



10-418 / 10-618 Machine Learning for Structured Data

Machine Learning Department
School of Computer Science
Carnegie Mellon University



Markov Chains

Matt Gormley
Lecture 19
Oct. 30, 2019

Reminders

- **Homework 3: Structured SVM**
 - **Out: Fri, Oct. 24**
 - **Due: Wed, Nov. 6 at 11:59pm**
- **Project Team Office Hours**
 - **Fri, Nov. 1,**
 - **GHC 5222, 1:45 – 2:50pm**
 - **informally chat with classmates / course staff about project ideas**

METROPOLIS-HASTINGS

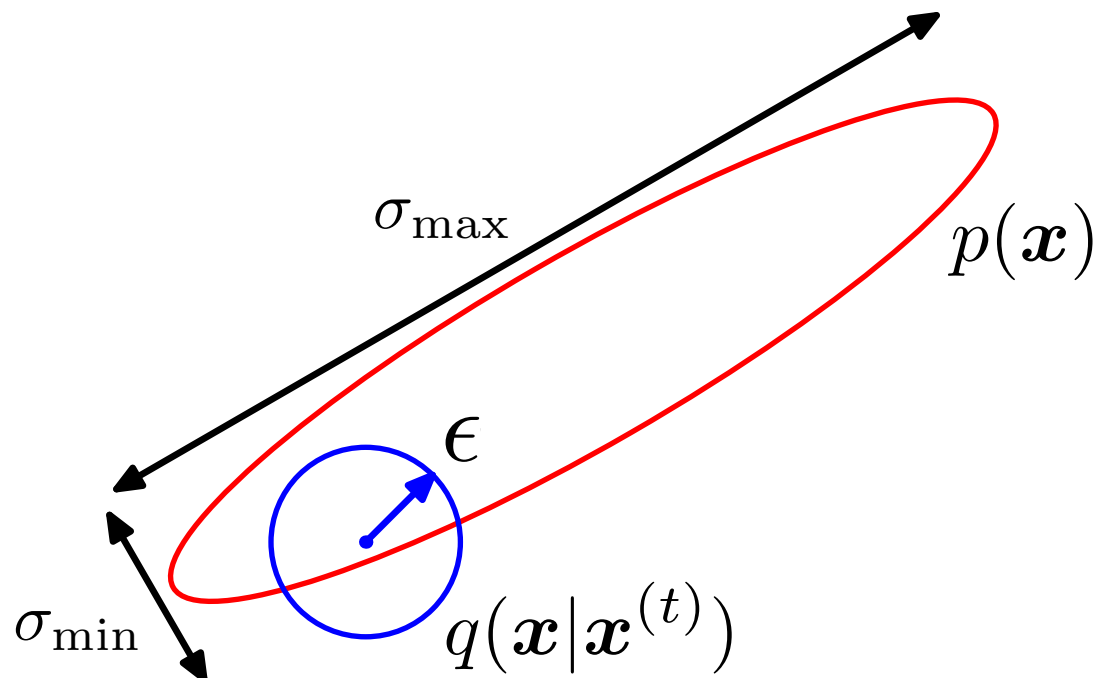
Metropolis-Hastings

Whiteboard

- Metropolis Algorithm
- Metropolis-Hastings Algorithm

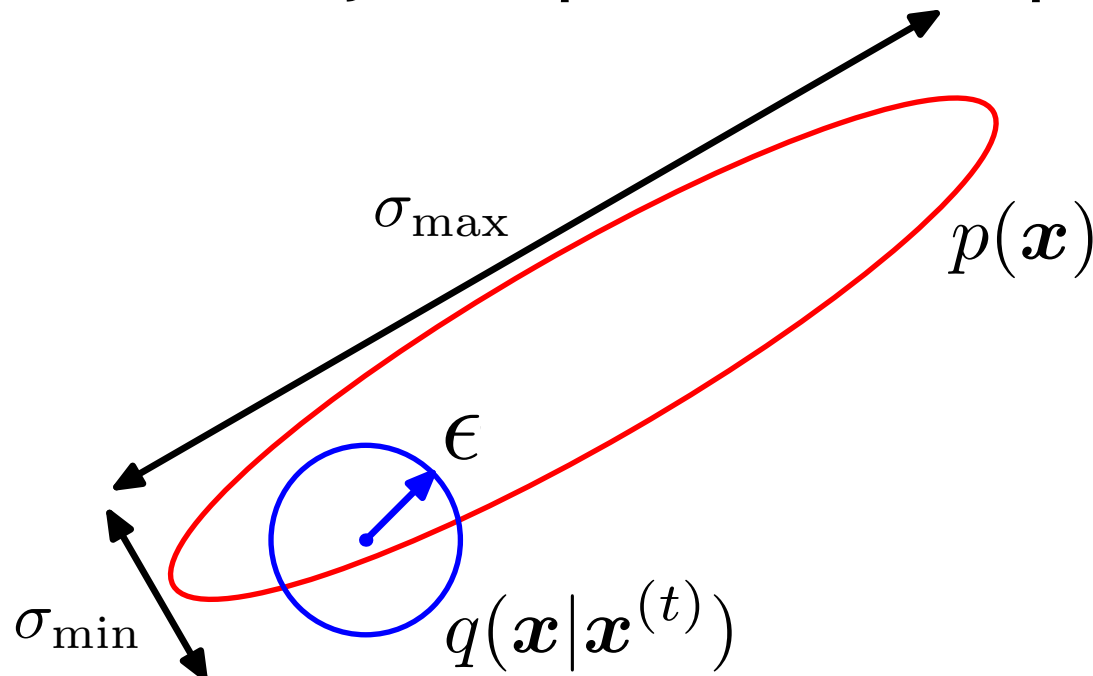
Random Walk Behavior of M-H

- For **Metropolis-Hastings**, a generic proposal distribution is: $q(x|x^{(t)}) = \mathcal{N}(0, \epsilon^2)$
- If ϵ is large, many rejections
- If ϵ is small, slow mixing



Random Walk Behavior of M-H

- For **Rejection Sampling**, the accepted samples are **independent**
- But for **Metropolis-Hastings**, the samples are **correlated**
- **Question:** How long must we wait to get effectively independent samples?



