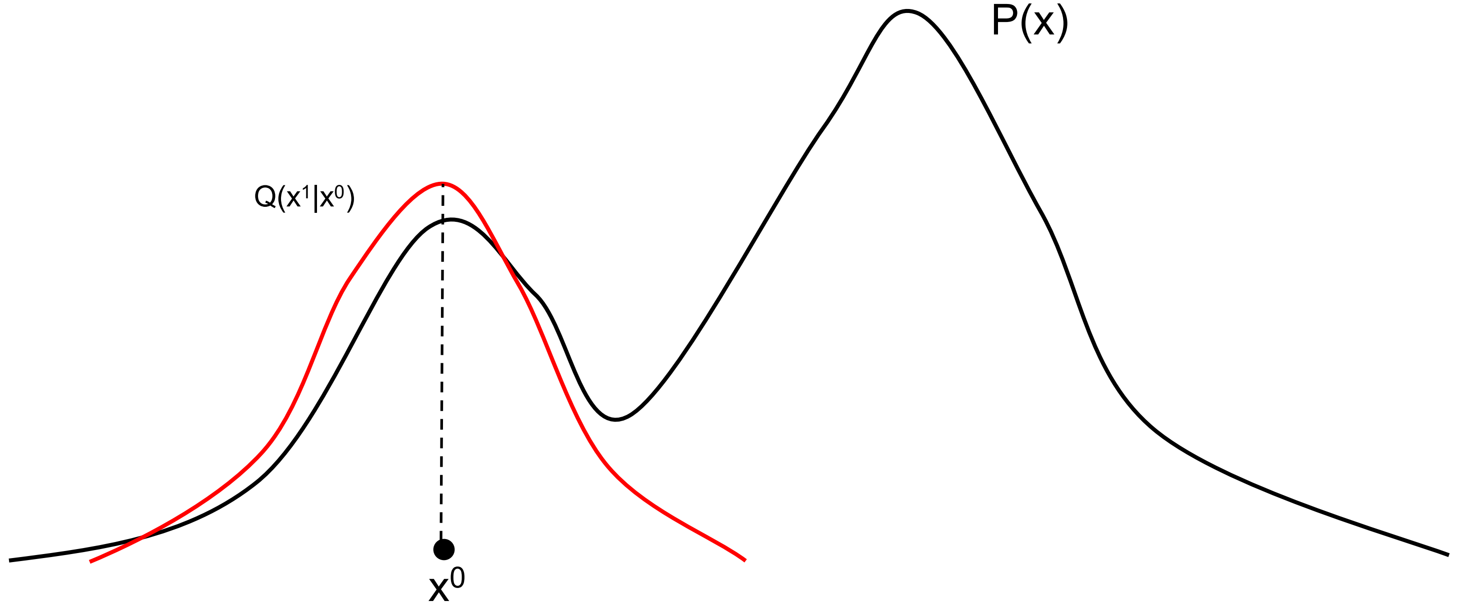
   $\Big\}$ Function Draw sample (x(t))
3. Take samples from P(x): Reset t=0, for t=1:N
   - x(t+1) ← Draw sample (x(t))
4. Monte Carlo Estimation using these N final samples

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 (it is symmetric)
  - We're trying to sample from a bimodal distribution P(x)

