Probabilistic Graphical Models & Probabilistic AI

Ben Lengerich

Lecture 14: Markov Chain Monte Carlo

March 13, 2025

Reading: See course homepage

Logistics

- Next week:
  - HW5 due Tuesday, **March 18.**
  - Midterm exam Thursday, March 20 **in-class**.
    - Study guide released.
- Looking ahead:
  - Project midway report due **April 11**.
    - **Updated expectations on course website.**

Today

- Approximate Inference, Monte Carlo Methods
- Markov Chain Monte Carlo
  - Metropolis-Hastings
  - Gibbs Sampling

Approximate Inference

A Brief Recap of our Roadmap

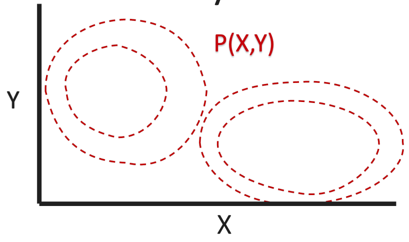
Prior Knowledge

Parametric Family P_θ

Regularizer / Prior

Disc / Gen Models - Lecture 3

Probability Distribution



Observations

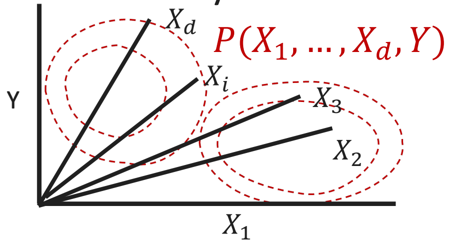
X	Y
0	0
0.1	1
0.2	1

Estimation $\rightarrow \hat{\theta}$

MLE / MAP - Lecture 2

A Brief Recap of our Roadmap

Probability Distribution



Observations

X	Y
0	0
0.1	1
0.2	1

Local Structure

Prior Knowledge

Graphs - Lectures 4,5

Parametric Family P_θ

Regularizer / Prior

Disc / Gen Models - Lecture 3

Estimation

MLE / MAP - Lecture 2

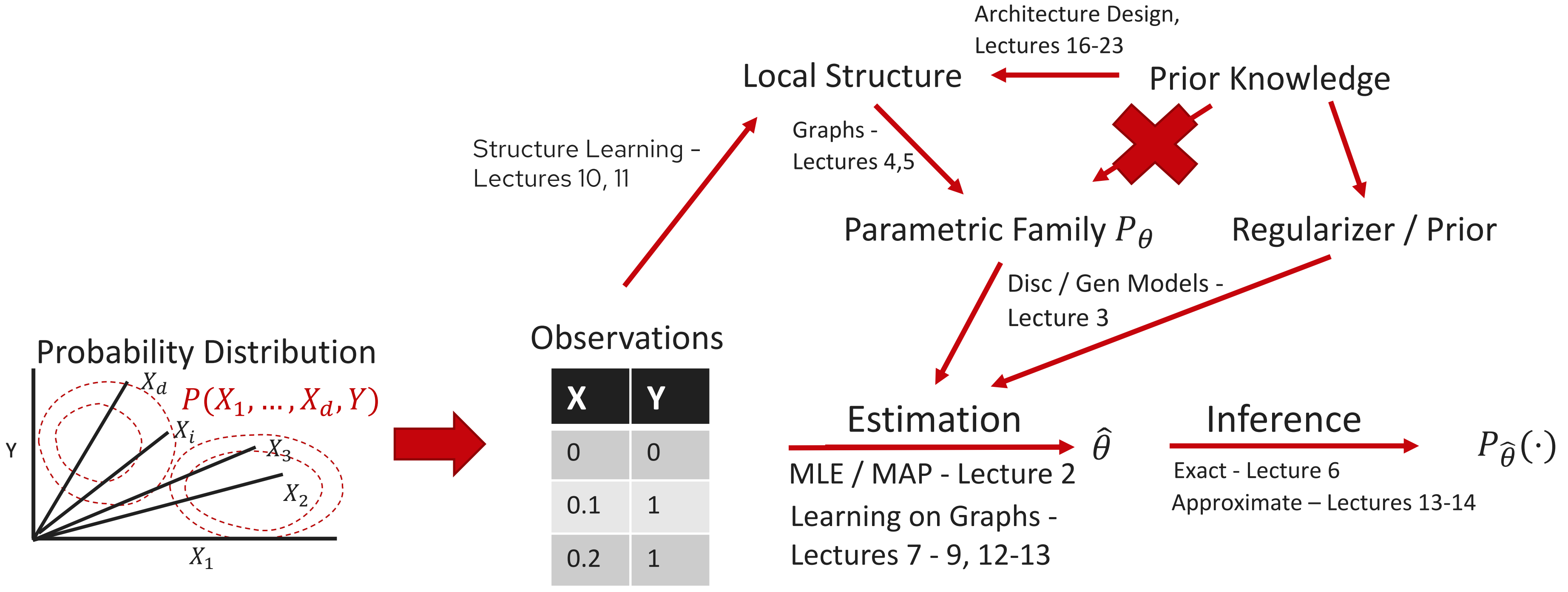
$\hat{\theta}$

Inference

Exact - Lecture 6

$P_{\hat{\theta}}(\cdot)$

A Brief Recap of our Roadmap



X	Y
0	0
0.1	1
0.2	1

Inference

- Inference
  - How do I answer questions/queries according to my model and/or based on observed data?

