Machine Learning Department
School of Computer Science
Carnegie Mellon University

# Exam 1 Review
# +
# MCMC

Matt Gormley
Lecture 12
Oct. 10, 2022

# Reminders

- Homework 2: Learning to Search for RNNs
  - Programming + Empirical Questions
    - Due: Mon, Oct 24 at 9:00am
  - Policy: 65 points or more on the autograder gives 100% autograder credit
- Homework 3: General Graph CRF Module
  - Out: Thu, Sep 29
  - Due: Mon, Oct 10 at 11:59pm
- Practice Problems 1
- Exam 1: Fri, Oct 14, in-class

# EXAM 1 LOGISTICS

# Exam 1

- **Time / Location**
  - **Time:** In-Class Exam
    **Fri, Oct. 14 at 1:25pm – 2:45pm**
  - **Location**: The same room as lecture/recitation.
    Please arrive a few minutes early.
  - Please watch Piazza carefully for announcements.
- **Logistics**
  - Covered material: Lecture 1 – Lecture 10
  - Format of questions:
    - Multiple choice
    - True / False (with justification)
    - Derivations
    - Short answers
    - Interpreting figures
    - Implementing algorithms on paper
    - Drawing
  - No electronic devices
  - You are allowed to **bring** one 8½ x 11 sheet of notes (front and back)

# Topics for Exam 1

- Search-Based Structured Prediction
  - Reductions to Binary Classification
  - Learning to Search
  - RNN-LMs
  - seq2seq models
- Graphical Model Representation
  - Directed GMs vs. Undirected GMs vs. Factor Graphs
  - Bayesian Networks vs. Markov Random Fields vs. Conditional Random Fields
- Graphical Model Learning
  - ~~Fully observed Bayesian Network learning~~
  - Fully observed MRF learning
  - Fully observed CRF learning
  - Parameterization of a GM
  - Neural potential functions
- Exact Inference
  - Three inference problems:
    (1) marginals
    (2) partition function
    (3) most probably assignment
  - Variable Elimination
  - Belief Propagation (sum-product and max-product)

# SAMPLE QUESTIONS

# Sample Questions

## Learning to Search

Suppose you are training a seq2seq model for supervised POS Tagging.

- Let the inputs to the encoder be $e_1$, $e_2$, $e_3$, …
- Let the inputs to the decoder be $d_1$, $d_2$, $d_3$, …
- Let the outputs of the decoder be $o_1$, $o_2$, $o_3$, …

1. (1 point) **Short Answer**: Describe in words what the inputs to the encoder would be. Assume you are training with Teacher Forcing.

2. (1 point) **Short Answer**: Describe in words what the inputs of the decoder would be. Assume you are training with Teacher Forcing.

3. (1 point) **Short Answer**: Describe in words what the outputs of the decoder would be. Assume you are training with Teacher Forcing.

# Sample Questions

## Learning to Search

Suppose you are training a seq2seq model for supervised POS Tagging.

- Let the inputs to the encoder be $e_1$, $e_2$, $e_3$, …
- Let the inputs to the decoder be $d_1$, $d_2$, $d_3$, …
- Let the outputs of the decoder be $o_1$, $o_2$, $o_3$, …

4. (1 point) **Short Answer**: Describe in words what the inputs to the encoder would be. Assume you are training with Scheduled Sampling. *(If your answer is the same as for Teacher Forcing, simply write "same".)*

5. (1 point) **Short Answer**: Describe in words what the inputs of the decoder would be. Assume you are training with Scheduled Sampling. *(If your answer is the same as for Teacher Forcing, simply write "same".)*

6. (1 point) **Short Answer**: Describe in words what the outputs of the decoder would be. Assume you are training with Scheduled Sampling. *(If your answer is the same as for Teacher Forcing, simply write "same".)*

# Sample Questions

## 6 Factor Graphs

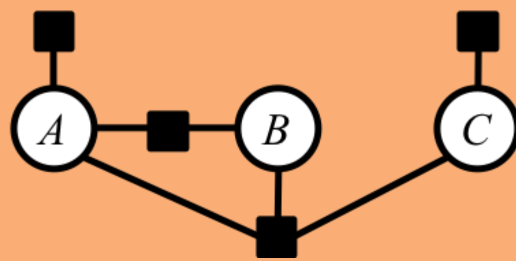



Figure 4: A factor graph over three binary random variables $A$, $B$, $C$, i.e. sampled values $a$, $b$, $c$ from the random variables are in $\{0, 1\}$. Assume the factors are named $\psi_A(a)$, $\psi_{A,B}(a, b)$, $\psi_{A,B,C}(a, b, c)$, and $\psi_C(c)$.

1. (2 points) **Short answer:** Consider the factor graph in Figure 4. Using the given factor names, write the partition function $Z$ that ensures the joint probability distribution $p(a, b, c)$ sums-to-one.

# Sample Questions

## 6 Factor Graphs

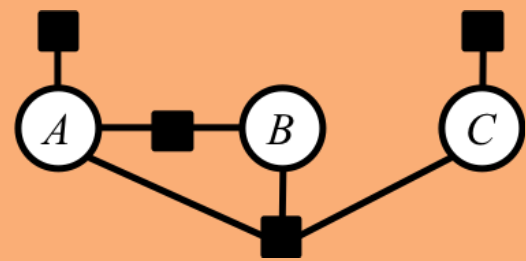



Figure 4: A factor graph over three binary random variables $A$, $B$, $C$, i.e. sampled values $a$, $b$, $c$ from the random variables are in $\{0, 1\}$. Assume the factors are named $\psi_A(a)$, $\psi_{A,B}(a, b)$, $\psi_{A,B,C}(a, b, c)$, and $\psi_C(c)$.

2. (2 points) **Short answer:** Using the given factor names, write the joint probability mass function $p(a, b, c)$ defined by the factor graph shown in Figure 4. *You may include the term $Z$ directly in your answer—no need to copy it from above.*

# Sample Questions

## 6 Factor Graphs

3. (2 points) **Drawing:** Suppose we have a joint probability distribution that factorizes as below:

$$p(w, x, y, z) \propto \psi_X(x)\psi_{X,Y}(x, y)\psi_{X,Y,Z}(x, y, z)\psi_{W,Z}(w, z)\psi_{Y,Z}(y, z)$$

where $\propto$ denotes *proportional to*. Draw the factor graph corresponding to this factorization of the joint distribution.

# Sample Questions

## 7 Inference in Graphical Models

Consider yet another factor graph consisting of two random variables $Q \in \{\text{red,green,blue}\}$, $R \in \{\text{pencil, crayon}\}$. Suppose we have the following factors:

| Q | $\psi_Q(q)$ |
|---|---|
| red | 3 |
| green | 1 |
| blue | 2 |

| Q | R | $\psi_{Q,R}(q,r)$ |
|---|---|---|
| red | pencil | 2 |
| red | crayon | 2 |
| green | pencil | 1 |
| green | crayon | 3 |
| blue | pencil | 4 |
| blue | crayon | 1 |

1. (2 points) **Short answer:** Draw a table containing all values of the function $s(q,r) = \psi_Q(q)\psi_{Q,R}(q,r)$. *You may use the integer abbreviations: red=1, green=2, blue=3, pencil=1, crayon=2.*

# Sample Questions

**Question:**

**Answer:**

## 7 Inference in Graphical Models

Consider yet another factor graph consisting of two random variables $Q \in \{\text{red,green,blue}\}$, $R \in \{\text{pencil, crayon}\}$. Suppose we have the following factors:

| Q | $\psi_Q(q)$ |
|---|---|
| red | 3 |
| green | 1 |
| blue | 2 |

| Q | R | $\psi_{Q,R}(q,r)$ |
|---|---|---|
| red | pencil | 2 |
| red | crayon | 2 |
| green | pencil | 1 |
| green | crayon | 3 |
| blue | pencil | 4 |
| blue | crayon | 1 |

2. (2 points) **Numerical answer:** What is the value of the partition function $Z$ for the joint distribution $p(q,r)$?