A: independent states in the M-H random walk are separated by roughly $(\sigma_{\max}/\sigma_{\min})^2$ steps

Whiteboard

- Gibbs Sampling as M-H
- Blocked Gibbs Sampling

Definitions and Theoretical Justification for MCMC

MARKOV CHAINS

Whiteboard

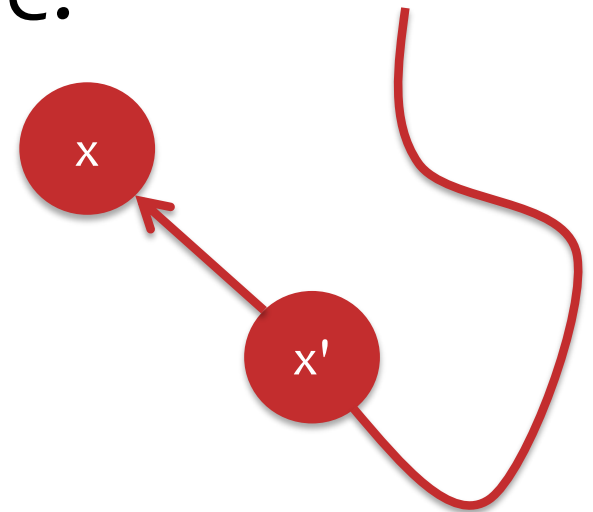
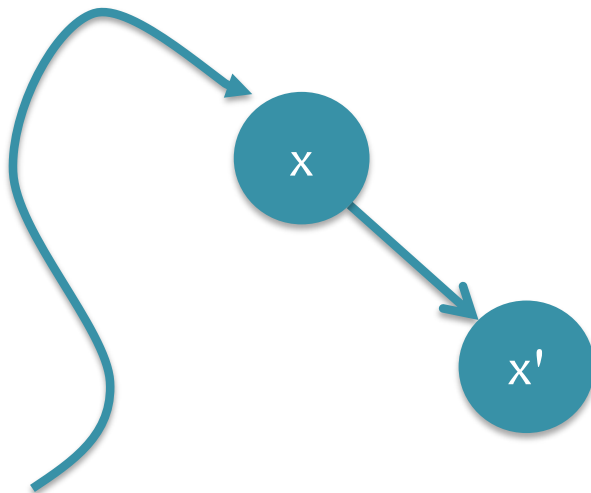
- Markov chains
- Transition probabilities
- Invariant distribution
- Equilibrium distribution
- Sufficient conditions for MCMC
- Markov chain as a WFSM

Detailed Balance

$$S(x' \leftarrow x)p(x) = S(x \leftarrow x')p(x')$$

Detailed balance means that, for each pair of states x and x' ,

arriving at x then x' and arriving at x' then x are equiprobable.



Whiteboard

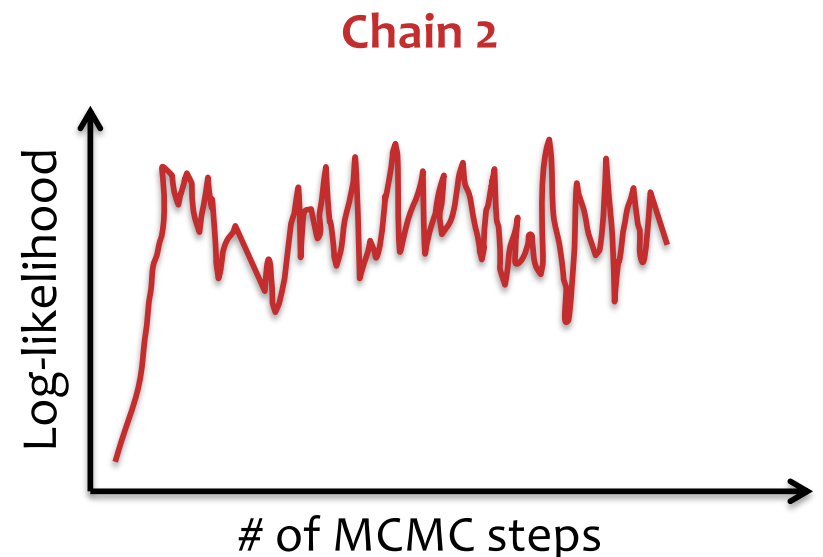
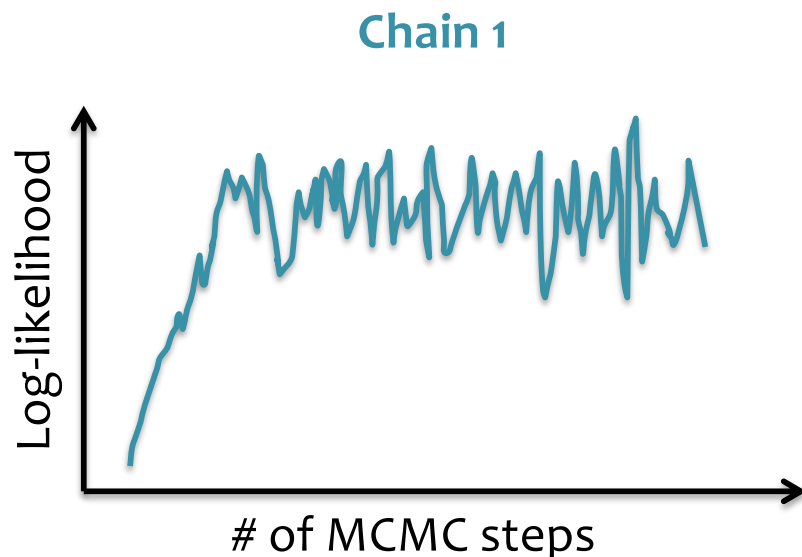
- Simple Markov chain example
- Constructing Markov chains
- Transition Probabilities for MCMC

Practical Issues

- **Question:** Is it better to move along one dimension or many?
- **Answer:** For **Metropolis-Hastings**, it is sometimes better to sample one dimension at a time
 - Q: Given a sequence of 1D proposals, compare rate of movement for **one-at-a-time** vs. **concatenation**.
- **Answer:** For **Gibbs Sampling**, sometimes better to sample a block of variables at a time
 - Q: When is it tractable to sample a block of variables?