$$\text{e.g. } P_M(X_i|D)$$

- We have seen exact inference:
  - $P_M(X_i|D)$ is factorized according to graph structure
  - Computational difficulty can be changed by variable elimination order

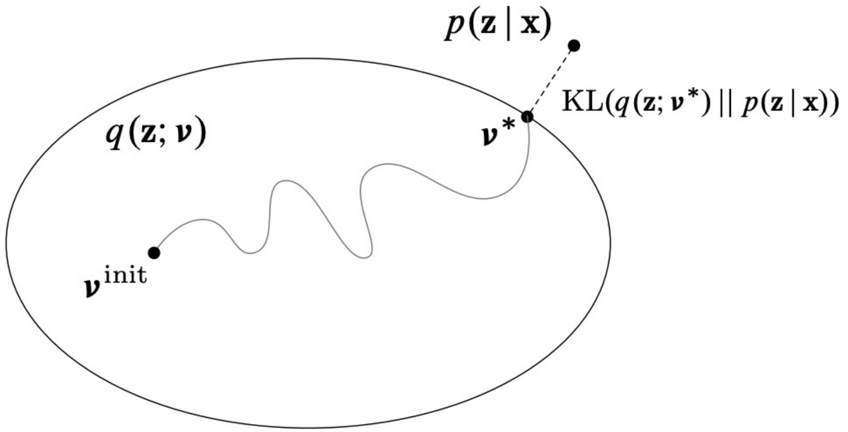
What should we do if $P_M(X_i|D)$ is a very complicated distribution?

→ Approximate inference

Approximate Inference

- Variational Inference
  - Mean-field: Replace $P_M(X_i|D)$ with:

$$\max_q \exp\left(E_{q(z)}[\log P(X,Z|D)] - E_{q(z)}[\log q(Z)]\right)$$



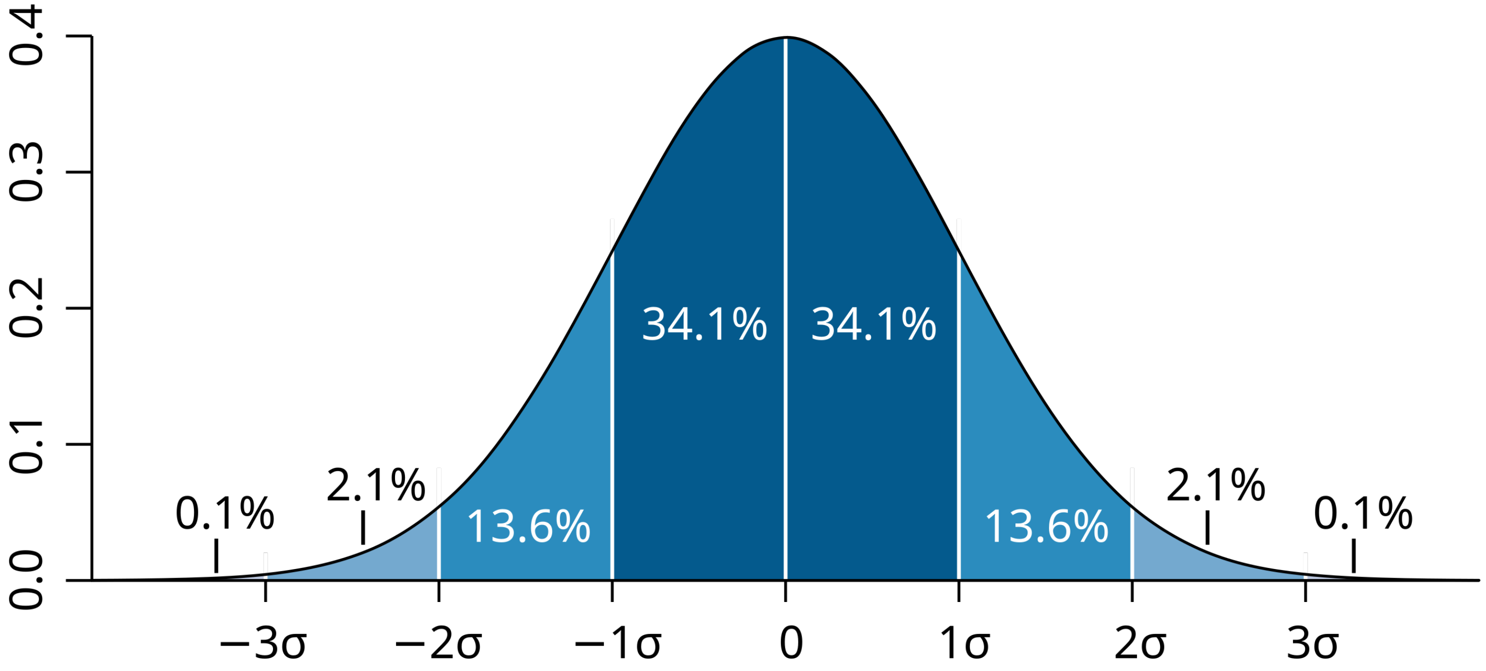
What should we do if the approximation class q is too far from the actual p?

→ Monte Carlo methods

Monte Carlo Methods

How to define a distribution?