# Sample Questions

## 7 Inference in Graphical Models

Consider yet another factor graph consisting of two random variables $Q \in \{\text{red,green,blue}\}$, $R \in \{\text{pencil, crayon}\}$. Suppose we have the following factors:

| Q | $\psi_Q(q)$ |
|---|---|
| red | 3 |
| green | 1 |
| blue | 2 |

| Q | R | $\psi_{Q,R}(q,r)$ |
|---|---|---|
| red | pencil | 2 |
| red | crayon | 2 |
| green | pencil | 1 |
| green | crayon | 3 |
| blue | pencil | 4 |
| blue | crayon | 1 |

3. (2 points) **Numerical answer:** What is the value of the joint probability $P(Q = green, R = crayon)$? *You may leave your answer in the form of an unsimplified fraction—no calculator necessary.*

# Sample Questions

## 7 Inference in Graphical Models

Consider yet another factor graph consisting of two random variables $Q \in \{\text{red,green,blue}\}$, $R \in \{\text{pencil, crayon}\}$. Suppose we have the following factors:

| Q | $\psi_Q(q)$ |
|---|---|
| red | 3 |
| green | 1 |
| blue | 2 |

| Q | R | $\psi_{Q,R}(q,r)$ |
|---|---|---|
| red | pencil | 2 |
| red | crayon | 2 |
| green | pencil | 1 |
| green | crayon | 3 |
| blue | pencil | 4 |
| blue | crayon | 1 |

4. (2 points) **Numerical answer:** What is the value of the marginal probability $P(Q = green)$? *You may leave your answer in the form of an unsimplified fraction—no calculator necessary.*

# Sample Questions

## 7 Inference in Graphical Models

Consider yet another factor graph consisting of two random variables $Q \in \{\text{red,green,blue}\}$, $R \in \{\text{pencil, crayon}\}$. Suppose we have the following factors:

| Q | $\psi_Q(q)$ |
|---|---|
| red | 3 |
| green | 1 |
| blue | 2 |

| Q | R | $\psi_{Q,R}(q,r)$ |
|---|---|---|
| red | pencil | 2 |
| red | crayon | 2 |
| green | pencil | 1 |
| green | crayon | 3 |
| blue | pencil | 4 |
| blue | crayon | 1 |

5. (2 points) **Short answer:** Suppose you run the Variable Elimination algorithm to eliminate the variable $Q$, resulting in a new factor graph with just one factor $m(r)$. Draw a table containing the values of this new factor.

# Sample Questions

## 7 Inference in Graphical Models

Consider yet another factor graph consisting of two random variables $Q \in \{\text{red,green,blue}\}$, $R \in \{\text{pencil, crayon}\}$. Suppose we have the following factors:

| Q | $\psi_Q(q)$ |
|---|---|
| red | 3 |
| green | 1 |
| blue | 2 |

| Q | R | $\psi_{Q,R}(q,r)$ |
|---|---|---|
| red | pencil | 2 |
| red | crayon | 2 |
| green | pencil | 1 |
| green | crayon | 3 |
| blue | pencil | 4 |
| blue | crayon | 1 |

6. (2 points) **Numerical answer:** What is the value of the marginal probability $P(R = crayon)$? *You may leave your answer in the form of an unsimplified fraction—no calculator necessary.*

# Sample Questions

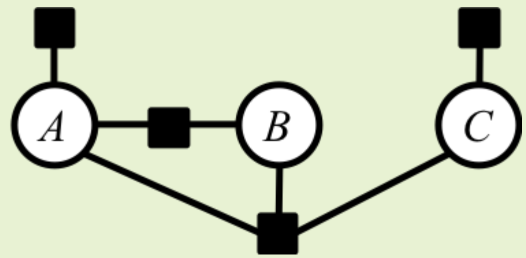



Figure 4: A factor graph over three binary random variables $A$, $B$, $C$, i.e. sampled values $a$, $b$, $c$ from the random variables are in $\{0, 1\}$. Assume the factors are named $\psi_A(a)$, $\psi_{A,B}(a, b)$, $\psi_{A,B,C}(a, b, c)$, and $\psi_C(c)$.

1. (1 point) **Drawing**: Suppose you are running the Variable Elimination algorithm. The first variable you eliminate is B. Draw the factor graph that results after you have eliminated variable B.

# Sample Questions

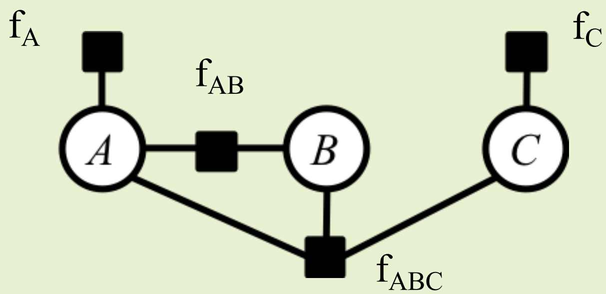


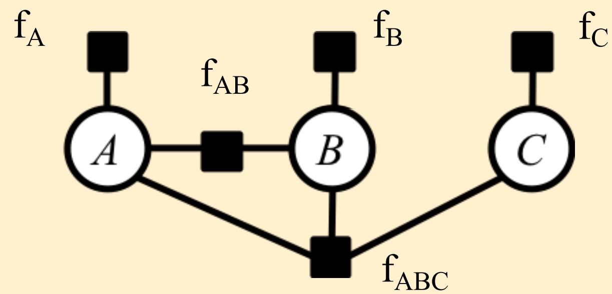

Figure 4: A factor graph over three binary random variables $A$, $B$, $C$, i.e. sampled values $a$, $b$, $c$ from the random variables are in $\{0, 1\}$. Assume the factors are named $\psi_A(a)$, $\psi_{A,B}(a, b)$, $\psi_{A,B,C}(a, b, c)$, and $\psi_C(c)$.

2. (1 point) **Numerical Answer**: Suppose you are running the Belief Propagation algorithm? How many messages are required to send a message from $f_{ABC}$ to C?

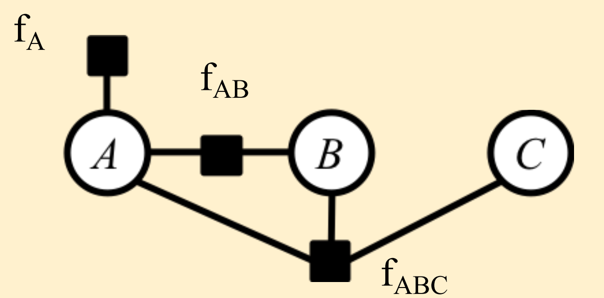
# Sample Questions

**Question:**

**Answer:**




1. (1 point) Is there a Bayesian Network that would convert to the factor graph shown above? Is yes, draw an example of such a Bayesian Network. If not, explain why not.




2. (1 point) Is there a Bayesian Network that would convert to the factor graph shown above? Is yes, draw an example of such a Bayesian Network. If not, explain why not.

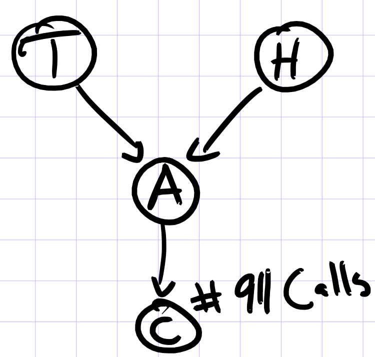
# Q&A

Metropolis, Metropolis-Hastings, Gibbs Sampling

# MCMC (BASIC METHODS)

# Sampling from a Joint Distribution

Ex: Tornado




$T \sim \text{Bernoulli}(\tau)$ $\quad \tau = 1/2$

$H \sim \text{Bernoulli}(\eta)$ $\quad \eta = 1/3$

$A \sim \text{Bernoulli}(\alpha_{H,T})$ $\quad \alpha = \begin{array}{c|cc} & T=0 & T=1 \\ \hline H=0 & 0 & 1/2 \\ H=1 & 1/2 & 1 \end{array}$

$C \sim \text{Unif}(\{1,\dots,63\}) + A * \text{Unif}(\{1,\dots,63\})$

← integer

| T | H | A | C |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

We can use these samples to estimate many different probabilities!