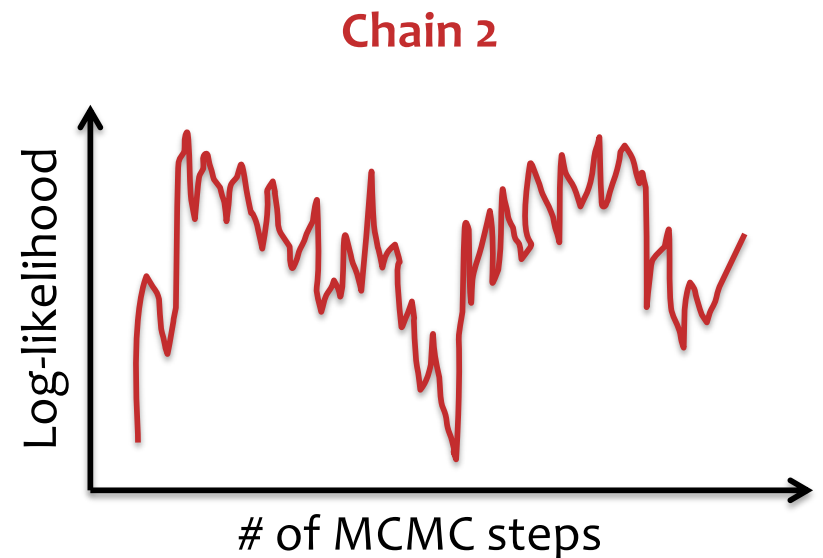
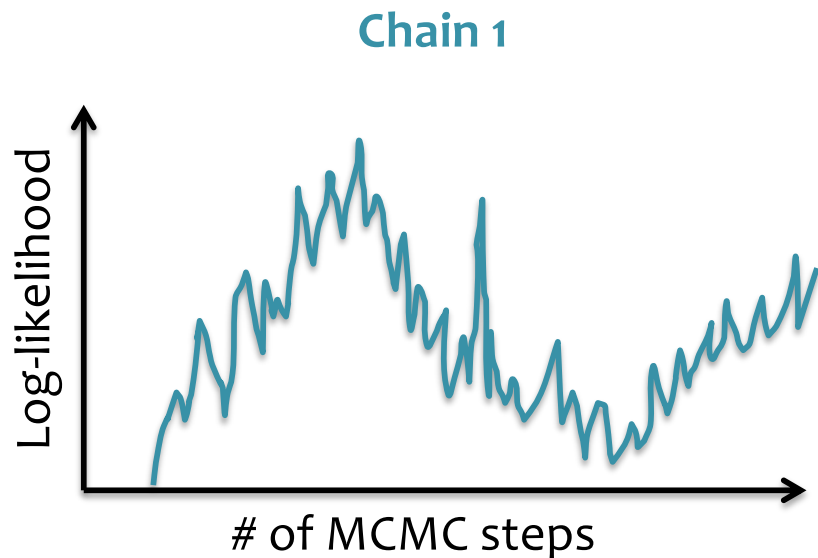
Practical Issues

- **Question:** How do we assess convergence of the Markov chain?
- **Answer:** It's not easy!
 - Compare statistics of multiple independent chains
 - Ex: Compare log-likelihoods



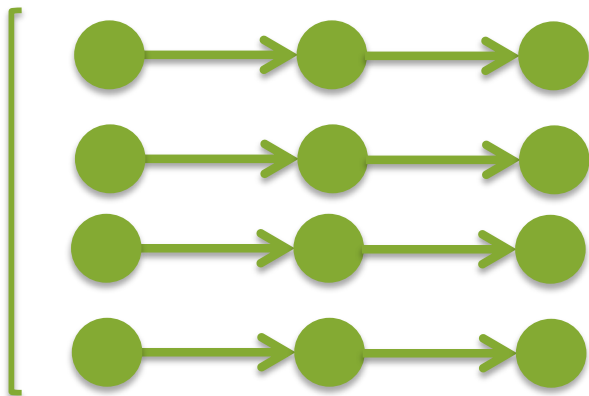
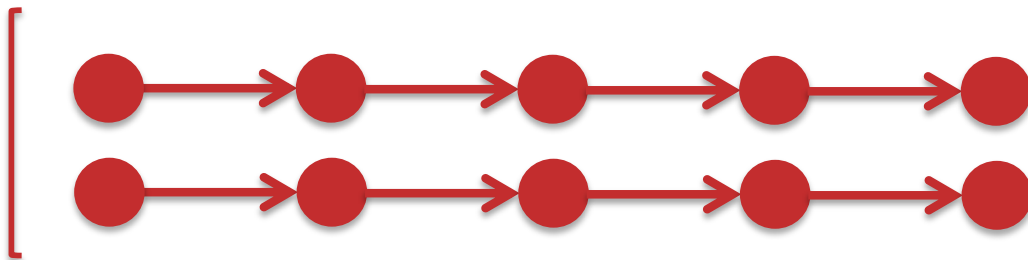
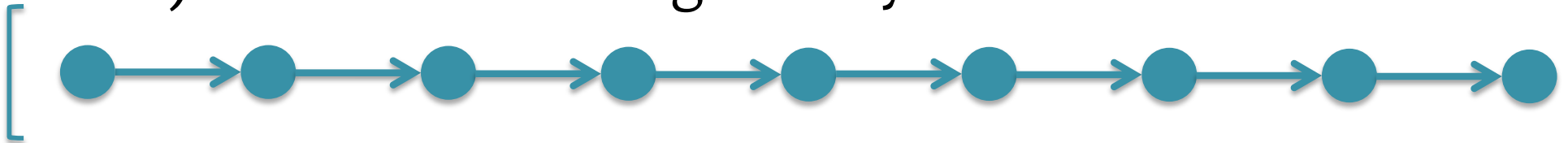
Practical Issues

- **Question:** How do we assess convergence of the Markov chain?
- **Answer:** It's not easy!
 - Compare statistics of multiple independent chains
 - Ex: Compare log-likelihoods



Practical Issues

- **Question:** Is one long Markov chain better than many short ones?
- **Note:** typical to discard initial samples (aka. “burn-in”) since the chain might not yet have mixed



- **Answer:** Often a balance is best:
 - Compared to one long chain: More independent samples
 - Compared to many small chains: Less samples discarded for burn-in
 - We can still parallelize
 - Allows us to assess mixing by comparing chains

MCMC Summary

- **Pros**
 - Very general purpose
 - Often easy to implement
 - Good theoretical guarantees as $t \rightarrow \infty$
- **Cons**
 - Lots of tunable parameters / design choices
 - Can be quite slow to converge
 - Difficult to tell whether it's working

Extra Slides

The remaining slides on auxiliary variable MCMC methods are extra slides that were not covered in lecture. They are left here in case you're curious to see two more examples of MCMC methods.

