PETUUM

Carnegie Mellon University
School of Computer Science

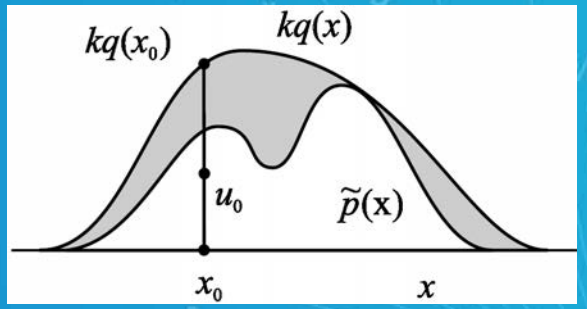
Probabilistic Graphical Models

Monte Carlo Methods

Eric Xing

Lecture 9, February 10, 2020

Reading: see class homepage



Approaches to inference

- Exact inference algorithms
  - The elimination algorithm
  - Message-passing algorithm (sum-product, belief propagation)
  - The junction tree algorithms

- Approximate inference techniques
  - Variational algorithms
    - Loopy belief propagation
    - Mean field approximation
  - Stochastic simulation / sampling methods
  - Markov chain Monte Carlo methods

How to represent a joint, or a marginal distribution?

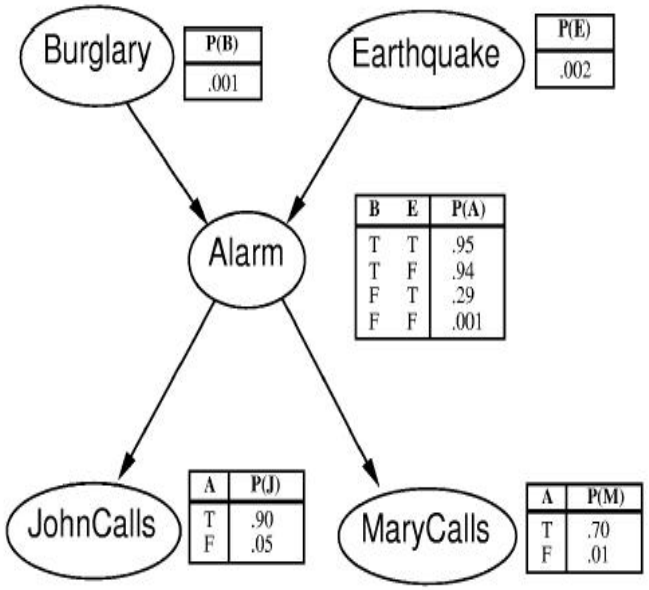
- Closed-form representation
  - E.g., $$(x_1, \ldots, x_p)^T \sim \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\Big(-\frac{1}{2}x-\mu)^T\Sigma^{-1}(x-\mu)\Big)$$

$$\mathrm{E}_p\big(f(x)\big) = \int f(x)p(x)dx$$

- Sample-based representation:

Example: naive sampling

- Construct samples according to probabilities given in a BN.



Alarm example: (Choose the right sampling sequence)
1) Sampling:P(B)=<0.001, 0.999> suppose it is false, B0. Same for E0. P(A|B0, E0)=<0.001, 0.999> suppose it is false...
2) Frequency counting: In the samples right,
P(J|A0)=P(J,A0)/P(A0)=<1/9, 8/9>.

E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J1
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E1	B0	A1	M1	J1
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0

Example: naive sampling

- Construct samples according to probabilities given in a BN.

Alarm example: (Choose the right sampling sequence)

3) what if we want to compute P(J|A1) ?
we have only one sample ...
P(J|A1)=P(J,A1)/P(A1)=<0, 1>.

4) what if we want to compute P(J|B1) ?
No such sample available!
P(J|A1)=P(J,B1)/P(B1) can not be defined.

For a model with hundreds or more variables, rare events will be very hard to garner evough samples even after a long time or sampling ...

E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J1
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E1	B0	A1	M1	J1
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0
E0	B0	A0	M0	J0

Monte Carlo methods

- Draw random samples from the desired distribution
- Yield a stochastic representation of a complex distribution
  - marginals and other expectations can be approximated using sample-based averages

$$E[f(x)] = \frac{1}{N}\sum_{t=1}^{N} f(x^{(t)})$$

- Asymptotically exact and easy to apply to arbitrary models
- Challenges:
  - how to draw samples from a given dist. (not all distributions can be trivially sampled)?
  - how to make better use of the samples (not all sample are useful, or equally useful, see an example later)?
  - how to know we've sampled enough?

Monte Carlo methods (cond.)

- Direct Sampling
  - We have seen it.
  - Very difficult to populate a high-dimensional state space
- Rejection Sampling
  - Create samples like direct sampling, only count samples which is consistent with given evidences.
- Likelihood weighting, ...
  - Sample variables and calculate evidence weight. Only create the samples which support the evidences.
- Markov chain Monte Carlo (MCMC)
  - Metropolis-Hasting
  - Gibbs

Rejection sampling

- Suppose we wish to sample from dist. $\Pi(X)=\Pi'(X)/Z$.
  - $\Pi(X)$ is difficult to sample, but $\Pi'(X)$ is easy to evaluate
  - Sample from a simpler dist $Q(X)$
  - Rejection sampling

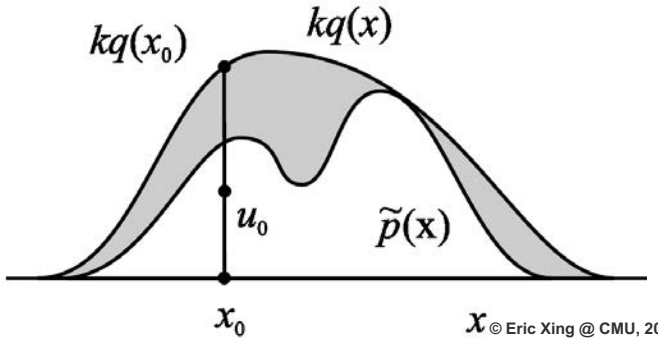
$$x^* \sim Q(X), \qquad \text{accept } x^* \text{ w.p. } \Pi'(x^*)/kQ(x^*)$$

  - Correctness:

$$p(x) = \frac{[\Pi'(x)/kQ(x)]Q(x)}{\int [\Pi'(x)/kQ(x)]Q(x)dx}$$

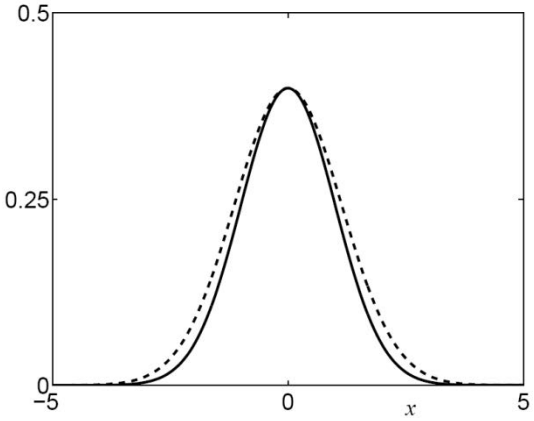
$$= \frac{\Pi'(x)}{\int \Pi'(x)dx} = \Pi(x)$$

  - Pitfall …

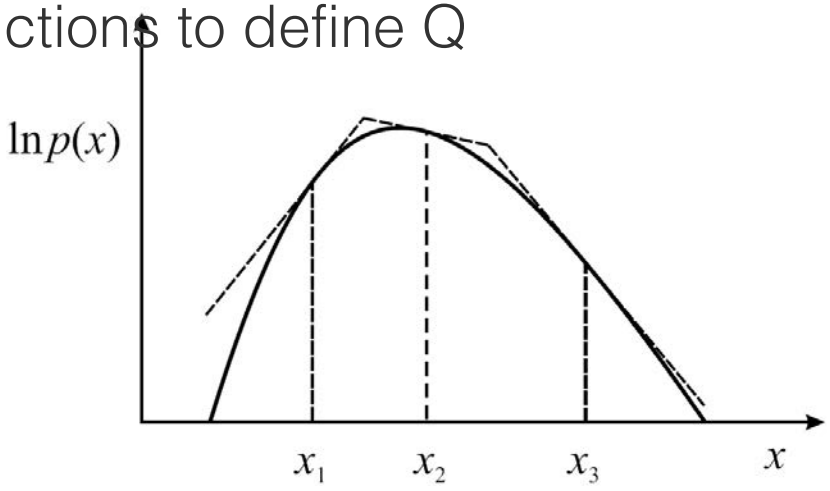


Rejection sampling

- Pitfall:
  - Using $Q=N(\mu,\sigma_q^{2/d})$ to sample $P=N(\mu,\sigma_p^{2/d})$
  - If σ_q exceeds σ_p by 1%, and dimensional=1000,
  - The optimal acceptance rate $k=(\sigma_q/\sigma_p)^d \approx 1/20{,}000$
  - Big waste of samples!

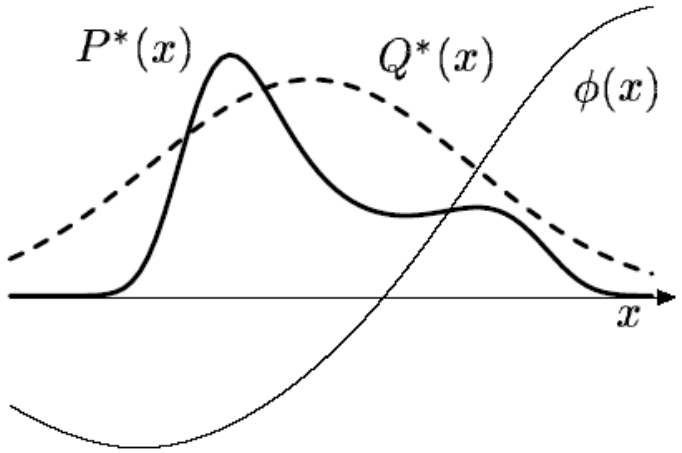


- Adaptive rejection sampling
  - Using envelope functions to define Q



Unnormalized importance sampling

- Suppose sampling from $P(\cdot)$ is hard.
- Suppose we can sample from a "simpler" proposal distribution $Q(\cdot)$ instead.
- If Q dominates P (i.e., $Q(x) > 0$ whenever $P(x) > 0$), we can sample from Q and reweight:



$$\langle f(X)\rangle = \int f(x)P(x)dx$$
$$= \int f(x)\frac{P(x)}{Q(x)}Q(x)dx$$
$$\approx \frac{1}{M}\sum_m f(x^m)\frac{P(x^m)}{Q(x^m)} \quad \text{where } x^m \sim Q(X)$$
$$= \frac{1}{M}\sum_m f(x^m)w^m$$

- What is the problem here?

