School of Computer Science
Carnegie Mellon

10-708 Probabilistic Graphical Models

Markov Chain Monte Carlo (MCMC)

Readings:
MacKay Ch. 29
Jordan Ch. 21

Matt Gormley
Lecture 16
March 14, 2016

Housekeeping

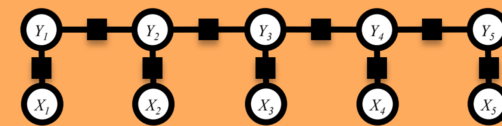
- **Homework 2**
  - Due March 16, 12:00 noon (extended)
- **Midway Project Report**
  - Due March 23, 12:00 noon

1. Data

$$\mathcal{D} = \{\boldsymbol{x}^{(n)}\}_{n=1}^{N}$$

2. Model

$$p(\boldsymbol{x} \mid \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \prod_{C \in \mathcal{C}} \psi_C(\boldsymbol{x}_C)$$



3. Objective

$$\ell(\theta; \mathcal{D}) = \sum_{n=1}^{N} \log p(\boldsymbol{x}^{(n)} \mid \boldsymbol{\theta})$$

5. Inference

1. Marginal Inference

$$p(\boldsymbol{x}_C) = \sum_{\boldsymbol{x}' : \boldsymbol{x}'_C = \boldsymbol{x}_C} p(\boldsymbol{x}' \mid \boldsymbol{\theta})$$

2. Partition Function

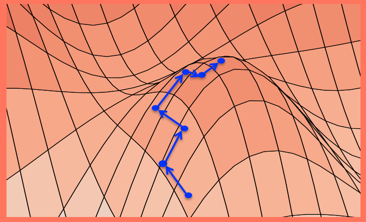
$$Z(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} \prod_{C \in \mathcal{C}} \psi_C(\boldsymbol{x}_C)$$

3. MAP Inference

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\; p(\boldsymbol{x} \mid \boldsymbol{\theta})$$

4. Learning

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \ell(\boldsymbol{\theta}; \mathcal{D})$$



It’s Pi Day 2016…

…so let’s compute π.

Properties of Monte Carlo

Estimator: $\int f(x)P(x)\,\mathrm{d}x \;\approx\; \hat{f} \;\equiv\; \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)}), \quad x^{(s)} \sim P(x)$

Estimator is unbiased:

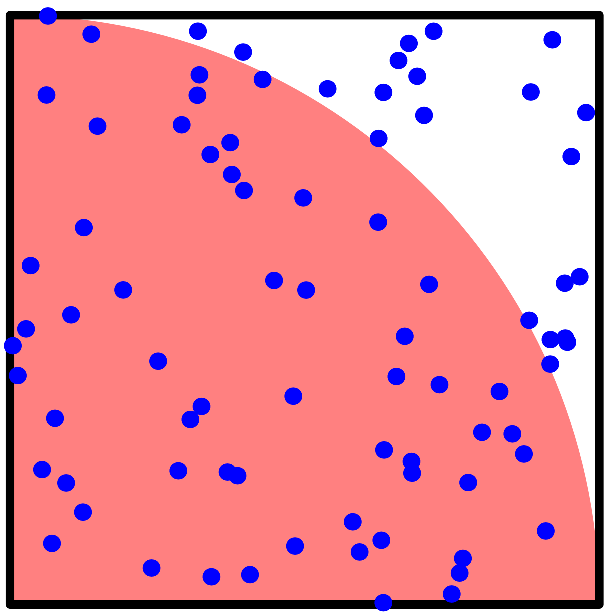
$$\mathbb{E}_{P(\{x^{(s)}\})}\left[\hat{f}\right] \;=\; \frac{1}{S}\sum_{s=1}^{S}\mathbb{E}_{P(x)}[f(x)] \;=\; \mathbb{E}_{P(x)}[f(x)]$$

Variance shrinks $\propto 1/S$:

$$\mathrm{var}_{P(\{x^{(s)}\})}\left[\hat{f}\right] \;=\; \frac{1}{S^2}\sum_{s=1}^{S}\mathrm{var}_{P(x)}[f(x)] \;=\; \mathrm{var}_{P(x)}[f(x)]\,/S$$

"Error bars" shrink like $\sqrt{S}$

A dumb approximation of π



$$P(x,y) = \begin{cases} 1 & 0<x<1 \text{ and } 0<y<1 \\ 0 & \text{otherwise} \end{cases}$$

$$\pi = 4 \iint \mathbb{I}\left((x^2+y^2)<1\right) P(x,y)\,\mathrm{d}x\,\mathrm{d}y$$

```
octave:1> S=12; a=rand(S,2); 4*mean(sum(a.*a,2)<1)
ans = 3.3333
octave:2> S=1e7; a=rand(S,2); 4*mean(sum(a.*a,2)<1)
ans = 3.1418
```

Aside: don't always sample!

"Monte Carlo is an extremely bad method; it should be used only when all alternative methods are worse."

— Alan Sokal, 1996

Example: numerical solutions to (nice) 1D integrals are fast

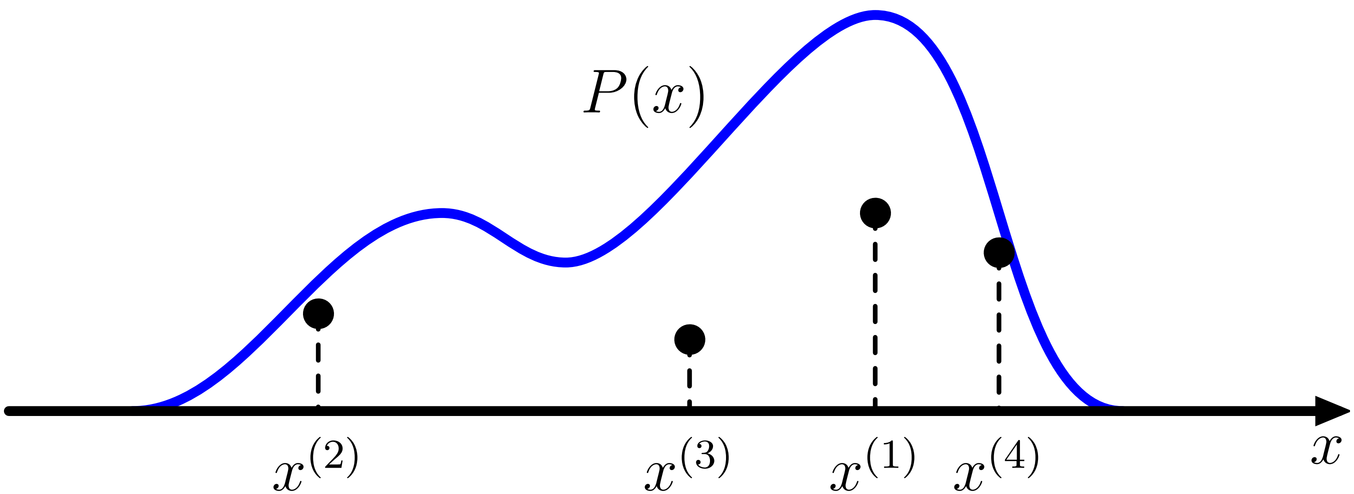
```
octave:1> 4 * quadl(@(x) sqrt(1-x.^2), 0, 1, tolerance)
```

Gives π to 6 dp's in 108 evaluations, machine precision in 2598.

(NB Matlab's `quadl` fails at zero tolerance)

Sampling from distributions

Draw points uniformly under the curve:



Probability mass to left of point $\sim$ Uniform[0,1]

Sampling from distributions

How to convert samples from a Uniform[0,1] generator:

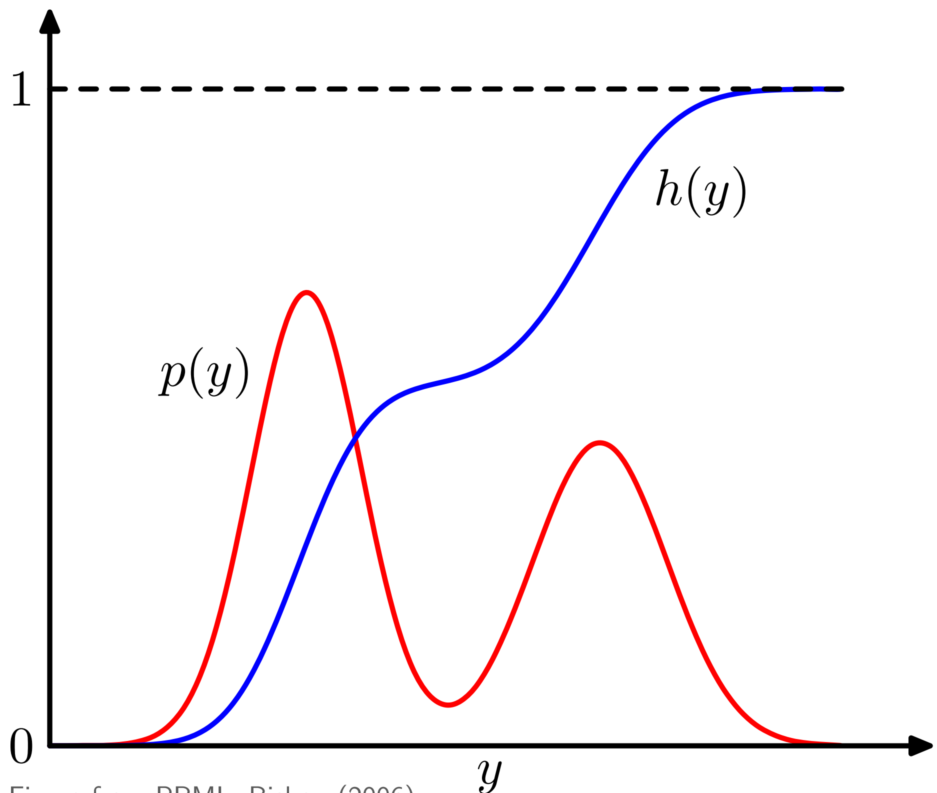


Figure from PRML, Bishop (2006)