# A Few Problems for a Factor Graph

Suppose we already have the parameters of a Factor Graph…

1. How do we compute the probability of a specific assignment to the variables?
   P(T=t, H=h, A=a, C=c)

2. How do we draw a sample from the joint distribution?
   t,h,a,c ~ P(T, H, A, C)

3. How do we compute marginal probabilities?
   P(A) = …

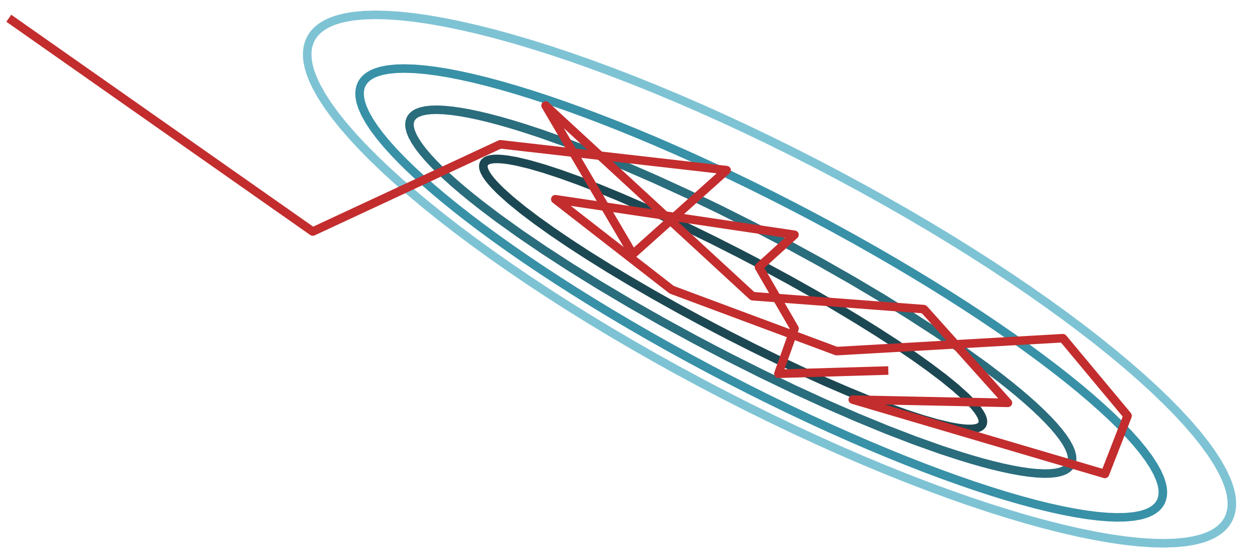
4. How do we draw samples from a conditional distribution?
   t,h,a ~ P(T, H, A | C = c)

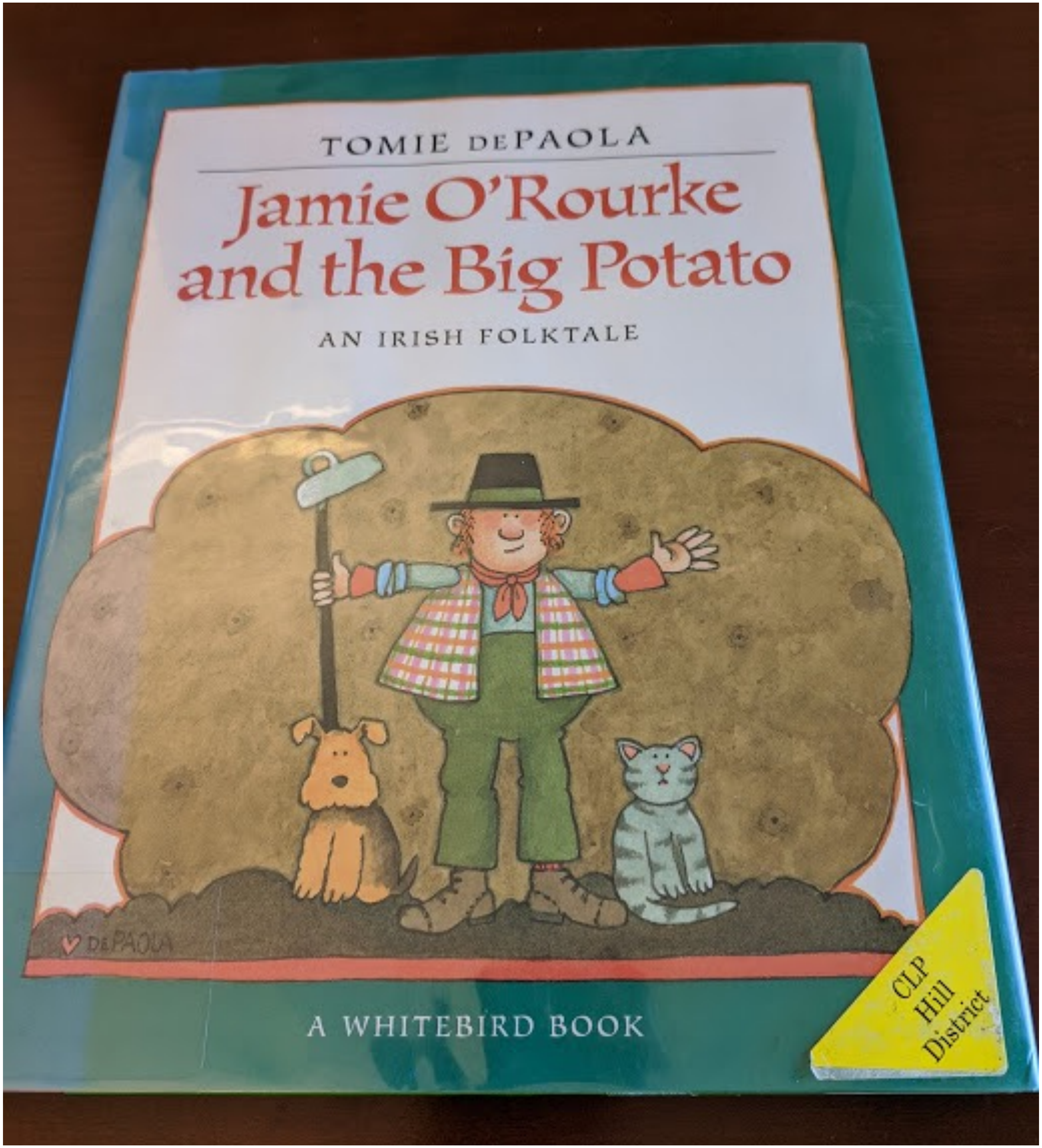
5. How do we compute conditional marginal probabilities?
   P(H | C = c) = …

Can we use samples ?

# MCMC

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




TOMIE DEPAOLA
Jamie O'Rourke and the Big Potato
AN IRISH FOLKTALE
DEPAOLA
A WHITEBIRD BOOK
CLP Hill District