Slice Sampling, Hamiltonian Monte Carlo

MCMC (AUXILIARY VARIABLE METHODS)

Auxiliary variables

The point of MCMC is to marginalize out variables, but one can introduce more variables:

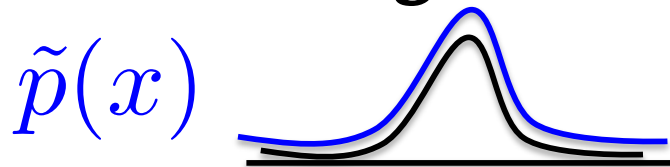
$$\int f(x)P(x) dx = \int f(x)P(x, v) dx dv$$
$$\approx \frac{1}{S} \sum_{s=1}^S f(x^{(s)}), \quad x, v \sim P(x, v)$$

We might want to do this if

- $P(x|v)$ and $P(v|x)$ are simple
- $P(x, v)$ is otherwise easier to navigate

Slice Sampling

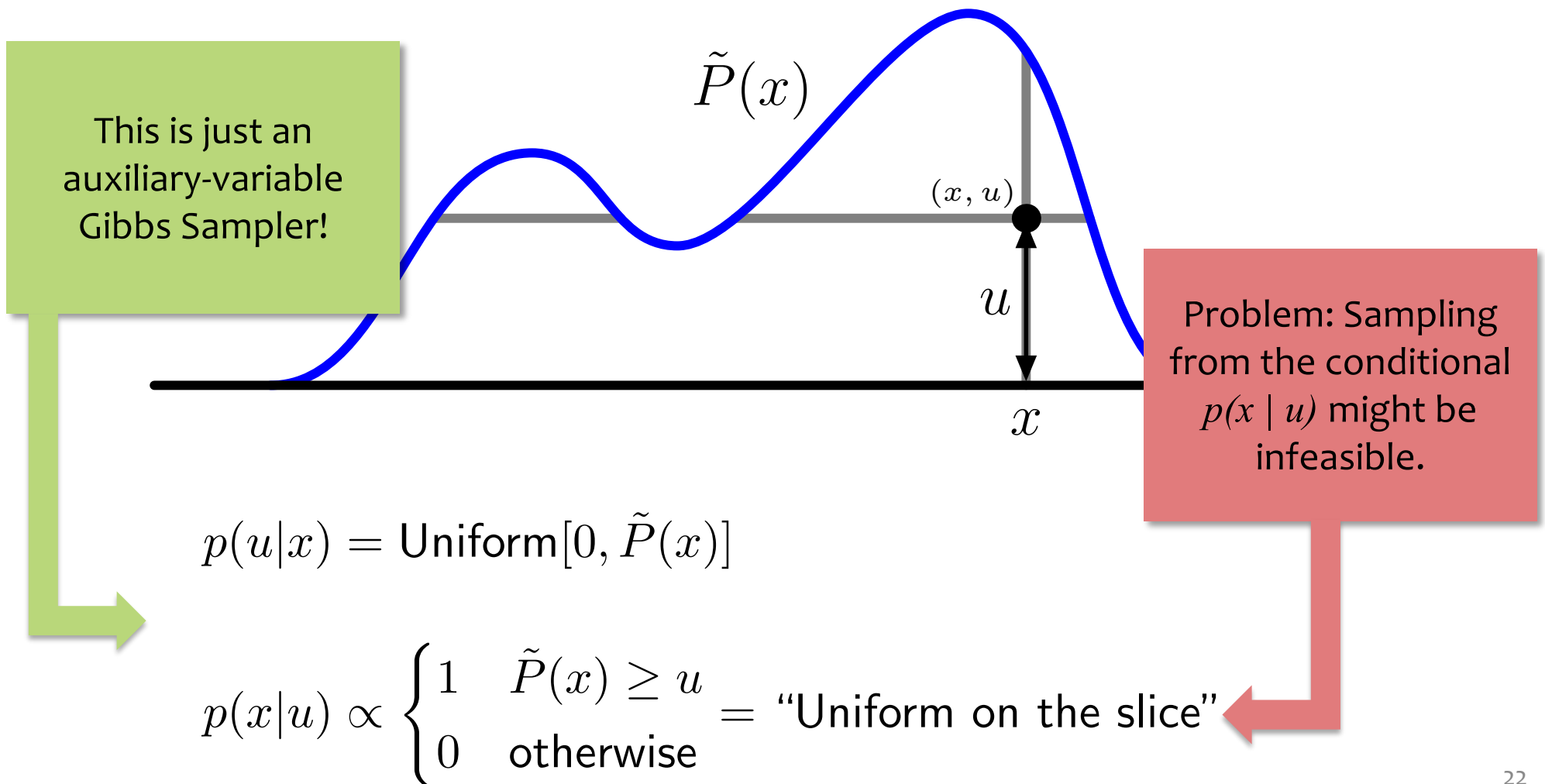
- Motivation:
 - Want **samples** from $p(x)$ and don't know the normalizer Z
 - Choosing a proposal at the correct **scale** is difficult
- Properties:
 - Similar to *Gibbs Sampling*: **one-dimensional** transitions in the state space
 - Similar to *Rejection Sampling*: (asymptotically) draws samples from the **region under the curve**