$h(y) = \int_{-\infty}^{y} p(y')\, \mathrm{d}y'$

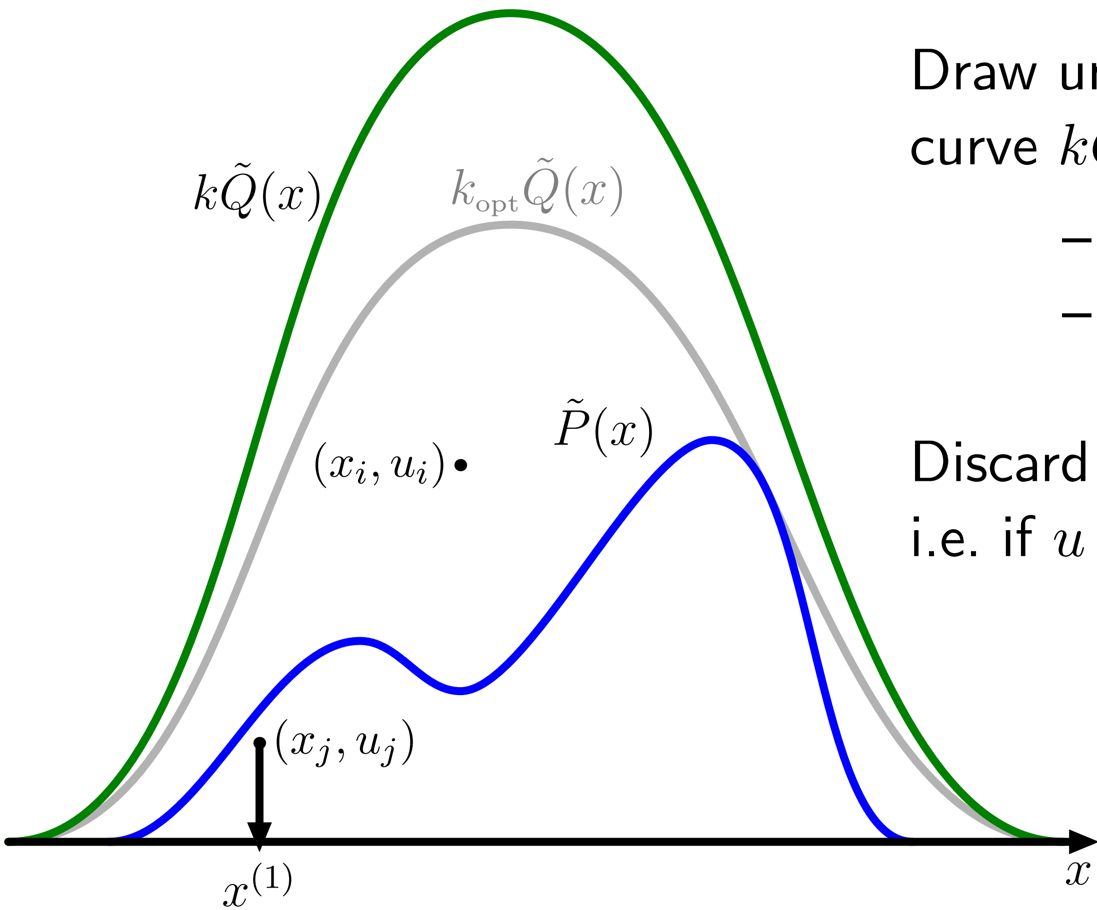
Draw mass to left of point:
$u \sim$ Uniform[0,1]

Sample, $y(u) = h^{-1}(u)$

Although we can't always compute and invert $h(y)$

Rejection sampling

Sampling underneath a $\tilde{P}(x) \propto P(x)$ curve is also valid



Draw underneath a simple curve $k\tilde{Q}(x) \geq \tilde{P}(x)$:

- Draw $x \sim Q(x)$
- height $u \sim$ Uniform$[0, k\tilde{Q}(x)]$

Discard the point if above $\tilde{P}$, i.e. if $u > \tilde{P}(x)$

Importance sampling

Computing $\tilde{P}(x)$ and $\tilde{Q}(x)$, then *throwing* x *away* seems wasteful
Instead rewrite the integral as an expectation under Q:

$$\int f(x)P(x)\,\mathrm{d}x \;=\; \int f(x)\frac{P(x)}{Q(x)}Q(x)\,\mathrm{d}x, \qquad (Q(x)>0 \text{ if } P(x)>0)$$

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

This is just simple Monte Carlo again, so it is unbiased.

Importance sampling applies when the integral is not an expectation.
Divide and multiply any integrand by a convenient distribution.

Importance sampling (2)

Previous slide assumed we could evaluate $P(x) = \tilde{P}(x)/\mathcal{Z}_P$

$$\int f(x)P(x)\,\mathrm{d}x \;\approx\; \frac{\mathcal{Z}_Q}{\mathcal{Z}_P}\,\frac{1}{S}\sum_{s=1}^{S} f(x^{(s)})\underbrace{\frac{\tilde{P}(x^{(s)})}{\tilde{Q}(x^{(s)})}}_{\tilde{r}^{(s)}}, \quad x^{(s)} \sim Q(x)$$

$$\approx \cancel{\frac{1}{S}}\sum_{s=1}^{S} f(x^{(s)})\frac{\tilde{r}^{(s)}}{\cancel{\frac{1}{S}}\sum_{s'}\tilde{r}^{(s')}} \;\equiv\; \sum_{s=1}^{S} f(x^{(s)})w^{(s)}$$

This estimator is **consistent** but **biased**

Exercise: Prove that $\mathcal{Z}_P/\mathcal{Z}_Q \approx \frac{1}{S}\sum_s \tilde{r}^{(s)}$

Summary so far

- Sums and integrals, often expectations, occur frequently in statistics
- **Monte Carlo** approximates expectations with a sample average
- **Rejection sampling** draws samples from complex distributions
- **Importance sampling** applies Monte Carlo to ‘any’ sum/integral

Pitfalls of Monte Carlo

Rejection & importance sampling scale badly with dimensionality

Example:

$$P(x) = \mathcal{N}(0, \mathbb{I}), \quad Q(x) = \mathcal{N}(0, \sigma^2\mathbb{I})$$

Rejection sampling:

Requires $\sigma \geq 1$. Fraction of proposals accepted $= \sigma^{-D}$

Importance sampling:

Variance of importance weights $= \left(\frac{\sigma^2}{2-1/\sigma^2}\right)^{D/2} - 1$

Infinite / undefined variance if $\sigma \leq 1/\sqrt{2}$

Outline

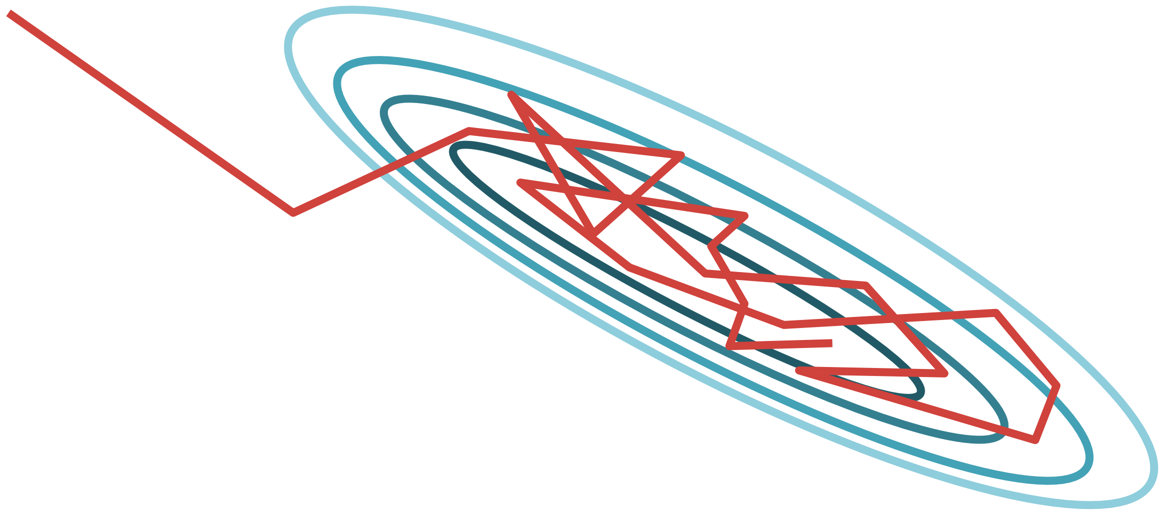
- **Review: Monte Carlo**
- **MCMC (Basic Methods)**
  - Metropolis algorithm
  - Metropolis-Hastings (M-H) algorithm
  - Gibbs Sampling
- **Markov Chains**
  - Transition probabilities
  - Invariant distribution
  - Equilibrium distribution
  - Markov chain as a WFSM
  - Constructing Markov chains
  - Why does M-H work?
- **MCMC (Auxiliary Variable Methods)**
  - Slice Sampling
  - Hamiltonian Monte Carlo