https://commons.wikimedia.org/wiki/File:Map_of_ireland.jpg

# Simulations of MCMC

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

https://chi-feng.github.io/mcmc-demo/

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

# GIBBS SAMPLING

# Gibbs Sampling

***Whiteboard***

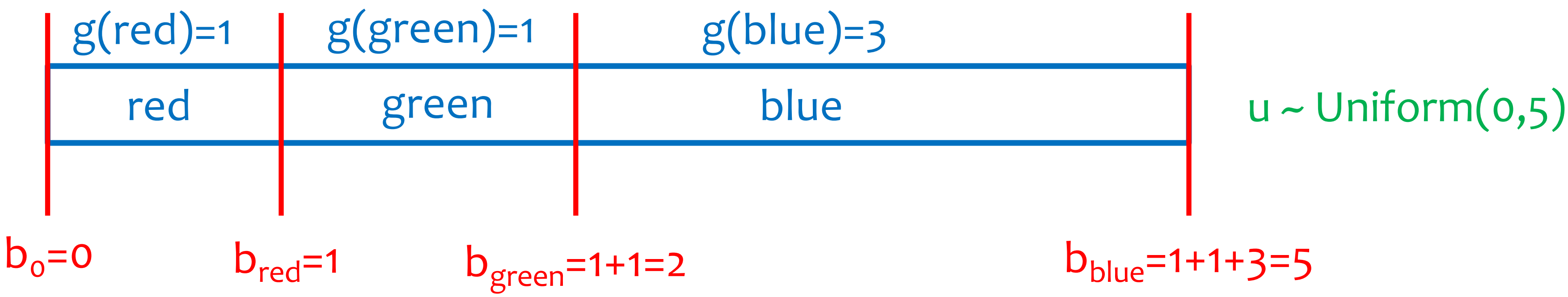
- Gibbs Sampling

# Sampling from a Discrete Distribution

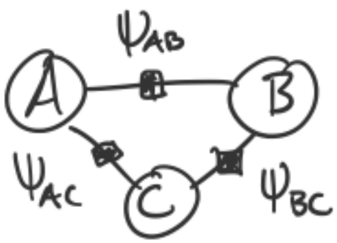
- To sample from a discrete distribution p(y) we only need a function proportional to it e.g., g(·) s.t. p(y) ∝ g(y)
- **Recipe:**
  - Define a bin cutoff $b_y$ for each value y ∈ {1, ..., V}

$$b_y = \sum_{t=1}^{y} g(t), \; \forall y \in \{1, \ldots, V\} \qquad b_0 = 0$$

  - Sample u ~ Uniform(0, $b_V$)
  - Return value y if u lands in bin [$b_{y-1}$, $b_y$,]



# Example: Gibbs Sampling




$A, B, C \in \{+, -\}$

| a | b | $\psi_{AB}(a,b)$ |
|---|---|---|
| + | + | 1 |
| + | − | 2 |
| − | + | 1 |
| − | − | 1 |

| a | c | $\psi_{AC}$ |
|---|---|---|
| + | + | 2 |
| + | − | 2 |
| − | + | 2 |
| − | − | 1 |

| b | c | $\psi_{BC}$ |
|---|---|---|
| + | + | 1 |
| + | − | 1 |
| − | + | 2 |
| − | − | 1 |

full conditionals:

might change at each iteration.

① $p(a \mid b, c) \propto \psi(a,b)\,\psi(a,c)$ → $g(a) =$ [+ | −]

② $p(b \mid a, c) \propto \psi(a,b)\,\psi(b,c)$ → $g(b) =$ [+ | −]

③ $p(c \mid a, b) \propto \psi(a,c)\,\psi(b,c)$ → $g(c) =$ [+ | −]

fixed while sampling

Algo: Initialize $a, b, c$ randomly $\in \{+, -\}$

For $i = 1, 2, 3, \ldots$

$a \sim p(a \mid b, c)$

$b \sim p(b \mid a, c)$

$c \sim p(c \mid a, b)$

# table entries: 2 or 8

$$p(a \mid b, c) = \frac{p(a,b,c)}{p(b,c)} \propto p(a,b,c)$$

# table entries: 8

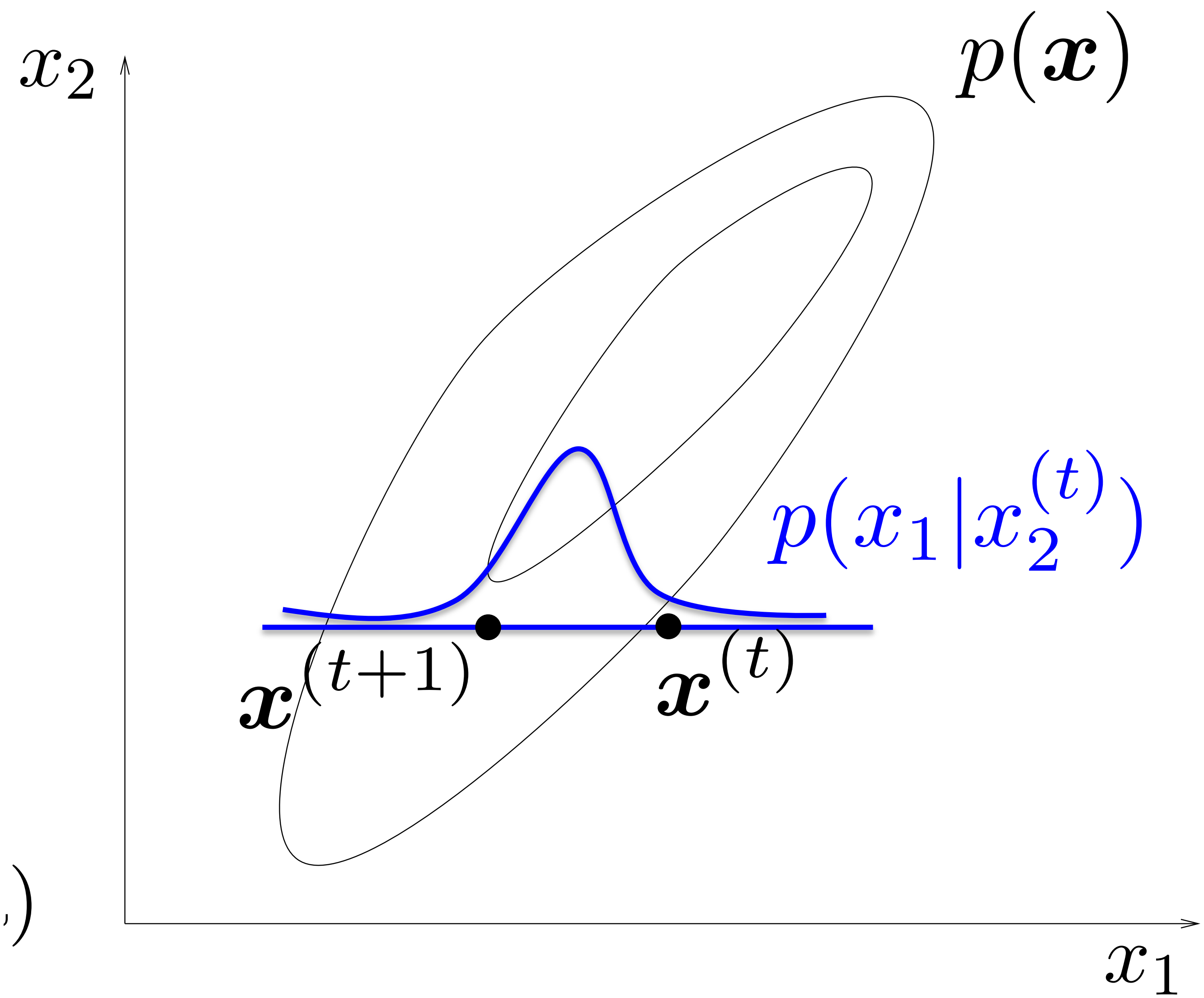
$$p(a,b,c) \triangleq \frac{1}{Z}\,\psi(a,b)\,\psi(a,c)\,\psi(b,c)$$