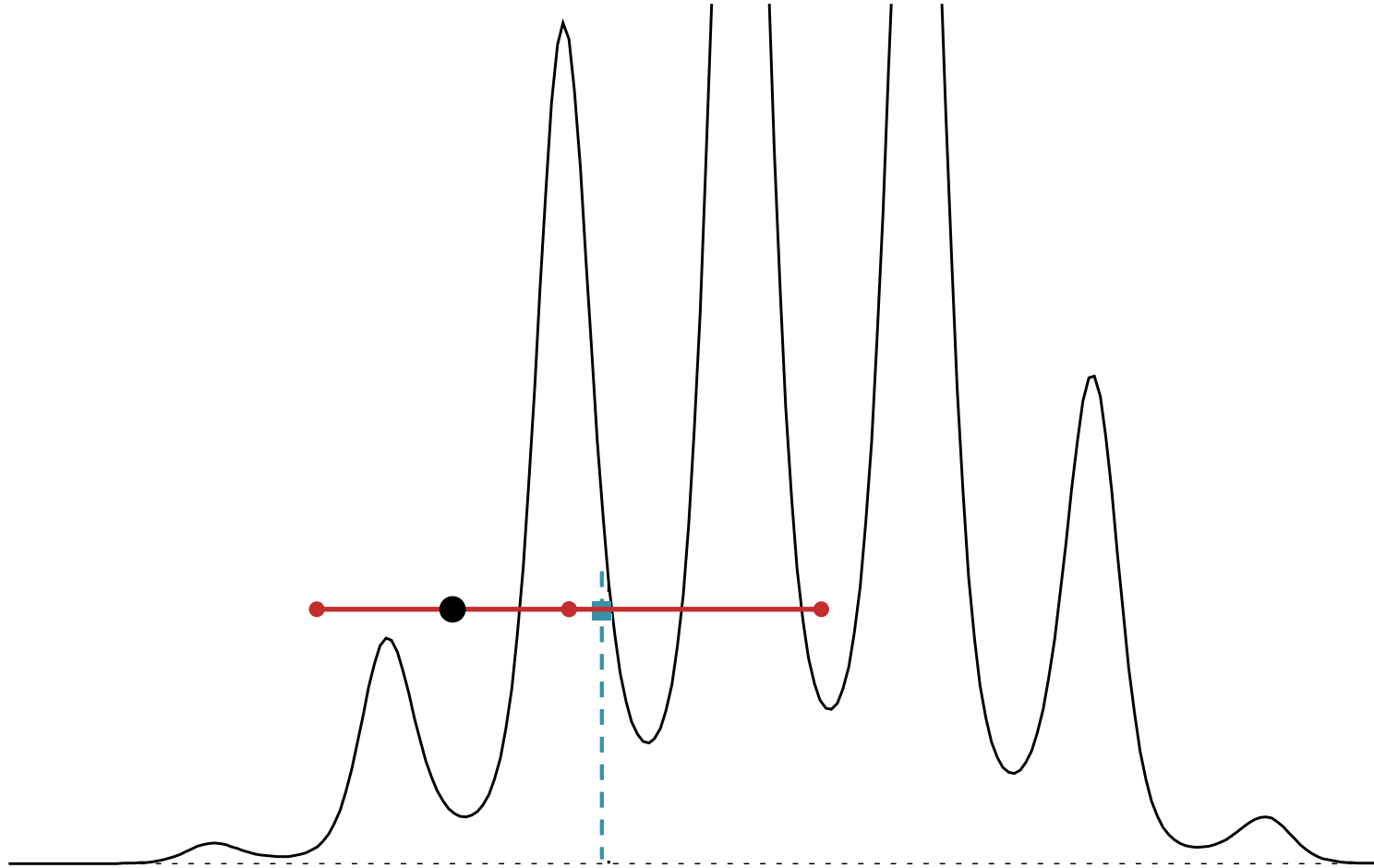
- An MCMC method with an **adaptive proposal**

Slice sampling idea

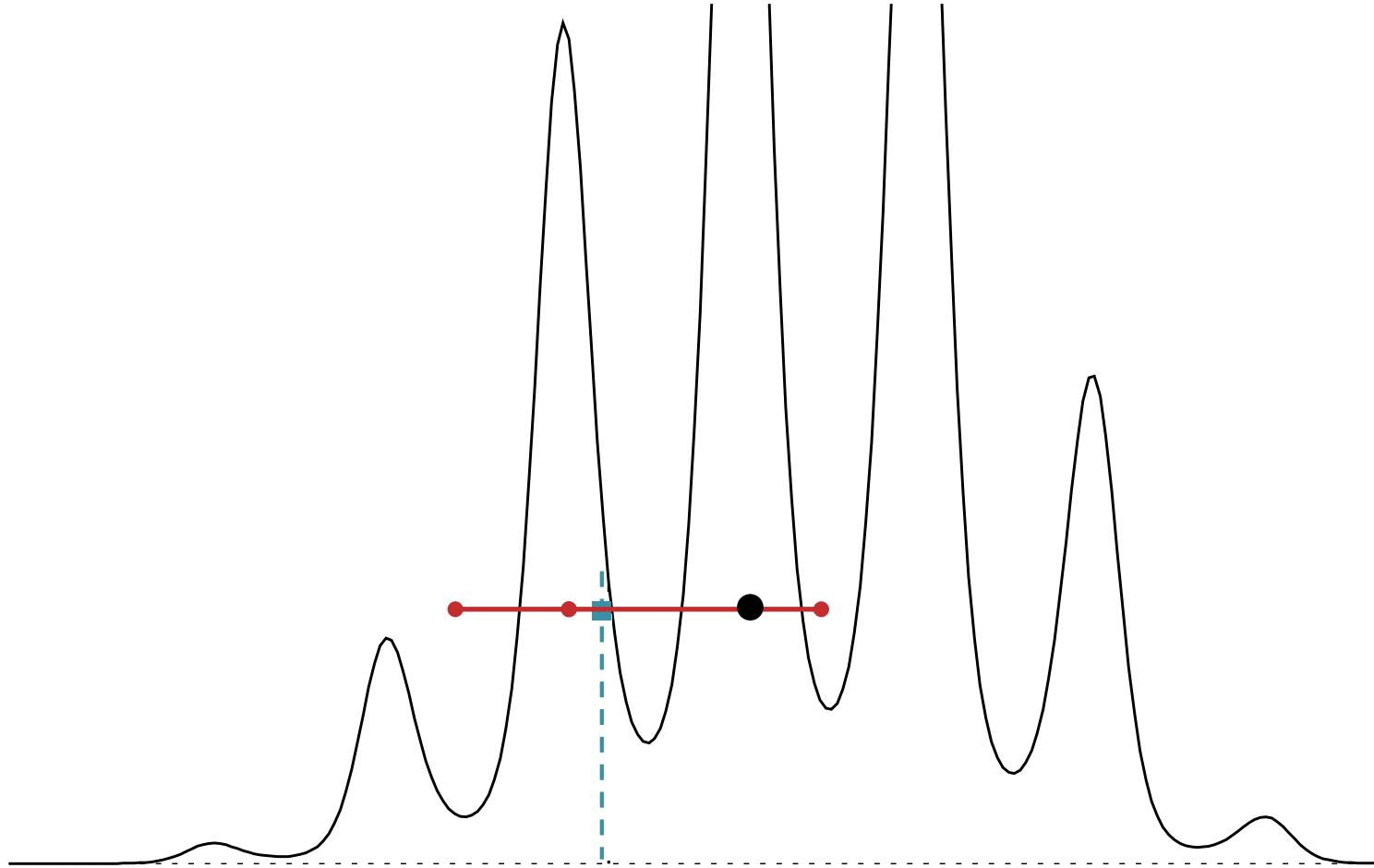
Sample point uniformly under curve $\tilde{P}(x) \propto P(x)$