- Parametric family with specific parameter values.
- Collection of samples



Monte Carlo methods: define dist by samples

- Draw random samples from desired distribution
- Yield a stochastic representation of desired distribution
  - $E_p[f(x)] = \frac{\sum_m f(X_m)}{|m|}$
- **Asymptotically exact**
- Challenges:
  - How to draw samples from desired distribution?
  - How to know we've sampled enough?

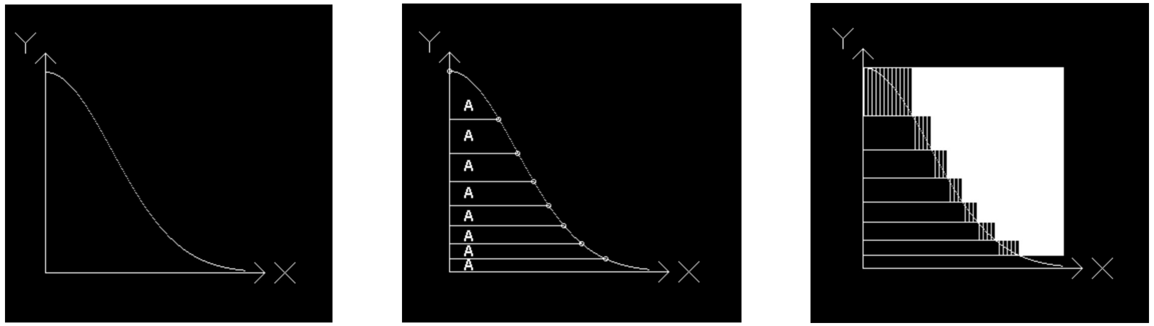
Why "Monte Carlo"?

- Stanislaw Ulam
  - Manhattan Project
  - Inspired by his uncle's gambling habits

Monte Carlo casino from "Goldeneye"

How to draw samples from a distribution?

- Suppose we have a generator function $g(\cdot)$ that gives us samples from $\text{Uniform}(0, 1)$
- How do we generate samples from Bernoulli(θ)?
  - Draw x from $g(\cdot)$. If $x > 1 - \theta \Rightarrow 1$, else 0.
- How do we generate samples from $N(\mu, \sigma^2)$?
  - Precompute k bins such that each bin has the same AUC.
  - Draw x from $g(\cdot)$. Map x to a bin.
  - Draw y from $g(\cdot)$. Scale y to the width of chosen bin and output y.



Monte Carlo Methods

- Direct sampling
- Rejection sampling
- Likelihood weighting
- Markov chain Monte Carlo (MCMC)

Rejection Sampling

- Instead of sampling from $P(X)$, sample x^* from $Q(X)$ and accept sample with probability:
  - $P_{accept}(x^*) = \frac{\mathrm{P}(\mathrm{x}^*)}{MQ(x^*)}$, where M is some constant such that $P(x) \leq MQ(x)\ \forall\ x$
- Works with un-normalized $P(X)$, too.

Unnormalized Importance Sampling

- Instead of hard **rejecting** samples, we can just **reweight** them:

$$E_P[f(X)] = \int_x P(x)f(x)dx = \int_x \frac{P(x)}{Q(x)}Q(x)f(x)dx = E_Q\left[\frac{P(x)}{Q(x)}f(x)\right]$$

- Approximate with empirical:

$$E_P[f(X)] \approx \frac{1}{n}\sum_{i=1,\dots,n} f(x_i)w(x_i)$$

where $x_i \sim Q$ and $w_i = \frac{P(x_i)}{Q(x_i)}$

What characteristic do we need for this to work?

Normalized Importance Sampling

- Instead of needing access to the normalized probability distribution P, we can also perform importance sampling with an un-normalized $\tilde{P} = aP$ by normalizing the weights according to the sample:
- $\widetilde{w_i} = \frac{w_i}{\sum_i w_i}$

Weighted resampling

- Problem of importance sampling:
  - Performance 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, ratio will be dominated by a few samples.
- Solution: use a heavy-tailed Q and weighted resampling.

Limitations of "simple" Monte Carlo

- Hard to get rare events in high-dimensional spaces
- We need a good proposal Q(x) that is not very different than P(x)

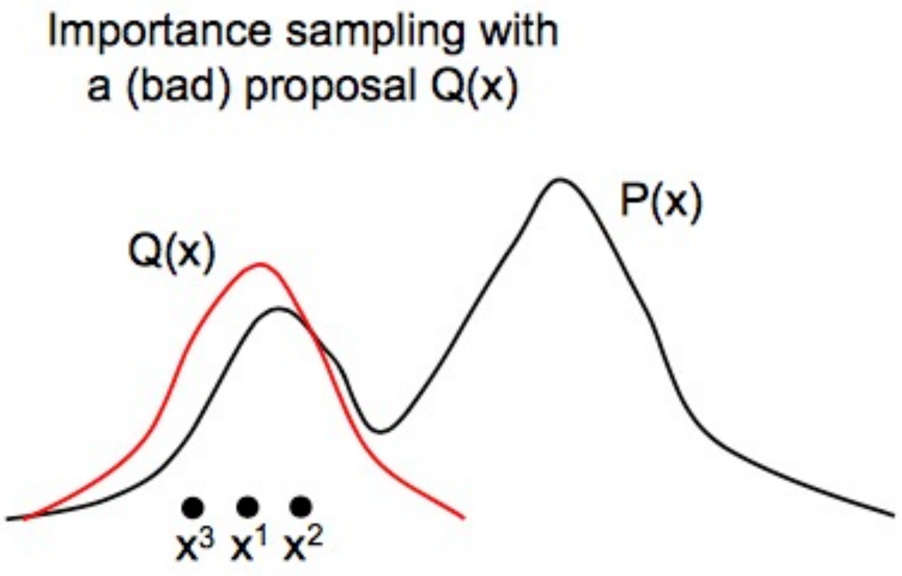
- What if we had an adaptive proposal Q(x)?