# Example: Gibbs Sampling

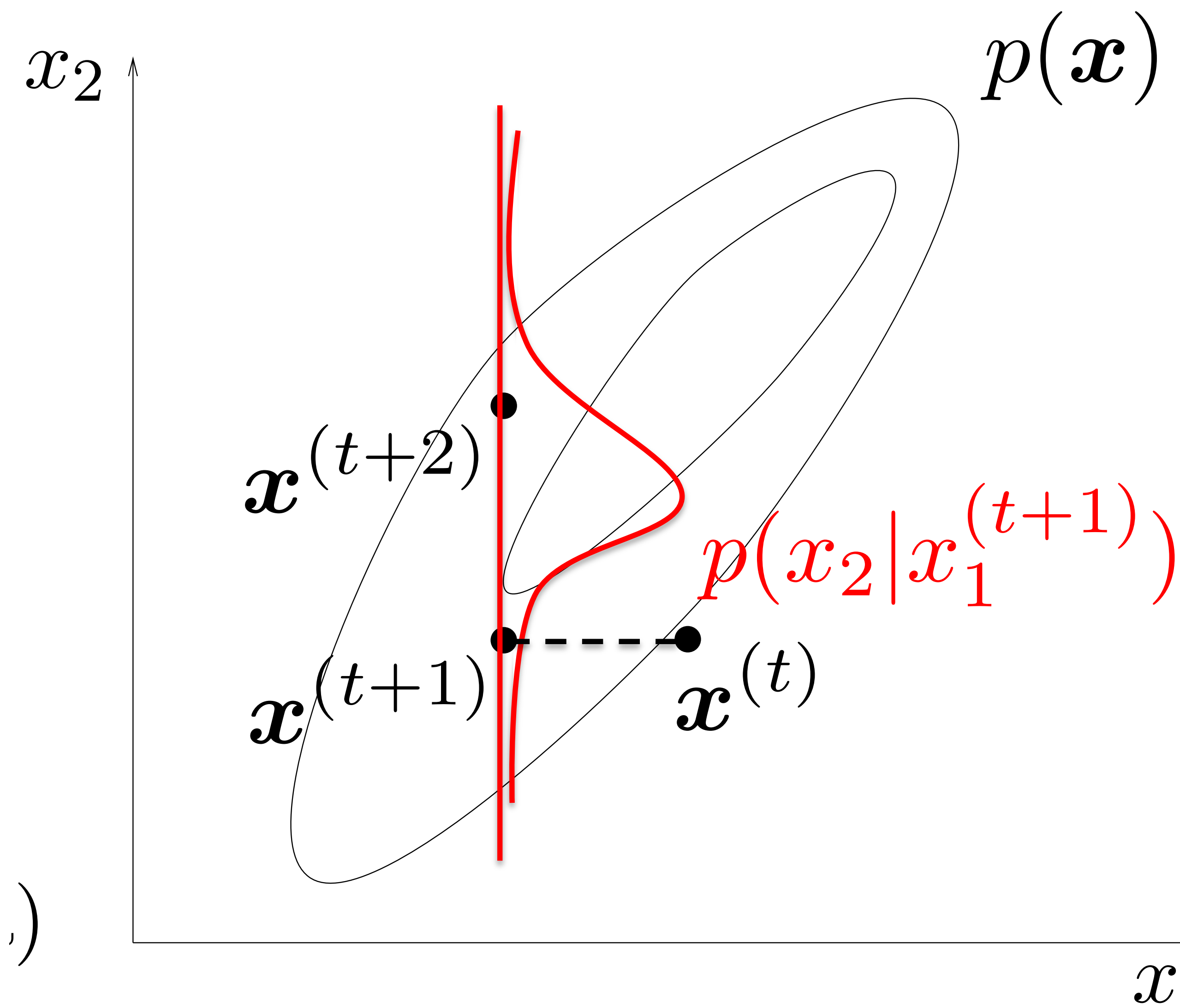
Example: 3-node Factor Graph

```
import numpy as np
import random

def sample01(g0, g1):
    u = random.uniform(0, g0 + g1)
    if u < g0:
        return 0
    else:
        return 1

def gibbs_sampling():
    # Define factor graph
    psi_ab = np.array([[1, 2], [1, 1]])
    psi_ac = np.array([[2, 2], [2, 1]])
    psi_bc = np.array([[1, 1], [2, 1]])

    # Initialize variable values
    a = random.choice([0,1])
    b = random.choice([0,1])
    c = random.choice([0,1])

    counts = np.array([[0, 0], [0, 0], [0, 0]])
    # Gibbs sampling
    for i in range(10):
        a = sample01(psi_ab[0,b] * psi_ac[0,c],
                     psi_ab[1,b] * psi_ac[1,c])
        b = sample01(psi_ab[a,0] * psi_bc[0,c],
                     psi_ab[a,1] * psi_bc[1,c])
        c = sample01(psi_ac[a,0] * psi_bc[b,0],
                     psi_ac[a,1] * psi_bc[b,1])
        print(a, b, c)
        counts[0, a] += 1
        counts[1, b] += 1
        counts[2, c] += 1

    print('p(a = 0) ~= %.2f' % (counts[0,0] / (counts[0,0] + counts[0,1])))
    print('p(b = 0) ~= %.2f' % (counts[1,0] / (counts[1,0] + counts[1,1])))
    print('p(c = 0) ~= %.2f' % (counts[2,0] / (counts[2,0] + counts[2,1])))

if __name__ == '__main__':
    gibbs_sampling()
```