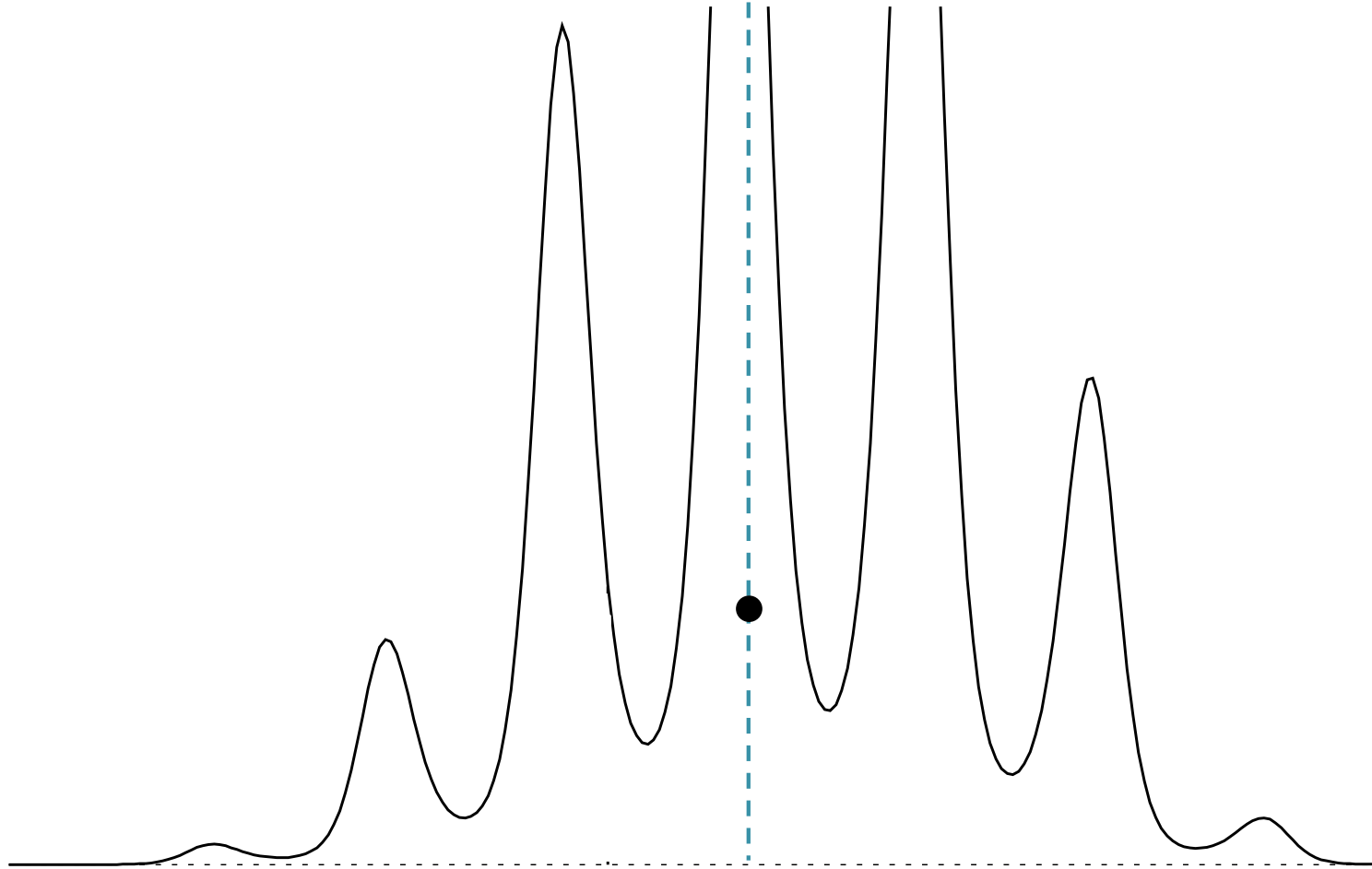
Slice Sampling



Slice Sampling



Slice Sampling



Slice Sampling

Goal: sample (x, u) given $(u^{(t)}, x^{(t)})$.

Part 1: Stepping Out

Sample interval (x_l, x_r) enclosing $x^{(t)}$.

Expand until endpoints are "outside" region under curve.

Part 2: Sample x (Shrinking)

Draw x from within the interval (x_l, x_r) , then accept or shrink.

Algorithm:

Slice Sampling

Goal: sample (x, u) given $(u^{(t)}, x^{(t)})$.

$u \sim \text{Uniform}(0, p(x^{(t)}))$

Part 1: Stepping Out

Sample interval (x_l, x_r) enclosing $x^{(t)}$.

$r \sim \text{Uniform}(u, w)$

$(x_l, x_r) = (x^{(t)} - r, x^{(t)} + w - r)$

Expand until endpoints are "outside" region under curve.

while($\tilde{p}(x_l) > u$) $\{x_l = x_l - w\}$

while($\tilde{p}(x_r) > u$) $\{x_r = x_r + w\}$

Part 2: Sample x (Shrinking)

Draw x from within the interval (x_l, x_r) , then accept or shrink.

Algorithm:

Slice Sampling

Goal: sample (x, u) given $(u^{(t)}, x^{(t)})$.

$u \sim \text{Uniform}(0, p(x^{(t)}))$

Part 1: Stepping Out

Sample interval (x_l, x_r) enclosing $x^{(t)}$.

$r \sim \text{Uniform}(u, w)$

$(x_l, x_r) = (x^{(t)} - r, x^{(t)} + w - r)$

Expand until endpoints are "outside" region under curve.

while($\tilde{p}(x_l) > u$) { $x_l = x_l - w$ }

while($\tilde{p}(x_r) > u$) { $x_r = x_r + w$ }

Part 2: Sample x (Shrinking)

while(true) {

Draw x from within the interval (x_l, x_r) , then accept or shrink.