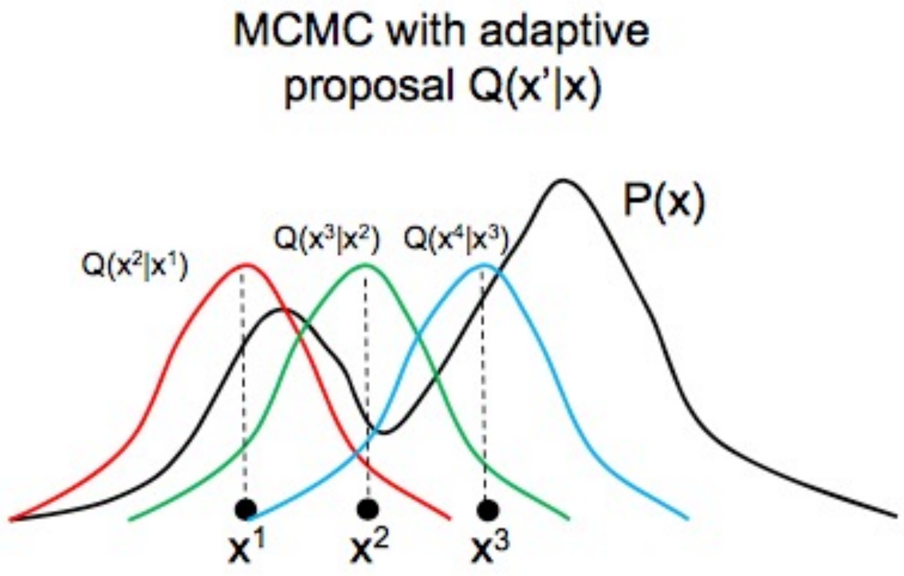
Markov Chain Monte Carlo (MCMC)

Markov Chain Monte Carlo

MCMC algorithms feature adaptive proposals

- Instead of Q(x') 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





Metropolis-Hastings

MCMC: Metropolis-Hastings

- Draw 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)
  - Acceptance prob: $A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$
- A(x'|x) is like a ration 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)
- A(x'|x) ensures that after sufficiently many draws, our samples come from the true distribution.

MCMC: Metropolis-Hastings

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 $u \sim \text{Uniform}(0,1)$ // draw acceptance threshold

   \- if $u < A(x^* \mid x) = \min\left(1, \frac{P(x^*)Q(x \mid x^*)}{P(x)Q(x^* \mid x)}\right)$
   - $x^{(t)} = x^*$ // transition

   \- else
   - $x^{(t)} = x$ // stay in current state

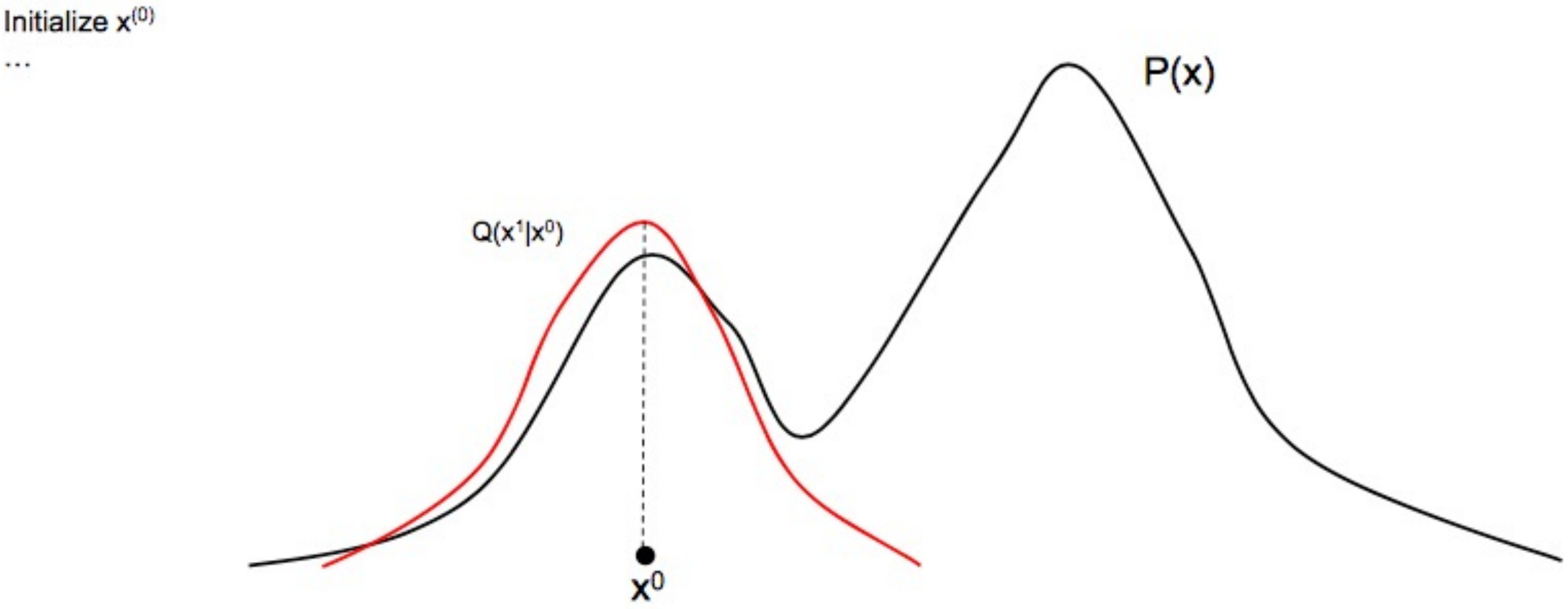
Function
Draw sample (x(t))