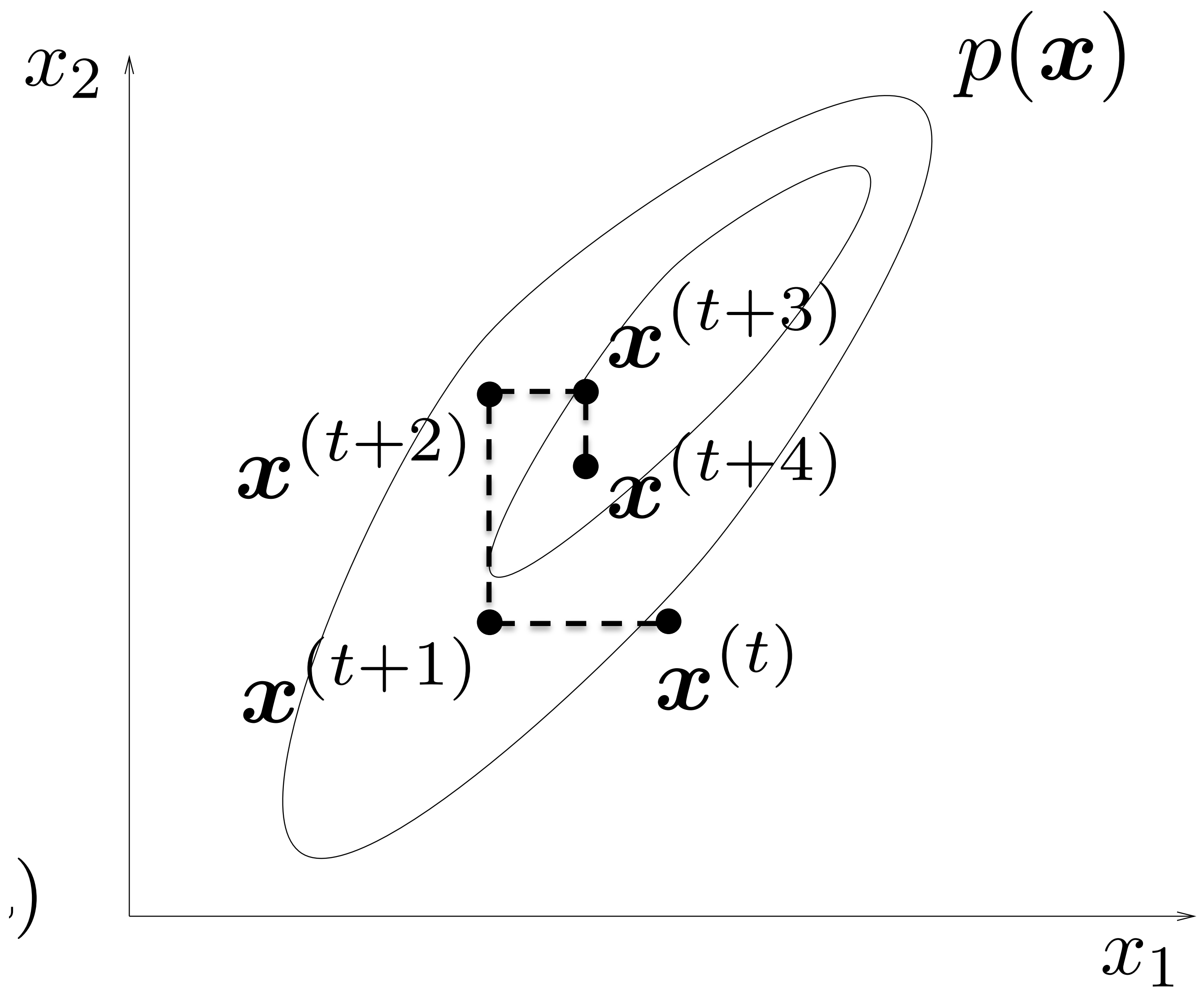
# Gibbs Sampling



# Gibbs Sampling



# Gibbs Sampling



# Gibbs Sampling

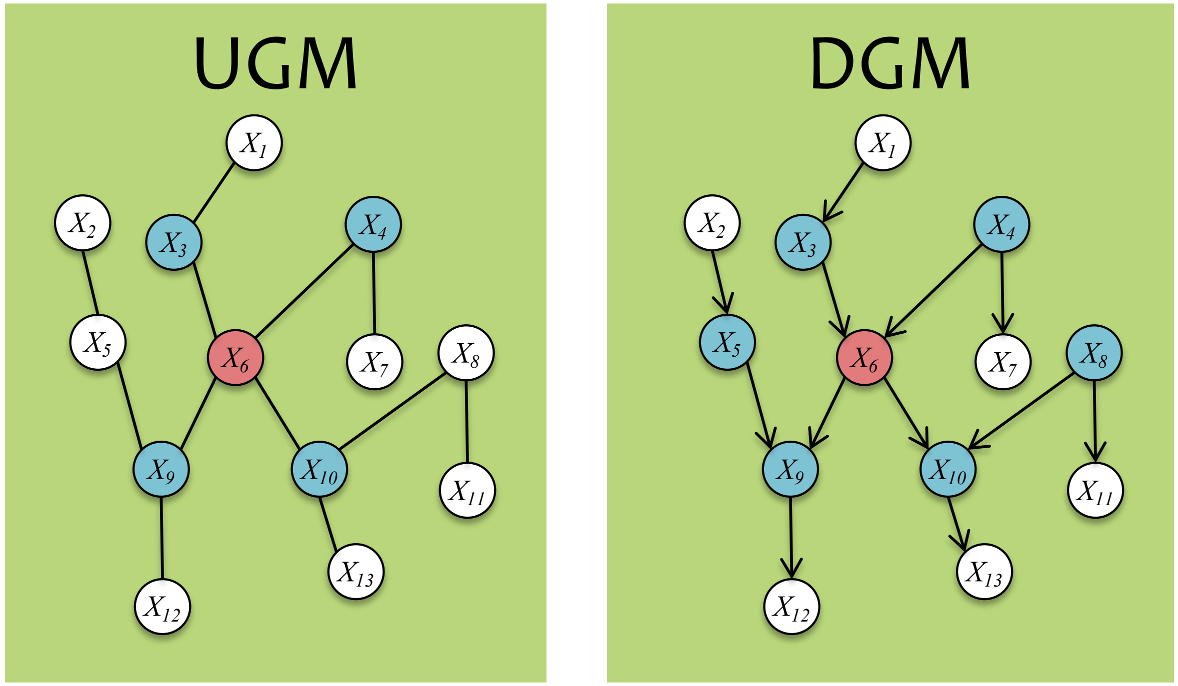
**Question:**
How do we draw samples from a conditional distribution?
$y_1, y_2, \ldots, y_J \sim p(y_1, y_2, \ldots, y_J \mid x_1, x_2, \ldots, x_J)$

**(Approximate) Solution:**
- Initialize $y_1^{(0)}, y_2^{(0)}, \ldots, y_J^{(0)}$ to arbitrary values
- For $t = 1, 2, \ldots$:
  - $y_1^{(t+1)} \sim p(y_1 \mid y_2^{(t)}, \ldots, y_J^{(t)}, x_1, x_2, \ldots, x_J)$
  - $y_2^{(t+1)} \sim p(y_2 \mid y_1^{(t+1)}, y_3^{(t)}, \ldots, y_J^{(t)}, x_1, x_2, \ldots, x_J)$
  - $y_3^{(t+1)} \sim p(y_3 \mid y_1^{(t+1)}, y_2^{(t+1)}, y_4^{(t)}, \ldots, y_J^{(t)}, x_1, x_2, \ldots, x_J)$
  - …
  - $y_J^{(t+1)} \sim p(y_J \mid y_1^{(t+1)}, y_2^{(t+1)}, \ldots, y_{J-1}^{(t+1)}, x_1, x_2, \ldots, x_J)$

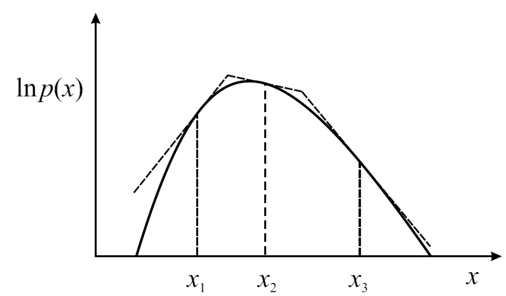
**Properties:**
- This will eventually yield samples from $p(y_1, y_2, \ldots, y_J \mid x_1, x_2, \ldots, x_J)$
- But it might take a long time -- just like other Markov Chain Monte Carlo methods

# Gibbs Sampling

**Full conditionals** only need to condition on the **Markov Blanket**




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



# 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

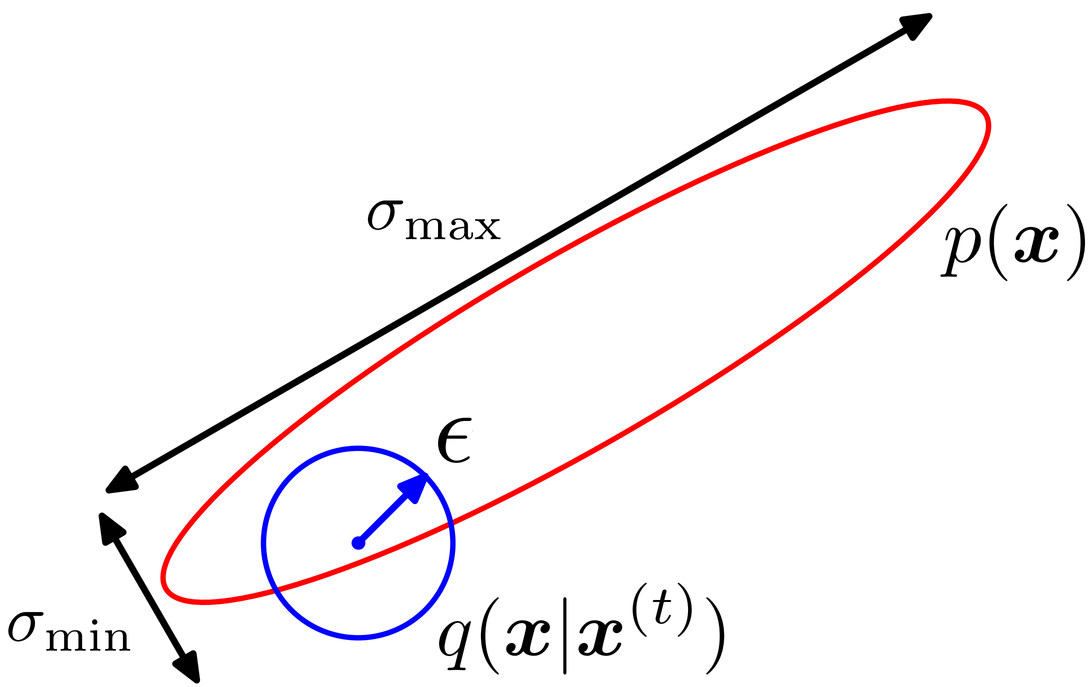
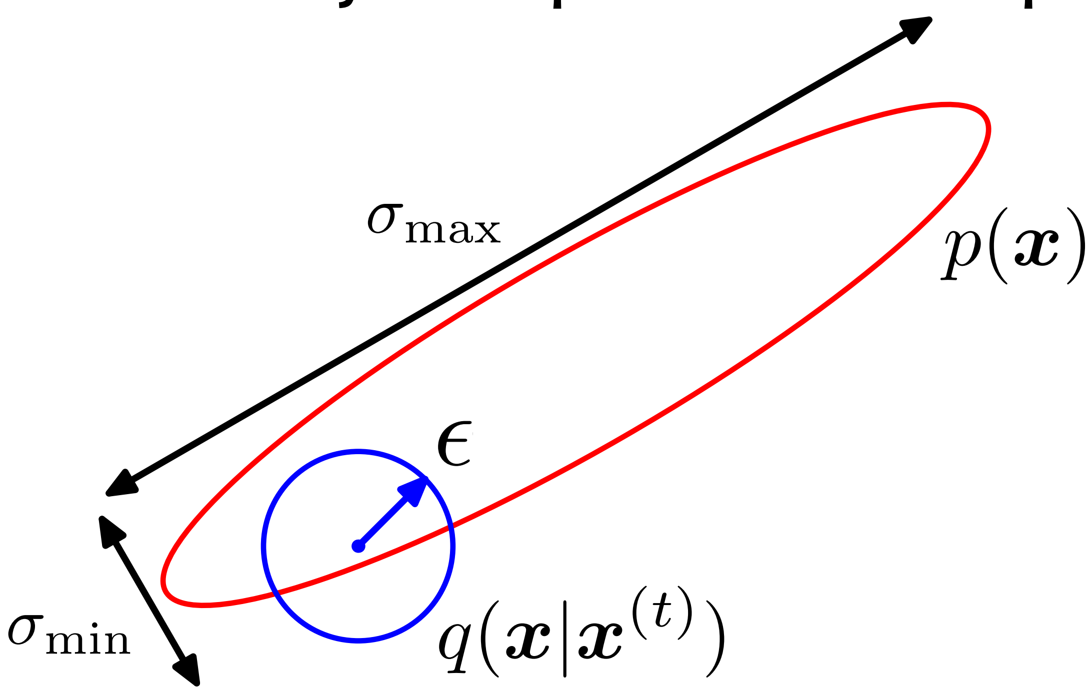