Normalized importance sampling

- Suppose we can only evaluate $P'(x) = \alpha P(x)$ (e.g. for an MRF).
- We can get around the nasty normalization constant α as follows:

$$\text{Let } r(X) = \frac{P'(x)}{Q(x)} \qquad \Rightarrow \quad \langle r(X) \rangle_Q = \int \frac{P'(x)}{Q(x)} Q(x) dx = \int P'(x) dx = \alpha$$

- Now

$$\langle f(X) \rangle_P = \int f(x) P(x) dx = \frac{1}{\alpha} \int f(x) \frac{P'(x)}{Q(x)} Q(x) dx$$

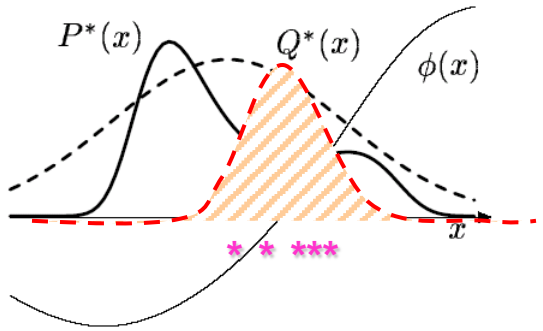
$$= \frac{\int f(x) r(x) Q(x) dx}{\int r(x) Q(x) dx}$$

$$\approx \frac{\sum_m f(x^m) r^m}{\sum_m r^m} \qquad \text{where } x^m \sim Q(X)$$

$$= \sum_m f(x^m) w^m \qquad \text{where } w^m = \frac{r^m}{\sum_m r^m}$$

Weighted resampling

- Problem of importance sampling: depends on how well Q matches P
  - If P(x)f(x) is strongly varying and has a significant proportion of its mass concentrated in a small region, r_m will be dominated by

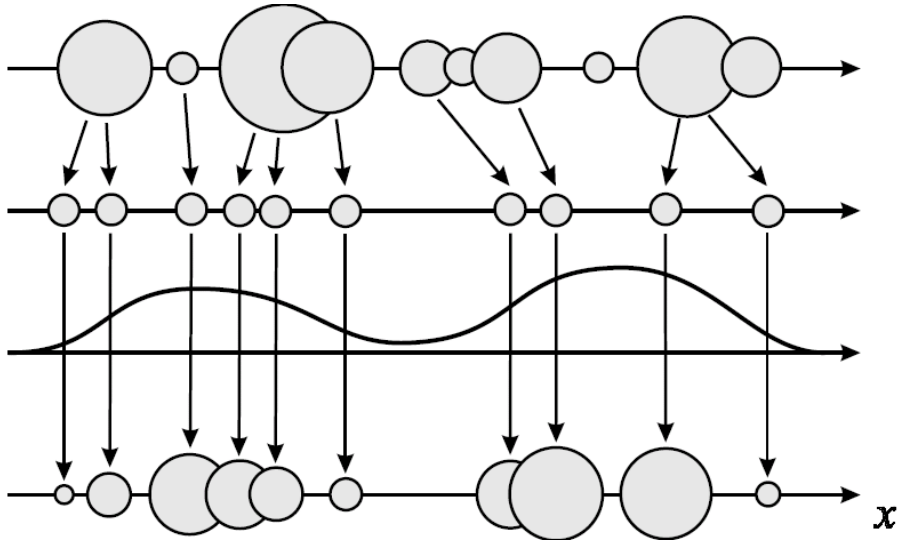


  - Note that if the high-prob mass region of Q falls into the low-prob mass region of P, the variance of $r^m = P(x^m)/Q(x^m)$ can be small even if the samples come from low-prob region of P and potentially erroneous .
- Solution
  - Use heavy tail Q.
  - Weighted resampling

$$w^m = \frac{P(x^m)/Q(x^m)}{\sum_l P(x^l)/Q(x^l)} = \frac{r^m}{\sum_m r^m}$$

Weighted resampling

- Sampling importance resampling (SIR):
  1. Draw N samples from Q: $X_1 \dots X_N$
  2. Construct weights: $w_1 \dots w_N$, $w^m = \frac{P(x^m)/Q(x^m)}{\sum_l P(x^l)/Q(x^l)} = \frac{r^m}{\sum_m r^m}$
  3. Sub-sample x from $\{X_1 \dots X_N\}$ w.p. $(w_1 \dots w_N)$



Recap of Monte Carlo so far

- 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 $p(x)$

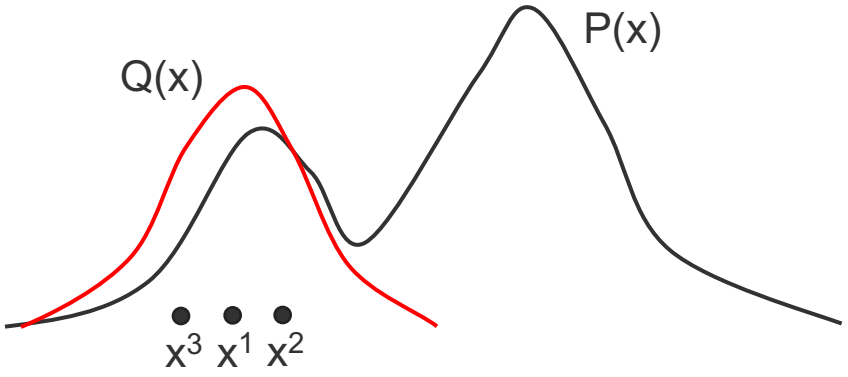
Limitations of Monte Carlo

- 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?

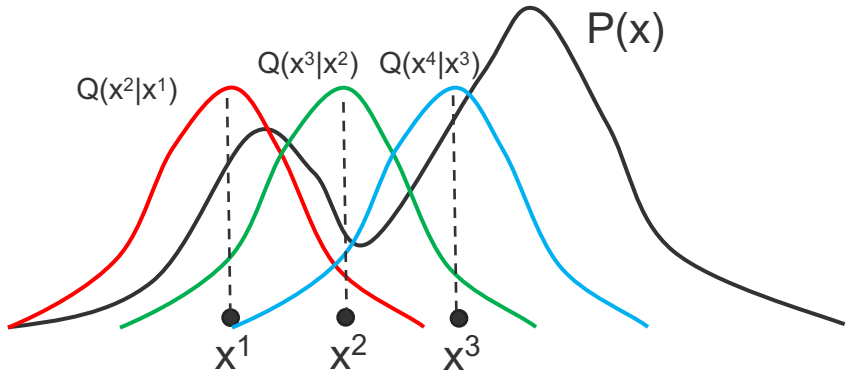
Markov Chain Monte Carlo

- 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 $u < A(x^*|x) = \min\left(1, \frac{P(x^*)Q(x|x^*)}{P(x)Q(x^*|x)}\right)$
     - $x^{(t)} = x^*$ // transition
   - else
     - $x^{(t)} = x$ // stay in current state
- Take samples from P(x): Reset t=0, for $t=1:N$
  - $x(t+1) \leftarrow$ Draw sample ($x(t)$)

Function
Draw sample ($x(t)$)

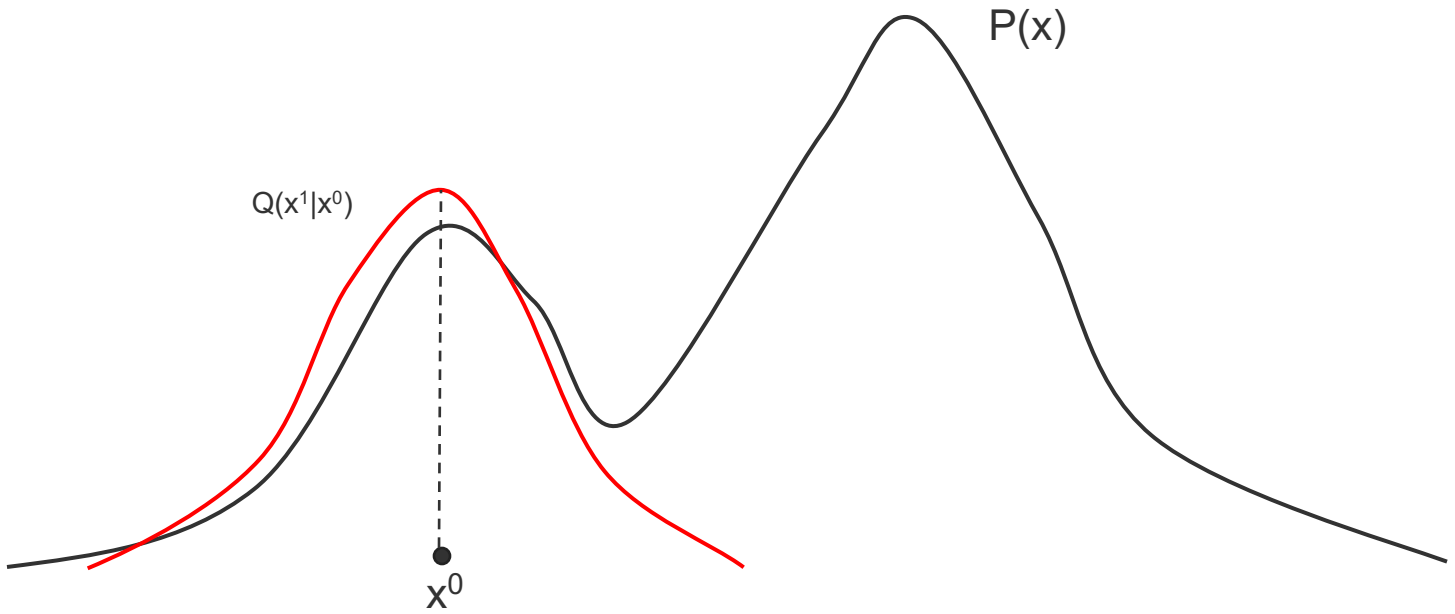
The MH Algorithm

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

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

Initialize $x^{(0)}$

...

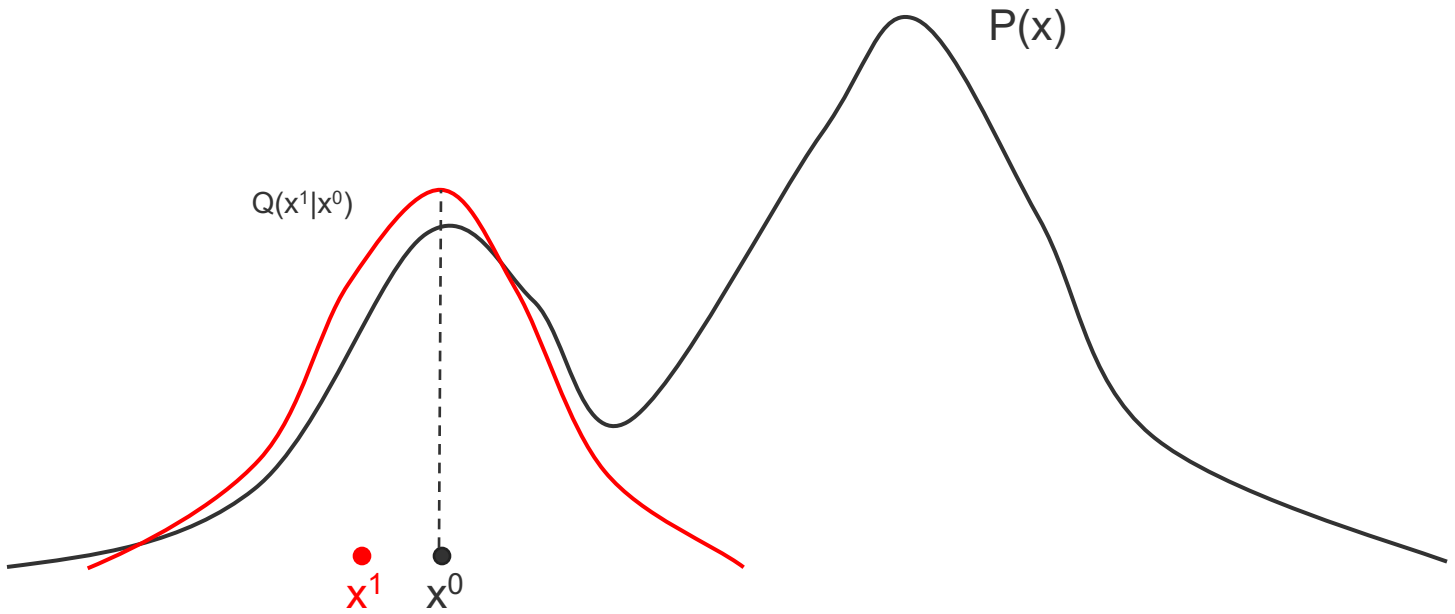


The MH Algorithm

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

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

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

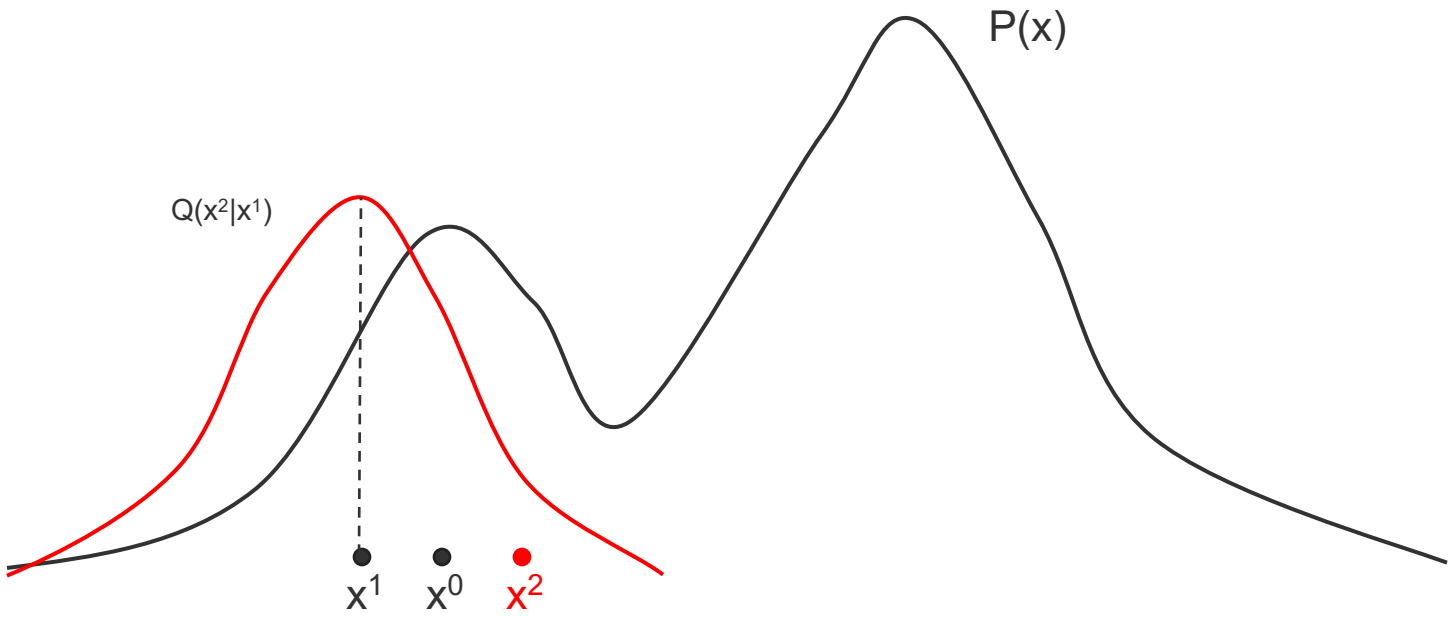


The MH Algorithm

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

- 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

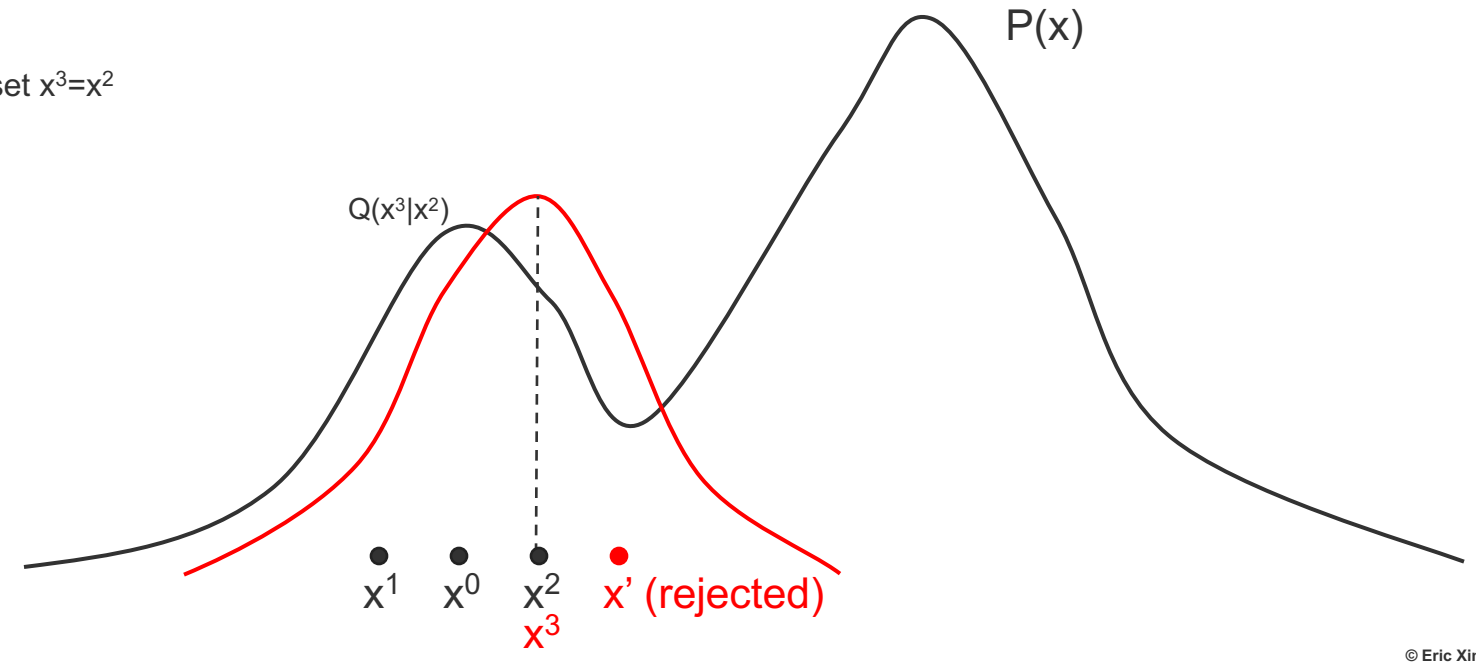


The MH Algorithm

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

- 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 $x^3=x^2$



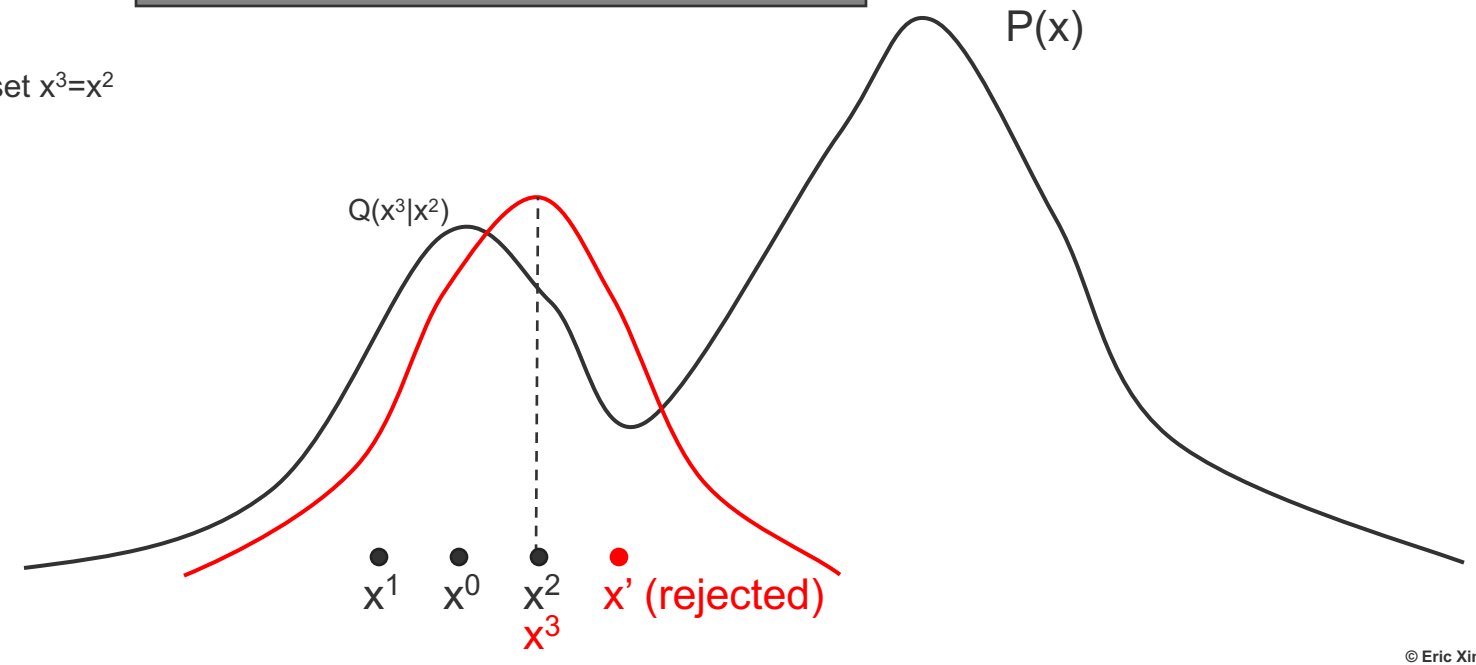
The MH Algorithm

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

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

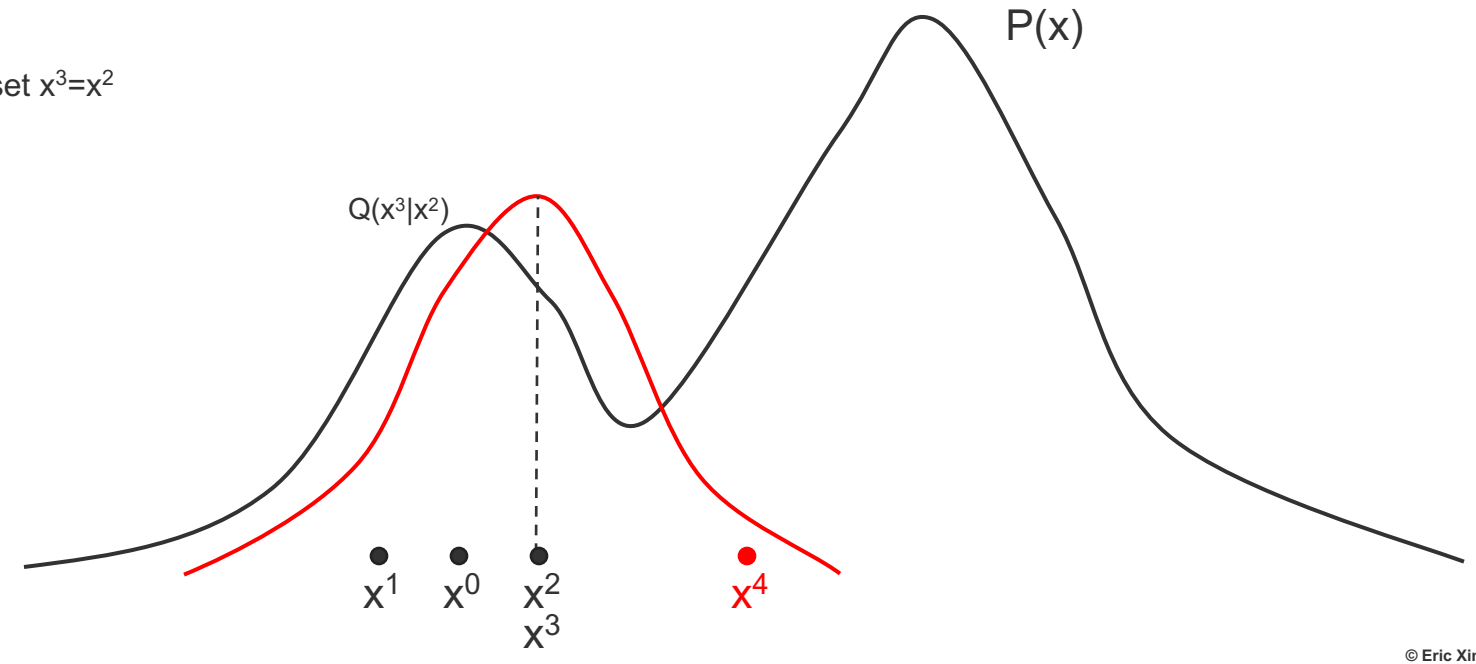


The MH Algorithm

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

- 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

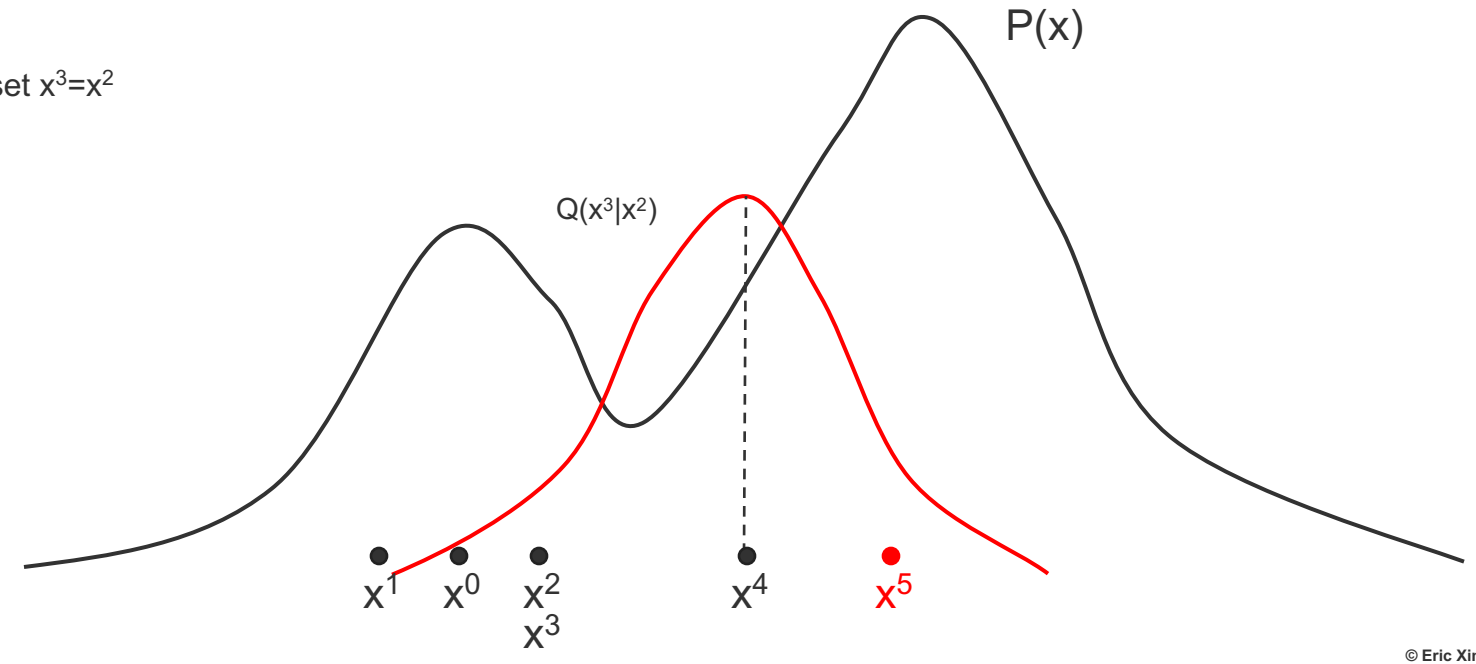


The MH Algorithm

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

- 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



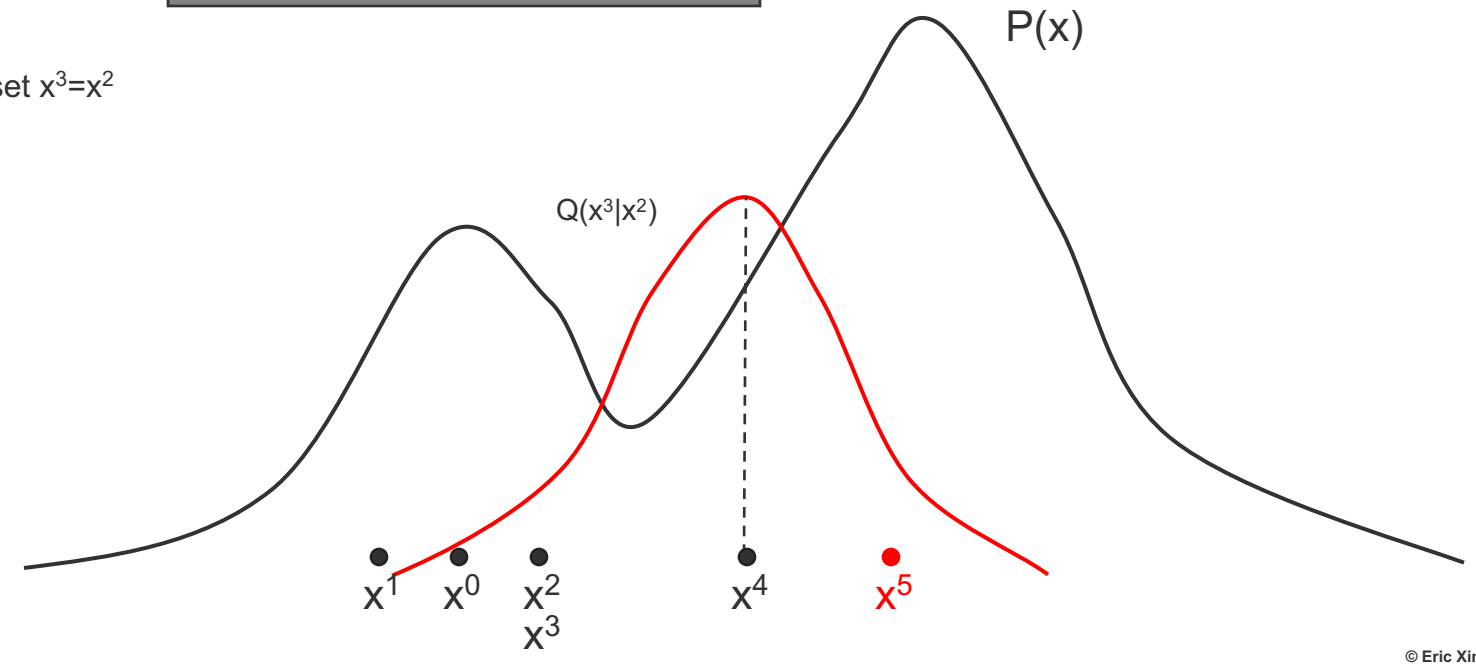
The MH Algorithm

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

- 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 $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$ 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 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 $P(x^{(t)} = x \mid x^{(t-1)})$

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 all 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:

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

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 $x_i \leftarrow x$

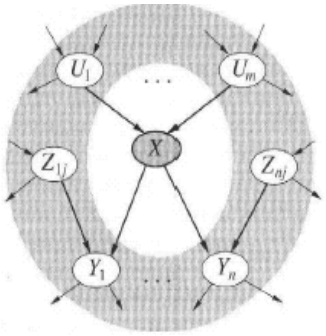
- When we update x_i, we <u>immediately</u> use its new value for sampling other variables x_j

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

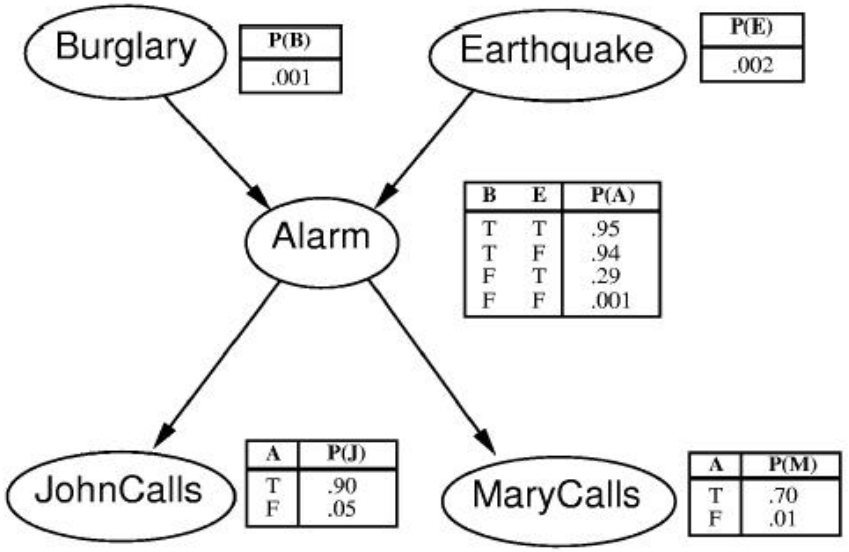
$$P(x_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i \mid MB(x_i))$$

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



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

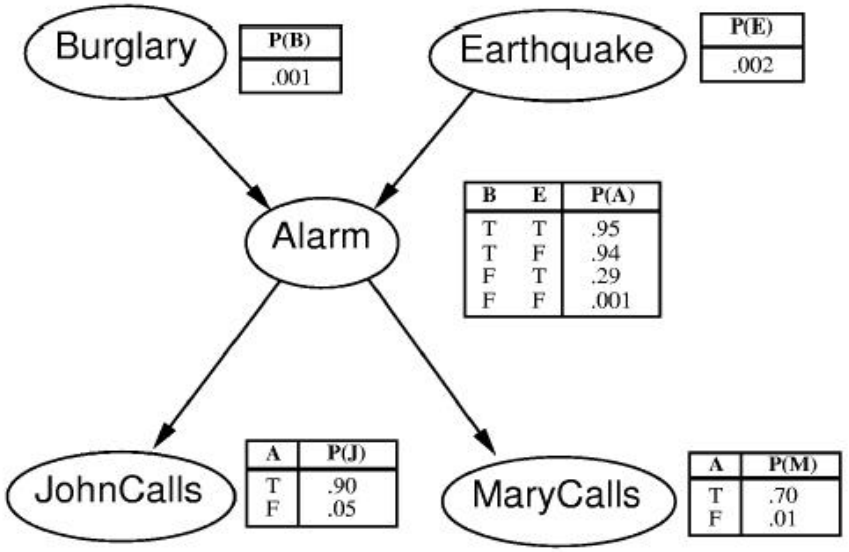
Gibbs Sampling: An Example



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,

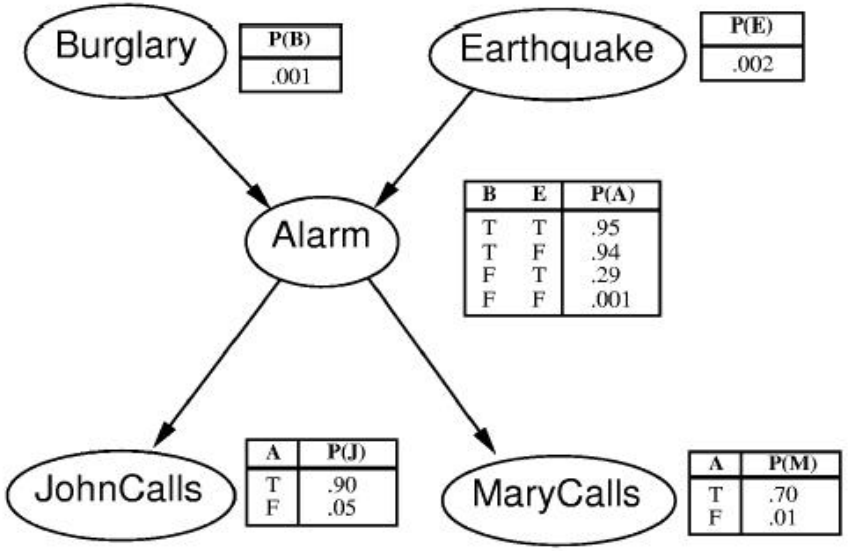
$$P(B \mid A,E) \propto P(A \mid B,E)P(B)$$

- (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,

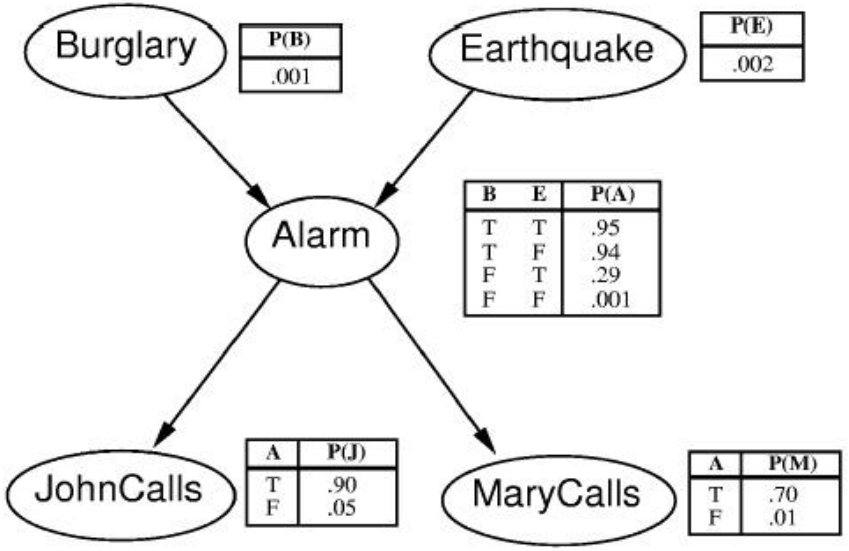
$$P(E \mid A, B) \propto P(A \mid B, E)P(E)$$

- (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,

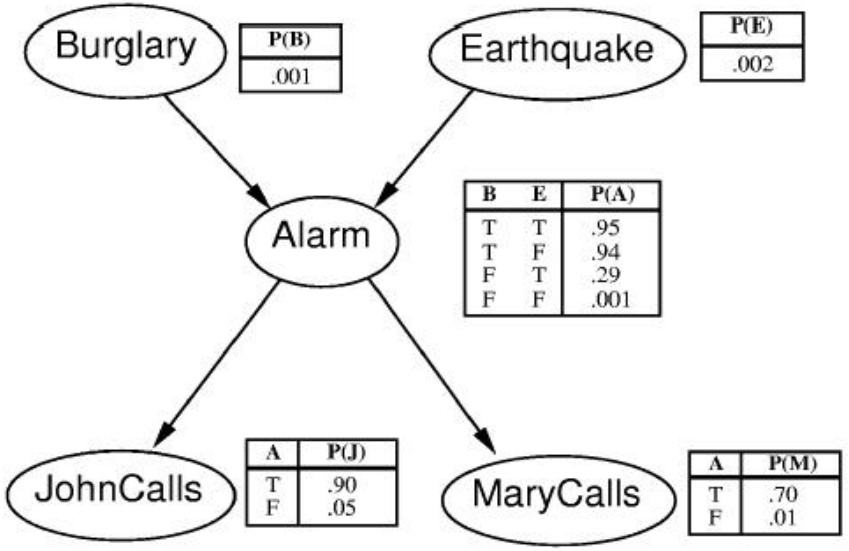
$$P(A \mid B,E,J,M) \propto P(J \mid A)P(M \mid A)P(A \mid B,E)$$

- (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					

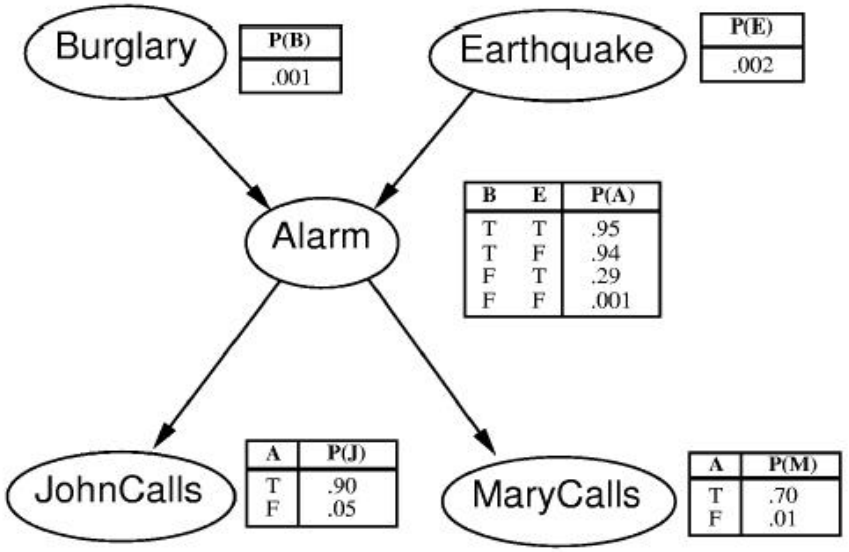
- Sampling P(J|A): No need to apply Bayes Rule

- 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					

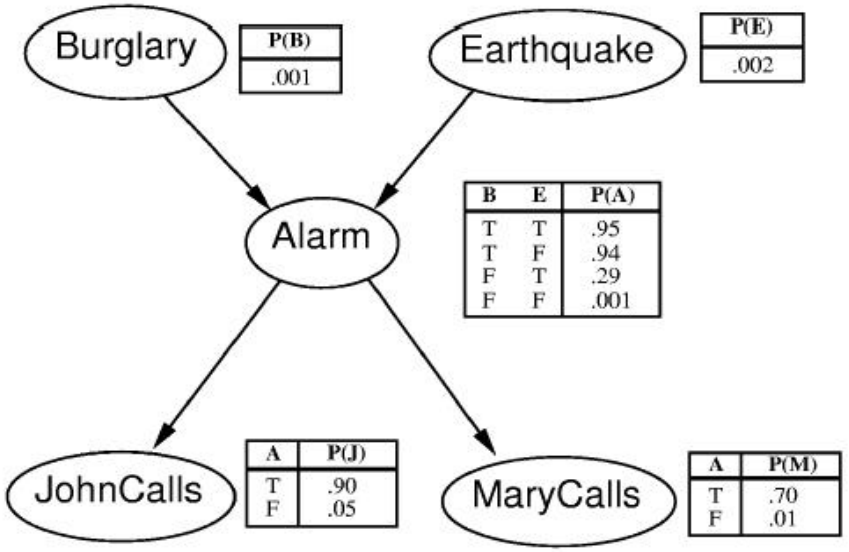
- Sampling P(M|A): No need to apply Bayes Rule

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

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

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

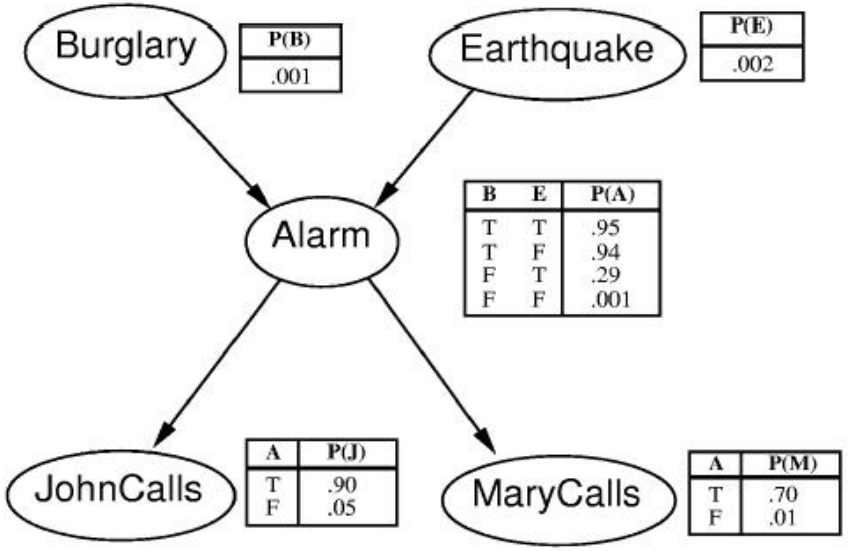
Gibbs Sampling: An Example



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)

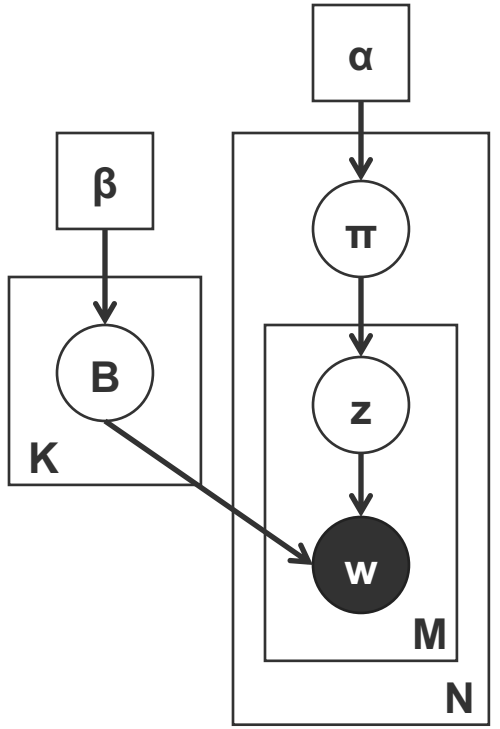
- Collapsed Gibbs sampling
  - Popular inference algorithm for topic models
  - Integrate out topic vectors π and topics B
  - Only need to sample word-topic assignments z

Algorithm:

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

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

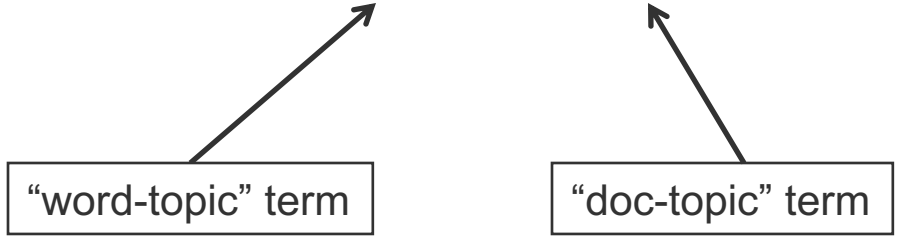
where $\mathbf{z}_{-i} = z_1^{(t+1)}, z_2^{(t+1)}, \ldots, z_{i-1}^{(t+1)}, z_{i+1}^{(t)}, \ldots, z_n^{(t)}$



Collapsed Gibbs sampling

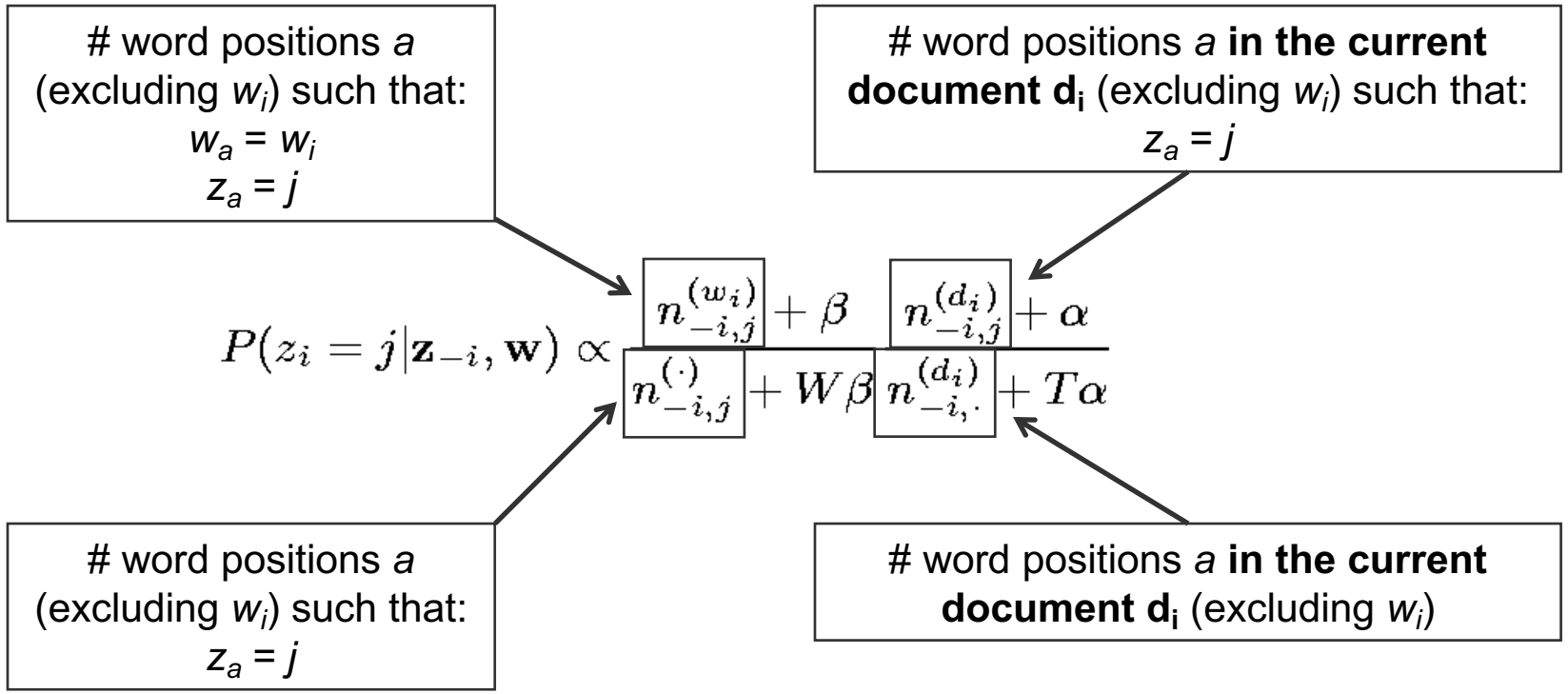
- What is $P(z_i|\mathbf{z}_{-i}, \mathbf{w})$?
  - It is a product of two Dirichlet-Multinomial conditional distributions:

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



Collapsed Gibbs sampling

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



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	iteration 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	iteration 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	iteration 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	iteration 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	iteration 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	iteration 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):

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

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

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

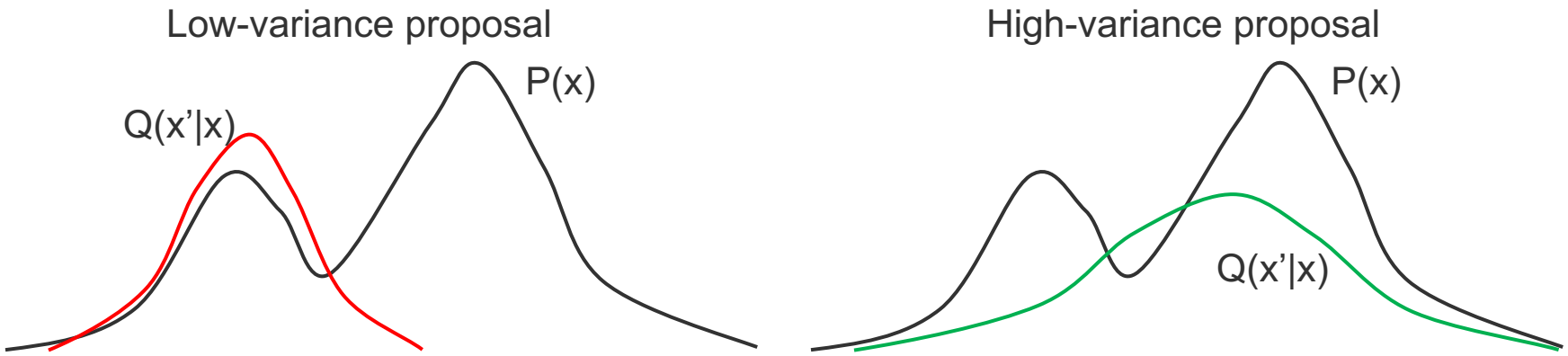
  - GS is simply MH with a proposal that is always accepted!

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

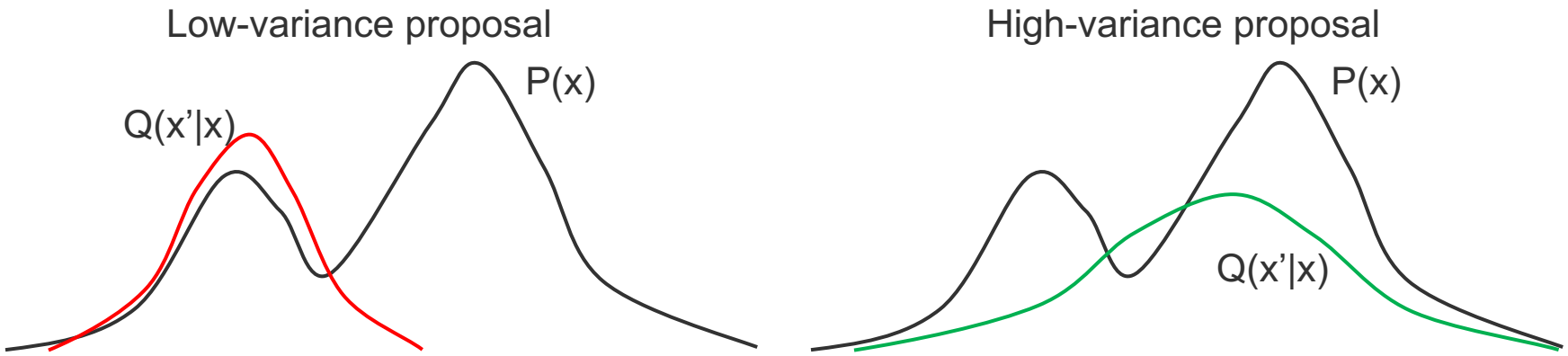
Acceptance Rate



- 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

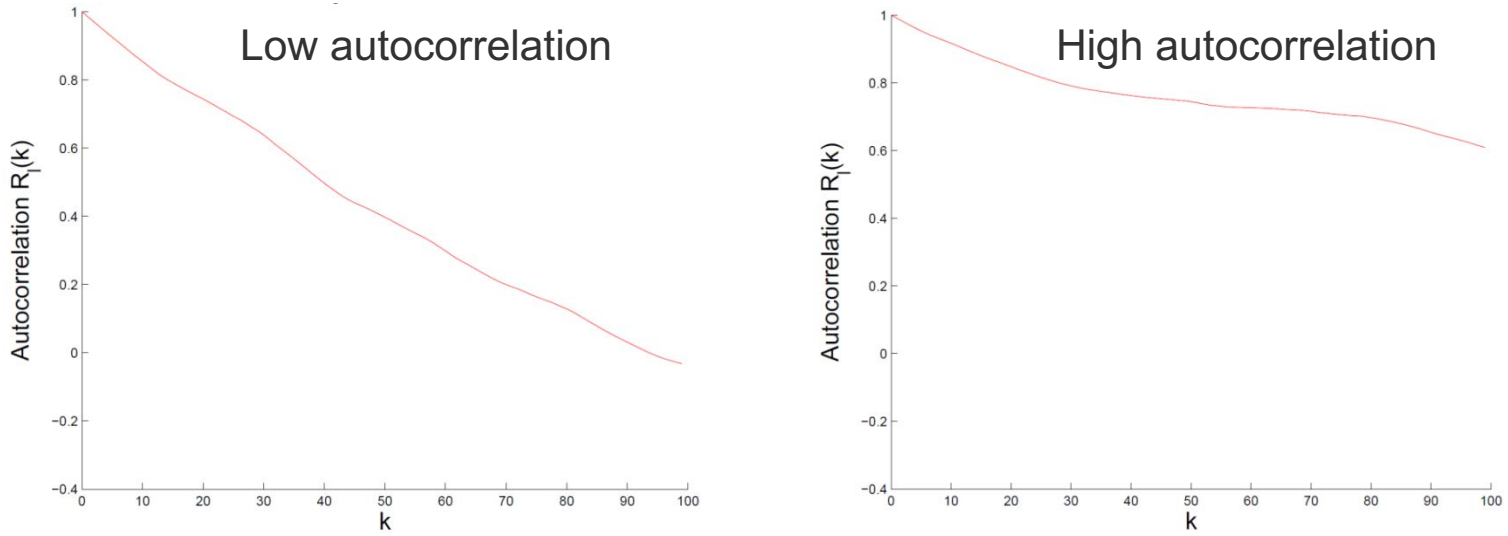


- Acceptance rate is the fraction of samples that MH accepts.
  - General guideline: proposals should have ~0.5 acceptance rate [1]

- 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

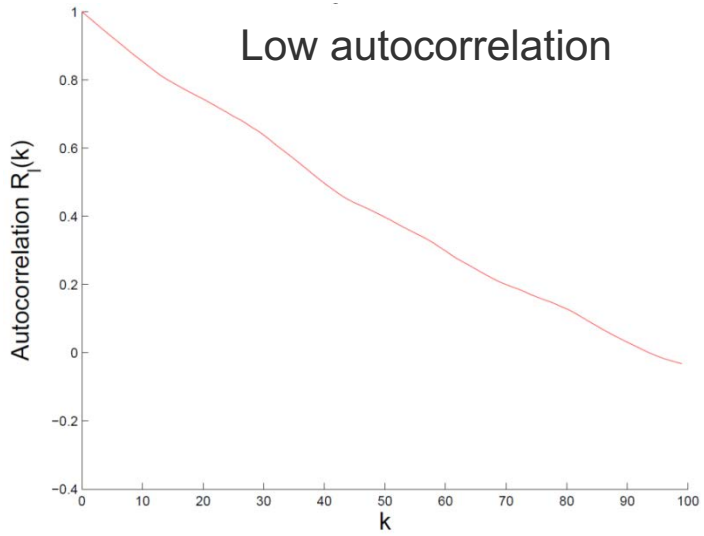


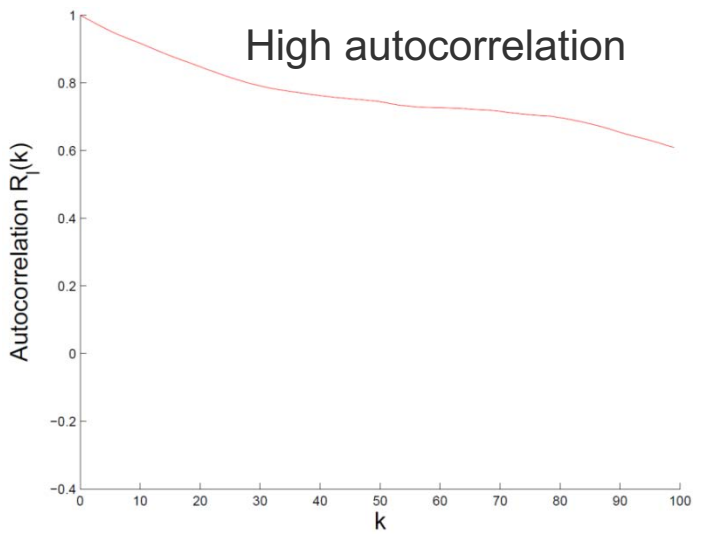
- MCMC chains always show autocorrelation (AC)
  - AC means that adjacent samples in time are highly correlated
- We quantify AC with the autocorrelation function of an r.v. x:

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

Autocorrelation function

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





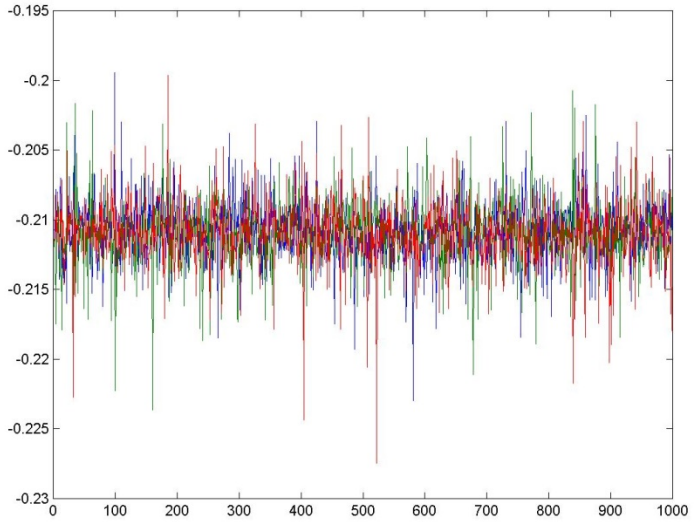
- The first-order AC $R_x(1)$ can be used to estimate the Sample Size Inflation Factor (SSIF):

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

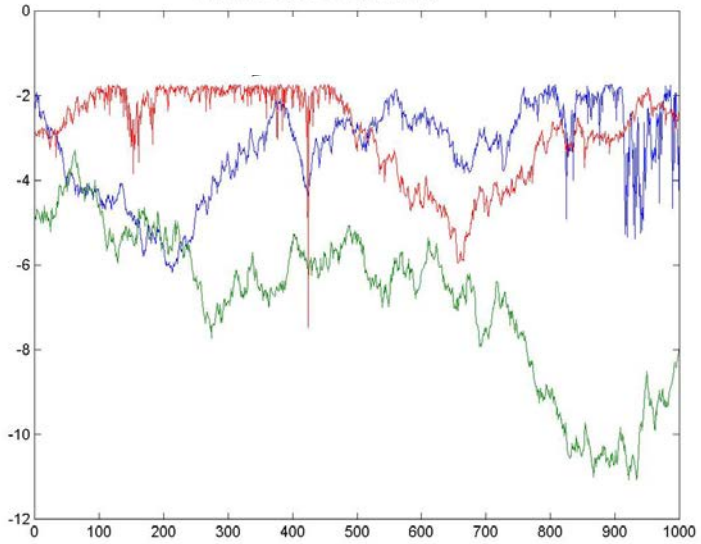
  - If we took n samples with SSIF s_x, then the effective sample size is n/s_x
  - High autocorrelation leads to smaller effective sample size!
  - We want proposals Q(x'|x) with low autocorrelation

Sample Values vs Time

Well-mixed chains

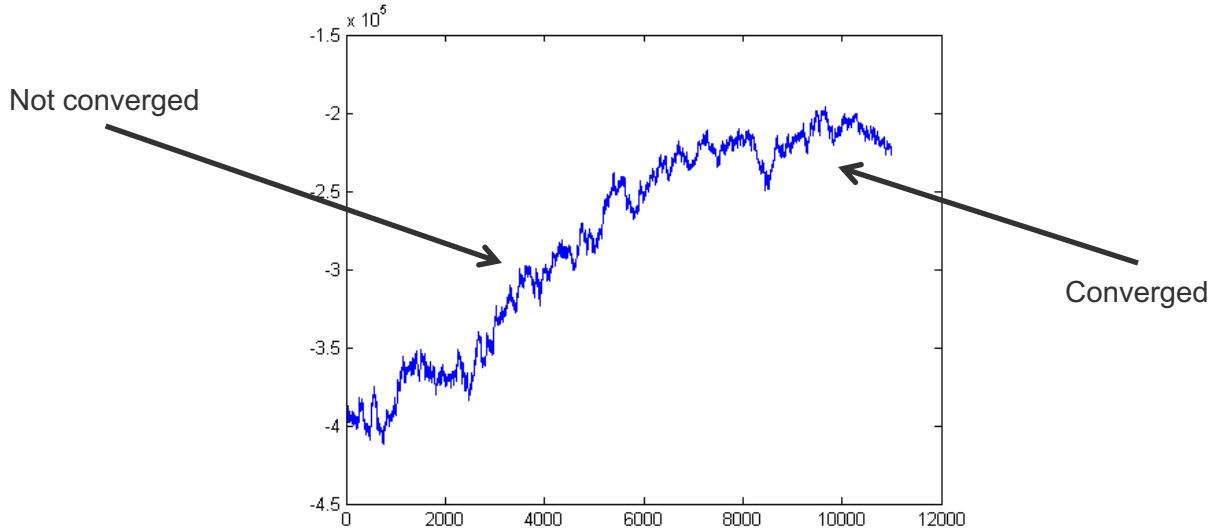


Poorly-mixed chains



- 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

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: Monte Carlo Methods

- Direct Sampling
  - Very difficult to populate a high-dimensional state space
- Rejection Sampling
  - Create samples like direct sampling, only count samples which is consistent with given evidences.
- Likelihood weighting, ...
  - Sample variables and calculate evidence weight. Only create the samples which support the evidences.
- Markov chain Monte Carlo (MCMC)
  - Metropolis-Hasting
  - Gibbs

Summary: Markov Chain Monte Carlo

- 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

PETUUM

Supplementary

Rao-Blackwellised sampling

- Sampling in high dimensional spaces causes high variance in the estimate.
- RB idea: sample some variables X_p, and conditional on that, compute expected value of rest X_d analytically:

$$
\begin{aligned}
E_{p(X|e)}[f(X)] &= \int p(x_p, x_d \mid e) f(x_p, x_d) dx_p dx_d \\
&= \int_{x_p} p(x_p \mid e) \left(\int_{x_d} p(x_d \mid x_p, e) f(x_p, x_d) dx_d \right) dx_p \\
&= \int_{x_p} p(x_p \mid e) E_{p(X_d|x_p,e)}[f(x_p, X_d)] dx_p \\
&= \frac{1}{M} \sum_m E_{p(X_d|x_p^m,e)}[f(x_p^m, X_d)], \qquad x_p^m \sim p(x_p \mid e)
\end{aligned}
$$

- This has lower variance, because of the identity:

$$
\mathrm{var}[\tau(X_p, X_d)] = \mathrm{var}[E[\tau(X_p, X_d) \mid X_p]] + E[\mathrm{var}[\tau(X_p, X_d) \mid X_p]]
$$

Rao-Blackwellised sampling

- Sampling in high dimensional spaces causes high variance in the estimate.
- RB idea: sample some variables X_p, and conditional on that, compute expected value of rest X_d analytically:

$$
\begin{aligned}
E_{p(X|e)}[f(X)] &= \int p(x_p, x_d \mid e) f(x_p, x_d) dx_p dx_d \\
&= \int_{x_p} p(x_p \mid e) \left(\int_{x_d} p(x_d \mid x_p, e) f(x_p, x_d) dx_d \right) dx_p \\
&= \int_{x_p} p(x_p \mid e) E_{p(X_d|x_p,e)}[f(x_p, X_d)] dx_p \\
&= \frac{1}{M} \sum_m E_{p(X_d|x_p^m,e)}[f(x_p^m, X_d)], \qquad x_p^m \sim p(x_p \mid e)
\end{aligned}
$$

- This has lower variance, because of the identity:

$$\mathrm{var}[\tau(X_p, X_d)] = \mathrm{var}[E[\tau(X_p, X_d) \mid X_p]] + E[\mathrm{var}[\tau(X_p, X_d) \mid X_p]]$$

- Hence $\mathrm{var}[E[\tau(X_p, X_d) \mid X_p]] \le \mathrm{var}[\tau(X_p, X_d)]$, so $\tau(X_p, X_d) = E[f(X_p, X_d) \mid X_p]$ is a lower variance estimator.