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

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

MCMC: Metropolis-Hastings example

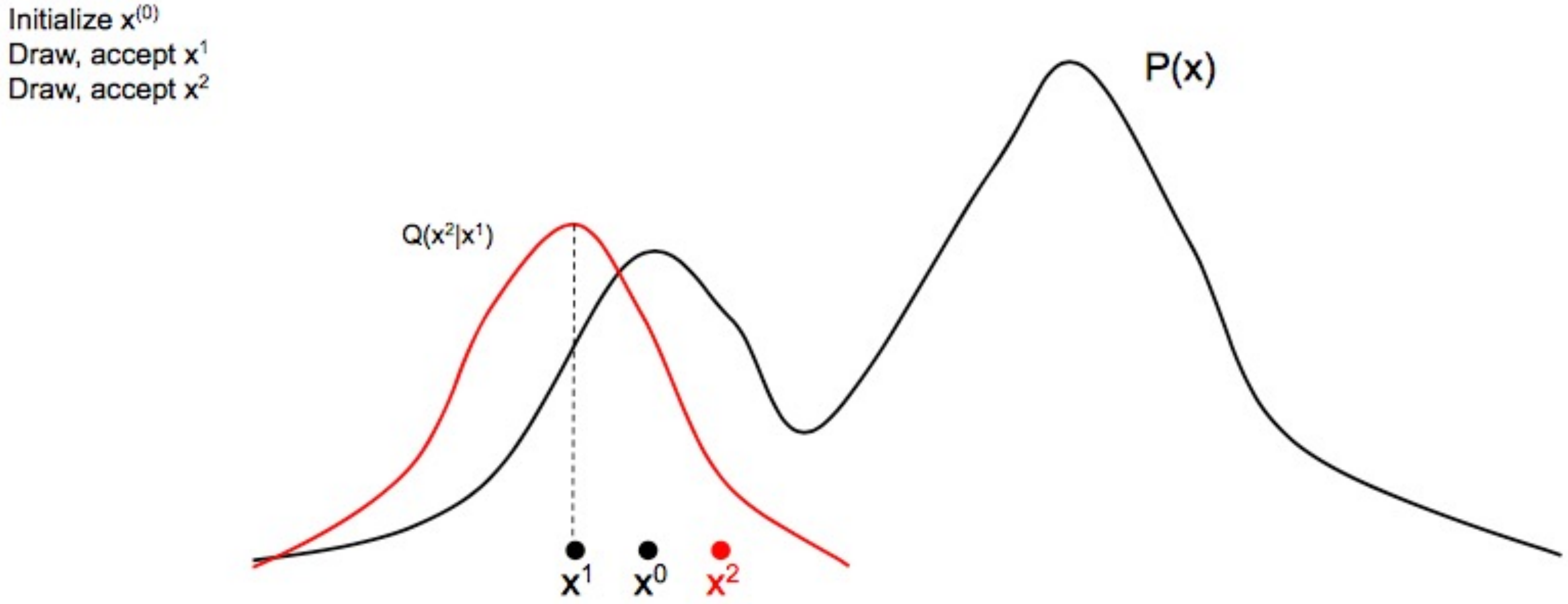
- We are trying to sample from a bimodal P(x)
- Let Q(x'|x) be a Gaussian centered on x



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

MCMC: Metropolis-Hastings example

- We are trying to sample from a bimodal P(x)
- Let Q(x′|x) be a Gaussian centered on x