Figure from Bishop (2006)

# 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

# Whiteboard

- Gibbs Sampling as M-H

Definitions and Theoretical Justification for MCMC

# MARKOV CHAINS

# Whiteboard

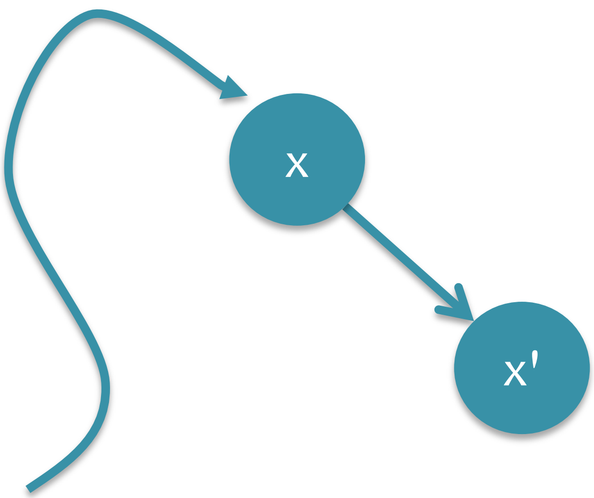
- Markov chains
- Transition probabilities
- Invariant distribution
- Equilibrium distribution
- Sufficient conditions for MCMC
- Markov chain as a WFSM

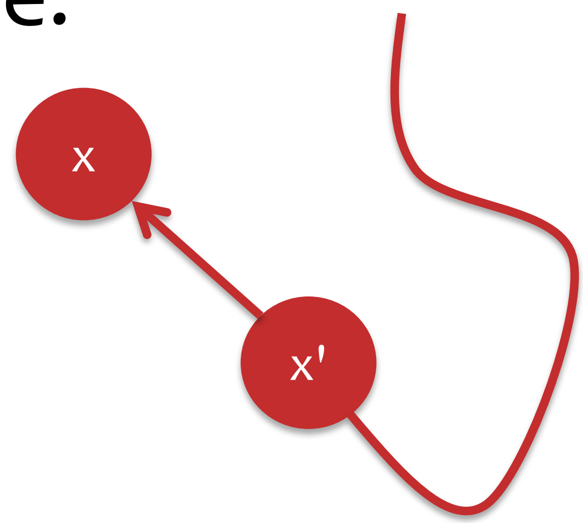
# Detailed Balance

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

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.



# 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?

# Blocked Gibbs Sampling

**Goal:**
Draw samples from a distribution $y_1, y_2, \ldots, y_J \sim p(y_1, y_2, \ldots, y_J)$

**Algorithm:**

- Initialize $y_1, y_2, \ldots, y_J$ to arbitrary values
- For $t = 1, 2, \ldots$ :

  for b in B: where $b \subseteq \{1, \ldots, J\}$

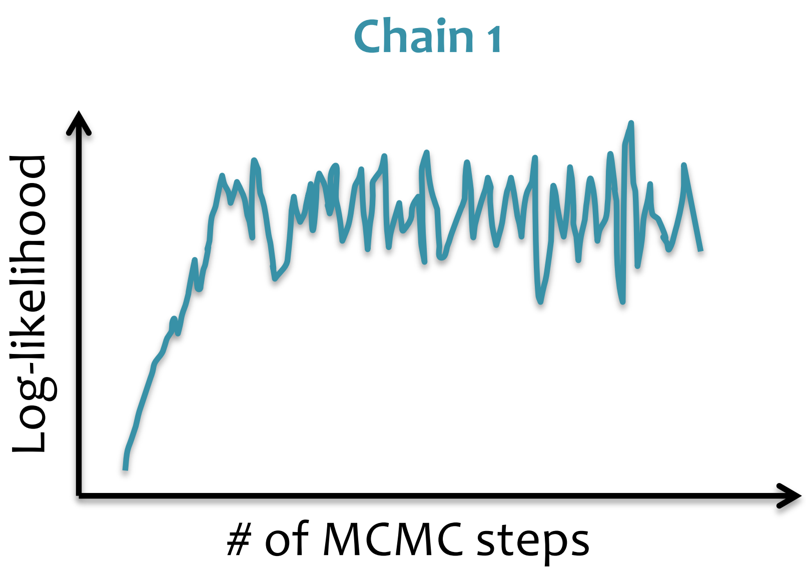
  $y_b \sim p(y_b \mid y_{\neg b})$
- Example: B = set of factors in a factor graph

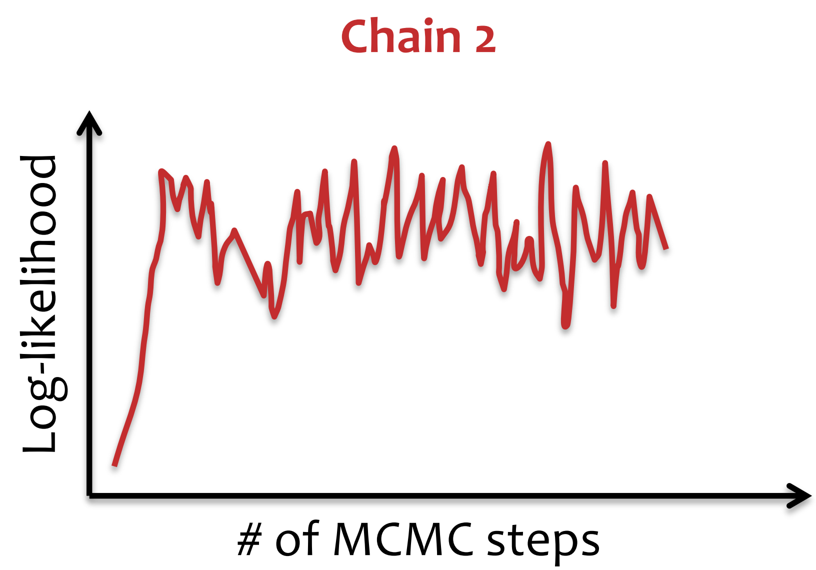
**Why use blocks?**

- As in Gibbs Sampler, this will eventually yield samples from $p(y_1, y_2, \ldots, y_J)$
- **Might improve mixing time** (i.e. "eventually" will be a bit sooner)