Initialize $x^{(0)}$
…

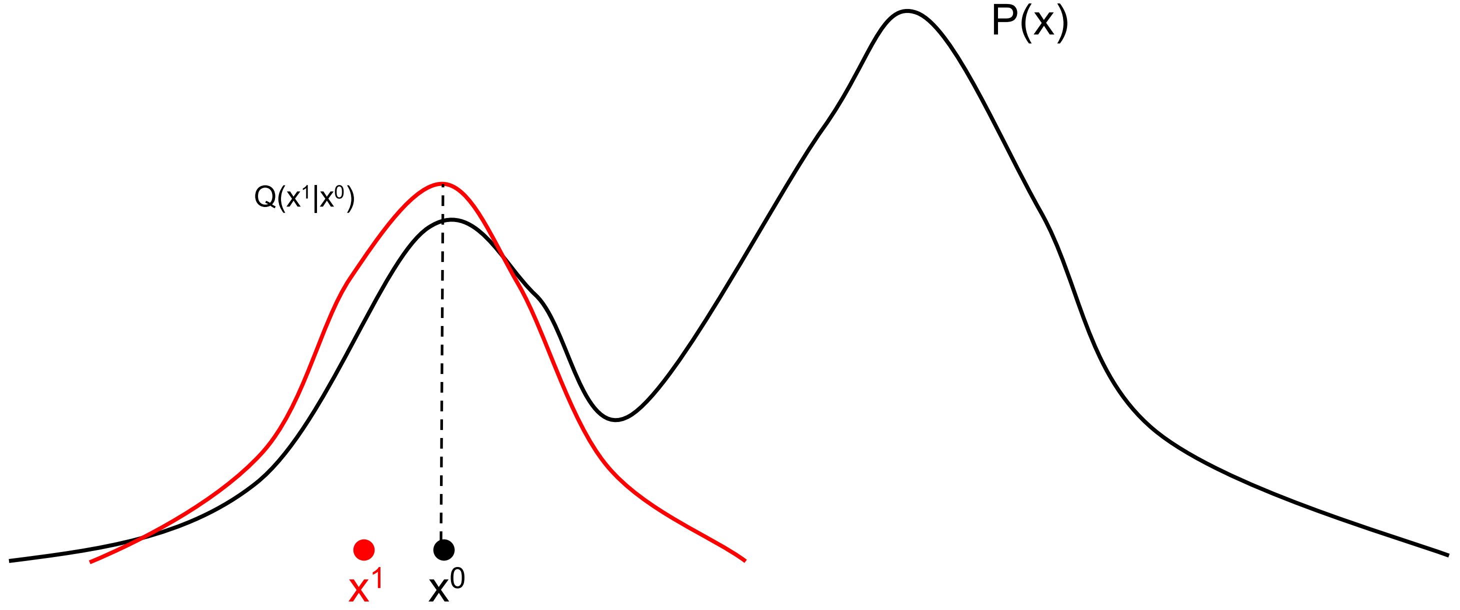


$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

The MH Algorithm

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

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

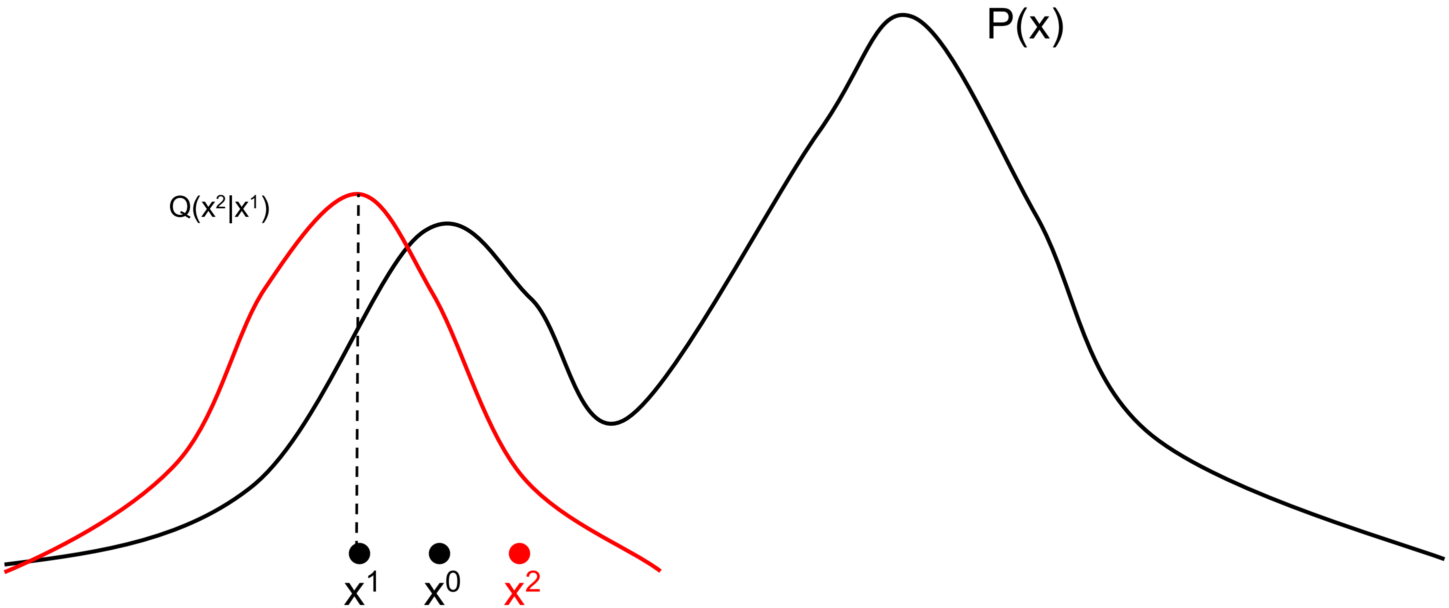


The MH Algorithm

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

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

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

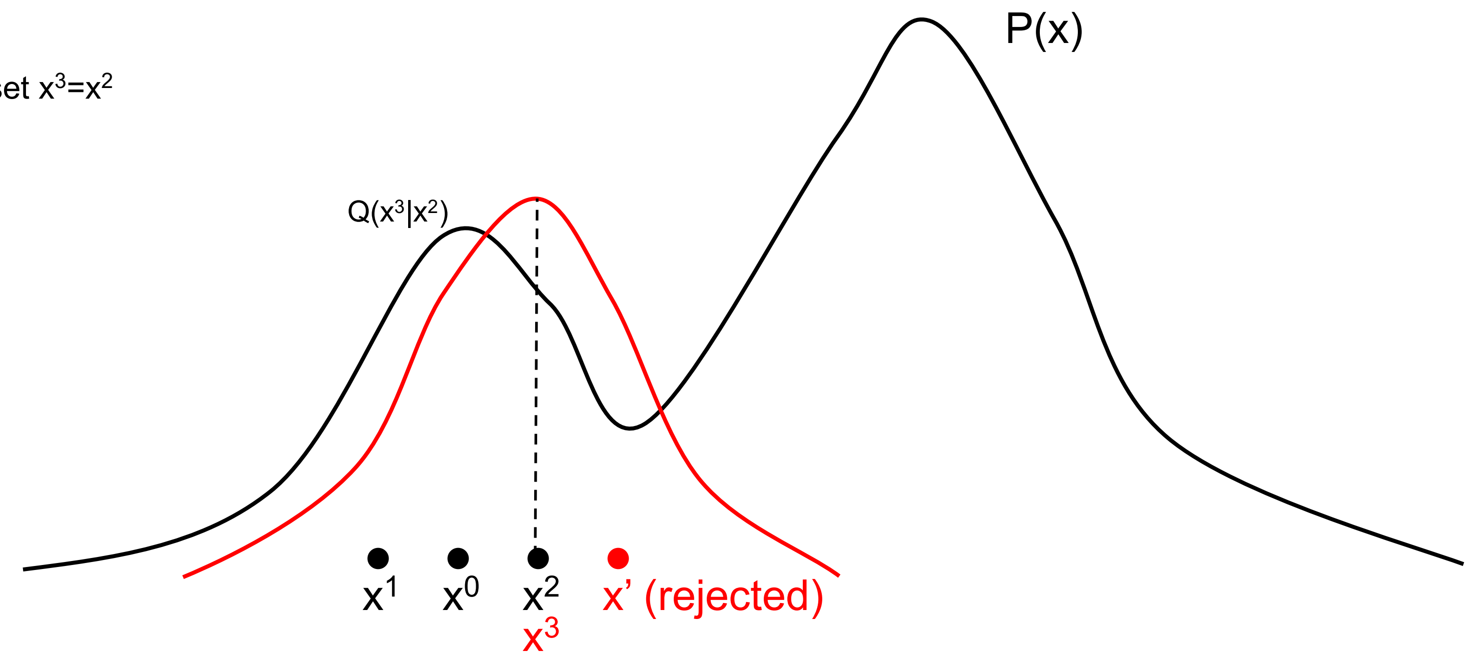


The MH Algorithm

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

- Example:
  - Let Q(x'|x) be a Gaussian centered on x (it is symmetric)
  - 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 $x^3=x^2$



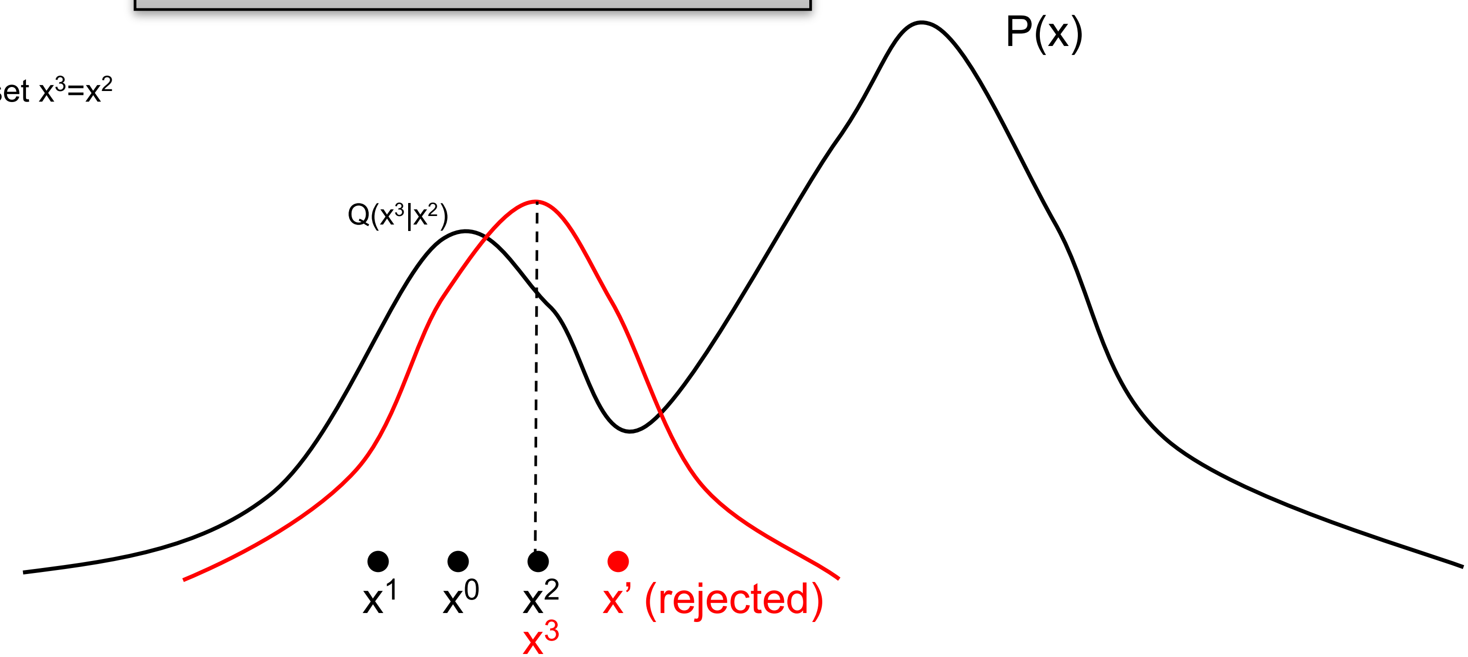
$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

The MH Algorithm

- Example:
  - Let Q(x'|x) be a Gaussian centered on x (it is symmetric)
  - 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!