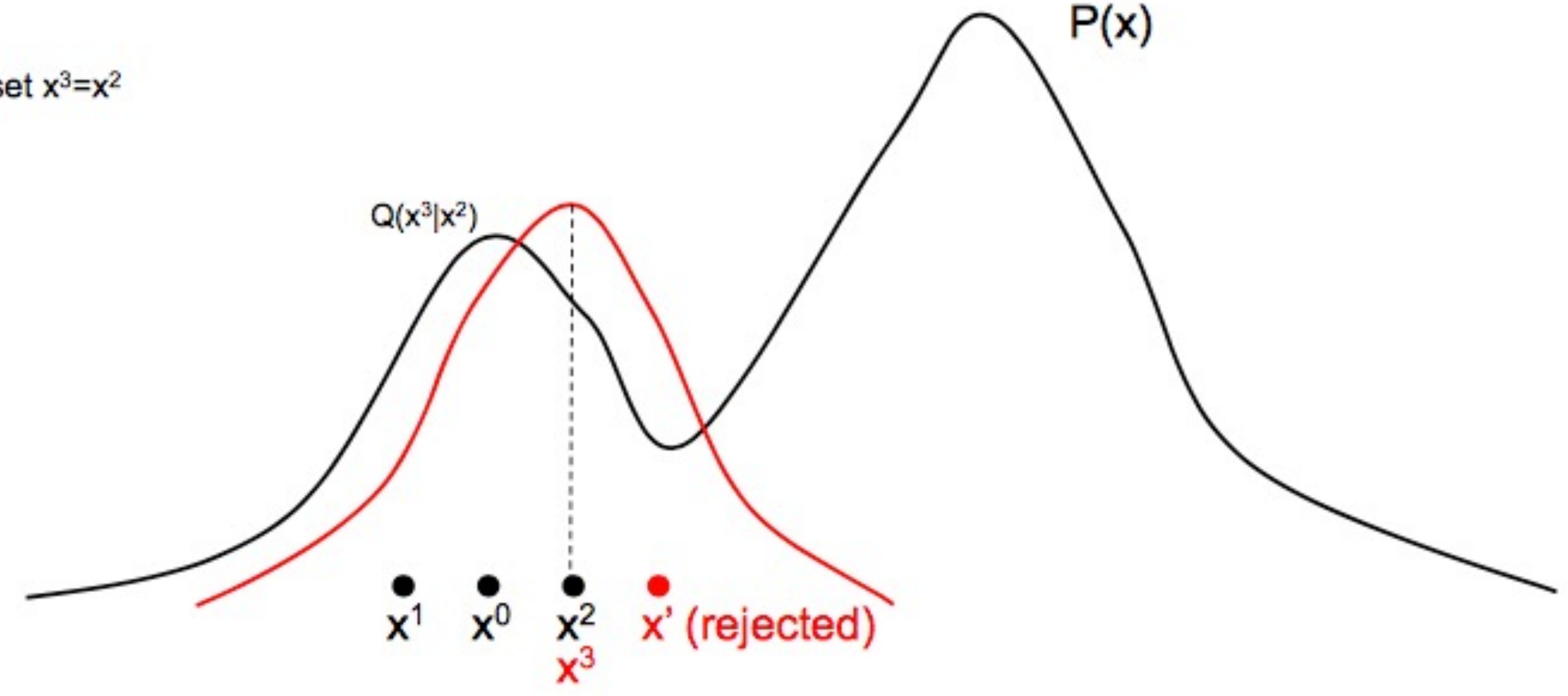


$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

MCMC: Metropolis-Hastings example

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

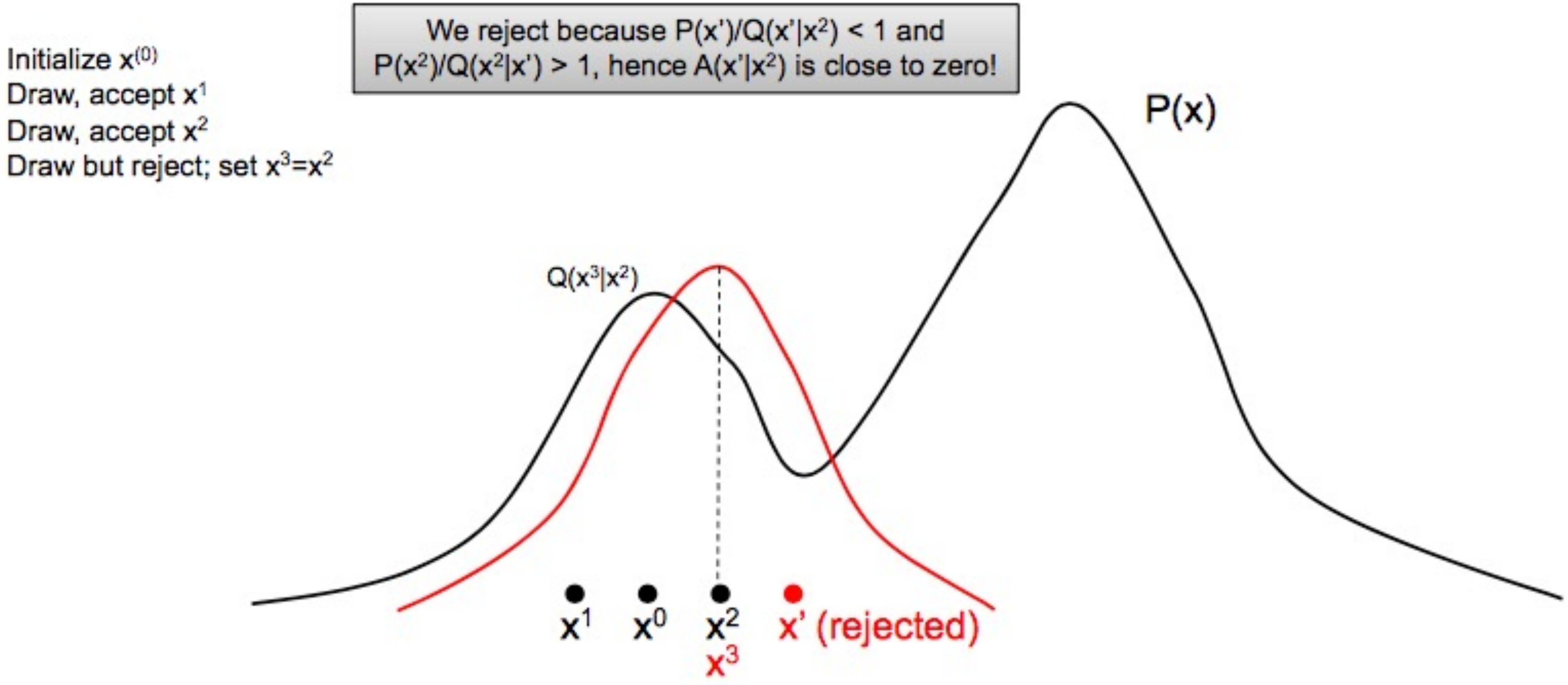
Initialize $x^{(0)}$
Draw, accept x^1
Draw, accept x^2
Draw but reject; set $x^3=x^2$