Metropolis, Metropolis-Hastings, Gibbs Sampling

MCMC (BASIC METHODS)

MCMC

- **Goal:** Draw approximate, correlated samples from a target distribution p(x)
- **MCMC:** Performs a biased random walk to explore the distribution



Simulations of MCMC

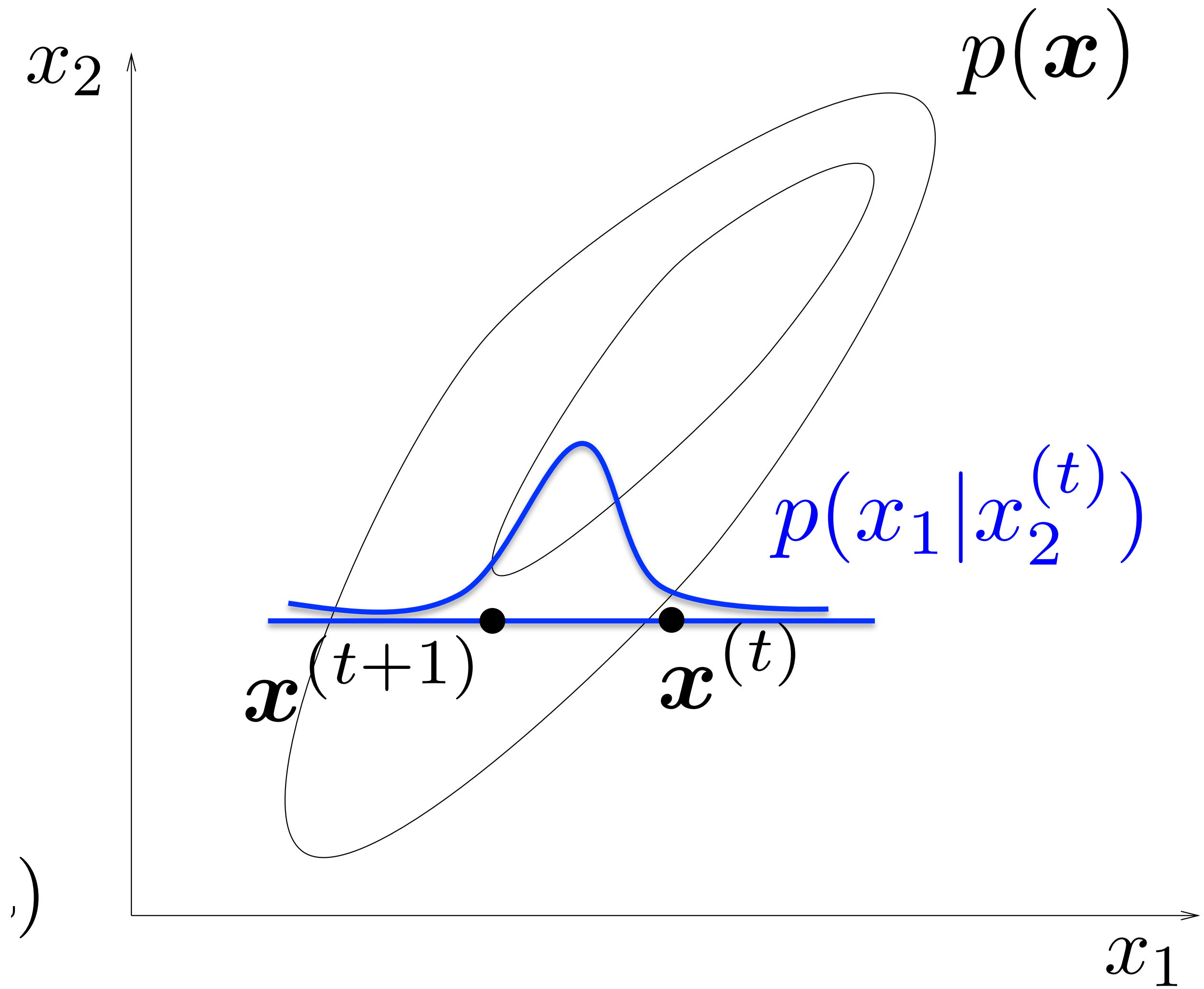
Visualization of Metroplis-Hastings, Gibbs Sampling, and Hamiltonian MCMC:

http://twiecki.github.io/blog/2014/01/02/visualizing-mcmc/

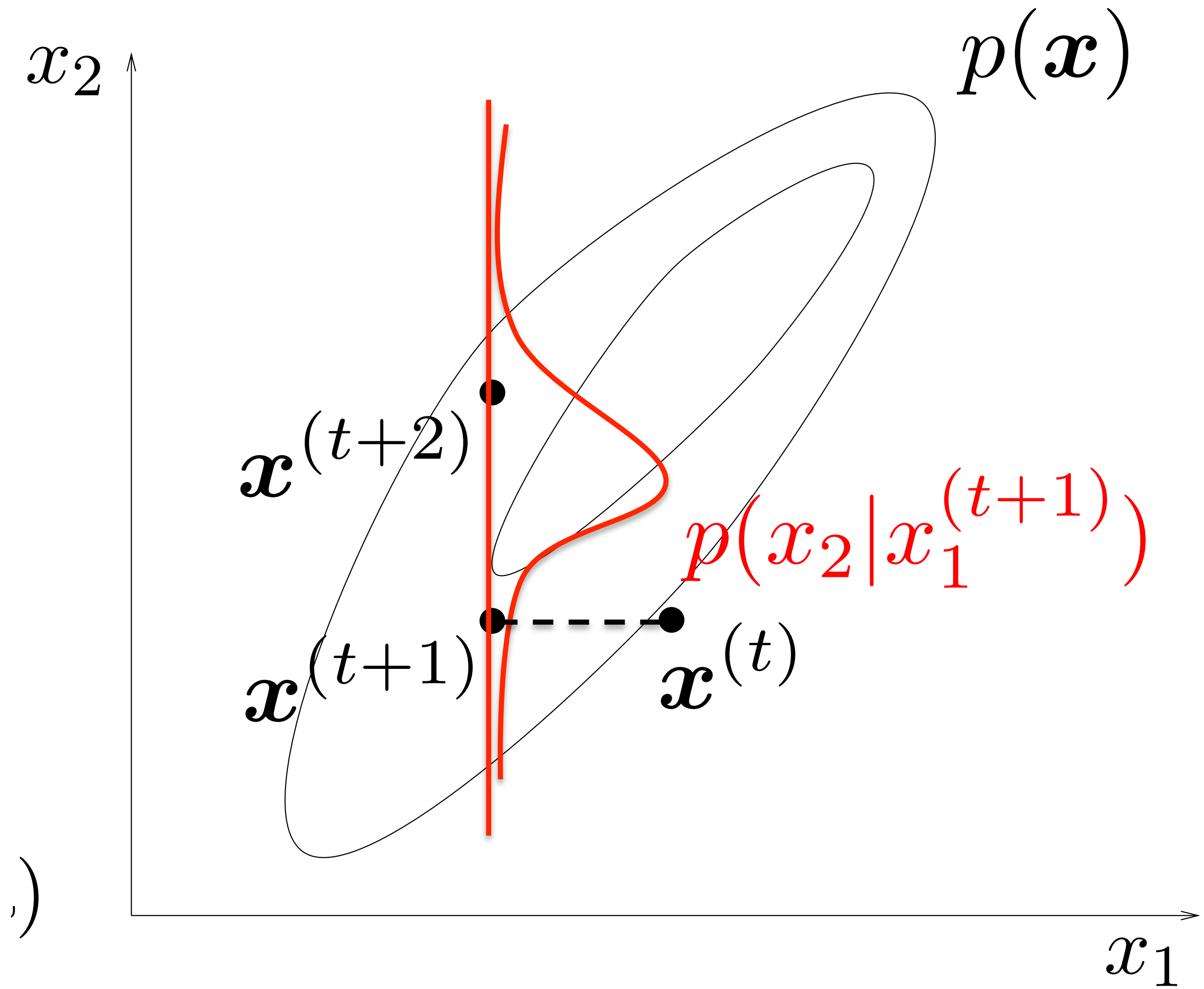
Whiteboard

- Metropolis Algorithm
- Metropolis-Hastings Algorithm
- Gibbs Sampling

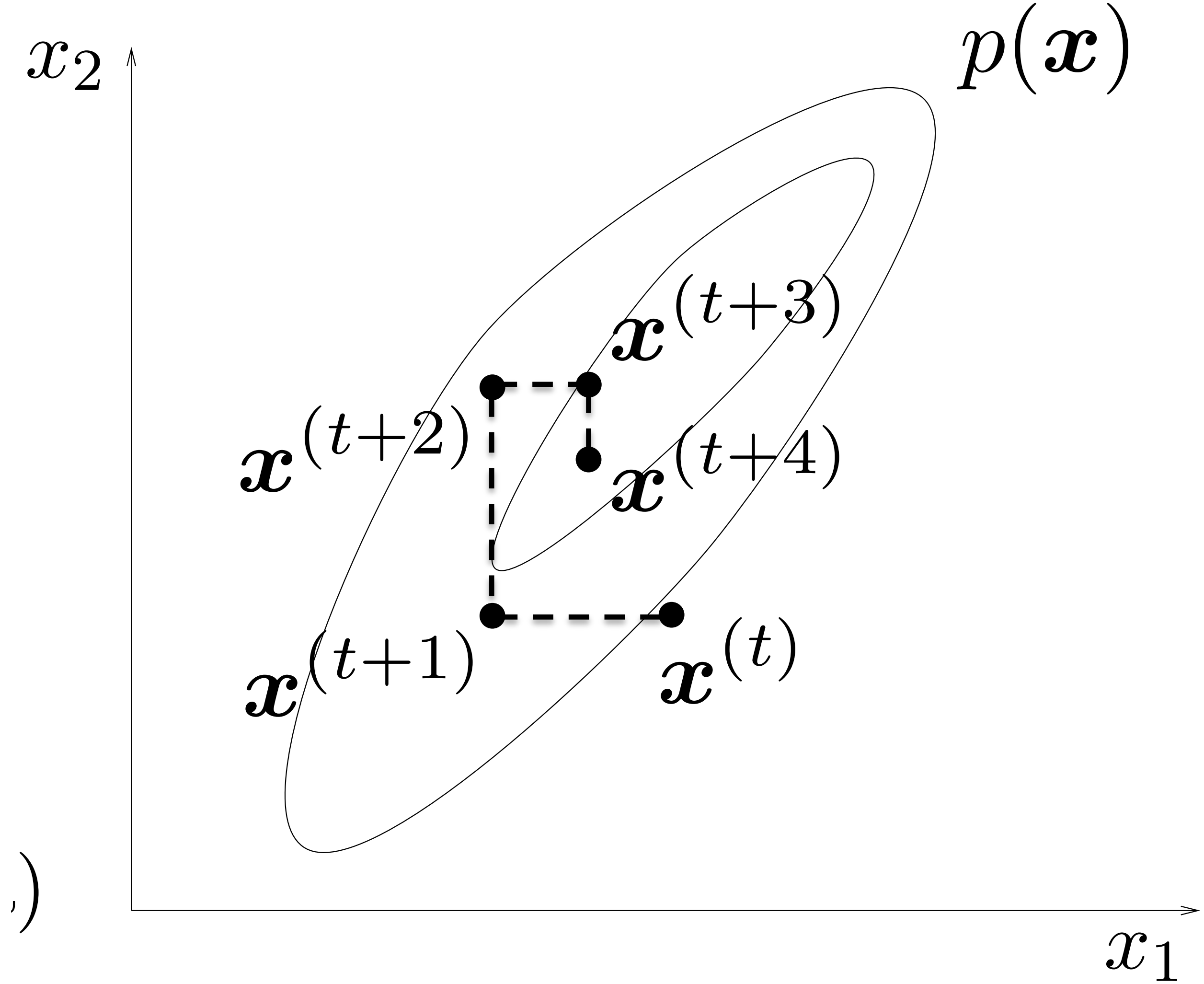
Gibbs Sampling



Gibbs Sampling



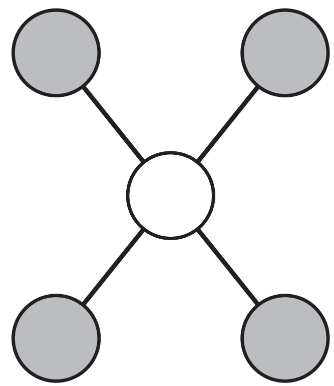
Gibbs Sampling



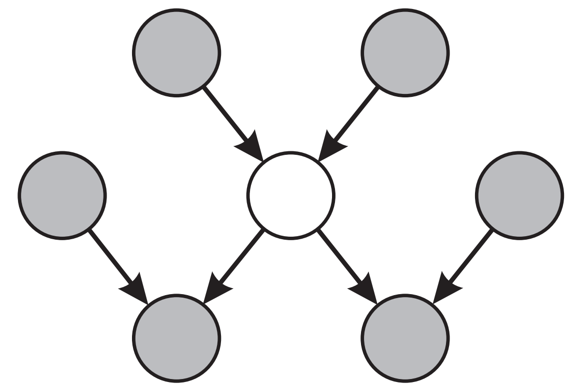
Gibbs Sampling

Full conditionals only need to condition on the Markov Blanket

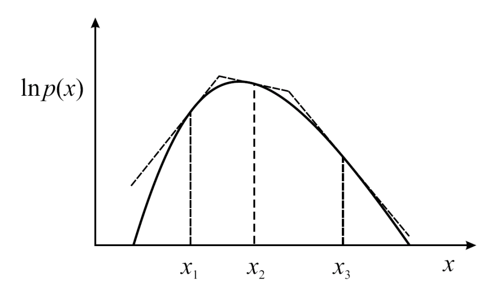
MRF



Bayes Net



- Must be "easy" to sample from conditionals
- Many conditionals are log-concave and are amenable to adaptive rejection sampling



Whiteboard

- Gibbs Sampling as M-H

Figure from Bishop (2006)

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

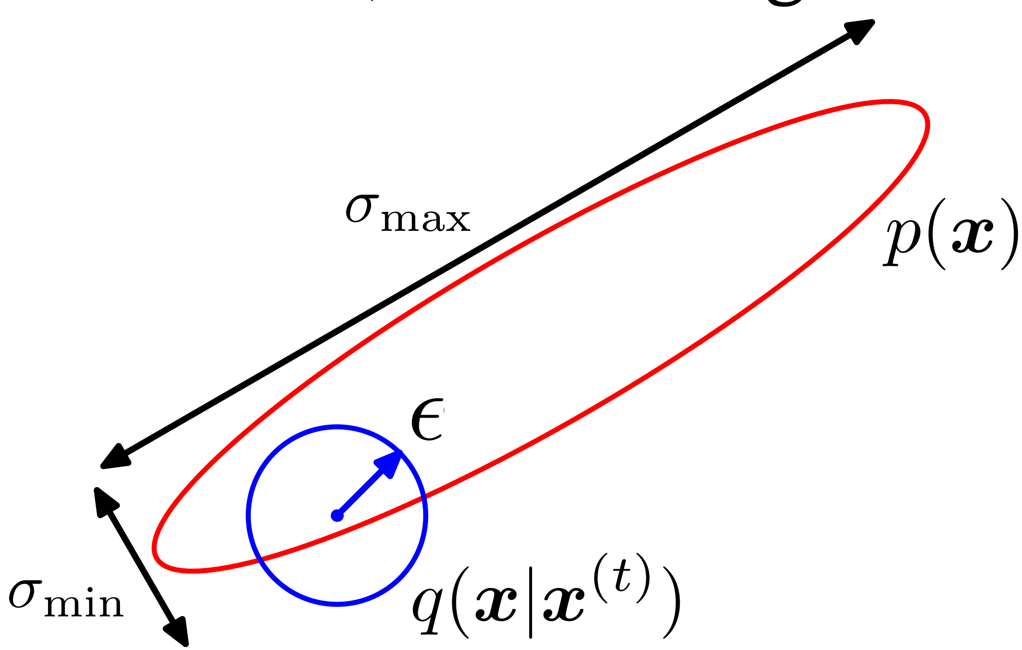
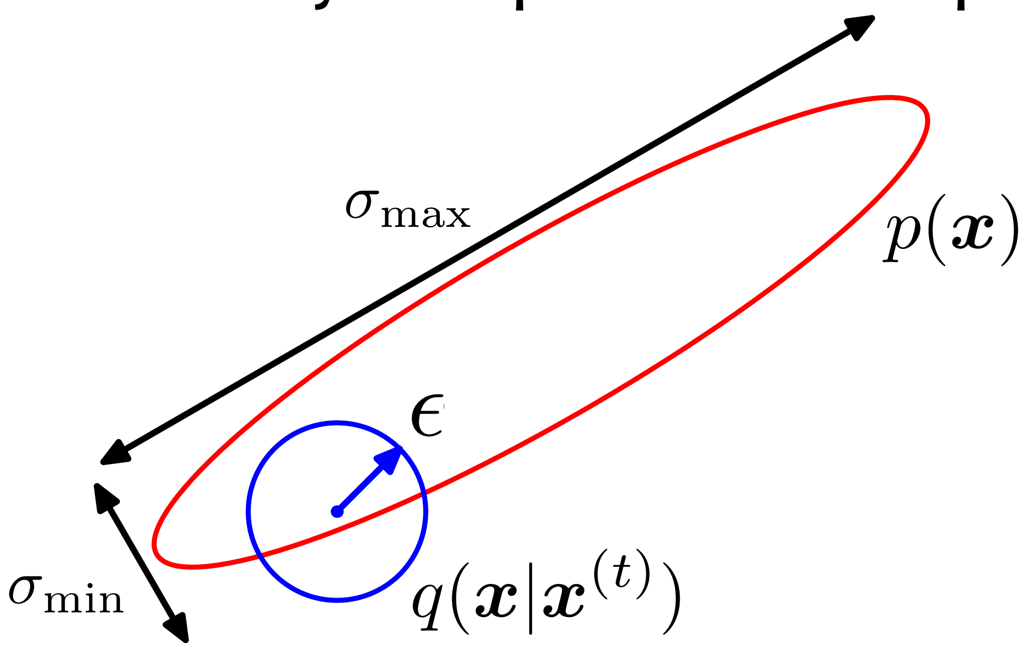


Figure from Bishop (2006)

Random Walk Behavior of M-H

- For Rejection Sampling, the accepted samples are 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

Definitions and Theoretical Justification for MCMC

MARKOV CHAINS

Whiteboard

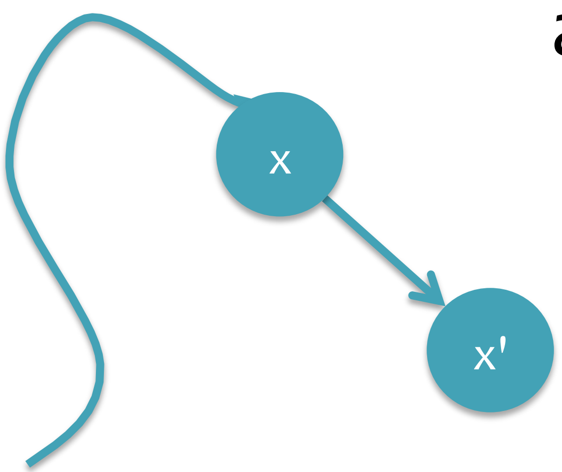
- Markov chains
- Transition probabilities
- Invariant distribution
- Equilibrium distribution
- Sufficient conditions for MCMC
- Markov chain as a WFSM

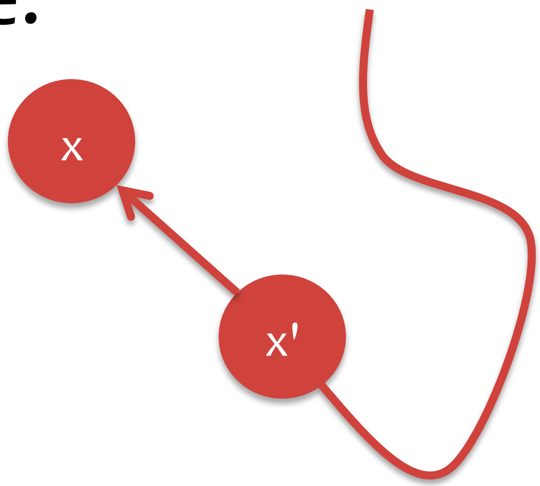
Detailed Balance

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

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

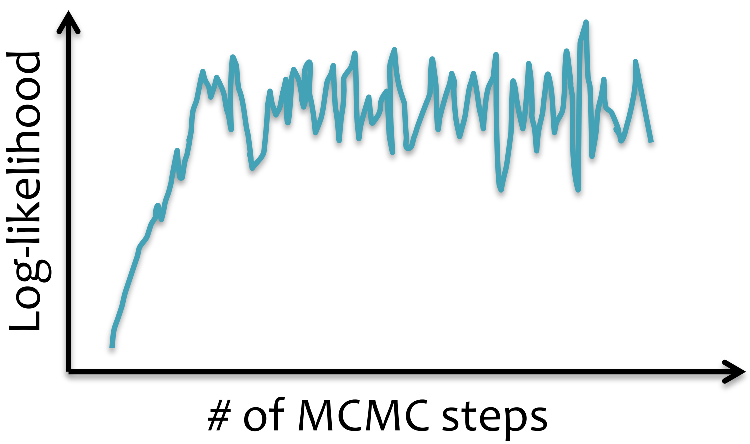
Practical Issues

- **Question:** Is it better to move along one dimension or many?
- **Answer:** For Metropolis-Hasings, 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?

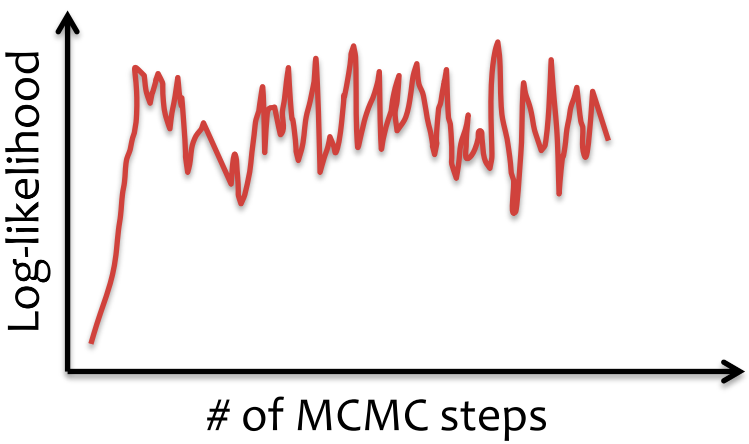
Practical Issues

- **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

Chain 1

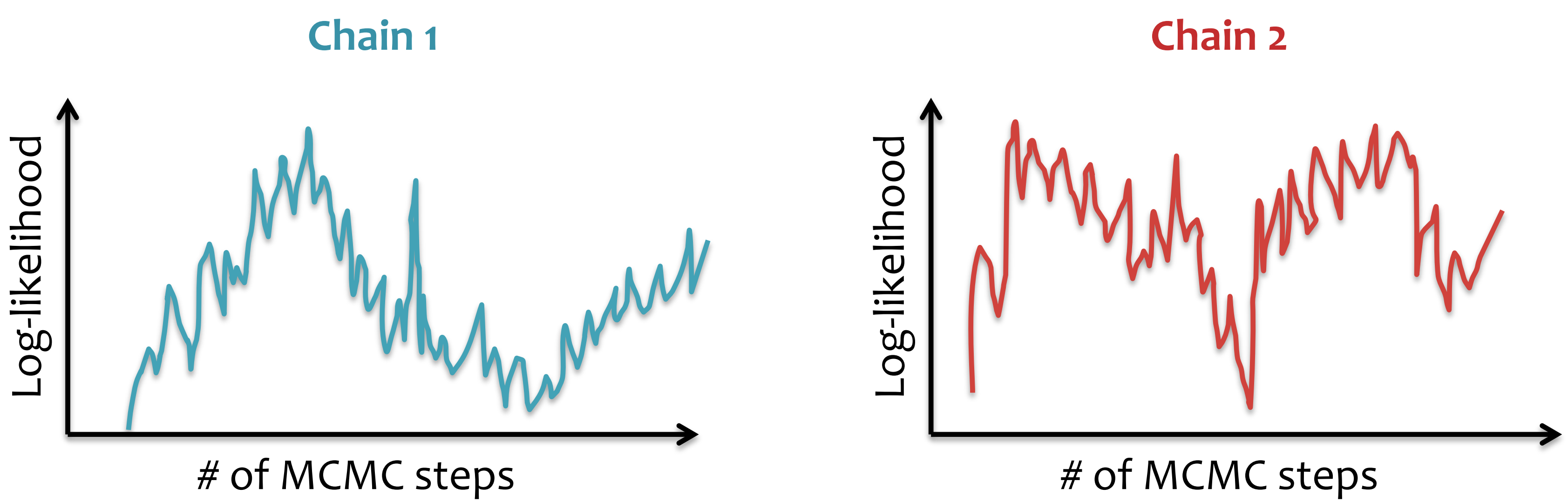


Chain 2



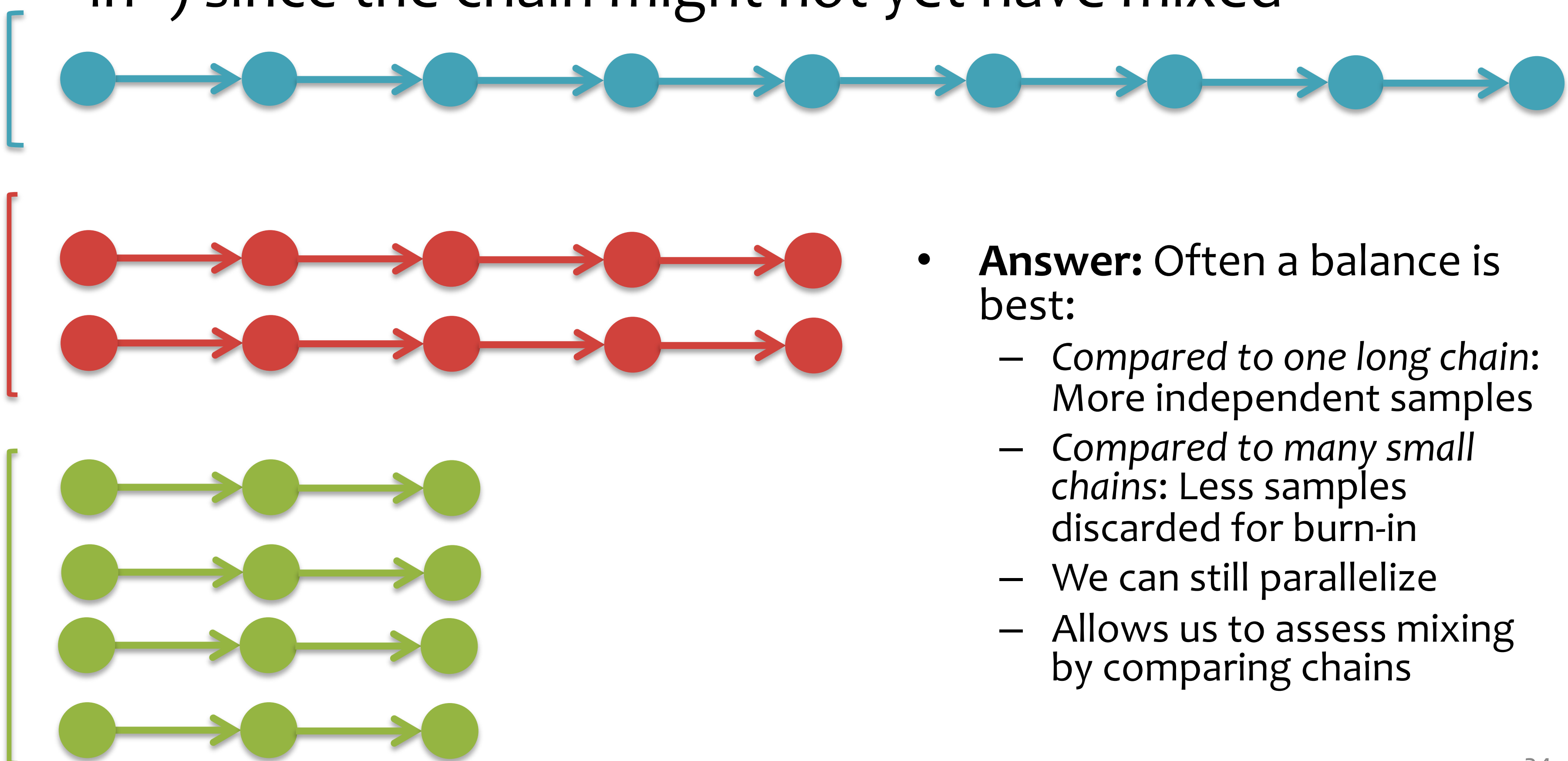
Practical Issues

- **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

Whiteboard

- Blocked Gibbs Sampling

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:

$$\int f(x)P(x)\,\mathrm{d}x = \int f(x)P(x,v)\,\mathrm{d}x\,\mathrm{d}v$$

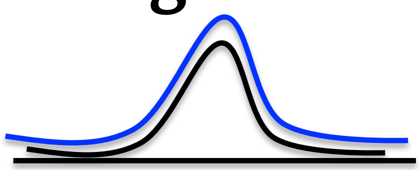
$$\approx \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)}), \quad x,v \sim P(x,v)$$

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**

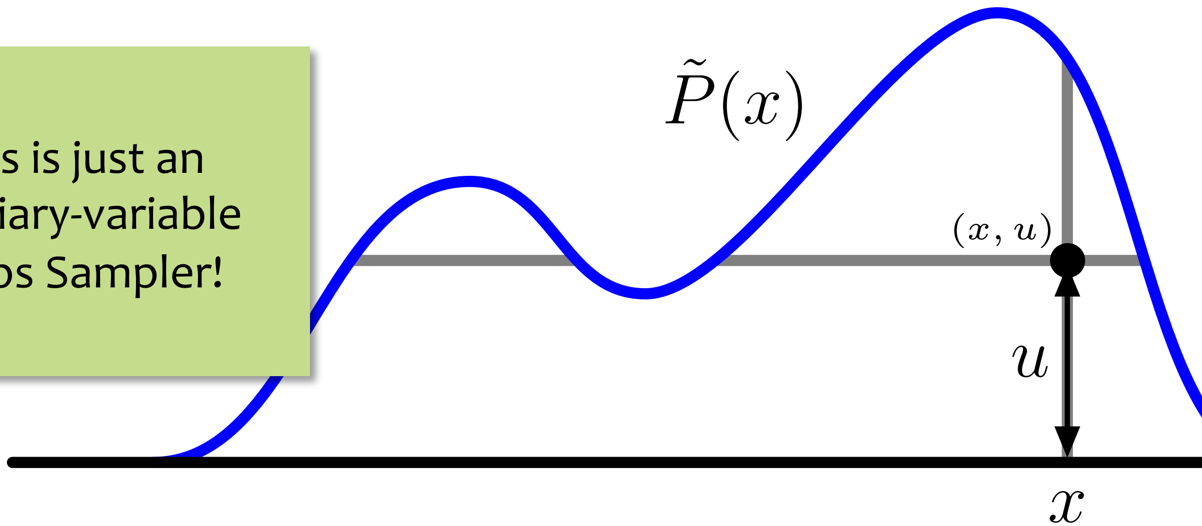
$\tilde{p}(x)$ 

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

Slice sampling idea

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

This is just an auxiliary-variable Gibbs Sampler!

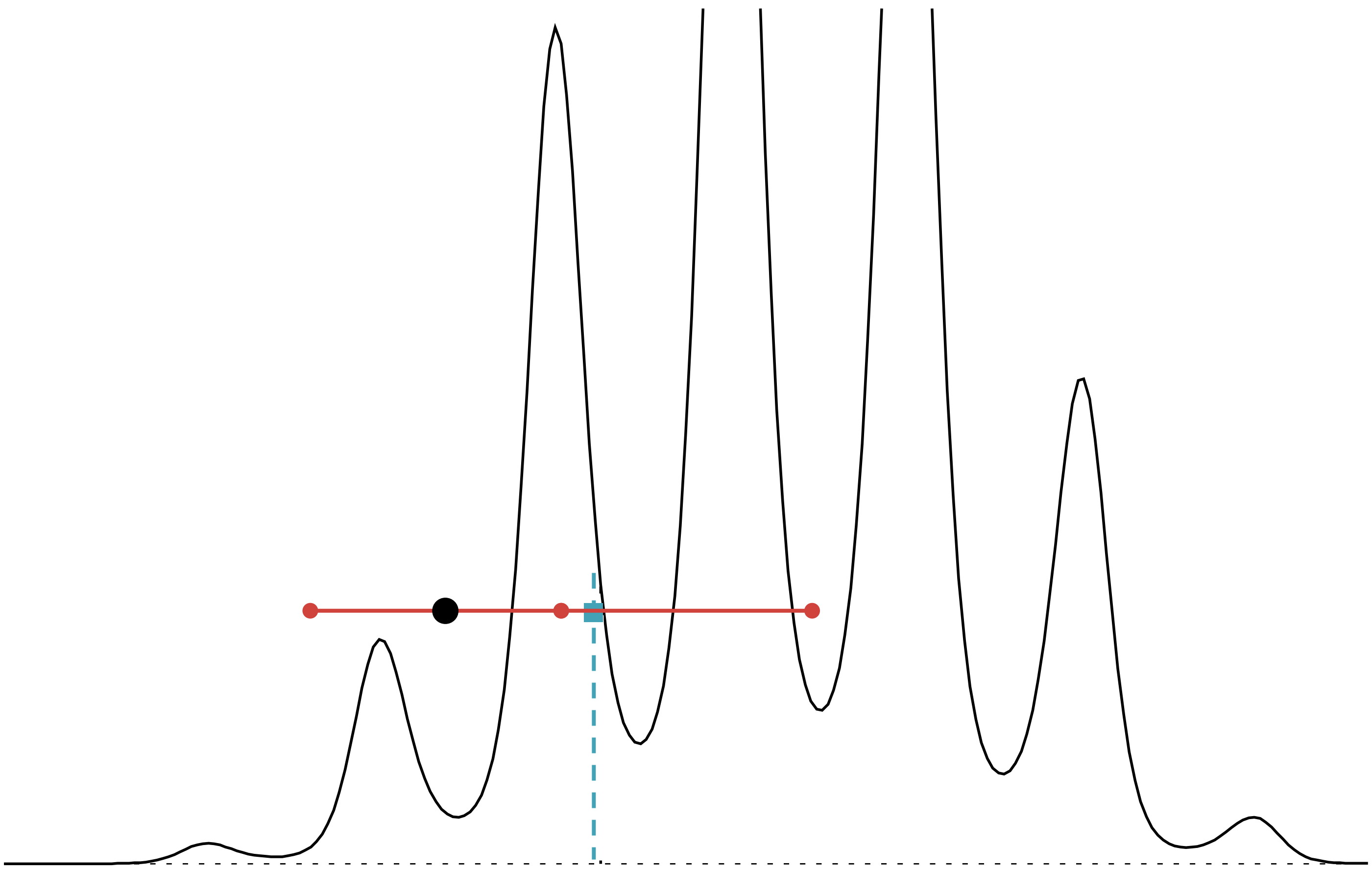


Problem: Sampling from the conditional $p(x \mid u)$ might be infeasible.

$$p(u|x) = \text{Uniform}[0, \tilde{P}(x)]$$

$$p(x|u) \propto \begin{cases} 1 & \tilde{P}(x) \geq u \\ 0 & \text{otherwise} \end{cases} = \text{"Uniform on the slice"}$$

Slice Sampling



Slice Sampling

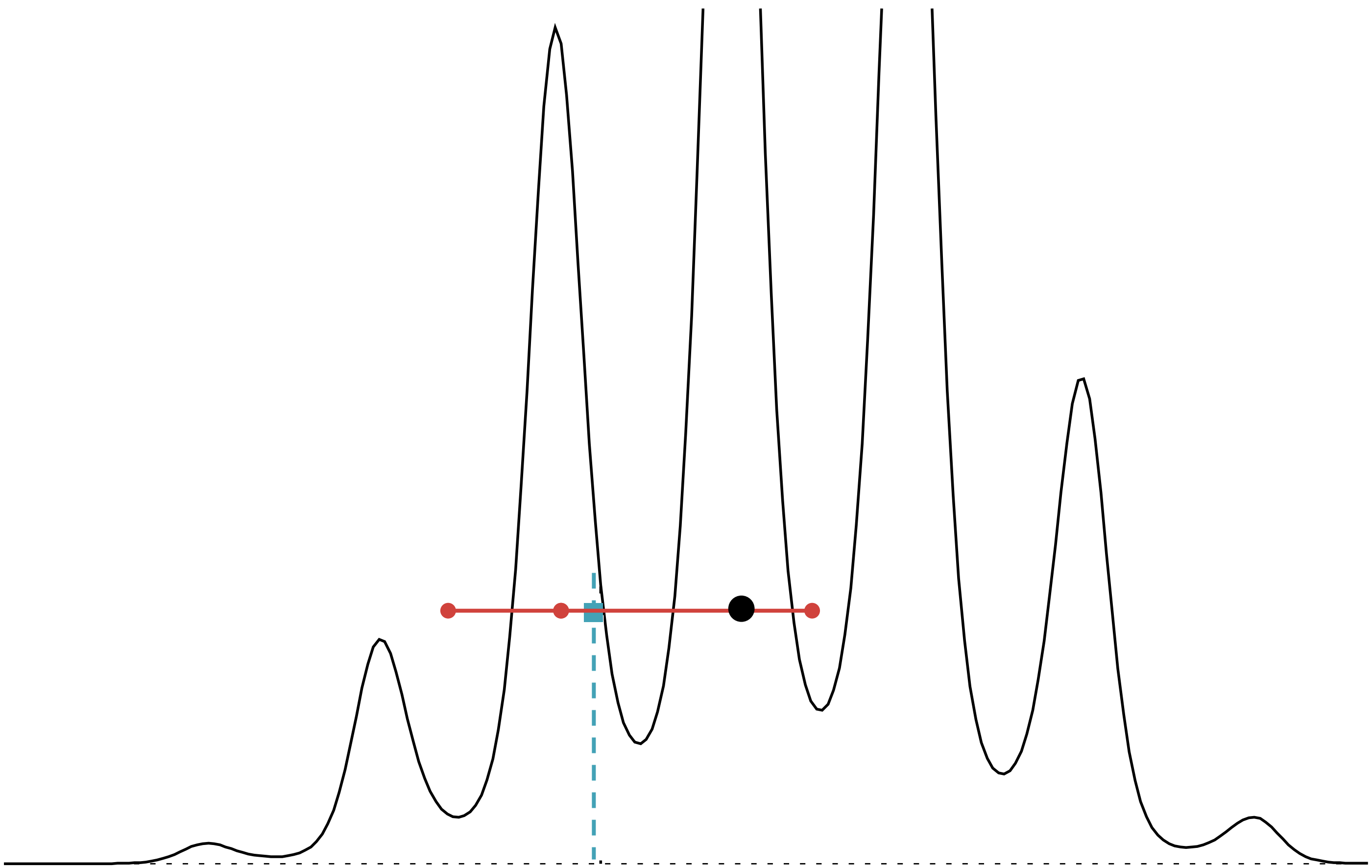
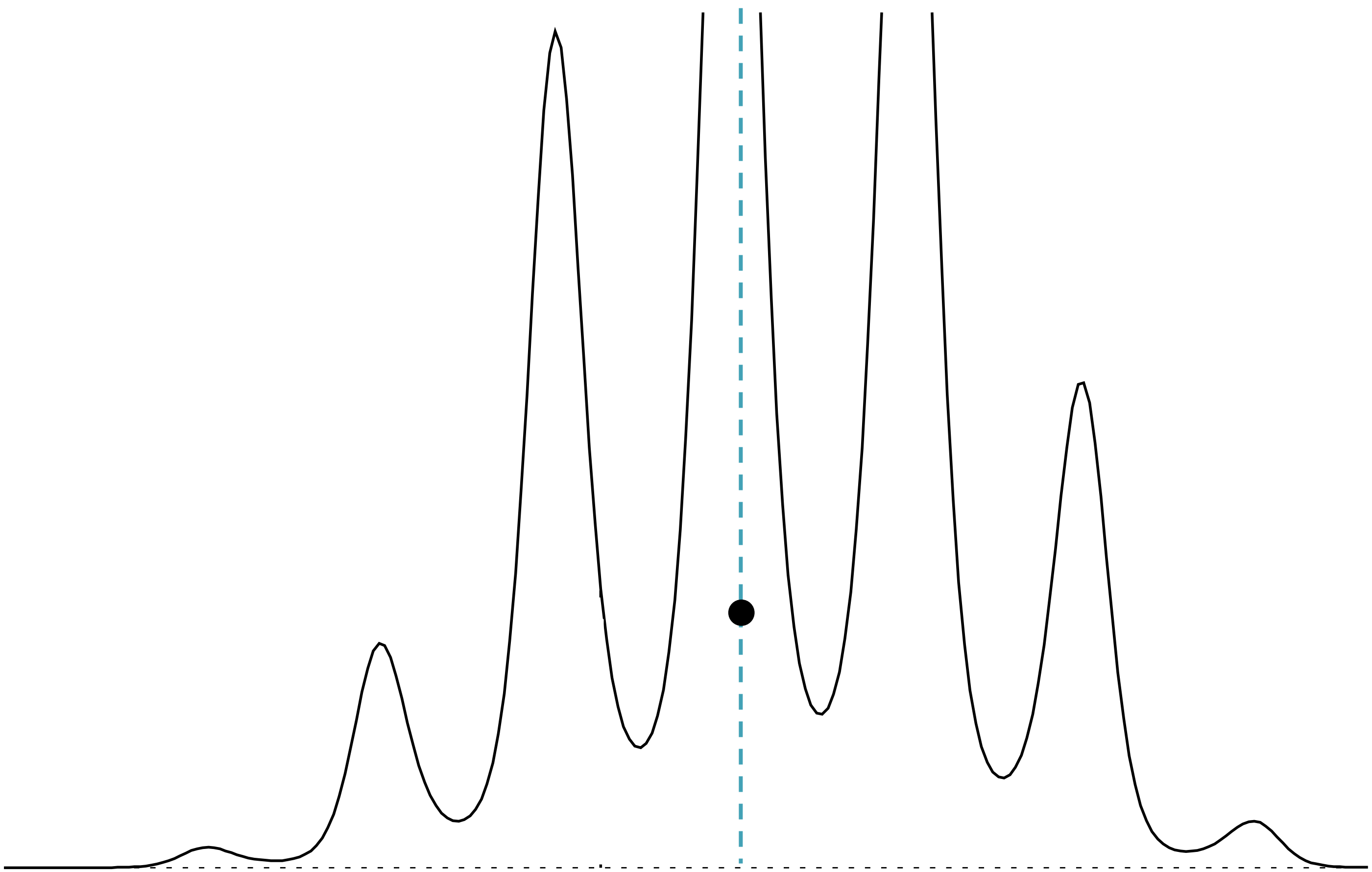


Figure adapted from MacKay Ch. 29

Slice Sampling



Slice Sampling

Algorithm:

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.

Slice Sampling

Algorithm:

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

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

Part 1: Stepping Out

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

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

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

Expand until endpoints are "outside" region under curve.

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

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

Part 2: Sample x (Shrinking)

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

Slice Sampling

Algorithm:

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.

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

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

Part 2: Sample x (Shrinking)

while(true) {

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

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

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

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

$\text{else}\{x_l = x\}$

}

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

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:

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

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

- We could use **random-walk M-H** to draw samples, but it seems a shame to **discard gradient information** $\nabla_{\boldsymbol{x}} E(\boldsymbol{x})$

- If we can evaluate it, the gradient tells us where to look for **high-probability regions**!

Background: Hamiltonian Dyanmics

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 Dyanmics

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

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

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

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

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

Hamiltonian function

Given a starting position $x^{(1)}$ and a starting momentum $p^{(1)}$ we can simulate the Hamiltonian dynamics of the system via:

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

Background: Hamiltonian Dyanmics

Parameters to tune:

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

Leapfrog Algorithm:

$$\text{for } \tau \text{ in } 1 \ldots L:$$

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

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

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

Figure from Neal (2011)

Background: Hamiltonian Dyanmics

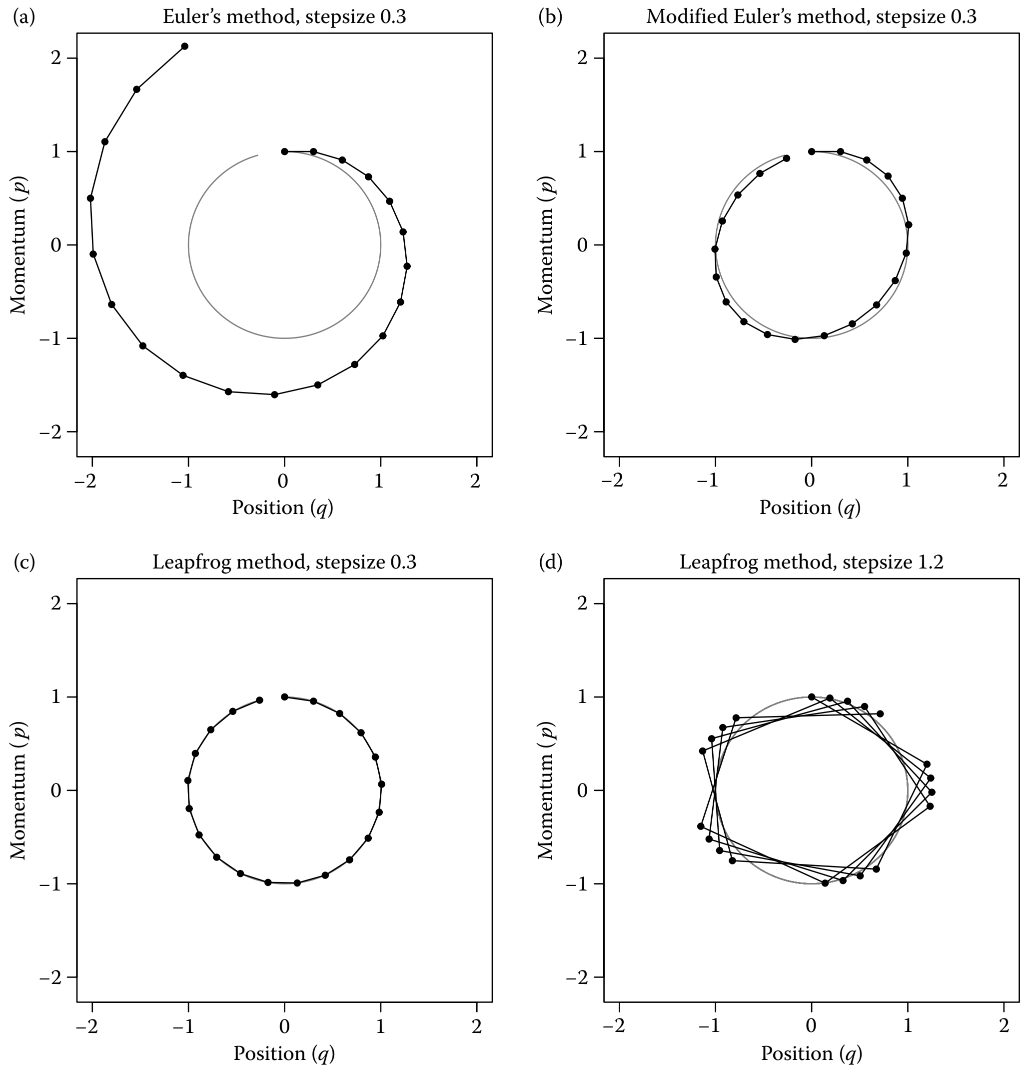


Figure from Neal (2011)

Hamiltonian Monte Carlo

Preliminaries

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

Define:

$$K(\boldsymbol{p}) = \boldsymbol{p}^T \boldsymbol{p}/2$$

$$H(\boldsymbol{x}, \boldsymbol{p}) = E(\boldsymbol{x}) + K(\boldsymbol{p})$$

$$\begin{aligned} p(\boldsymbol{x}, \boldsymbol{p}) &= \exp\{-H(\boldsymbol{x}, \boldsymbol{p})\}/Z_H \\ &= \exp\{-E(\boldsymbol{x}\} \exp\{-K(\boldsymbol{p})\}/Z_H \end{aligned}$$

Note:
Since p(x,p) is separable…

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

Target dist.

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

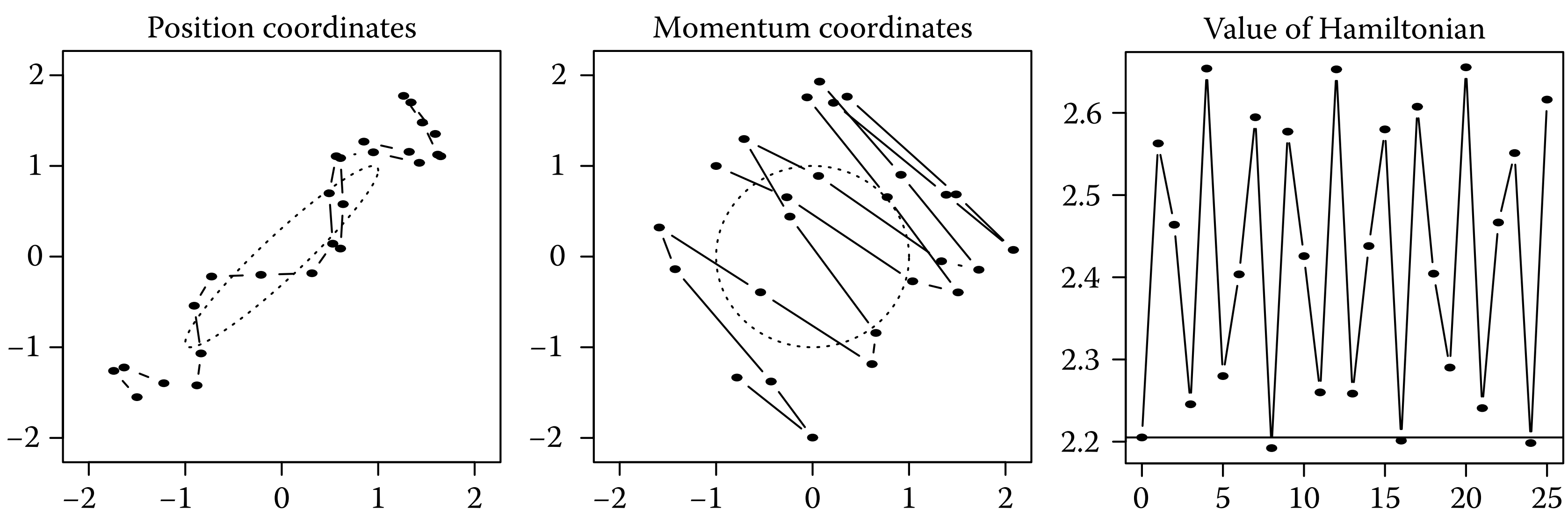
Gaussian

Whiteboard

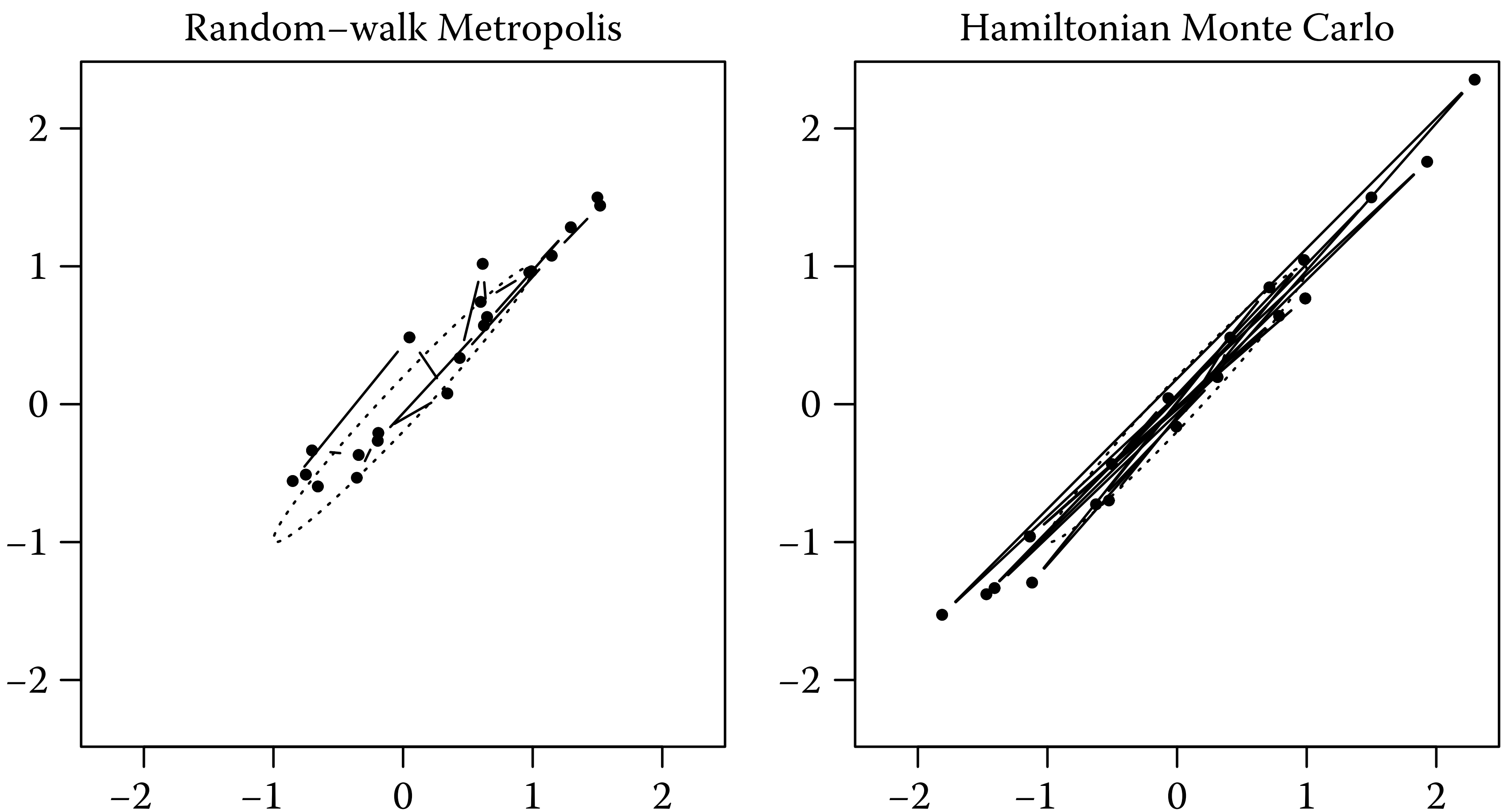
- Hamiltonian Monte Carlo algorithm (aka. Hybrid Monte Carlo)

Figure from Neal (2011)

Hamiltonian Monte Carlo



M-H vs. HMC


Random−walk Metropolis
Hamiltonian Monte Carlo
2
1
0
−1
−2
−2
−1
0
1
2

Simulations of MCMC

Visualization of Metroplis-Hastings, Gibbs Sampling, and Hamiltonian MCMC:

http://twiecki.github.io/blog/2014/01/02/visualizing-mcmc/

MCMC Summary

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