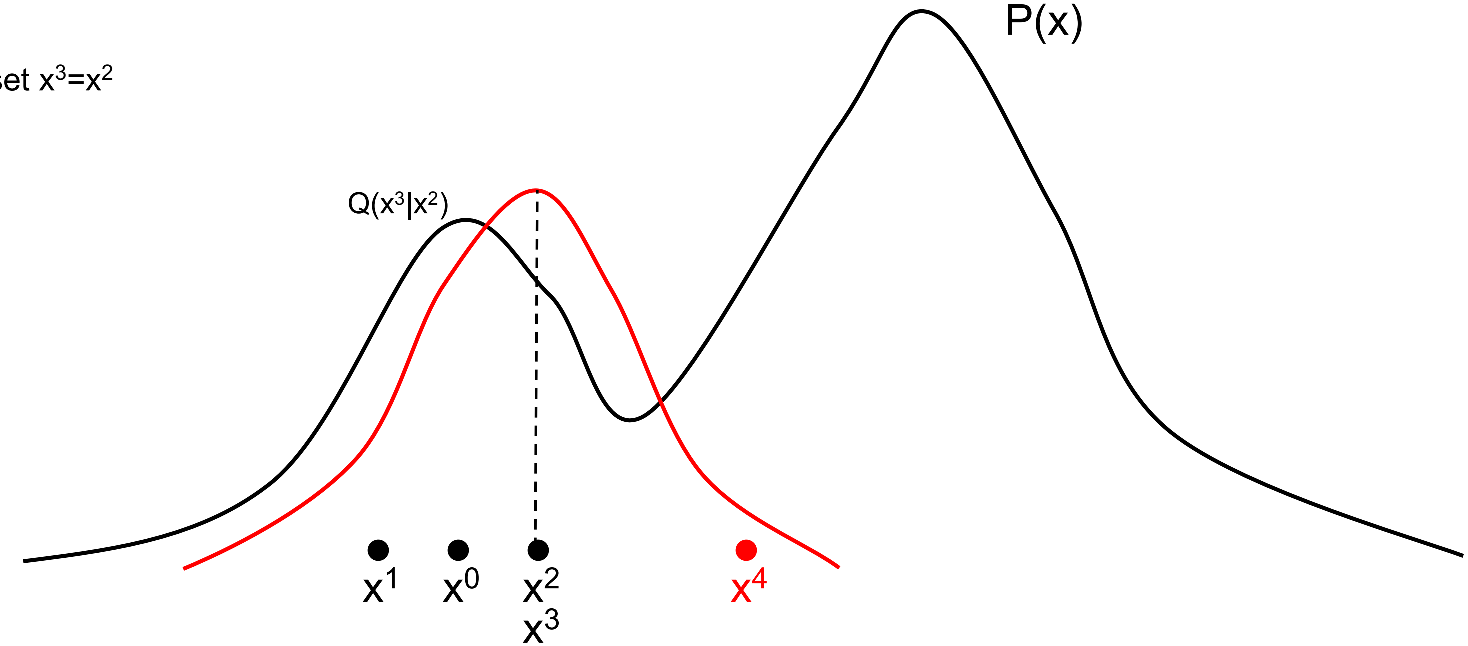


$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

The MH Algorithm

- Example:
  - Let Q(x'|x) be a Gaussian centered on x (it is symmetric)
  - 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

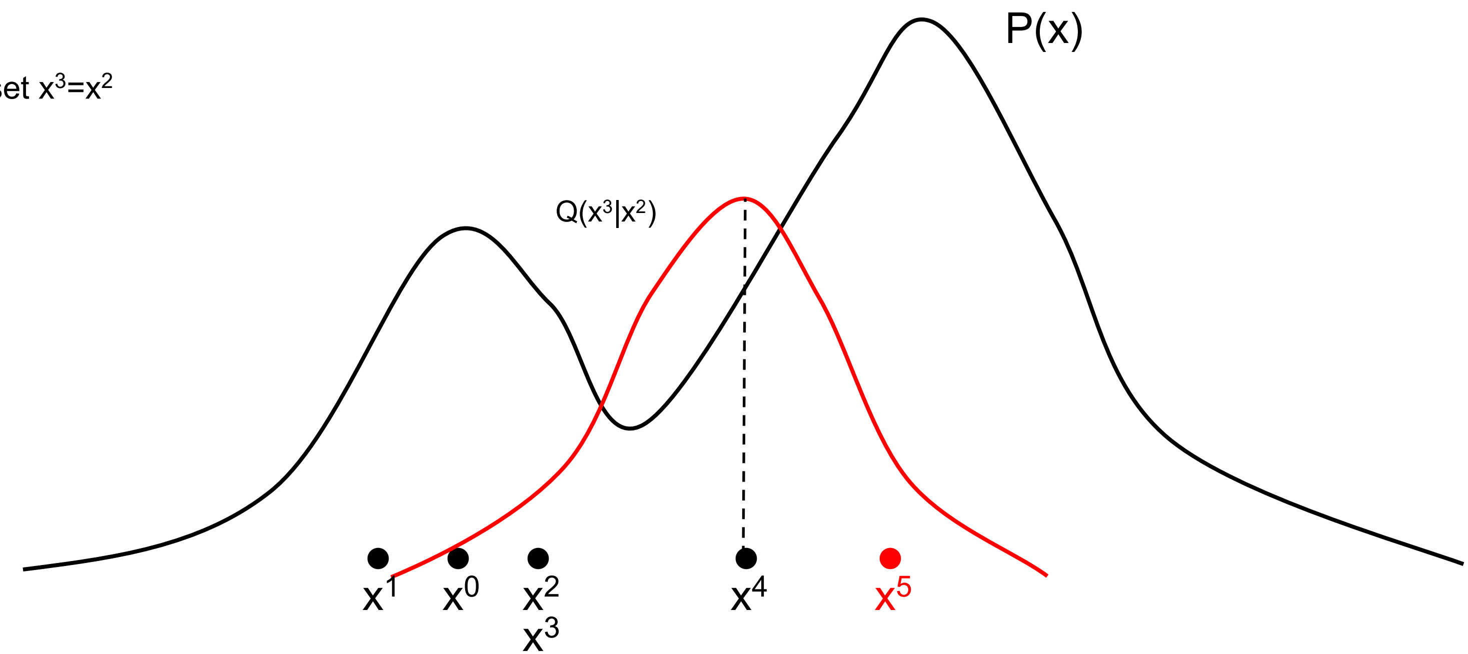


The MH Algorithm

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

- Example:
  - Let Q(x'|x) be a Gaussian centered on x (it is symmetric)
  - 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