MCMC: Metropolis-Hastings example

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

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

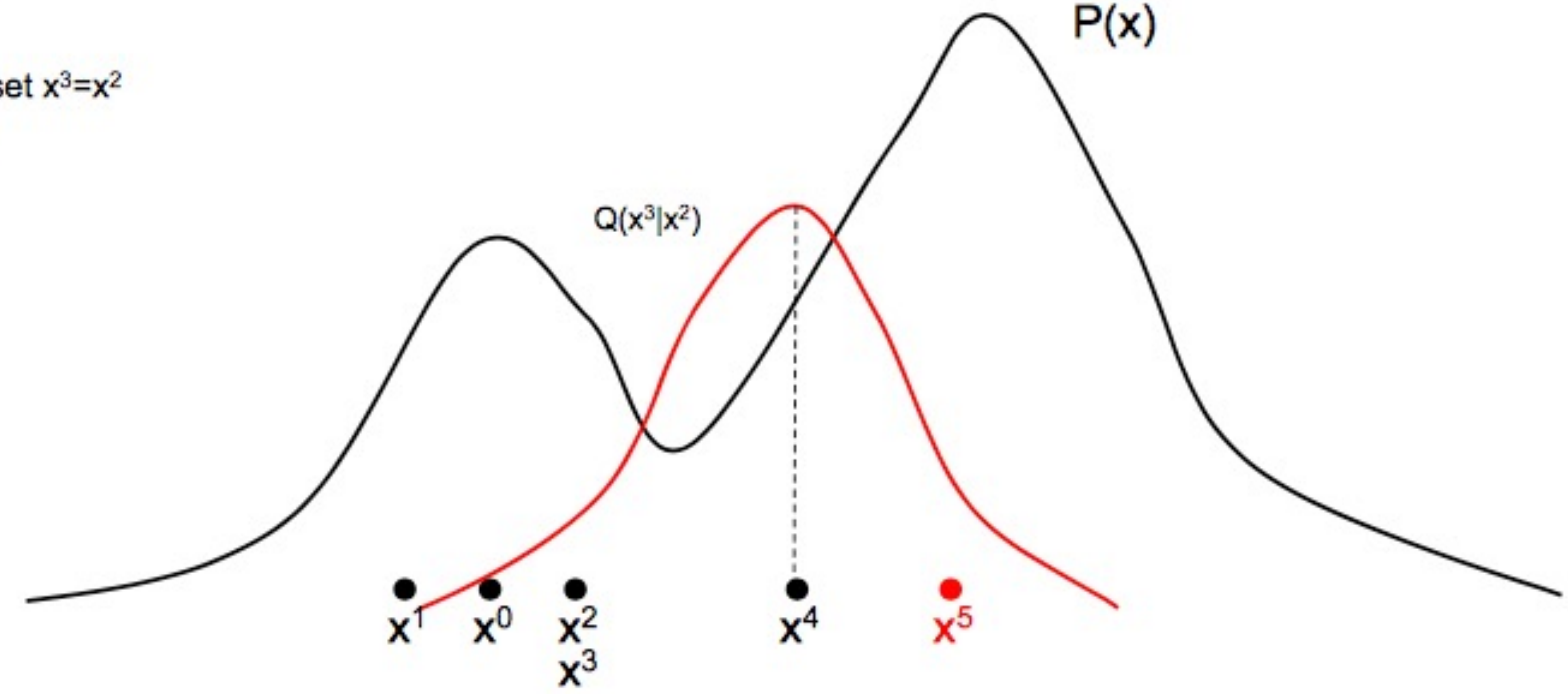


$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

MCMC: Metropolis-Hastings example

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

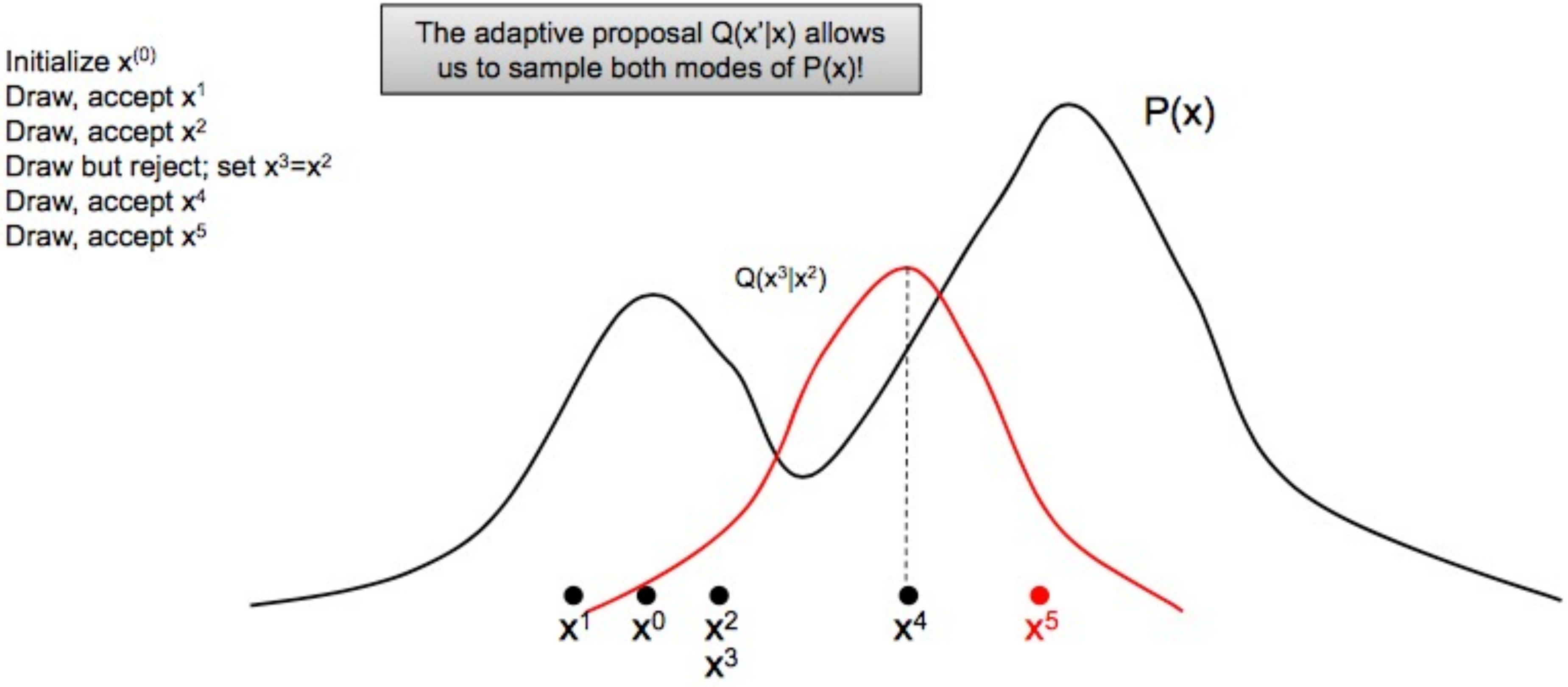
Initialize $x^{(0)}$
Draw, accept x^1
Draw, accept x^2
Draw but reject; set $x^3=x^2$
Draw, accept x^4
Draw, accept x^5



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

MCMC: Metropolis-Hastings example

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



MCMC: Some theory

- The MH algorithm has a burn-in period
  - Initial samples are not truly from P
- 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)?
- Why Markov Chain?

MCMC: Some theory

- Stationary distributions are of great importance in MCMC. Some notions
  - **Irreducible**: an MC is irreducible if you can get from any state x to any other state x' with probability x > 0 in a finite number of steps
  - **Aperiodic:** an MC is aperiodic if you can return to any state x at any time
  - **Ergodic (or regular):** an MC is ergodic if it is irreducible and aperiodic

- Ergodicity is important: it implies you can reach the stationary distribution no matter the initial distribution.

MCMC: Some theory