$x \sim \text{Uniform}(x_l, x_r)$

if($\tilde{p}(x) > u$) { break }

else if($x > x^{(t)}$) { $x_r = x$ }

else { $x_l = x$ }

}

$x^{(t+1)} = x, u^{(t+1)} = u$

Algorithm:

Slice Sampling

Multivariate Distributions

– Resample each variable x_i **one-at-a-time** (just like Gibbs Sampling)

– Does not require sampling from

$$p(x_i | \{x_j\}_{j \neq i})$$

– Only need to evaluate a quantity **proportional** to the conditional

$$p(x_i | \{x_j\}_{j \neq i}) \propto \tilde{p}(x_i | \{x_j\}_{j \neq i})$$

Hamiltonian Monte Carlo

- Suppose we have a distribution of the form:

$$p(\mathbf{x}) = \exp\{-E(\mathbf{x})\} / Z$$

where $\mathbf{x} \in \mathcal{R}^N$

- We could use **random-walk M-H** to draw samples, but it seems a shame to **discard gradient information** $\nabla_{\mathbf{x}} E(\mathbf{x})$
- If we can evaluate it, the gradient tells us where to look for **high-probability regions!**

Background: Hamiltonian Dynamics


Applications:

- Following the motion of atoms in a fluid through time
- Integrating the motion of a solar system over time
- Considering the evolution of a galaxy (i.e. the motion of its stars)
- “molecular dynamics”
- “N-body simulations”

Properties:

- Total energy of the system $H(x,p)$ stays constant
- Dynamics are reversible

Important for
detailed balance



Background: Hamiltonian Dynamics

Let $\mathbf{x} \in \mathcal{R}^N$ be a position

$\mathbf{p} \in \mathcal{R}^N$ be a momentum

Potential energy: $E(\mathbf{x})$

Kinetic energy: $K(\mathbf{p}) = \mathbf{p}^T \mathbf{p} / 2$

Total energy: $H(\mathbf{x}, \mathbf{p}) = E(\mathbf{x}) + K(\mathbf{p})$



Hamiltonian function

Given a starting position $x^{(l)}$ and a starting momentum $p^{(l)}$ we can simulate the Hamiltonian dynamics of the system via:

1. Euler's method
2. Leapfrog method
3. etc.

Background: Hamiltonian Dynamics

Parameters to tune:

1. Step size, ϵ
2. Number of iterations, L

Leapfrog Algorithm:

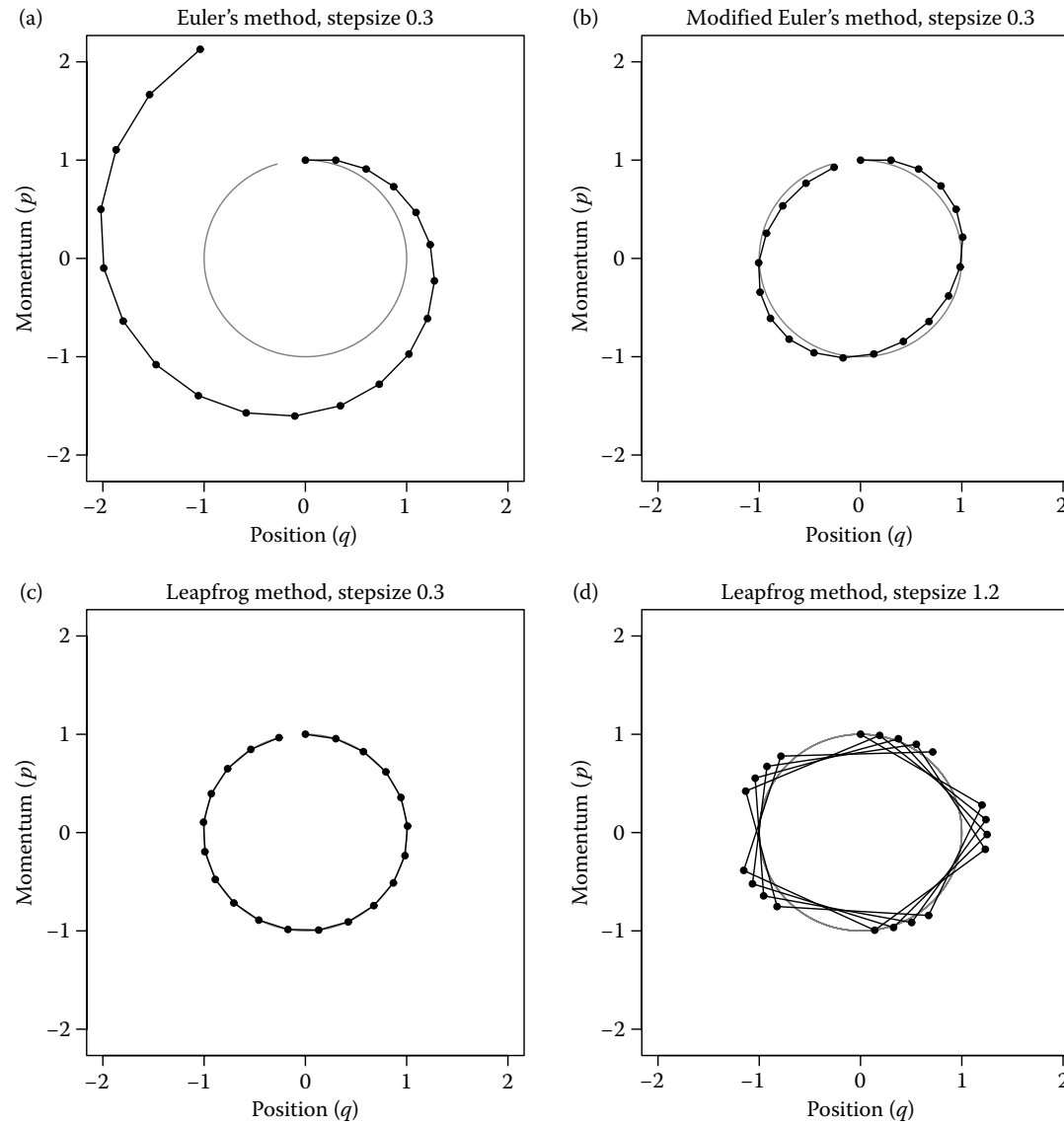
for τ in $1 \dots L$:

$$\mathbf{p} = \mathbf{p} - \frac{\epsilon}{2} \nabla_{\mathbf{x}} E(\mathbf{x})$$

$$\mathbf{x} = \mathbf{x} + \epsilon \mathbf{p}$$

$$\mathbf{p} = \mathbf{p} - \frac{\epsilon}{2} \nabla_{\mathbf{x}} E(\mathbf{x})$$

Background: Hamiltonian Dynamics



Hamiltonian Monte Carlo

Preliminaries

Goal: $p(\mathbf{x}) = \exp\{-E(\mathbf{x})\}/Z$ where $\mathbf{x} \in \mathcal{R}^N$

Define: $K(\mathbf{p}) = \mathbf{p}^T \mathbf{p}/2$
 $H(\mathbf{x}, \mathbf{p}) = E(\mathbf{x}) + K(\mathbf{p})$

$$p(\mathbf{x}, \mathbf{p}) = \exp\{-H(\mathbf{x}, \mathbf{p})\}/Z_H$$

$$= \exp\{-E(\mathbf{x})\} \exp\{-K(\mathbf{p})\}/Z_H$$

Note:

Since $p(\mathbf{x}, \mathbf{p})$ is separable...

$$\Rightarrow \sum_{\mathbf{p}} p(\mathbf{x}, \mathbf{p}) = \exp\{-E(\mathbf{x})\}/Z$$

Target dist.

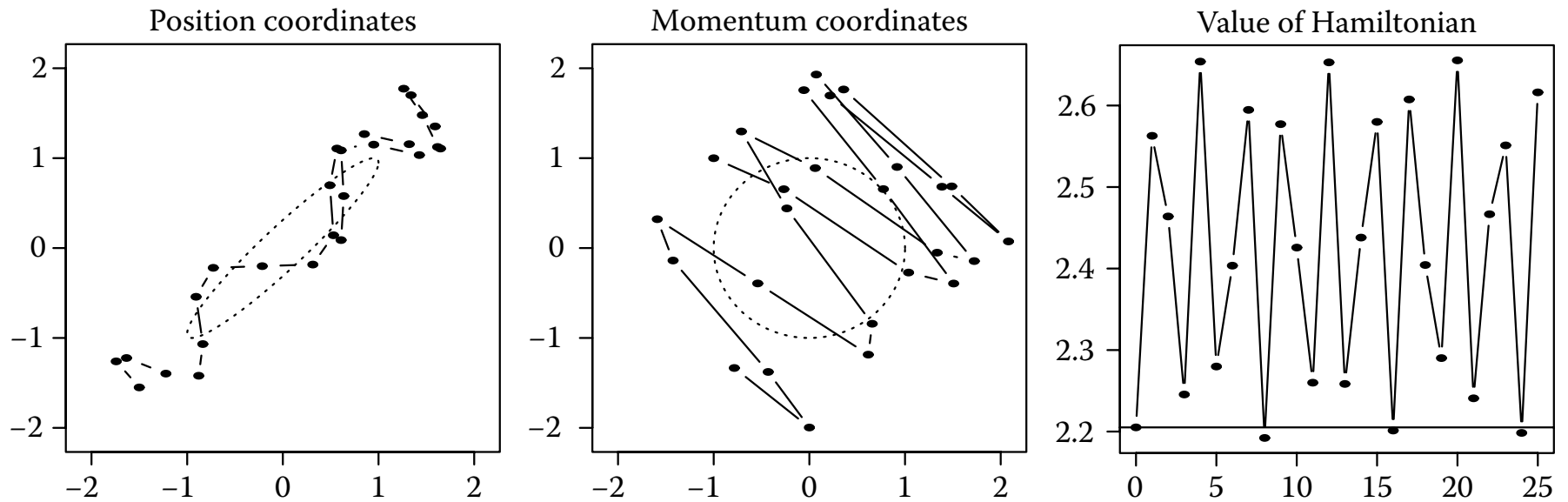
$$\Rightarrow \sum_{\mathbf{x}} p(\mathbf{x}, \mathbf{p}) = \exp\{-K(\mathbf{p})\}/Z_K$$

Gaussian

Whiteboard

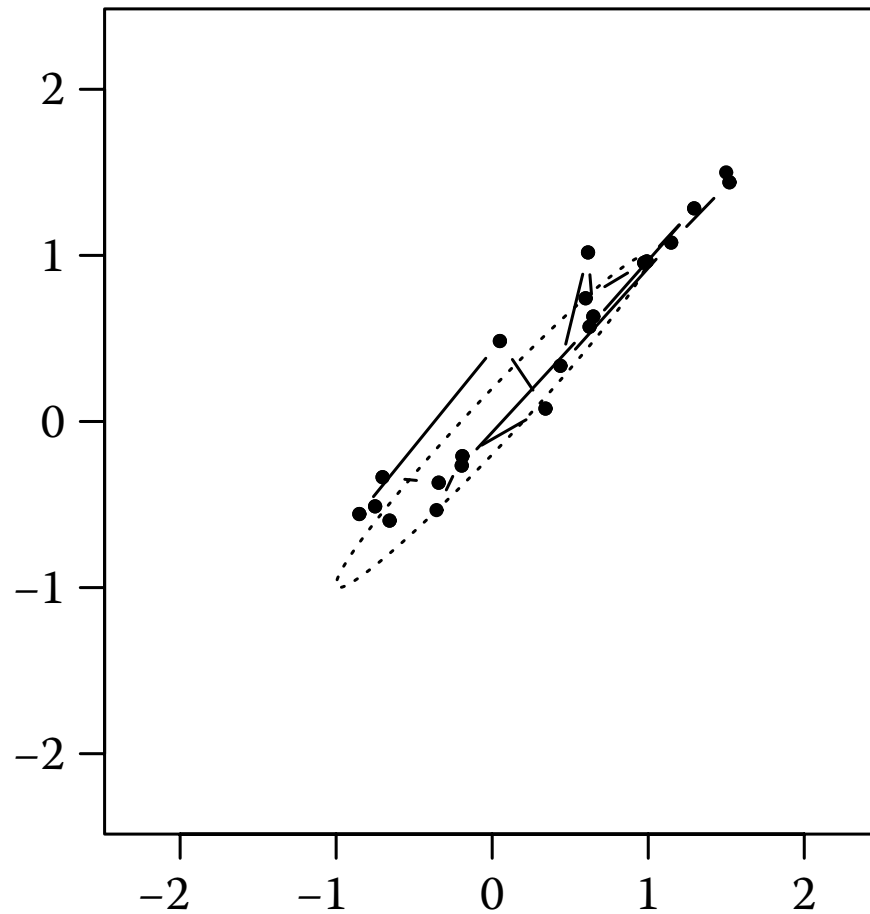
- Hamiltonian Monte Carlo algorithm
(aka. Hybrid Monte Carlo)

Hamiltonian Monte Carlo



M-H vs. HMC

Random-walk Metropolis



Hamiltonian Monte Carlo