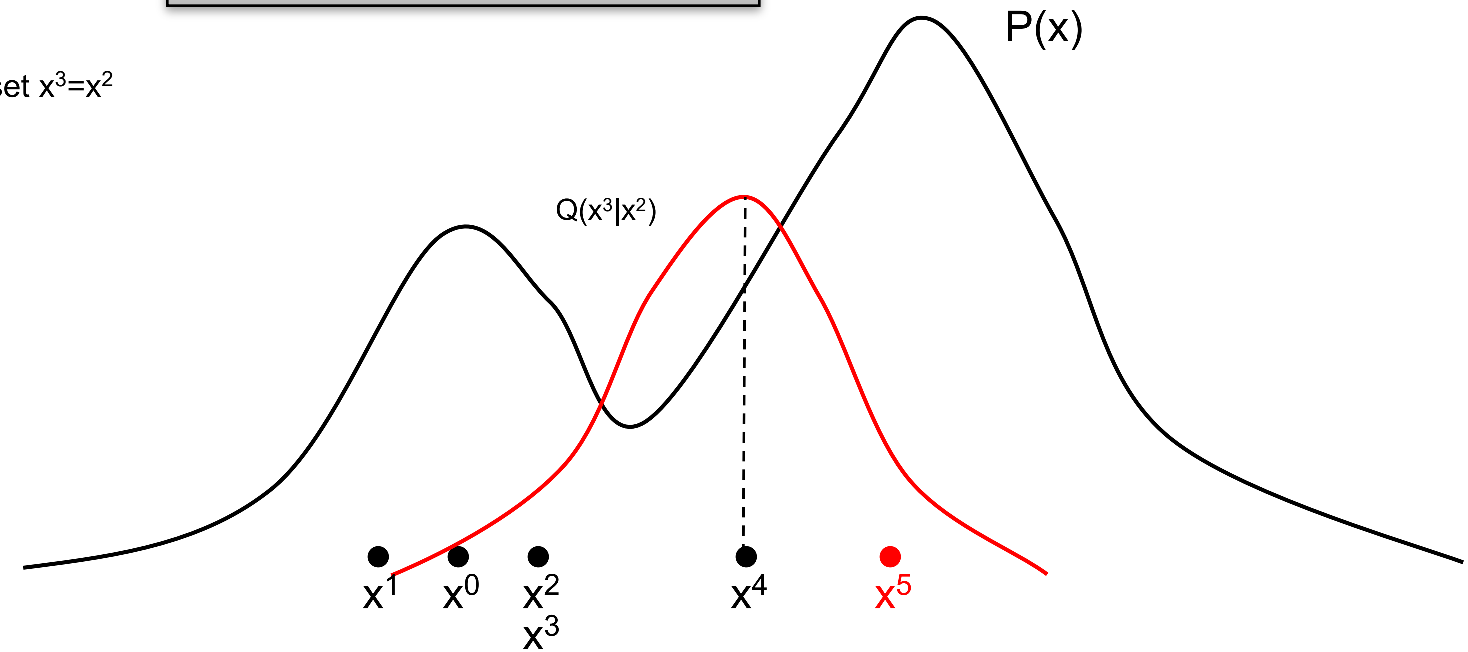
The MH Algorithm

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

- Example:
  - Let Q(x'|x) be a Gaussian centered on x (it is symmetric)
  - 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)!



Agenda

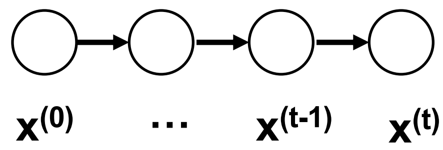
- Quick Recap
- Markov Chain Monte Carlo (MCMC)
  - Theoretical Aspects of MCMC
- Gibbs Sampling and Practical MCMC

Theoretical Aspects of MCMC

- The MH algorithm has a "burn-in"/"warm-up" period. We throw away all the samples we get from this period. Why?
- 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)?
- What are good, general-purpose, proposal distributions?

Markov Chains

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

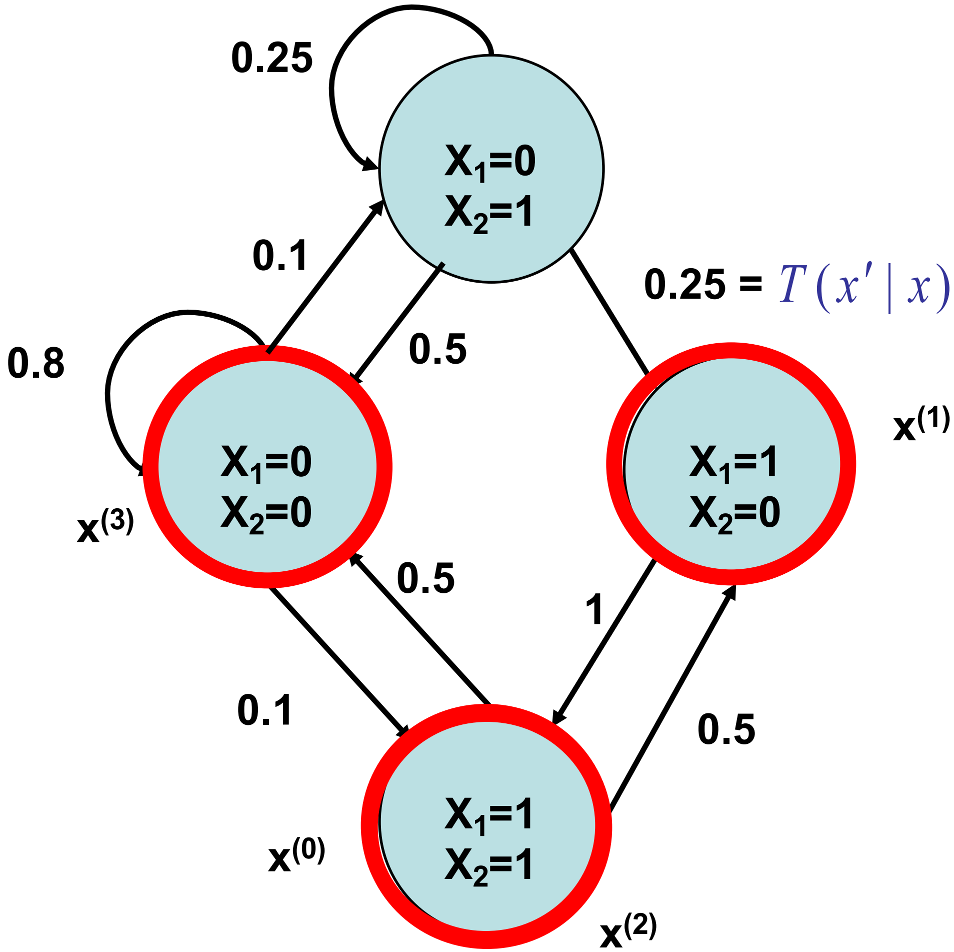


$$P(x^{(t)} = x \mid x^{(1)},\ldots,x^{(t-1)}) = P(x^{(t)} = x \mid x^{(t-1)})$$

  - $P(x^{(t)} = x \mid x^{(t-1)})$ is known as the transition kernel (just a matrix for discrete random variables)
  - The whole process is completely determined by the transition kernel and the initial state. The next state depends only on the preceding state
  - Note: the random variable $x^{(i)}$ can be vectors
    - We define $x^{(t)}$ to be the t-th sample of all variables in our model

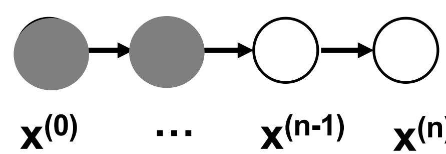
- We study homogeneous Markov Chains, in which the transition kernel $P(x^{(t)} = x' \mid x^{(t-1)} = x)$ is fixed with time
  - To emphasize this, we will call the kernel $T(x' \mid x)$, where x is the previous state and x' is the next state

Markov Chains



0.25 = $T(x' \mid x)$ = T (x' = (1,0) | x=(0,1))

Markov Chain Sampling = simulating the dynamics of a Markov Chain



Randomly pick an outgoing edge (sample $x^{(1)}$ given $x^{(0)}$ =(1,1))

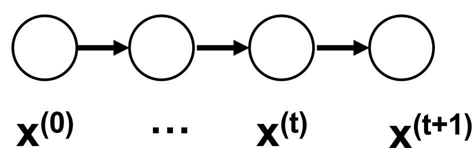
Initialize the simulation in one state (or randomly) $x^{(0)}$

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
    - Here 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 transit the entire distribution:

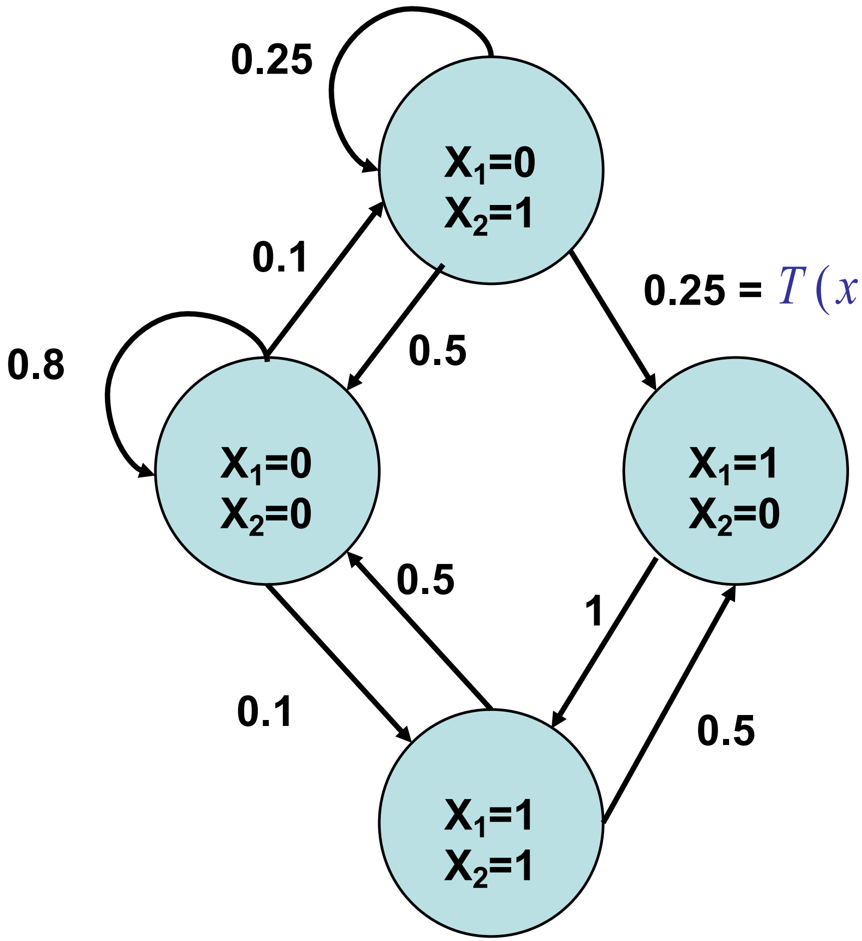
$$\pi^{(t+1)}(x') = \sum_x \pi^{(t)}(x) T(x' \mid x)$$

    - At time t, state x has probability mass $\pi^{(t)}(x)$. The transition probability redistributes this mass to other states x'.



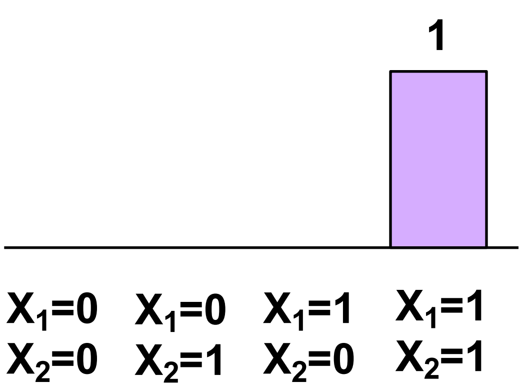
$$\pi^{(0)}(x_0) \quad \pi^{(t)}(x) \quad \pi^{(t+1)}(x')$$

Markov Chains



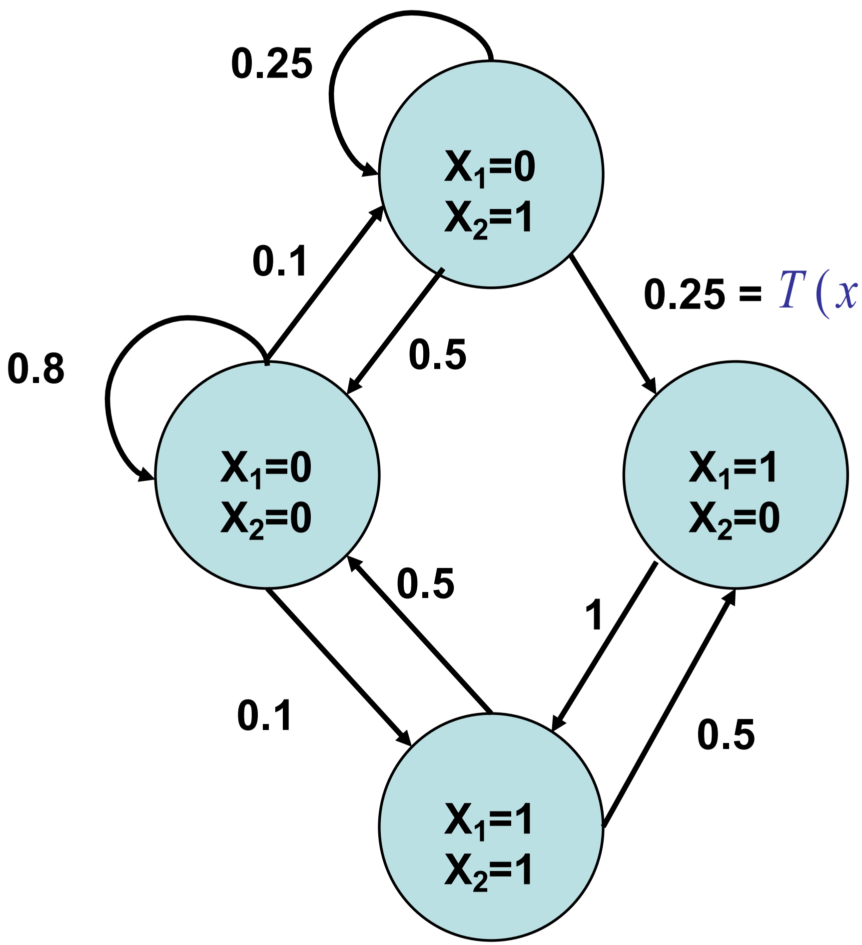
0.25 = $T(x' \mid x)$ = T (x' = (1,0) | x=(0,1))

$\pi^{(0)}(x)$



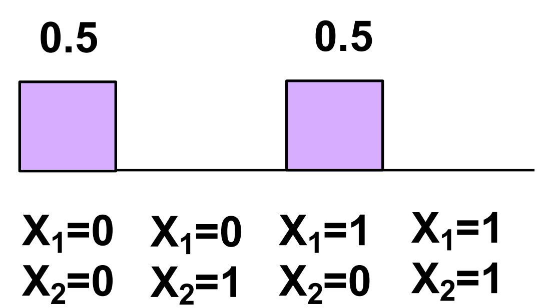
Initialize the simulation in one state $x^{(0)}$

Markov Chains



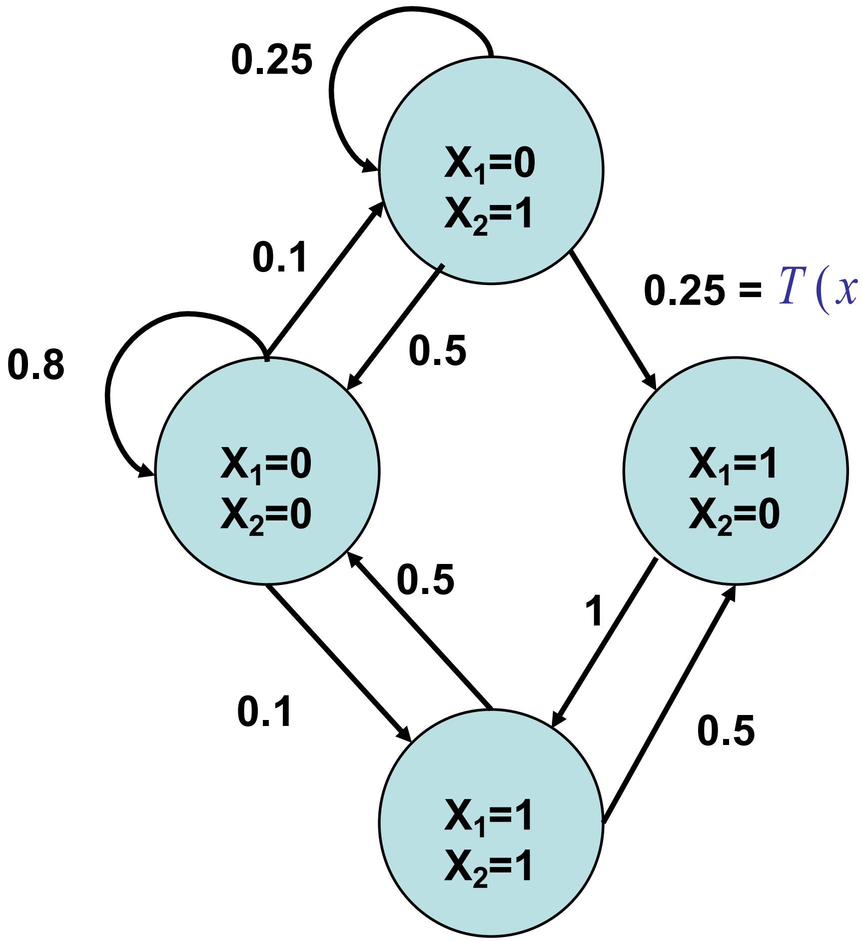
0.25 = $T(x' \mid x)$ = T (x' = (1,0) | x=(0,1))

$\pi^{(1)}(x)$



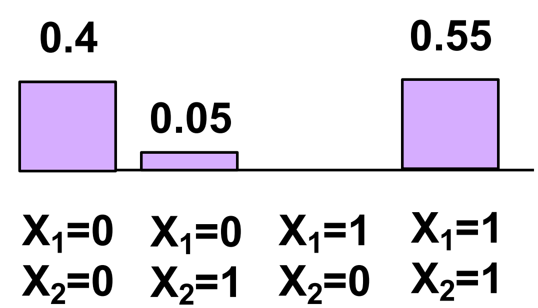
Initialize the simulation in one state $x^{(0)}$

Markov Chains



$0.25 = T(x' \mid x)$ = T (x’ = (1,0) | x=(0,1))

$\pi^{(2)}(x)$



stationary if it does not change

Initialize the simulation in one state $x^{(0)}$

Stationary Distribution

- $\pi(x)$ is stationary if it does not change under the transition kernel $T(x' \mid x)$

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

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

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

  - This is saying under the distribution $\pi(x)$, the probability of x'→x is the same as x→x'

- Theorem: $\pi(x)$ is a stationary distribution of the MC if it is reversible

Stationary Distribution

- $\pi(x)$ 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 MH 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 MH is reversible, i.e. stationary distribution exists:
  - 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 MH Work?

if $A(x'|x) < 1$ then $\dfrac{P(x)Q(x'|x)}{P(x')Q(x|x')} > 1$ 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

Why Does MH Work?

- P(x) is its unique stationary distribution.
- However, the *mixing time*, or how long it takes to **reach** something close the stationary distribution, can’t be guaranteed.

Agenda

- Quick Recap
- Markov Chain Monte Carlo (MCMC)
  - Theoretical Aspects of MCMC
- Gibbs Sampling and Practical MCMC

Gibbs Sampling

- Gibbs Sampling is a special case of the MH algorithm
- Gibbs Sampling samples each random variable one at a time. Therefore, it has reasonable computation and memory requirements

Gibbs Sampling Algorithm

- Suppose the model contains variables $x_1,\ldots,x_n$
- Initialize starting values for $x_1,\ldots,x_n$
- 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

Gibbs Sampling is 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:

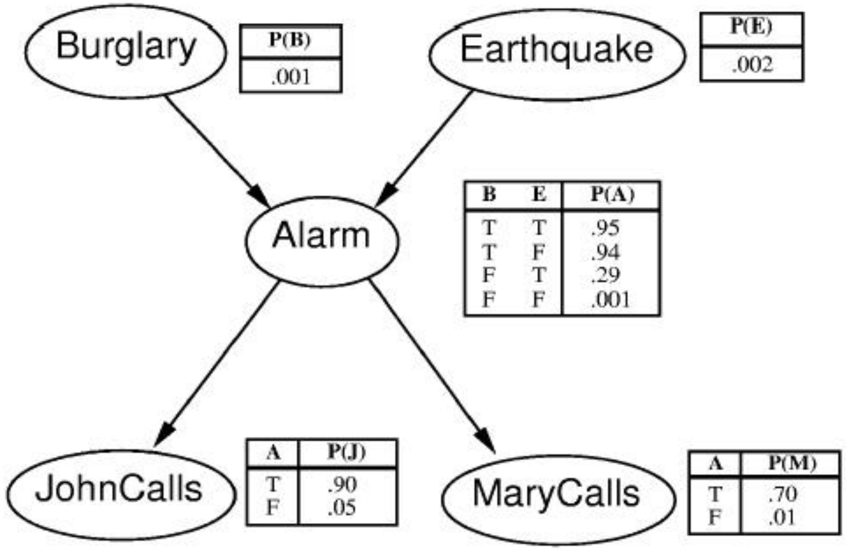
$$A(x_i', \mathbf{x}_{-i} \mid x_i, \mathbf{x}_{-i}) = \min\left(1, \frac{P(x_i', \mathbf{x}_{-i})Q(x_i, \mathbf{x}_{-i} \mid x_i', \mathbf{x}_{-i})}{P(x_i, \mathbf{x}_{-i})Q(x_i', \mathbf{x}_{-i} \mid x_i, \mathbf{x}_{-i})}\right)$$

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

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

GS is simply MH with a proposal that is always accepted

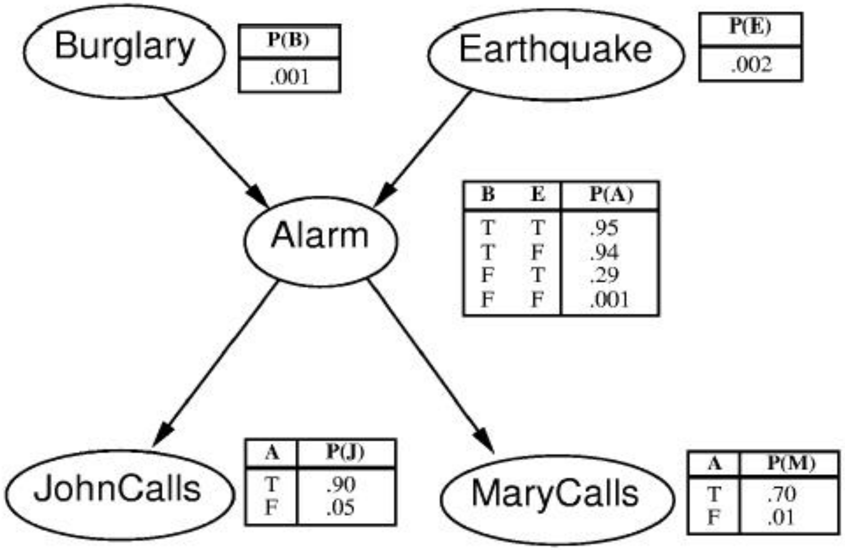
Gibbs Sampling: An Example



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,

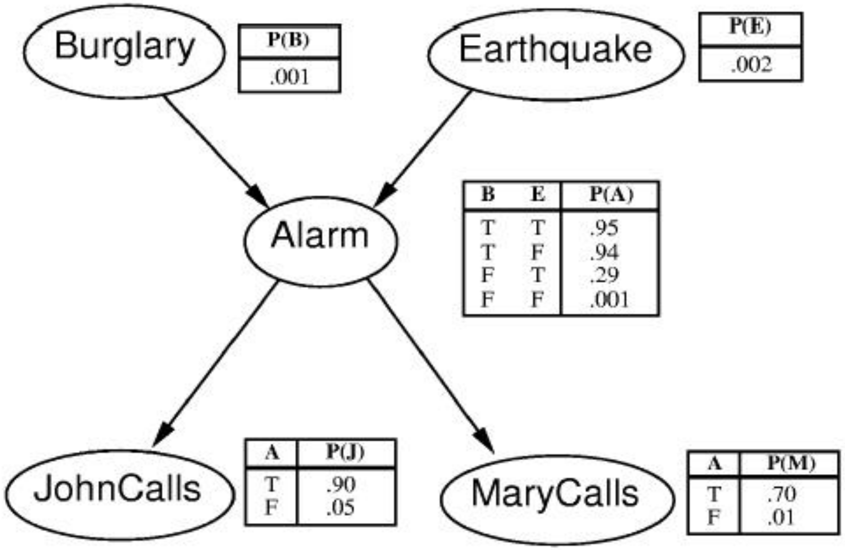
$$P(B \mid A, E) \propto P(A \mid B, E) P(B)$$

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

$$P(B = T \mid A = F, E = F) \propto (0.06)(0.001) = 0.00006$$

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

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,

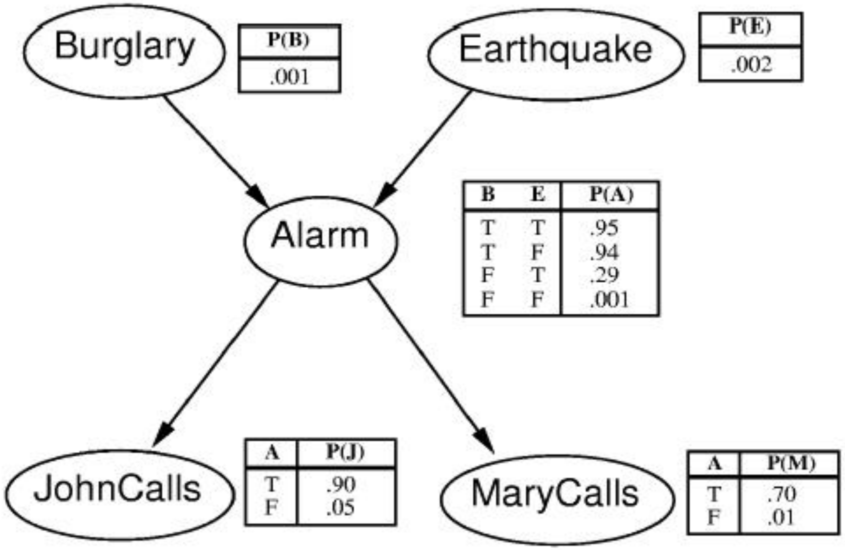
$$P(E \mid A, B) \propto P(A \mid B, E) P(E)$$

- (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,

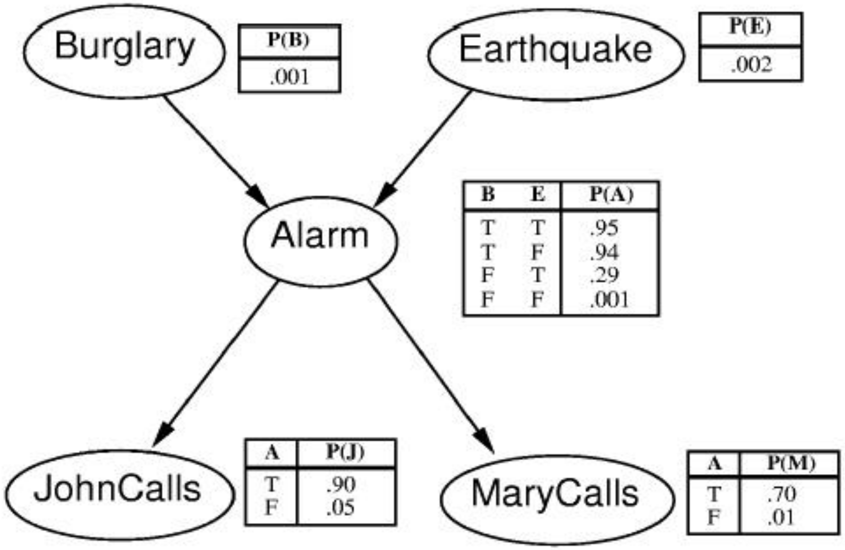
$$P(A \mid B, E, J, M) \propto P(J \mid A) P(M \mid A) P(A \mid B, E)$$

- (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



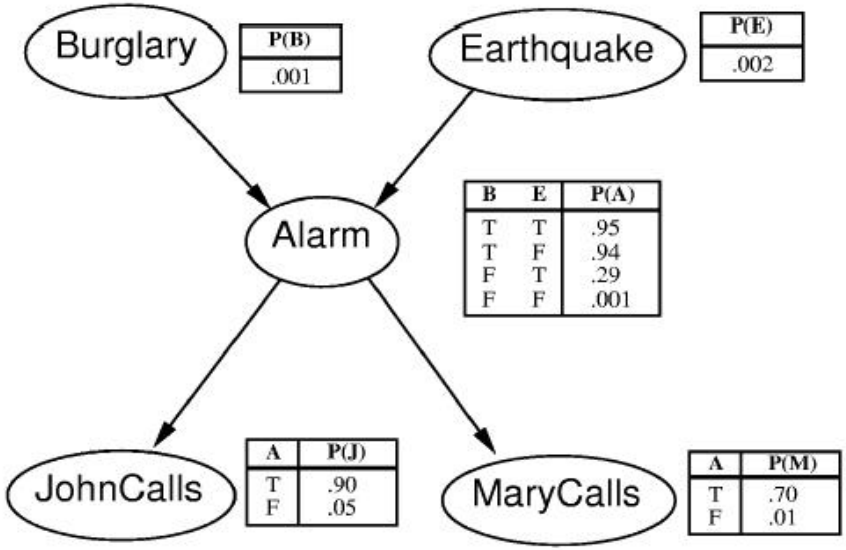
t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F	T	
2					
3					
4					

- 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



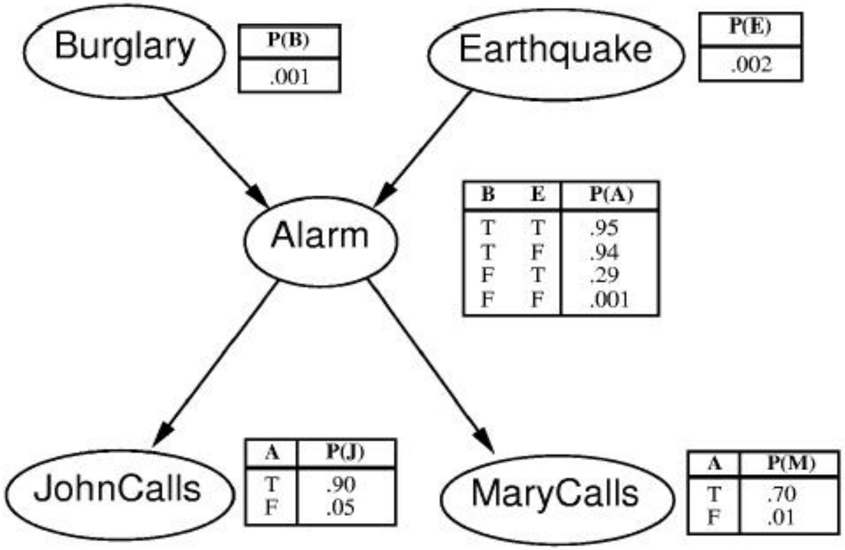
t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F	T	F
2					
3					
4					

- Sampling P(M|A): No need to apply Bayes Rule
- A = F, so we compute the following, and sample