- Reversible (detailed balance): an MC is reversible if there exists a distribution π(x) such that the detailed balance condition holds

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

- Reversible MCs always have a stationary distribution

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

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

Why does Metropolis-Hastings work?

- We draw a sample x’ according to Q(x’|x) and then accept/reject according to A(x’|x). Hence the transition kernel is:

$$T(x'|x) = Q(x'|x)A(x'|x)$$

- We can prove that MH satisfies detailed balance.

  - Recall that

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

$$\frac{Q(x,x')}{Q(x')}$$

$$\frac{Q(x,x')}{Q(x)}$$

$$\frac{P(x')Q(x,x')Q(x)}{P(x)Q(x,x')Q(x')} \rightarrow \frac{P(x')Q(x)}{P(x)Q(x')}$$

Why does Metropolis-Hastings work?

- Since MH satisfies detailed balance:
  - The MH algorithm leads to a stationary distribution P(x)
  - We defined P(x) to be the true distribution of x
  - Thus, MH eventually converges to the true distribution

Gibbs Sampling

- Gibbs Sampling is an MCMC algorithm that samples each random variable of a graphical model, one at a time
- GS is fairly easy to derive for many graphical models
- GS has reasonable computation and memory requirements (because we sample one r.v. at a time)

Gibbs Sampling

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 $x_i \leftarrow x$

Questions?

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON