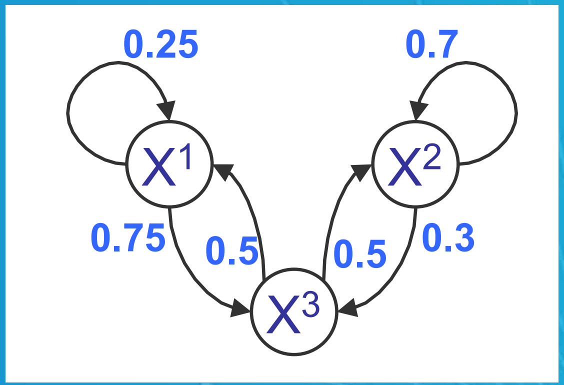
PETUUM

Carnegie Mellon University
School of Computer Science

Probabilistic Graphical Models

Approximate Inference: Markov Chain Monte Carlo



Eric Xing

Lecture 14, March 4, 2019

Reading: see class homepage

Recap of Monte Carlo

- Monte Carlo methods are algorithms that:
  - Generate samples from a given probability distribution $p(x)$
  - Estimate expectations of functions $E[f(x)]$ under a distribution $p(x)$

- Why is this useful?
  - Can use samples of $p(x)$ to approximate $p(x)$ itself
    - Allows us to do graphical model inference when we can't compute $p(x)$
  - Expectations $E[f(x)]$ reveal interesting properties about $p(x)$
    - e.g. means and variances of $p(x)$

Limitations of Monte Carlo

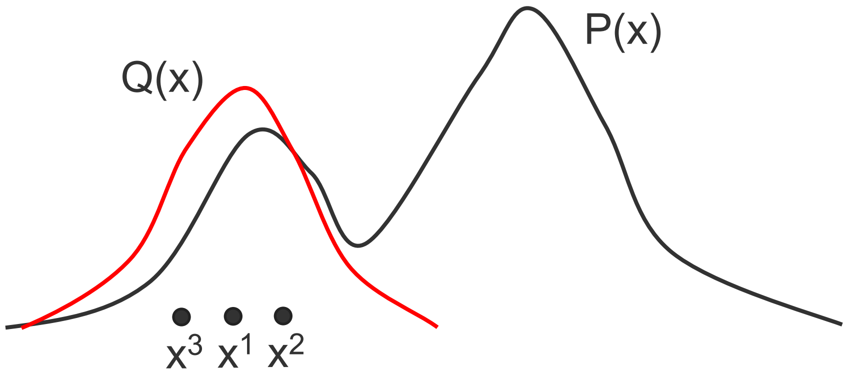
- Direct sampling
  - Hard to get rare events in high-dimensional spaces
  - Infeasible for MRFs, unless we know the normalizer Z
- Rejection sampling, Importance sampling
  - Do not work well if the proposal Q(x) is very different from P(x)
  - Yet constructing a Q(x) similar to P(x) can be difficult
    - 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?

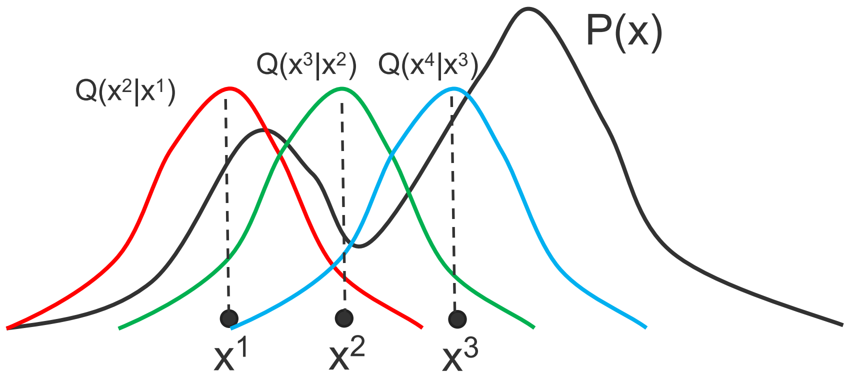
Markov Chain Monte Carlo

- 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

$$\left.\right\}\ \text{Function Draw sample } (x(t))$$

Function
Draw sample (x(t))

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

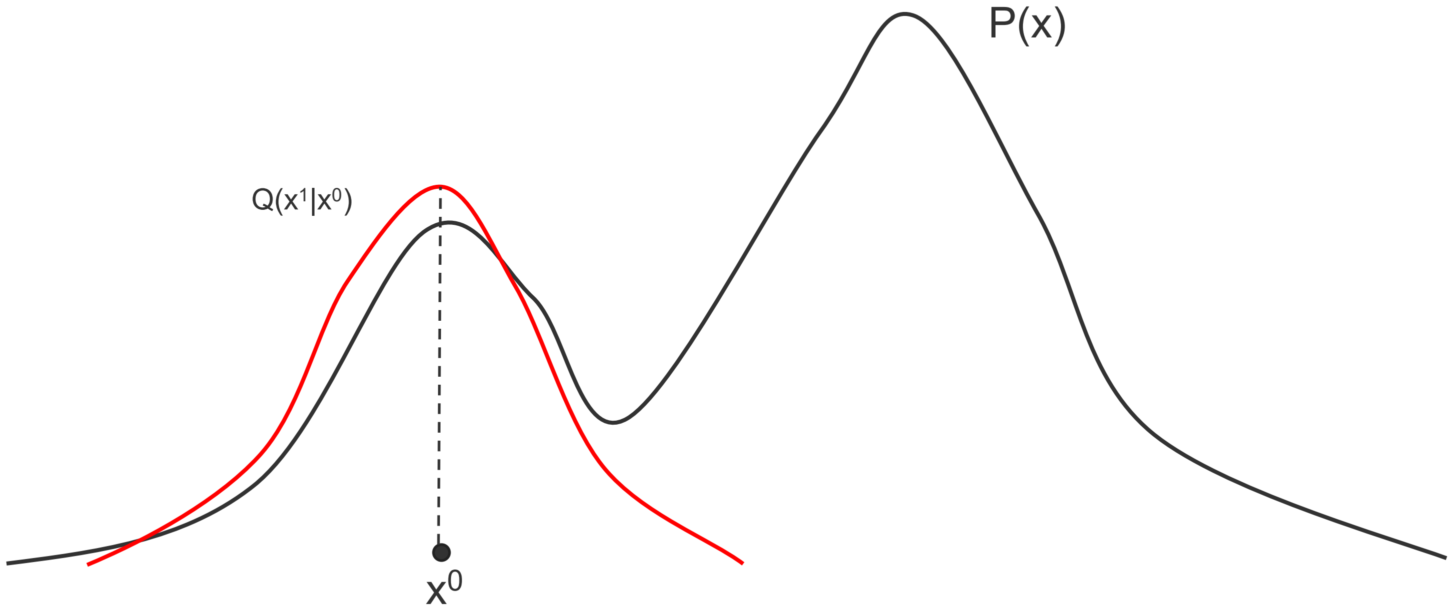
The MH Algorithm

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

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

Initialize $x^{(0)}$

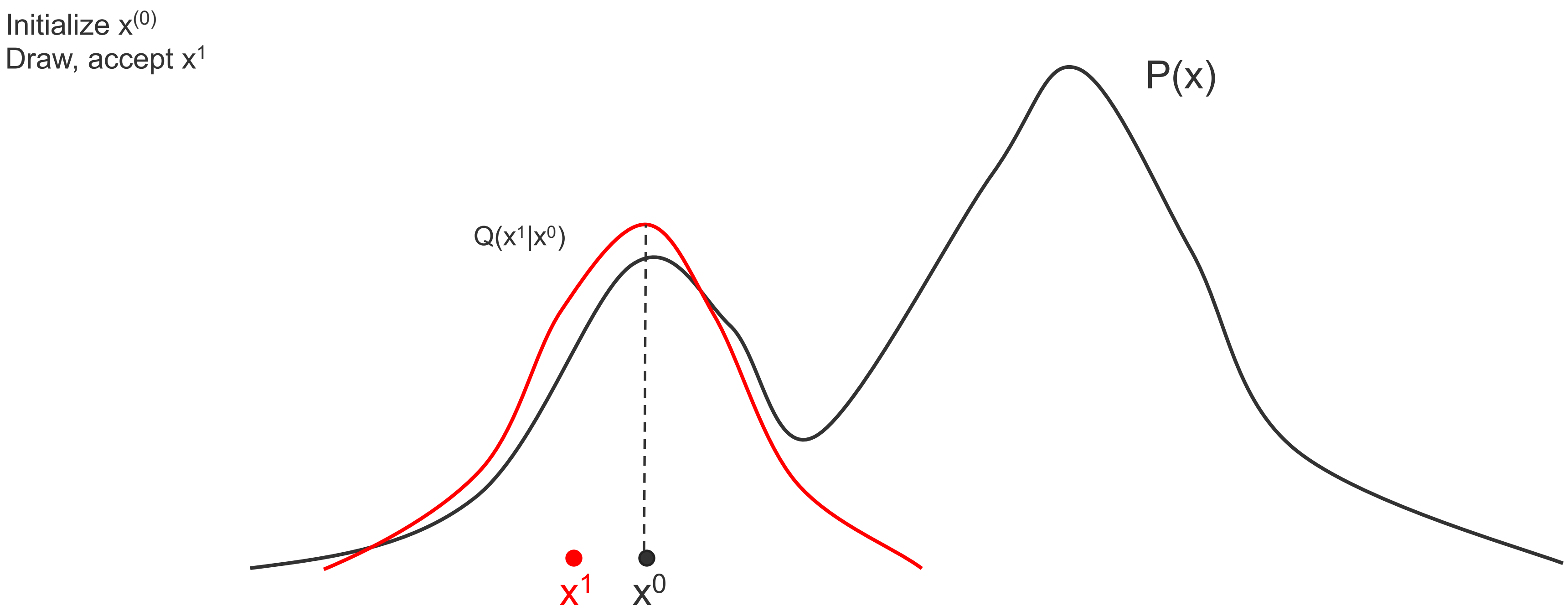
…



The MH Algorithm

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

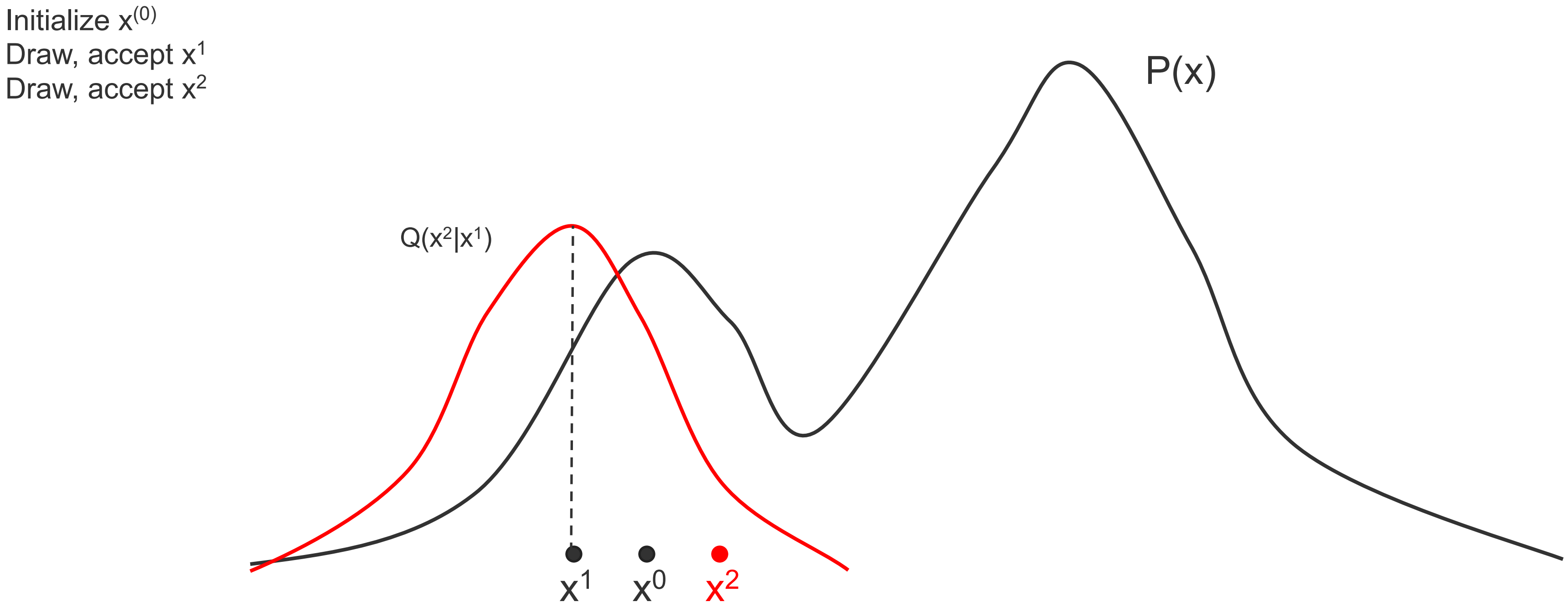
- Example:
  - Let Q(x'|x) be a Gaussian centered on x
  - We're trying to sample from a bimodal distribution P(x)



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 $P(x)$

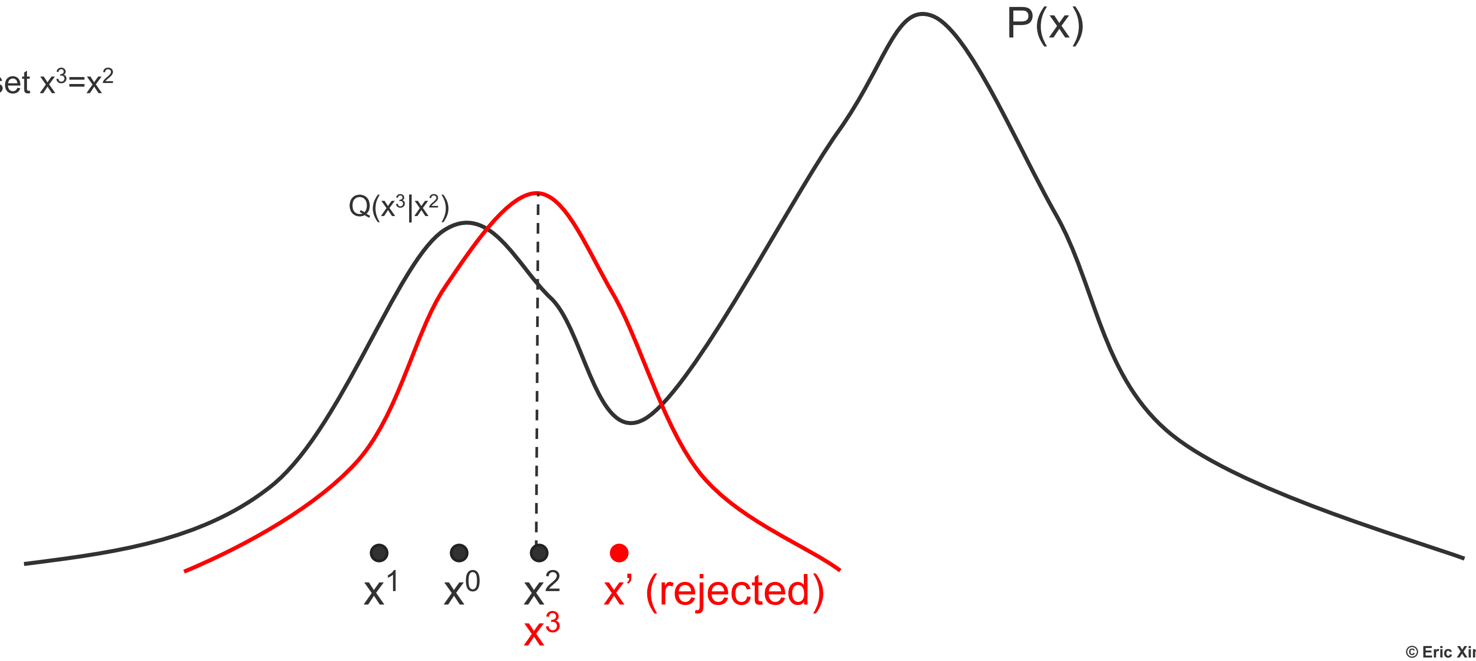


The MH Algorithm

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

- 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 $x^3=x^2$



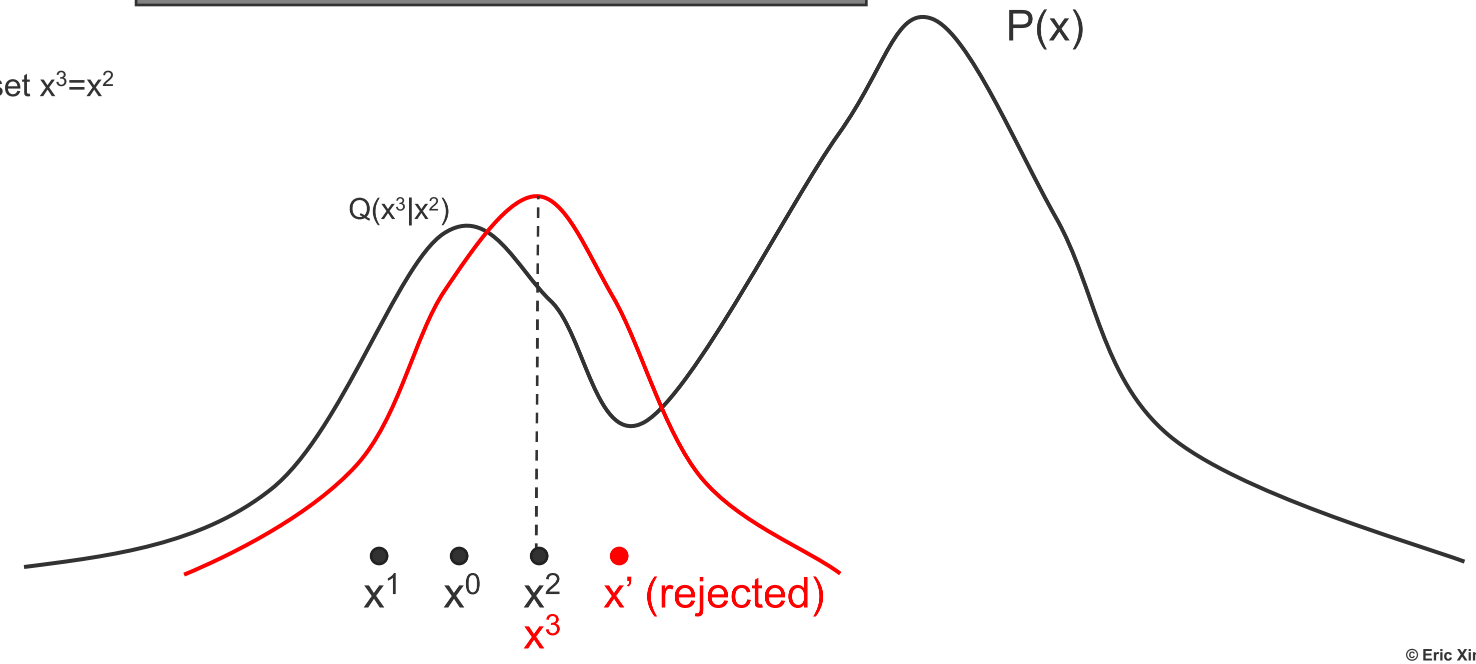
$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

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 $x^3=x^2$

We reject because $P(x')/Q(x'|x^2) < 1$ and $P(x^2)/Q(x^2|x') > 1$, hence $A(x'|x^2)$ is close to zero!

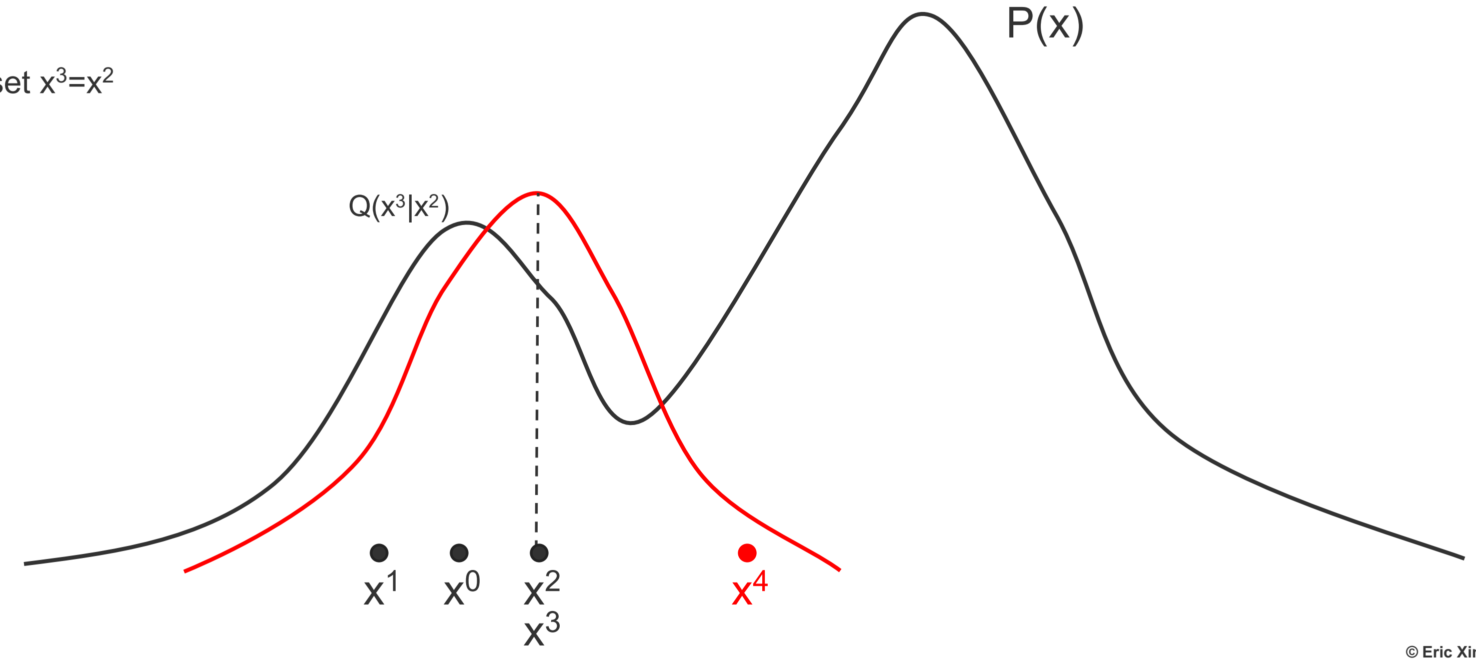


The MH Algorithm

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

- 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

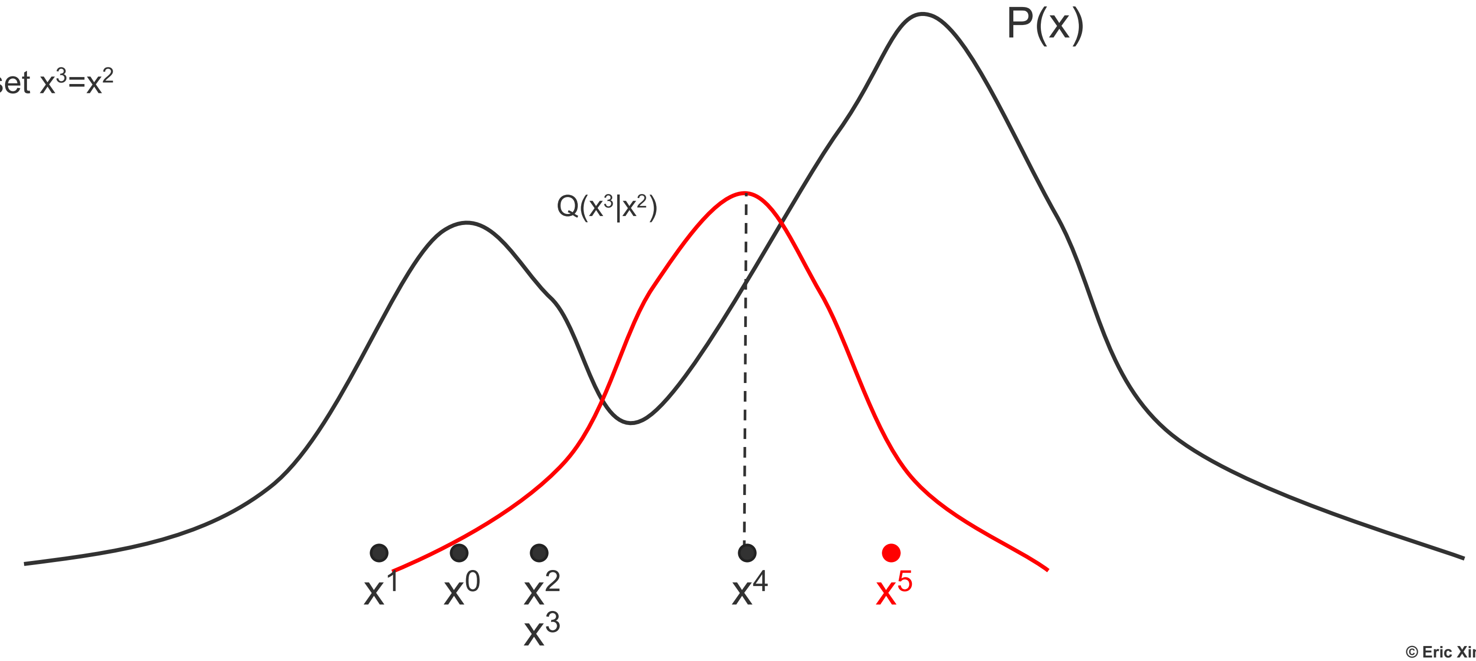


$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

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



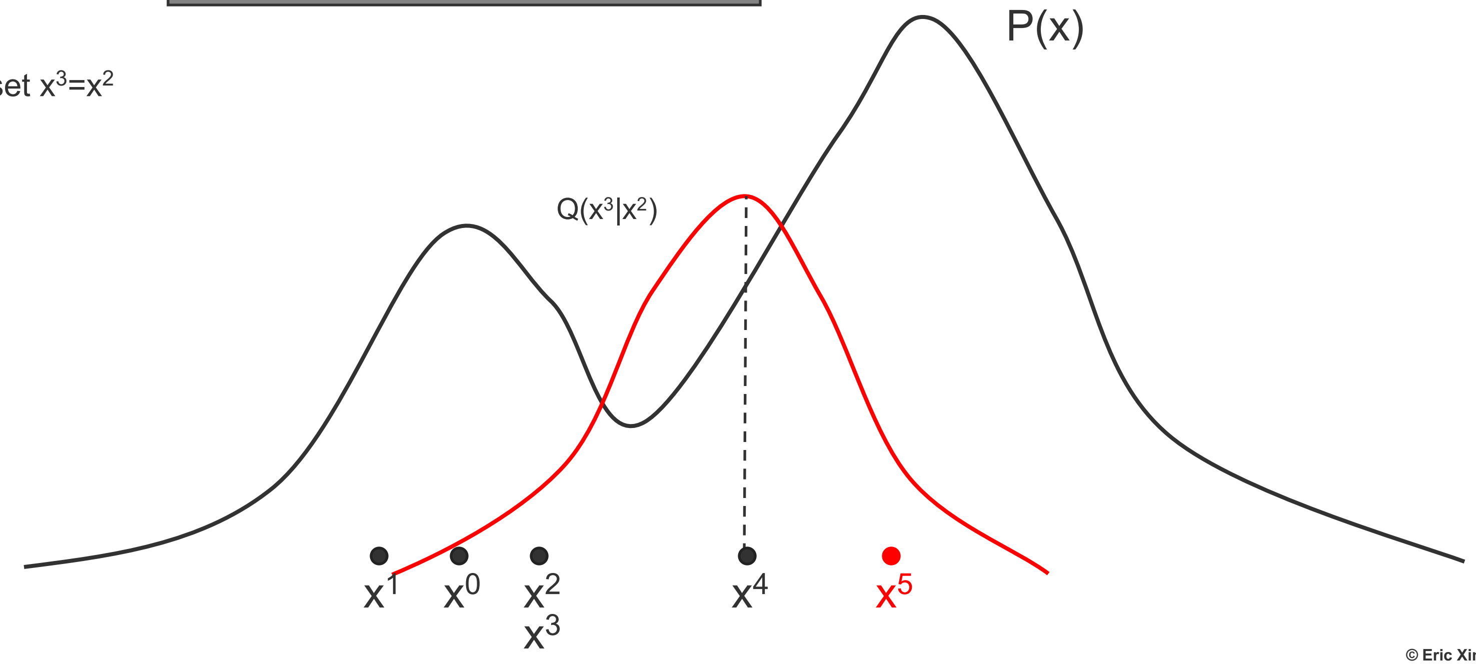
The MH Algorithm

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

- 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

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)?
- What is the connection between Markov Chains and MCMC?

Markov Chains

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

$$P(x^{(n)}=x\,|\,x^{(1)},\dots,x^{(n-1)})=P(x^{(n)}=x\,|\,x^{(n-1)})$$

  - $P(x^{(n)}=x\,|\,x^{(n-1)})$ 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 is fixed with time $P(x^{(t)}=x\,|\,x^{(t-1)})$
  - To emphasize this, we will call the kernel $T(x'\,|\,x)$, where x is the previous state and x' is the next state

MC 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: $\pi(x') = \sum_x \pi(x) T(x' \mid x)$, for all x'

MC 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
  - 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

MC 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'

- Reversible MCs always have a stationary distribution! 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 satisfies detailed balance
  - 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 $\frac{P(x)Q(x' \mid x)}{P(x')Q(x \mid x')} > 1$ 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!

Caveats

- Although MH eventually converges to the true distribution P(x), we have no guarantees as to when this will occur
  - The burn-in period represents the un-converged part of the Markov Chain – that's why we throw those samples away!
  - Knowing when to halt burn-in is an art. We will look at some techniques later in this lecture.

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

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 from $P(x_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$, i.e. the conditional distribution of x_i given the current values of all other variables
        2. Update $x_i \leftarrow x$

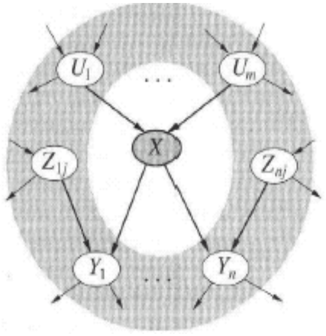
- When we update x_i, we <u>immediately</u> use its new value for sampling other variables x_j

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

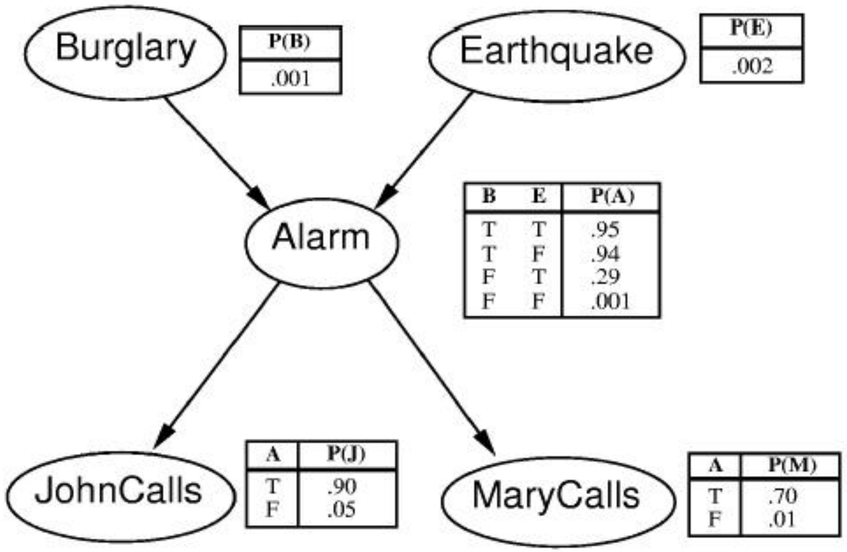
$$P(x_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i \mid MB(x_i))$$

- For a BN, the Markov Blanket of x is the set c parents, children, and co-parents



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

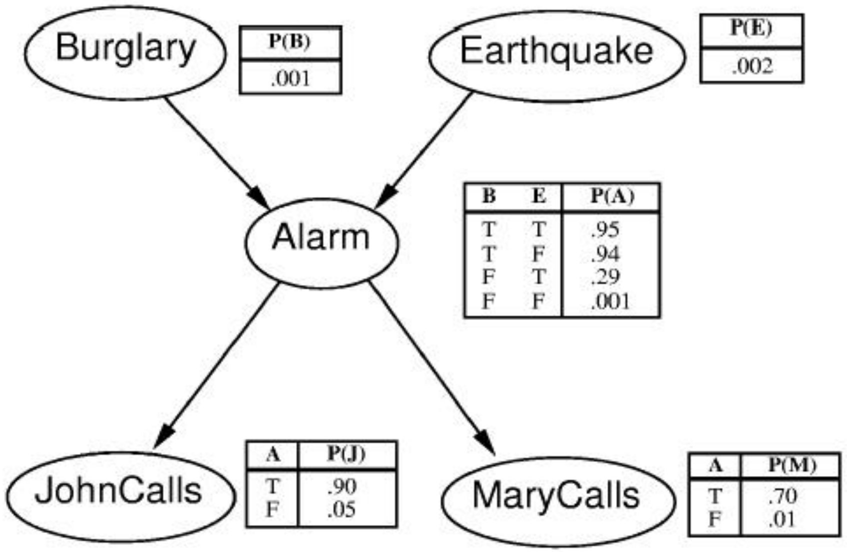
Gibbs Sampling: An Example



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,

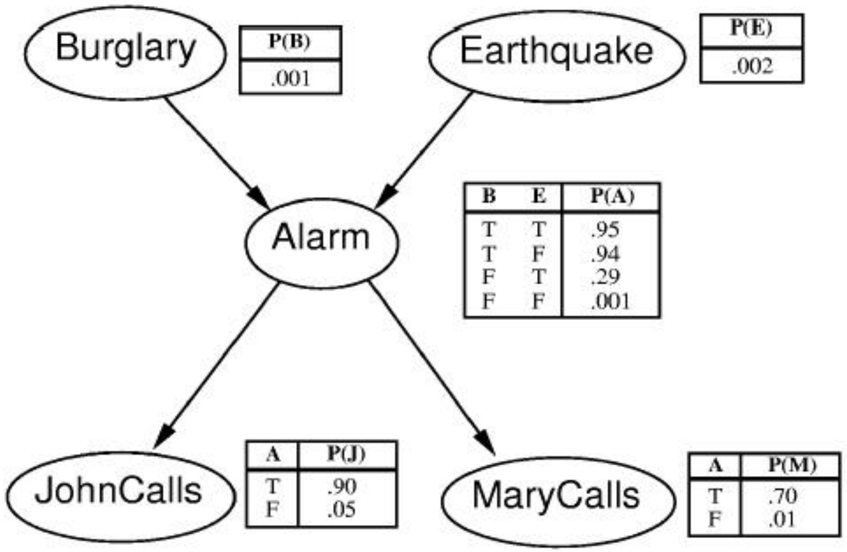
$$P(B \mid A, E) \propto P(A \mid B, E) P(B)$$

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

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

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

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,

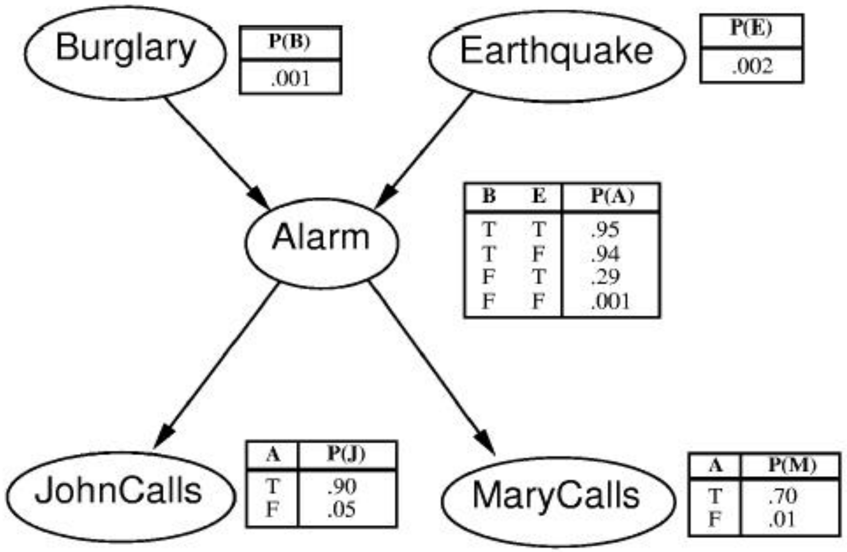
$$P(E \mid A, B) \propto P(A \mid B, E) P(E)$$

- (A,B) = (F,F), so we compute the following, and sample E = T

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

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

Gibbs Sampling: An Example



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

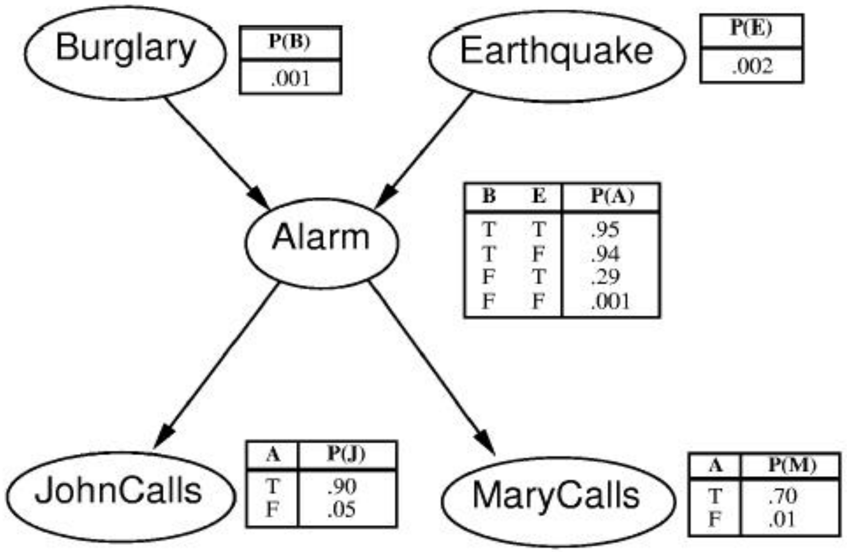
- Sampling P(A|B,E,J,M): Using Bayes Rule,

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

- (B,E,J,M) = (F,T,F,F), so we compute the following, and sample A = F

$$P(A=T \mid B=F,E=T,J=F,M=F) \propto (0.1)(0.3)(0.29) = 0.0087$$
$$P(A=F \mid B=F,E=T,J=F,M=F) \propto (0.95)(0.99)(0.71) = 0.6678$$

Gibbs Sampling: An Example

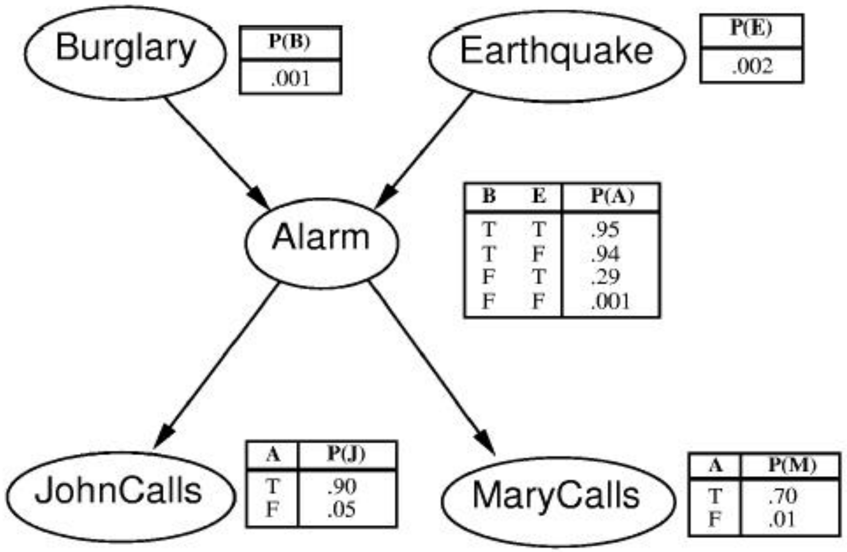


t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F	T	
2					
3					
4					

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

$$P(J=T \mid A=F) \propto 0.05$$
$$P(J=F \mid A=F) \propto 0.95$$

Gibbs Sampling: An Example

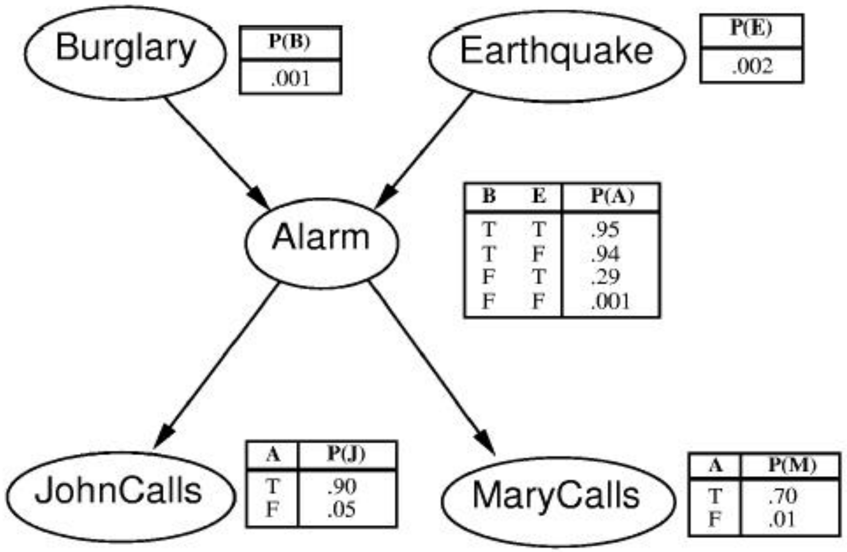


t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F	T	F
2					
3					
4					

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

$$P(M = T \mid A = F) \propto 0.01$$
$$P(M = F \mid A = F) \propto 0.99$$

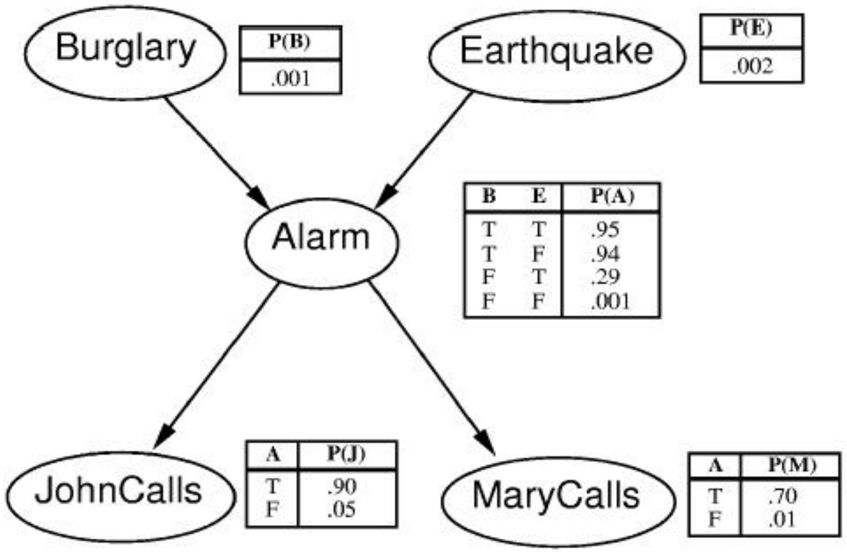
Gibbs Sampling: An Example



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

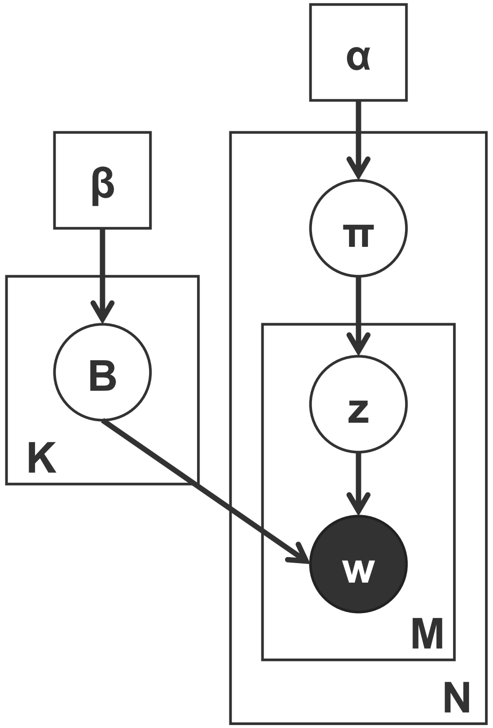


t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F	T	F
2	F	T	T	T	T
3	T	F	T	F	T
4	T	F	T	F	F

- 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.

Topic Models: Collapsed Gibbs
(Tom Griffiths & Mark Steyvers)

- Collapsed Gibbs sampling
  - Popular inference algorithm for topic models
  - Integrate out topic vectors π and topics B
  - Only need to sample word-topic assignments z



Algorithm:

For all variables $\mathbf{z} = z_1, z_2, \ldots, z_n$

Draw $z_i^{(t+1)}$ from $P(z_i|\mathbf{z}_{-i}, w)$

where $\mathbf{z}_{-i} = z_1^{(t+1)}, z_2^{(t+1)}, \ldots, z_{i-1}^{(t+1)}, z_{i+1}^{(t)}, \ldots, z_n^{(t)}$

Collapsed Gibbs sampling

- What is $P(z_i|\mathbf{z}_{-i}, w)$?
  - It is a product of two Dirichlet-Multinomial conditional distributions:

$$P(z_i = j|\mathbf{z}_{-i}, \mathbf{w}) \propto \underbrace{\frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + W\beta}}_{\text{“word-topic” term}} \underbrace{\frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}}_{\text{“doc-topic” term}}$$

Collapsed Gibbs sampling

- What is $P(z_i|\mathbf{z}_{-i}, \mathbf{w})$?
  - It is a product of two Dirichlet-Multinomial conditional distributions:

word positions *a* (excluding w_i) such that:
$w_a = w_i$
$z_a = j$

word positions *a* **in the current document d_i** (excluding w_i) such that:
$z_a = j$

$$P(z_i = j|\mathbf{z}_{-i}, \mathbf{w}) \propto \frac{\boxed{n_{-i,j}^{(w_i)}} + \beta}{\boxed{n_{-i,j}^{(\cdot)}} + W\beta} \frac{\boxed{n_{-i,j}^{(d_i)}} + \alpha}{\boxed{n_{-i,\cdot}^{(d_i)}} + T\alpha}$$

word positions *a* (excluding w_i) such that:
$z_a = j$

word positions *a* **in the current document d_i** (excluding w_i)

Collapsed Gibbs illustration

			iteration
			1
i	w_i	d_i	z_i
1	MATHEMATICS	1	2
2	KNOWLEDGE	1	2
3	RESEARCH	1	1
4	WORK	1	2
5	MATHEMATICS	1	1
6	RESEARCH	1	2
7	WORK	1	2
8	SCIENTIFIC	1	1
9	MATHEMATICS	1	2
10	WORK	1	1
11	SCIENTIFIC	2	1
12	KNOWLEDGE	2	1
.	.	.	.
.	.	.	.
.	.	.	.
50	JOY	5	2

Collapsed Gibbs illustration

			iteration	
			1	2
i	w_i	d_i	z_i	z_i
1	MATHEMATICS	1	2	?
2	KNOWLEDGE	1	2	
3	RESEARCH	1	1	
4	WORK	1	2	
5	MATHEMATICS	1	1	
6	RESEARCH	1	2	
7	WORK	1	2	
8	SCIENTIFIC	1	1	
9	MATHEMATICS	1	2	
10	WORK	1	1	
11	SCIENTIFIC	2	1	
12	KNOWLEDGE	2	1	
.	.	.	.	
.	.	.	.	
.	.	.	.	
50	JOY	5	2	

Collapsed Gibbs illustration

			iteration	
			1	2
i	w_i	d_i	z_i	z_i
1	MATHEMATICS	1	2	?
2	KNOWLEDGE	1	2	
3	RESEARCH	1	1	
4	WORK	1	2	
5	MATHEMATICS	1	1	
6	RESEARCH	1	2	
7	WORK	1	2	
8	SCIENTIFIC	1	1	
9	MATHEMATICS	1	2	
10	WORK	1	1	
11	SCIENTIFIC	2	1	
12	KNOWLEDGE	2	1	
.	.	.	.	
.	.	.	.	
.	.	.	.	
50	JOY	5	2	

$$P(z_i = j | \mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + W\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}$$

Collapsed Gibbs illustration

			iteration	
			1	**2**
i	w_i	d_i	z_i	z_i
1	MATHEMATICS	1	2	?
2	KNOWLEDGE	1	2	
3	RESEARCH	1	1	
4	WORK	1	2	
5	MATHEMATICS	1	1	
6	RESEARCH	1	2	
7	WORK	1	2	
8	SCIENTIFIC	1	1	
9	MATHEMATICS	1	2	
10	WORK	1	1	
11	SCIENTIFIC	2	1	
12	KNOWLEDGE	2	1	
.	.	.	.	
.	.	.	.	
.	.	.	.	
50	JOY	5	2	

$$P(z_i = j | \mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + W\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}$$

Collapsed Gibbs illustration

			iteration	
			1	2
i	w_i	d_i	z_i	z_i
1	MATHEMATICS	1	2	2
2	KNOWLEDGE	1	2	?
3	RESEARCH	1	1	
4	WORK	1	2	
5	MATHEMATICS	1	1	
6	RESEARCH	1	2	
7	WORK	1	2	
8	SCIENTIFIC	1	1	
9	MATHEMATICS	1	2	
10	WORK	1	1	
11	SCIENTIFIC	2	1	
12	KNOWLEDGE	2	1	
.	.	.	.	
.	.	.	.	
.	.	.	.	
50	JOY	5	2	

$$P(z_i = j | \mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + W\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}$$

Collapsed Gibbs illustration

			iteration 1	2
i	w_i	d_i	z_i	z_i
1	MATHEMATICS	1	2	2
2	KNOWLEDGE	1	2	1
3	RESEARCH	1	1	?
4	WORK	1	2	
5	MATHEMATICS	1	1	
6	RESEARCH	1	2	
7	WORK	1	2	
8	SCIENTIFIC	1	1	
9	MATHEMATICS	1	2	
10	WORK	1	1	
11	SCIENTIFIC	2	1	
12	KNOWLEDGE	2	1	
.	.	.	.	
.	.	.	.	
.	.	.	.	
50	JOY	5	2	

$$P(z_i = j | \mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + W\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}$$

Collapsed Gibbs illustration

			iteration 1	2
i	w_i	d_i	z_i	z_i
1	MATHEMATICS	1	2	2
2	KNOWLEDGE	1	2	1
3	RESEARCH	1	1	1
4	WORK	1	2	?
5	MATHEMATICS	1	1	
6	RESEARCH	1	2	
7	WORK	1	2	
8	SCIENTIFIC	1	1	
9	MATHEMATICS	1	2	
10	WORK	1	1	
11	SCIENTIFIC	2	1	
12	KNOWLEDGE	2	1	
.	.	.	.	
.	.	.	.	
.	.	.	.	
50	JOY	5	2	

$$P(z_i = j | \mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + W\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}$$

Collapsed Gibbs illustration

			iteration 1	2
i	w_i	d_i	z_i	z_i
1	MATHEMATICS	1	2	2
2	KNOWLEDGE	1	2	1
3	RESEARCH	1	1	1
4	WORK	1	2	2
5	MATHEMATICS	1	1	?
6	RESEARCH	1	2	
7	WORK	1	2	
8	SCIENTIFIC	1	1	
9	MATHEMATICS	1	2	
10	WORK	1	1	
11	SCIENTIFIC	2	1	
12	KNOWLEDGE	2	1	
.	.	.	.	
.	.	.	.	
.	.	.	.	
50	JOY	5	2	

$$P(z_i = j | \mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + W\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}$$

Collapsed Gibbs illustration

			iteration			
			1	**2**	**…**	**1000**
i	w_i	d_i	z_i	z_i		z_i
1	MATHEMATICS	1	2	2		2
2	KNOWLEDGE	1	2	1		2
3	RESEARCH	1	1	1		2
4	WORK	1	2	2		1
5	MATHEMATICS	1	1	2		2
6	RESEARCH	1	2	2		2
7	WORK	1	2	2		2
8	SCIENTIFIC	1	1	1	…	1
9	MATHEMATICS	1	2	2		2
10	WORK	1	1	2		2
11	SCIENTIFIC	2	1	1		2
12	KNOWLEDGE	2	1	2		2
.	.	.	.	.		.
.	.	.	.	.		.
.	.	.	.	.		.
50	JOY	5	2	1		1

$$P(z_i = j | \mathbf{z}_{-i}, \mathbf{w}) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(\cdot)} + W\beta} \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}$$

Gibbs Sampling is a special case of MH

- The GS proposal distribution is

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

  - Where $\mathbf{x}_{-i}$ denotes all variables except x_i
- Applying MH to this proposal, we find that samples are always accepted (which is exactly what GS does):

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

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

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

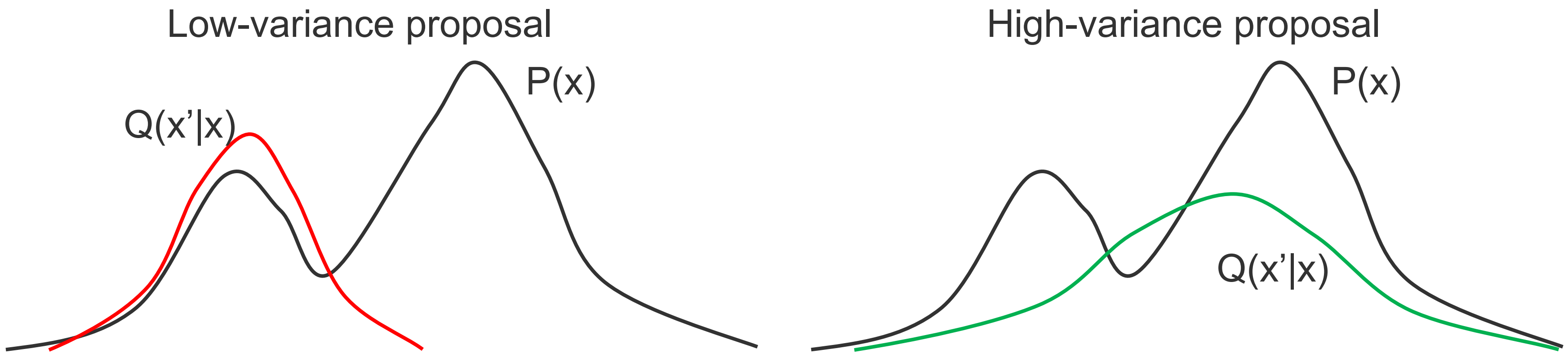
  - GS is simply MH with a proposal that is always accepted!

Practical Aspects of MCMC

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

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

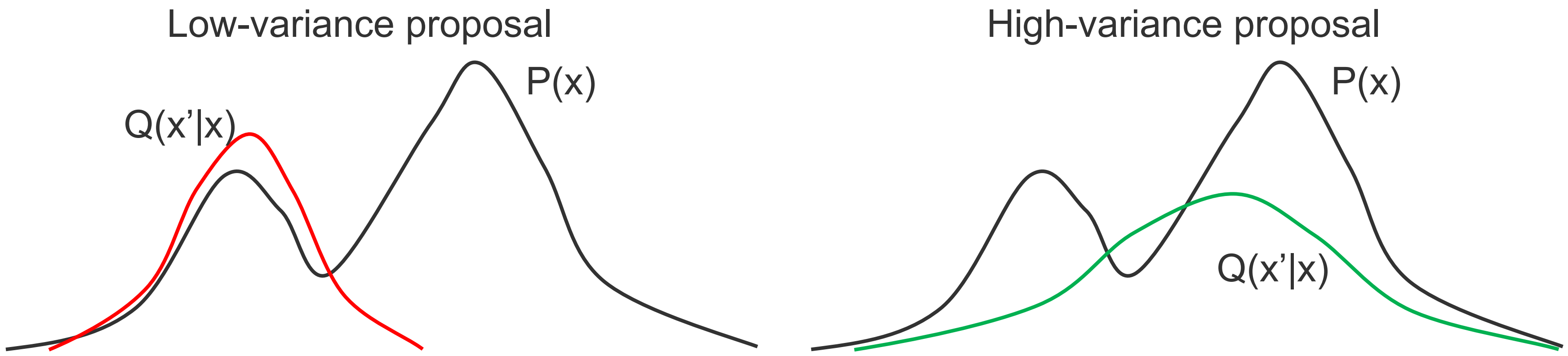
Acceptance Rate



- 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

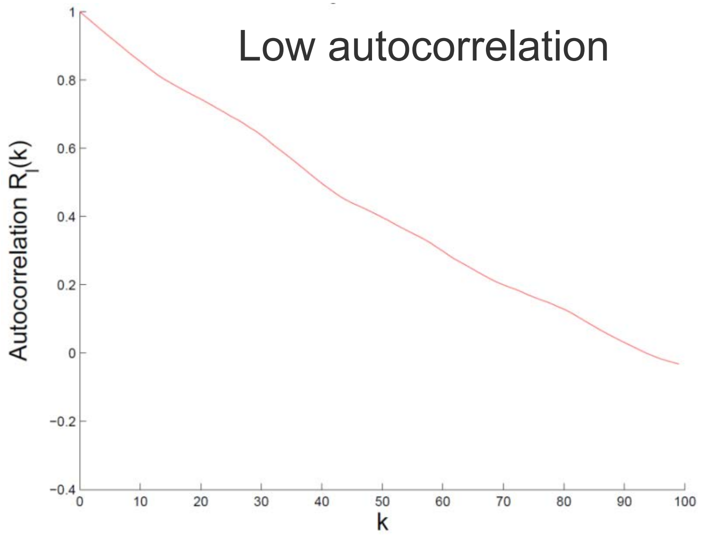


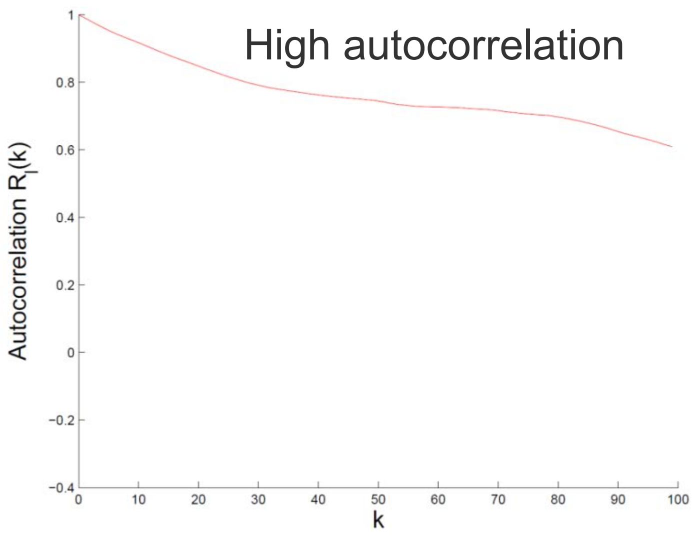
- Acceptance rate is the fraction of samples that MH accepts.
  - General guideline: proposals should have ~0.5 acceptance rate [1]

- 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"

Autocorrelation function



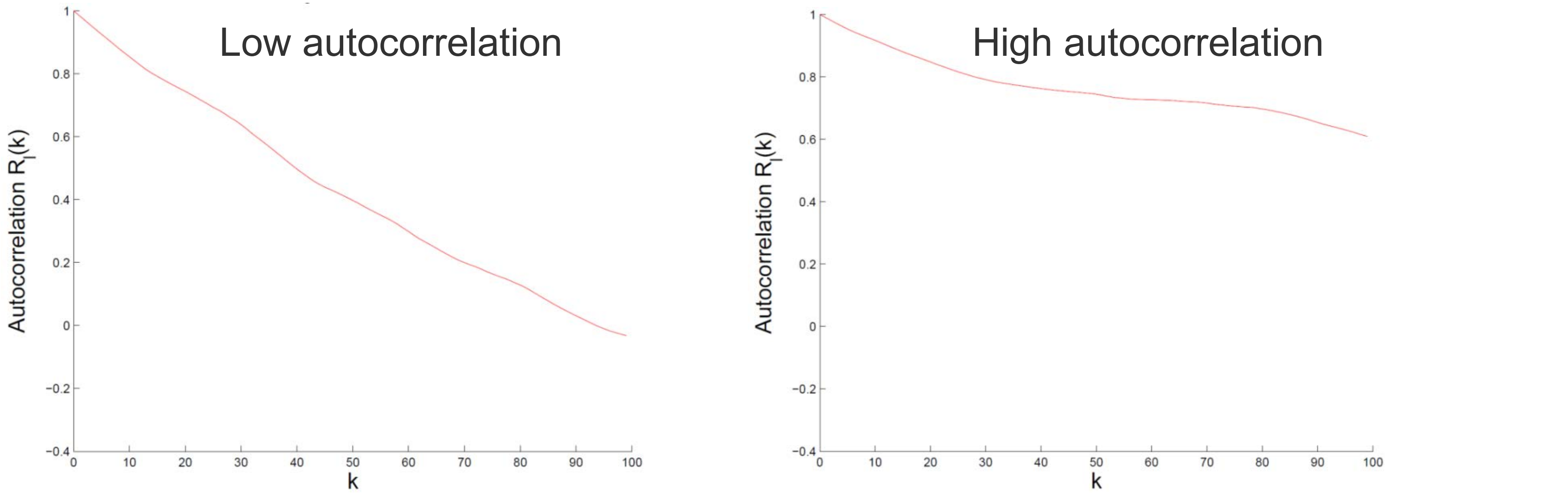


- MCMC chains always show autocorrelation (AC)
  - AC means that adjacent samples in time are highly correlated
- We quantify AC with the autocorrelation function of an r.v. x:

$$R_x(k) = \frac{\sum_{t=1}^{n-k}(x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^{n-k}(x_t - \bar{x})^2}$$

Autocorrelation function

$$R_x(k) = \frac{\sum_{t=1}^{n-k} (x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^{n-k} (x_t - \bar{x})^2}$$

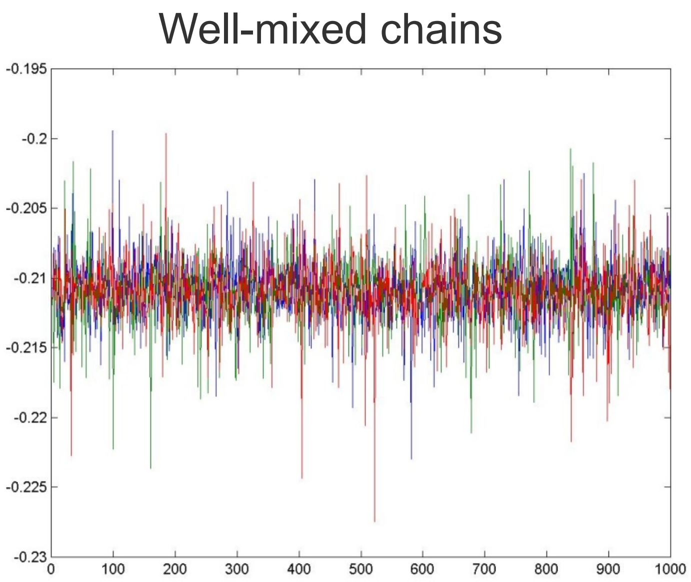


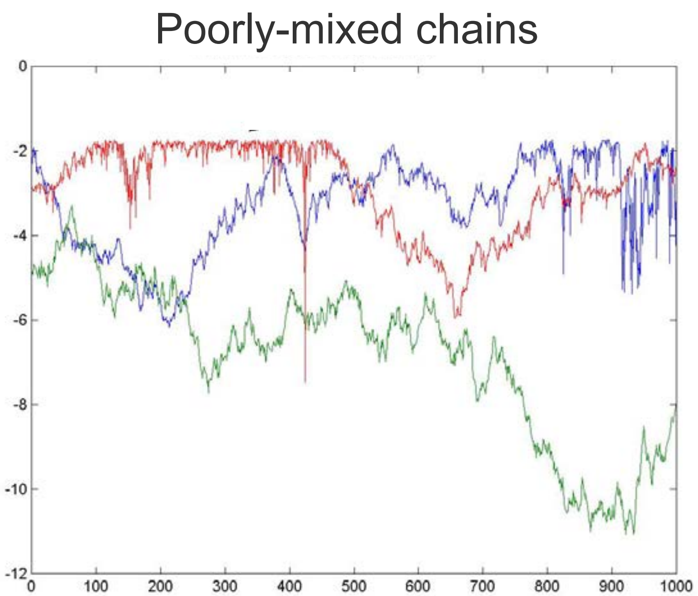
- The first-order AC $R_x(1)$ can be used to estimate the Sample Size Inflation Factor (SSIF):

$$s_x = \frac{1 + R_x(1)}{1 - R_x(1)}$$

  - If we took n samples with SSIF s_x, then the effective sample size is n/s_x
  - High autocorrelation leads to smaller effective sample size!
  - We want proposals $Q(x'|x)$ with low autocorrelation

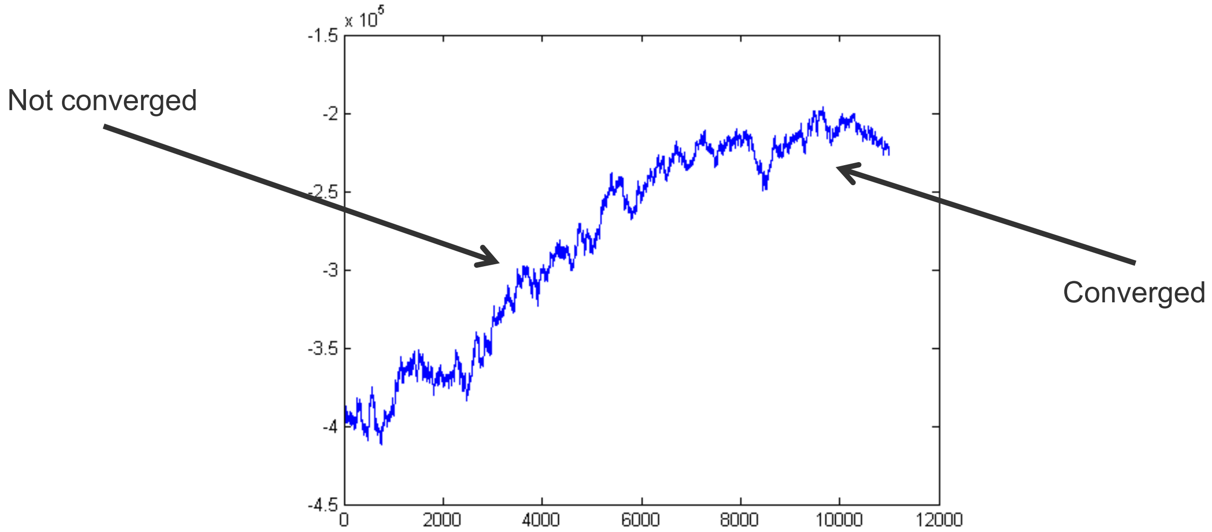
Sample Values vs Time





- 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

Log-likelihood vs Time



- Many graphical models are high-dimensional
  - Hard to visualize all r.v. chains at once

- Instead, plot the complete log-likelihood vs. time
  - The complete log-likelihood is an r.v. that depends on all model r.v.s
  - Generally, the log-likelihood will climb, then eventually plateau

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

Supplementary

Optimization in MCMC

Random walk in MCMC

$P(X)$ $\qquad\qquad$ $Q(X_{new}|X_{old})$

$$\min\{\ 1,\quad \frac{P(X_{new})Q(X_{old}|X_{new})}{P(X_{old})Q(X_{new}|X_{old})}\}$$

Hamiltonian Monte Carlo

- Hamiltonian Dynamics (1959)
  - Deterministic System
- Hybrid Monte Carlo (1987)
  - United MCMC and molecular Dynamics
- Statistical Application (1993)
  - Inference in Neural Networks
  - Improves acceptance rate
  - Uncorrelated SamplesType equation here.

Target distribution:

$$P(x) = \frac{e^{-\boldsymbol{E}(\boldsymbol{x})}}{\boldsymbol{Z}}$$

The Hamiltonian:

$$H(x,p) = E(x) + K(p)$$

$$\dot{x} = p \quad \dot{p} = -\frac{\partial E(x)}{\partial x} \qquad K(p) = p^{\mathrm{T}} p/2$$

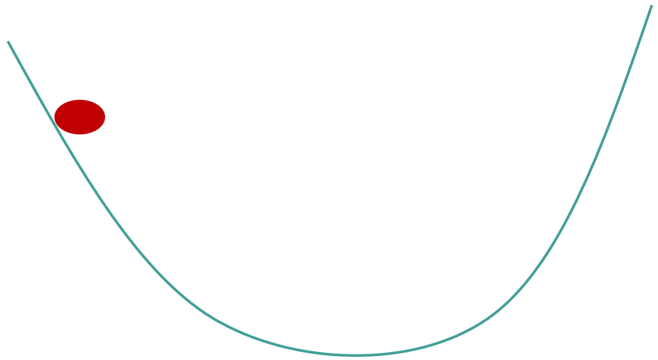
Auxiliary distribution:

$$P_H(x,p) = \frac{e^{-E(x)-K(p)}}{Z_H}$$

Hamiltonian Dynamics

- Position vector x, Momentum vector p
- Kinetic Energy $K(p)$
- Potential Energy $U(x)$
- Define $H(p, x) = K(p) + U(x)$

Hamiltonian Dynamics



- Position vector x, Momentum vector p
- Kinetic Energy $K(p)$
- Potential Energy $U(x)$
- Define $H(p,x) = K(p) + U(x)$
- Hamiltonian Dynamics

$$\frac{dx_i}{dt} = \frac{\partial H}{\partial p_i}$$

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial x_i}$$

Alternative notation

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}$$

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}$$

Hamiltonian Dynamics: Example

- Kinetic Energy $K(p) = \frac{|p|^2}{2}$
- Potential Energy $U(q) = \frac{q^2}{2}$
- So

$$\frac{dq}{dt} = p, \quad \frac{dp}{dt} = -q$$

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}$$

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}$$

- And

$$q(t) = r\cos(a+t), \quad p(t) = -r\sin(a+t)$$

How to compute updates: Euler's Method

$$p_i(t+\varepsilon) = p_i(t) + \varepsilon \frac{dp_i}{dt}(t) = p_i(t) - \varepsilon \frac{\partial U}{\partial q_i}(q(t))$$

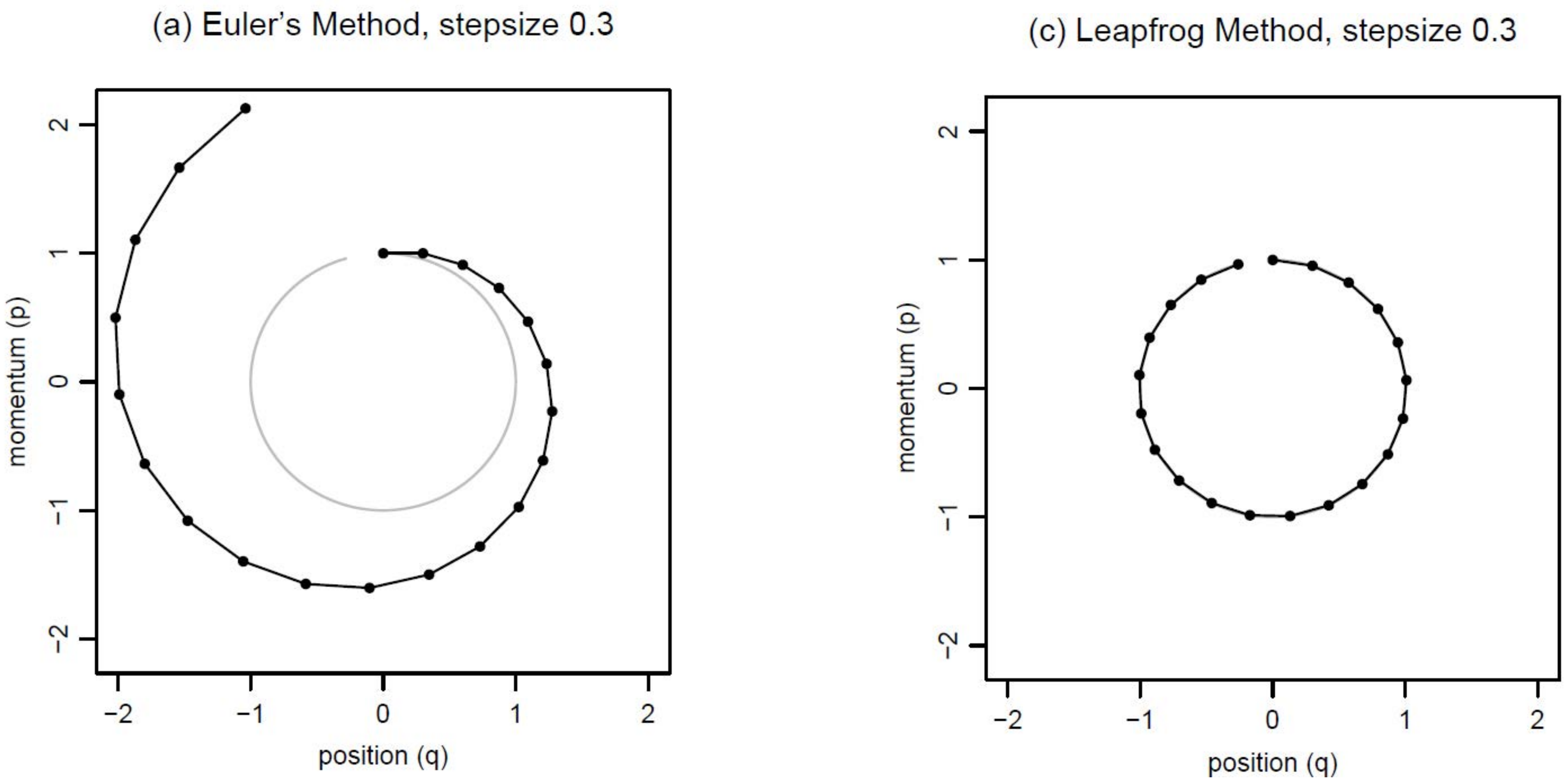
$$q_i(t+\varepsilon) = q_i(t) + \varepsilon \frac{dq_i}{dt}(t) = q_i(t) + \varepsilon \frac{p_i(t)}{m_i}$$

How to compute updates: Leapfrog Method

- The updates looks like

$$
\begin{aligned}
p_i(t+\varepsilon/2) &= p_i(t) - (\varepsilon/2)\frac{\partial U}{\partial q_i}(q(t)) \\
q_i(t+\varepsilon) &= q_i(t) + \varepsilon\frac{p_i(t+\varepsilon/2)}{m_i} \\
p_i(t+\varepsilon) &= p_i(t+\varepsilon/2) - (\varepsilon/2)\frac{\partial U}{\partial q_i}(q(t+\varepsilon))
\end{aligned}
$$

Leapfrog Vs Euler



$$q(t) = r\cos(a+t), \quad p(t) = -r\sin(a+t)$$

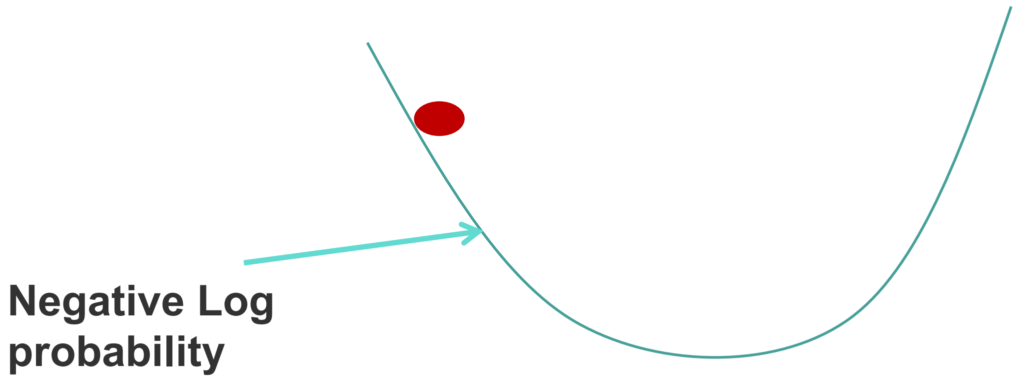
MCMC from Hamiltonian Dynamics

- Let q be variable of interest
- Define:

$$P(q,p) = \frac{1}{Z} \exp(-U(q)/T) \exp(-K(p)/T)$$

  - where $U(q) = -\log\left[\pi(q)L(q|D)\right] \qquad K(p) = \sum_{i=1}^{d} \frac{p_i^2}{2m_i}$

- Key Idea: Use Hamiltonian dynamics to propose next step.



MCMC from Hamiltonian Dynamics

- Given q_0 (starting state)
- Draw $p \sim N(0,1)$
- Use L steps of leapfrog to propose next state
- Accept / reject based on change in Hamiltonian

MCMC from Hamiltonian Dynamics

```
p = rnorm(length(q),0,1)
```

MCMC from Hamiltonian Dynamics

```
p = rnorm(length(q),0,1)
p = p - epsilon * grad_U(q) / 2
```

MCMC from Hamiltonian Dynamics

```
p = rnorm(length(q),0,1)
p = p - epsilon * grad_U(q) / 2
# Alternate full steps for position and momentum
for (i in 1:L)
{
    q = q + epsilon * p
    if (i!=L) p = p - epsilon * grad_U(q)
}
```

MCMC from Hamiltonian Dynamics

```
p = rnorm(length(q),0,1)
p = p - epsilon * grad_U(q) / 2

# Alternate full steps for position and momentum
for (i in 1:L)
{
    q = q + epsilon * p
    if (i!=L) p = p - epsilon * grad_U(q)
}

p = p - epsilon * grad_U(q) / 2        p = -p
```

MCMC from Hamiltonian Dynamics

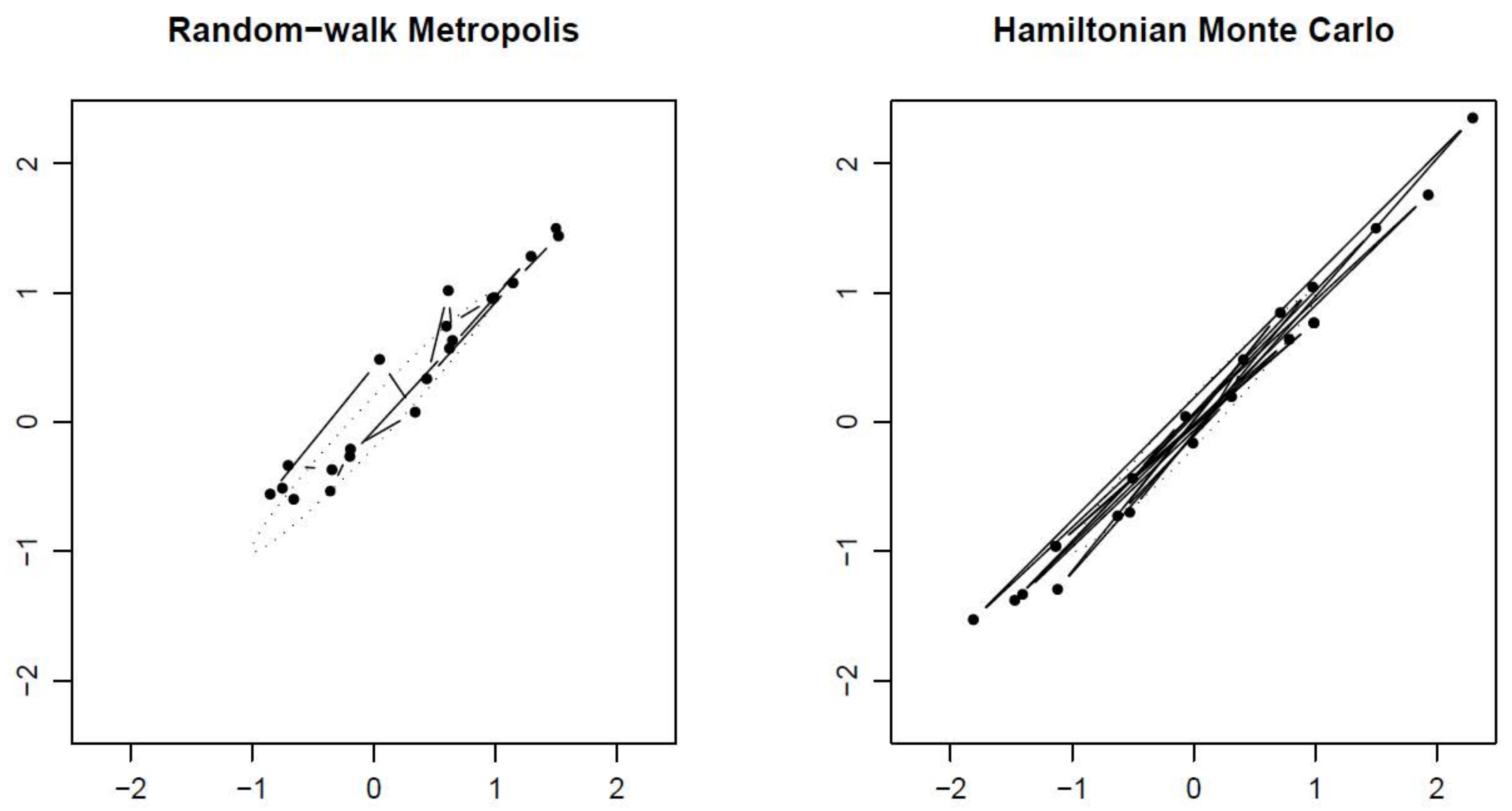
```
p = rnorm(length(q),0,1)
p = p - epsilon * grad_U(q) / 2
# Alternate full steps for position and momentum
for (i in 1:L)
{
    q = q + epsilon * p
    if (i!=L) p = p - epsilon * grad_U(q)
}
p = p - epsilon * grad_U(q) / 2        p = -p
Accept or reject the state at end of trajectory
```

$$\min\left[1, \exp(-U(q^*) + U(q) - K(p^*) + K(p))\right]$$

MCMC from Hamiltonian Dynamics

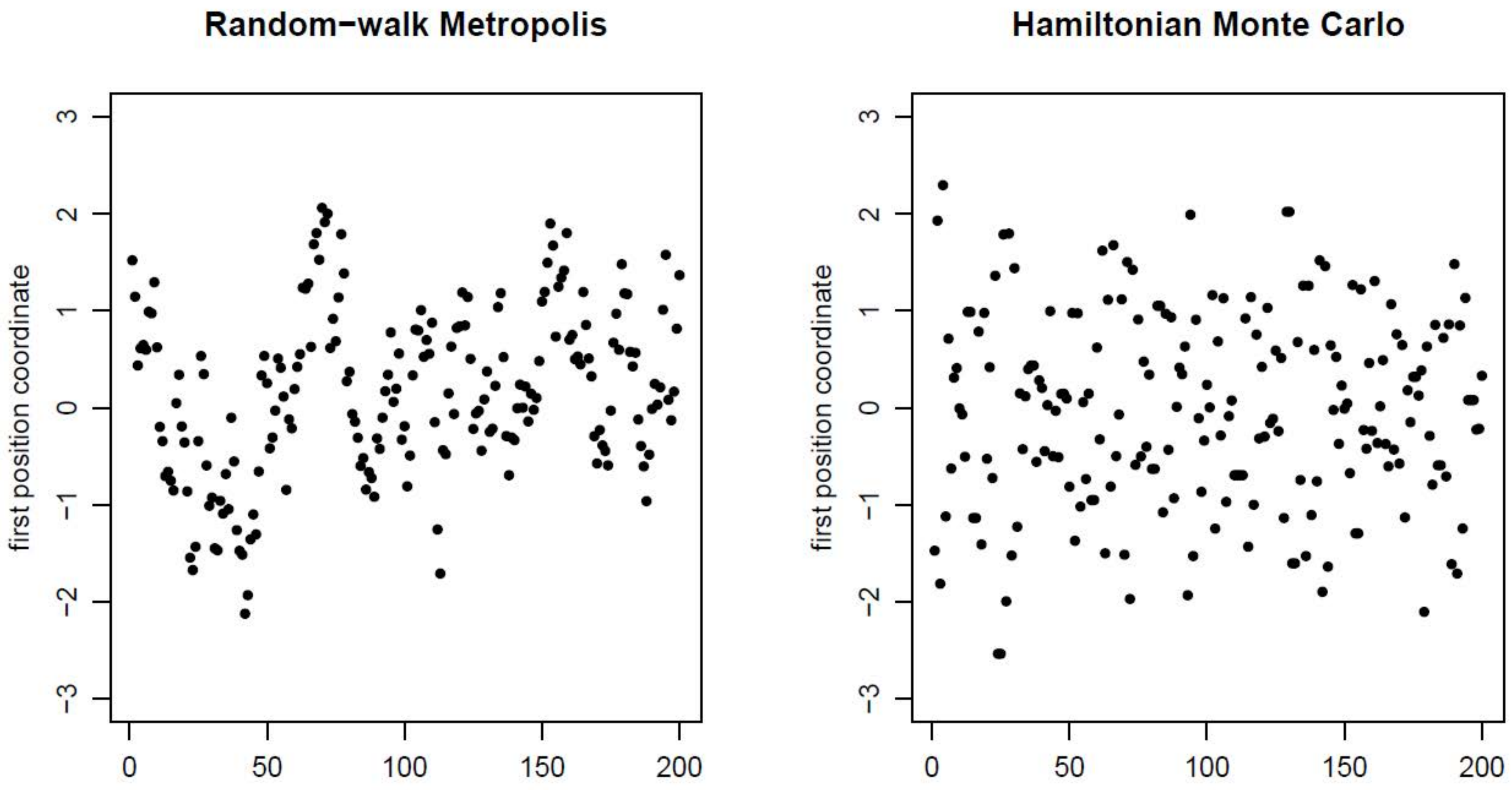
- Detailed balance satisfied
- Ergodic
- canonical distribution invariant

2D Gaussian Example



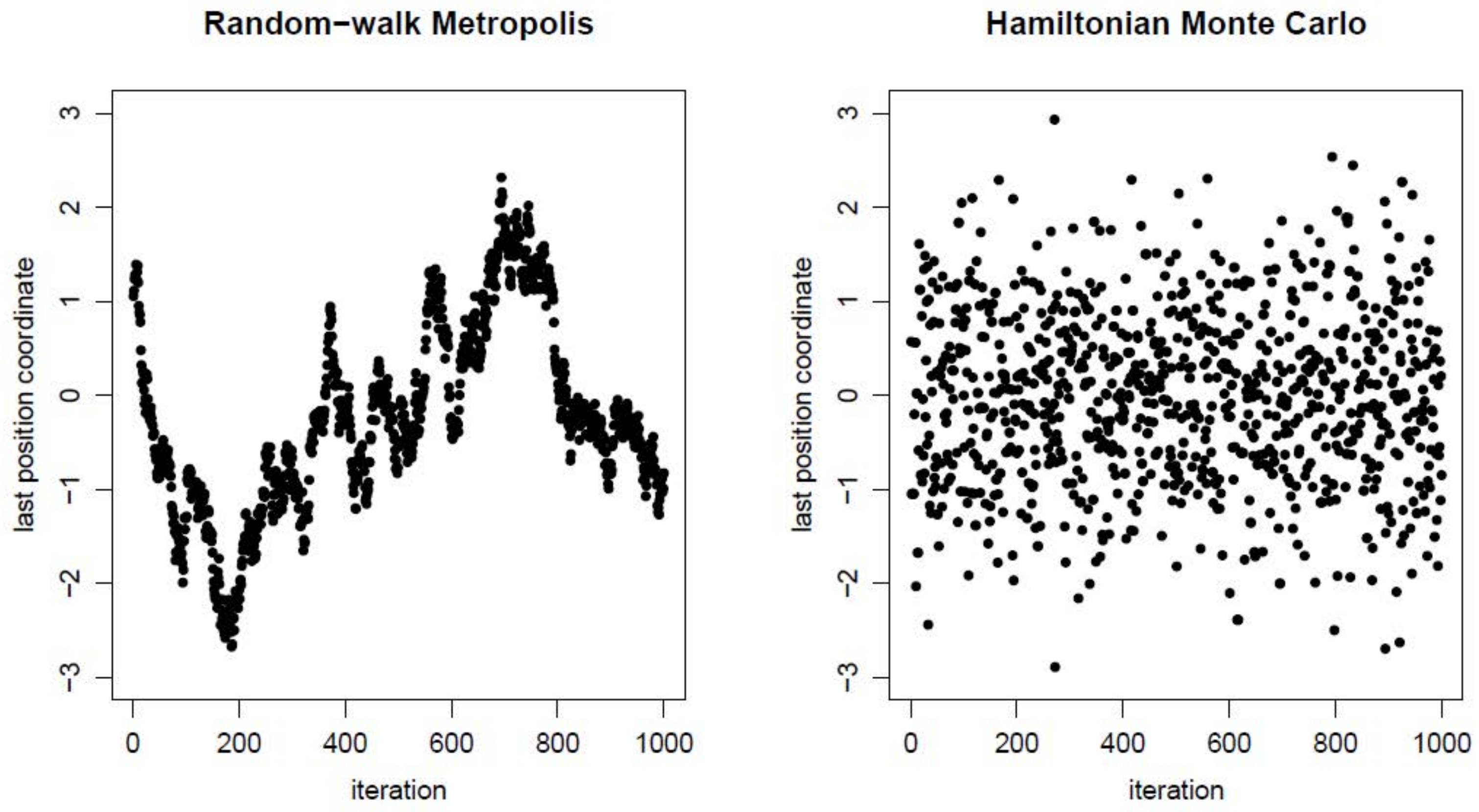
Twenty iterations of the random-walk Metropolis method (with 20 updates per iteration) and of the Hamiltonian Monte Carlo method (with 20 leapfrog steps per trajectory) for a 2D Gaussian distribution with marginal standard deviations of one and correlation 0.98. Only the two position coordinates are plotted, with ellipses drawn one standard deviation away from the mean.

2D Gaussian Example



Two hundred iterations, starting with the twenty iterations shown above, with only the first position coordinate plotted.

100D Gaussian Example



Acceptance Rate

- 2D example HMC : 91% Random Walk: 63%
- 100D example HMC: 87% Random Walk: 25%

Langevin Dynamics

Leapfrog

$$
\begin{aligned}
p_i(t+\varepsilon/2) &= p_i(t) - (\varepsilon/2)\frac{\partial U}{\partial q_i}(q(t)) \\
q_i(t+\varepsilon) &= q_i(t) + \varepsilon\frac{p_i(t+\varepsilon/2)}{m_i} \\
p_i(t+\varepsilon) &= p_i(t+\varepsilon/2) - (\varepsilon/2)\frac{\partial U}{\partial q_i}(q(t+\varepsilon))
\end{aligned}
$$

$$
\begin{aligned}
q_i^* &= q_i - \frac{\varepsilon^2}{2}\frac{\partial U}{\partial q_i}(q) + \varepsilon p_i \\
p_i^* &= p_i - \frac{\varepsilon}{2}\frac{\partial U}{\partial q_i}(q) - \frac{\varepsilon}{2}\frac{\partial U}{\partial q_i}(q^*)
\end{aligned}
$$

accept q^* as the new state with probability

$$
\min\left[1,\ \exp\left(-(U(q^*)-U(q)) - \frac{1}{2}\sum_i((p_i^*)^2 - p_i^2)\right)\right]
$$

Stochastic Langevin Dynamics

- For large datasets hard to compute the whole gradient

$$q_i^* = q_i - \frac{\varepsilon^2}{2}\frac{\partial U}{\partial q_i}(q) + \varepsilon p_i$$

$$\boxed{U(q) = -\log\Big[\pi(q)L(q|D)\Big]}$$

Stochastic Gradient Langevin Dynamics

- For large datasets hard to compute the whole gradient

$$q_i^* = q_i - \frac{\varepsilon^2}{2} \underbrace{\frac{\partial U}{\partial q_i}(q)}_{\text{Calculate using subset of data}} + \varepsilon p_i$$

$$\boxed{U(q) = -\log\left[\pi(q) L(q|D)\right]}$$

Stochastic Gradient Langevin Dynamics: Bayesian Models

- Posterior $p(\theta|\mathbf{X}) \propto p(\theta) \prod_{i=1}^{N} p(x_i|\theta)$

- SGLD update:

$$\Delta\theta_t = \frac{h_t}{2}\left(\nabla \log p(\theta_t) + \frac{N}{n}\sum_{i=1}^{n} \nabla \log p(x_{ti}|\theta_t)\right) + \eta_t$$

$$\eta_t \sim N(0, h_t)$$

$$\boxed{\begin{aligned} q_i^* &= q_i - \frac{\varepsilon^2}{2}\frac{\partial U}{\partial q_i}(q) + \varepsilon p_i \\ U(q) &= -\log\left[\pi(q) L(q|D)\right] \end{aligned}}$$

Stochastic Gradient Langevin Dynamics

- High variance in stochastic gradient

- Take help from the optimization community

Conclusion

- HMC can improve acceptance rate and give better mixing
- Stochastic variants can be used to improve performance in large dataset scenarios
- HMC may not be used for discrete variable

Towards better proposal

- $Q(X_{new}|X_{old})$ determines when the chain converges
- Idea: Variational approximation of P(X) be the proposal distribution

Variational Inference: Recap

- Interested in posterior of parameters $P(\theta|x)$
- Using Jensen's Inequality

$$\boldsymbol{log(p(x|\theta)} \geq \boldsymbol{E_{q(z)}[log(p(x|\theta)]} - \boldsymbol{E_{q(z)}[log\big(q(z)\big)]}$$

- Choose $q(z|\lambda)$ where λ is the variational parameter
- Replace $p(x|\theta)$ with $p(x|\theta,\xi)$ where ξ is another set of variational parameters
- Using this we can easily obtain un-normalized bound for posterior

$$\boldsymbol{P(\theta|x)} \geq \boldsymbol{P^{est}(\theta|x,\lambda,\xi)}$$

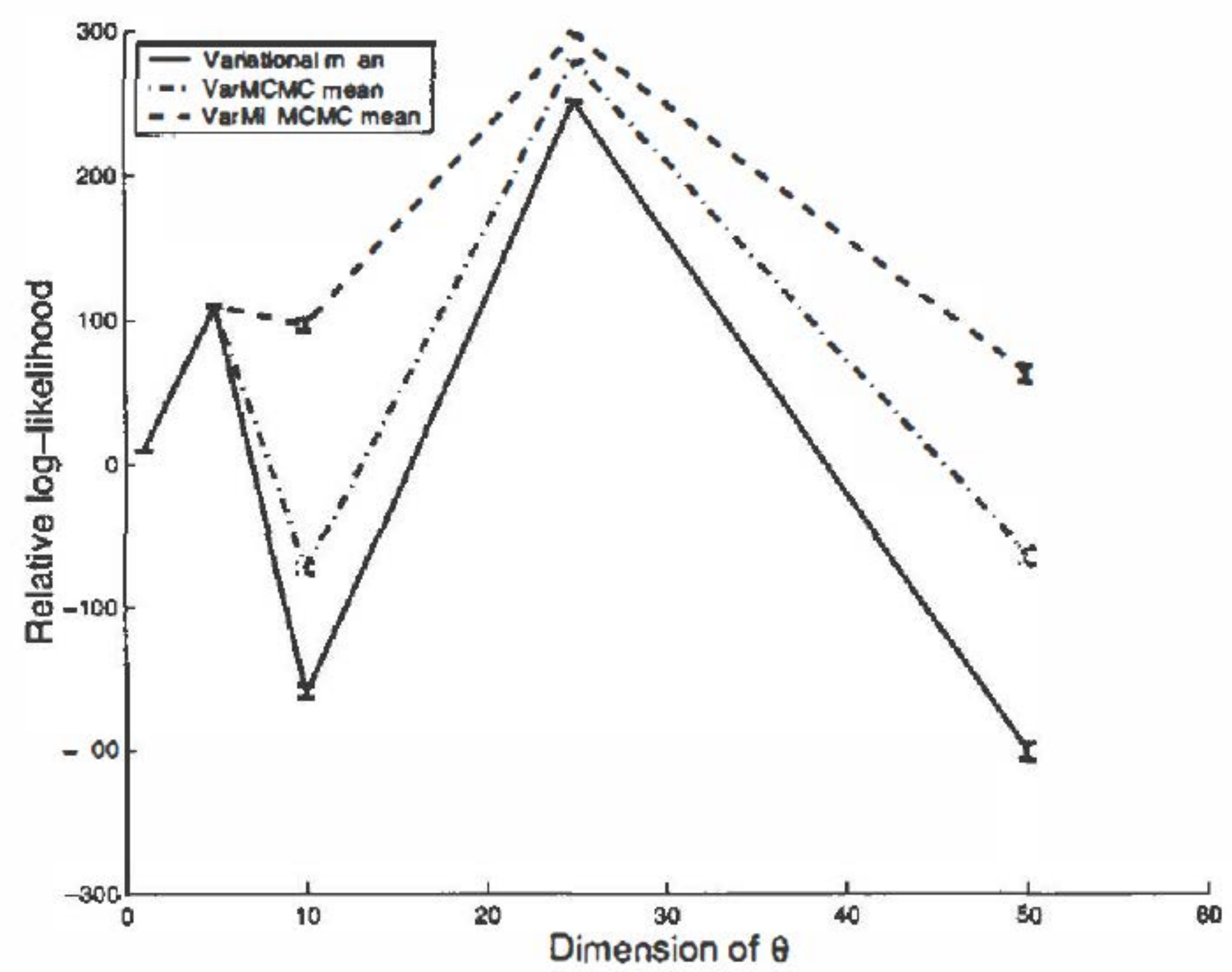
Variational MCMC

- Idea: Variational approximation of P(X) be the proposal distribution
- $Q(\theta_{new}|\theta_{old}) = \boldsymbol{P}^{\boldsymbol{est}}(\boldsymbol{\theta}|\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\xi})$
- Issues:
  - Low acceptance in high dimensions
  - Works well if $\boldsymbol{P}^{\boldsymbol{est}}$ is close to P

Variational MCMC

- Design the proposal in blocks to take care of correlated variables
- Use a mixture of random walk and variational approximation as a proposal distribution
- Now can use stochastic variational methods in estimating $\boldsymbol{P}^{\boldsymbol{est}}(\boldsymbol{\theta}|\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\xi})$

Variational MCMC



Conclusion

- Adapting proposal distribution can be helpful in
  - Increasing mixing
  - Decreasing time to convergence
  - Increasing acceptance rate
  - Getting uncorrelated information