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

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

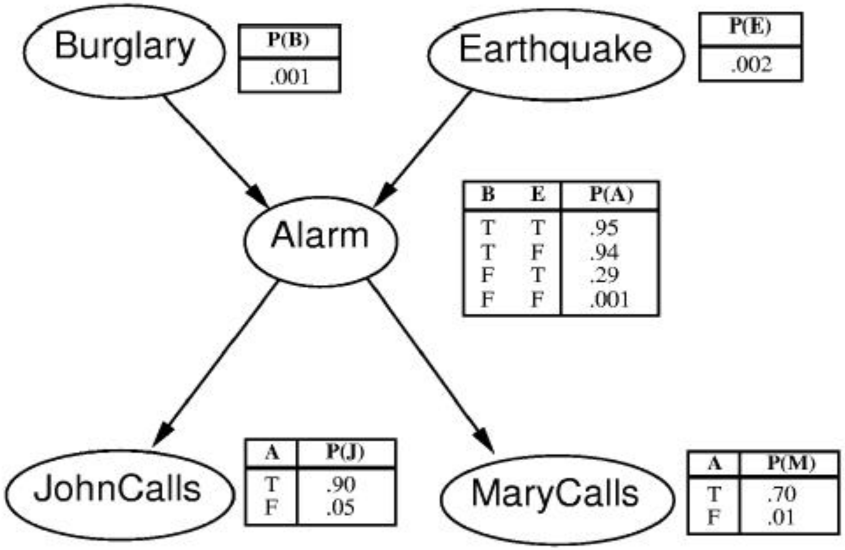
Gibbs Sampling: An Example



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



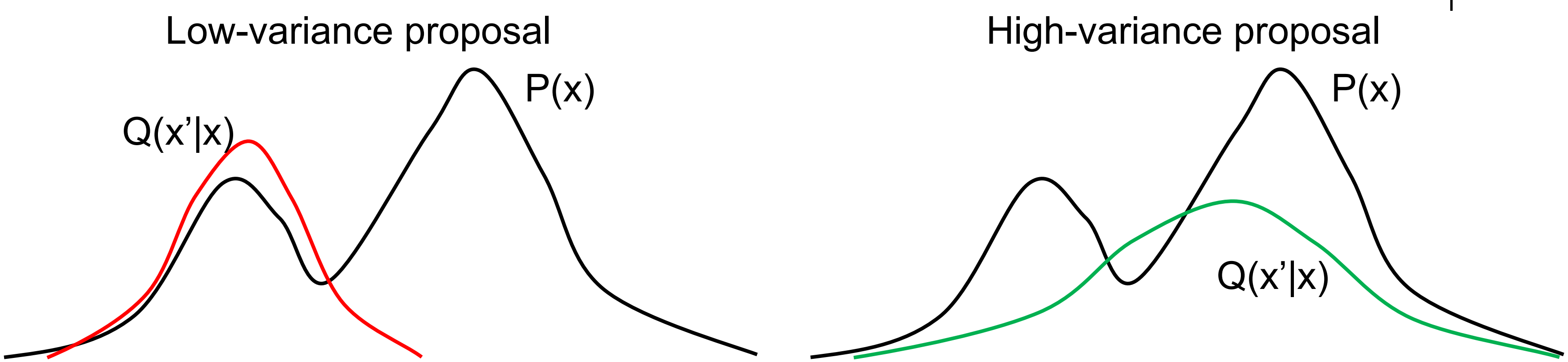
t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F	T	F
2	F	T	T	T	T
3	T	F	T	F	T
4	T	F	T	F	F

- 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.

Practical Aspects of MCMC

- How do we know if our proposal Q(x'|x) is good or not?
  - Monitor the acceptance rate

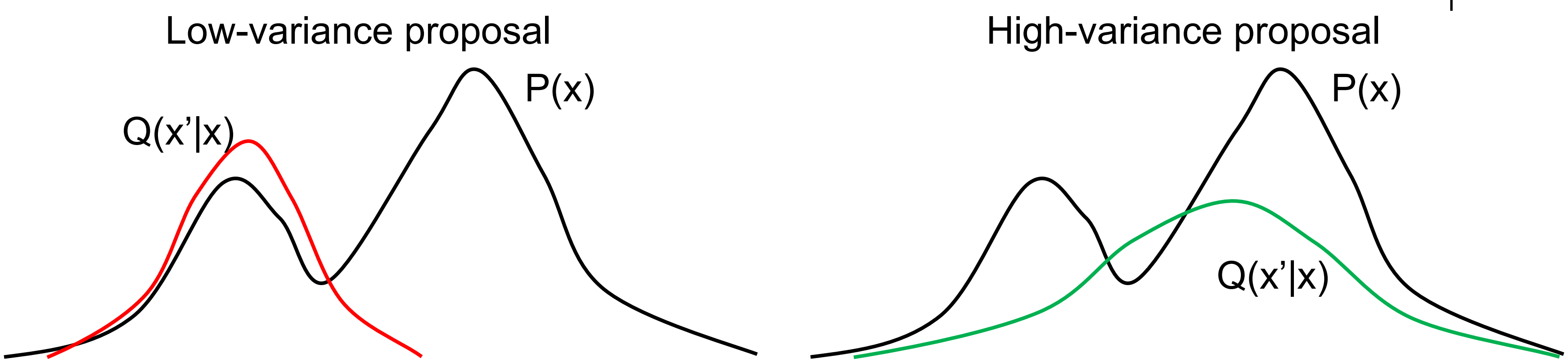
Acceptance Rate



- Choosing the proposal Q(x'|x) is a tradeoff:
  - "Narrow", low-variance proposals have high acceptance, but take many iterations to explore P(x) fully because the proposed x are too close
  - "Wide", high-variance proposals have the potential to explore much of P(x), but many proposals are rejected which slows down the sampler

- A good Q(x'|x) proposes distant samples x' with a sufficiently high acceptance rate

Acceptance Rate



- Acceptance rate is the fraction of samples that MH accepts.
  - General guideline: proposals should have ~0.5 acceptance rate [1]

- Gaussian special case:
  - If both P(x) and Q(x'|x) are Gaussian, the optimal acceptance rate is ~0.45 for D=1 dimension and approaches ~0.23 as D tends to infinity [2]

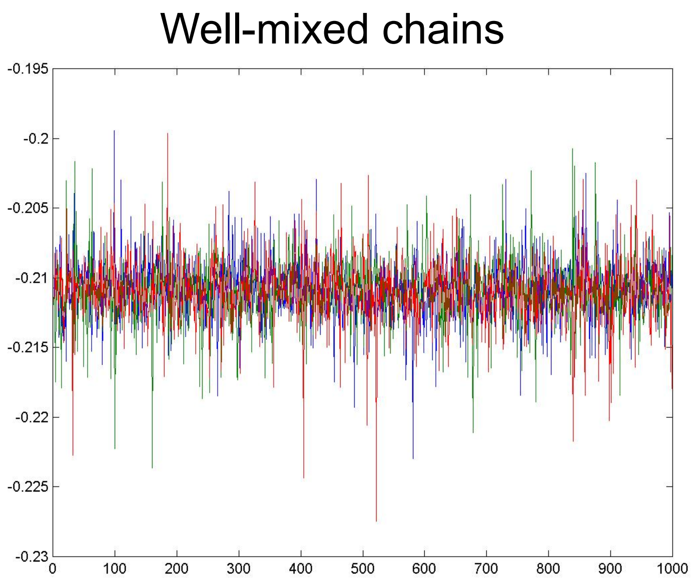
[1] Muller, P. (1993). "A Generic Approach to Posterior Integration and Gibbs Sampling"
[2] Roberts, G.O., Gelman, A., and Gilks, W.R. (1994). "Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms"

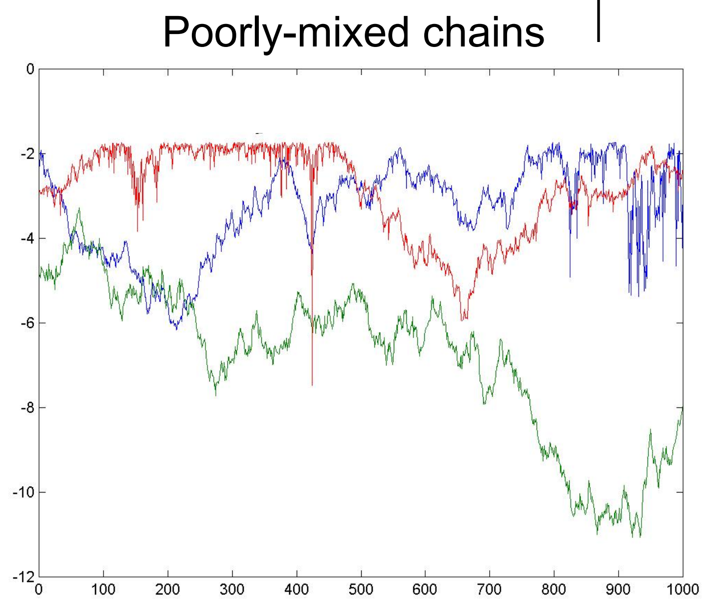
Practical Aspects of MCMC

- How do we know if our proposal Q(x'|x) is any good?
  - Monitor the acceptance rate

- How do we know when to stop burn-in?
  - Plot the sample values vs time

Sample Values vs Time





- 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
- In practice, we usually start with multiple chains

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) is not easy
- Gibbs sampling sets the proposal Q(x'|x) to the conditional distribution P(x'|x)
  - Acceptance rate is 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

Thank you!
Q & A