# Practical Issues

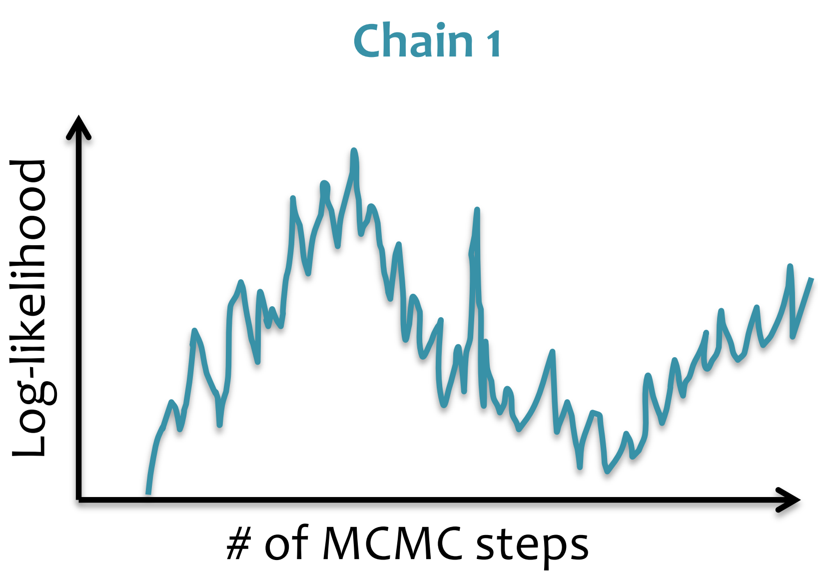
- **Question:** How do we assess convergence of the Markov chain?
- **Answer:** It's not easy!
  - Compare statistics of multiple independent chains
  - Ex: Compare log-likelihoods

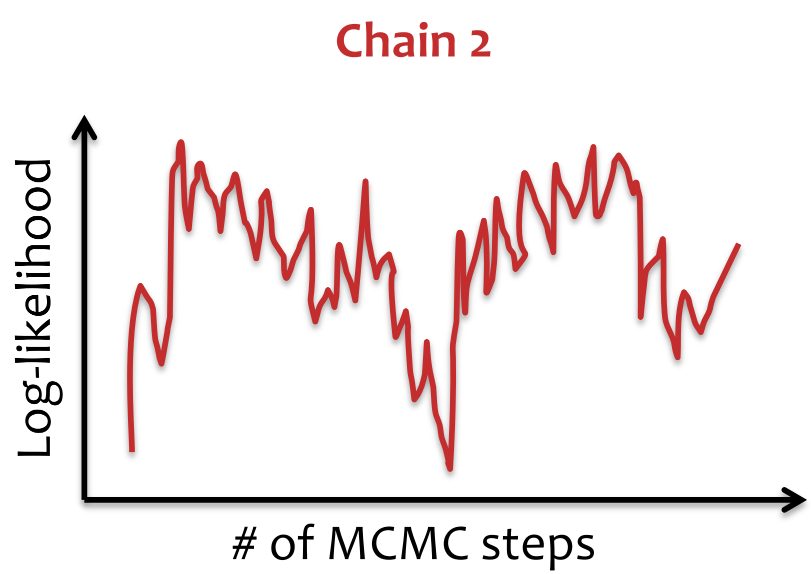


# Practical Issues

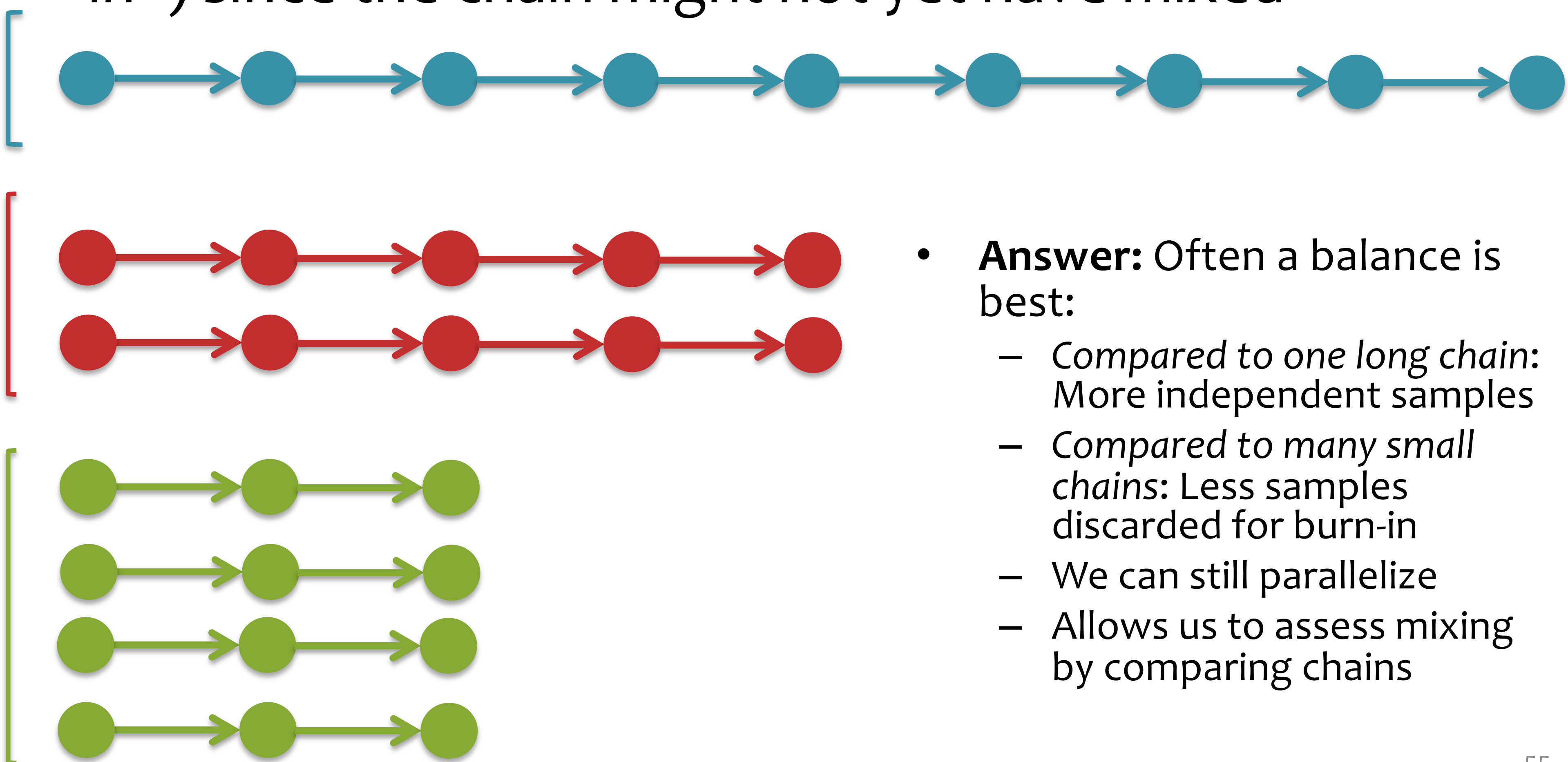
- **Question:** How do we assess convergence of the Markov chain?
- **Answer:** It's not easy!
  – Compare statistics of multiple independent chains
  – Ex: Compare log-likelihoods



# 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

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:**

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

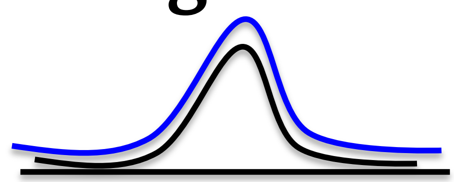
$$\approx \frac{1}{S}\sum_{s=1}^{S} f(x^{(s)}), \quad x, v \sim P(x,v)$$

**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**

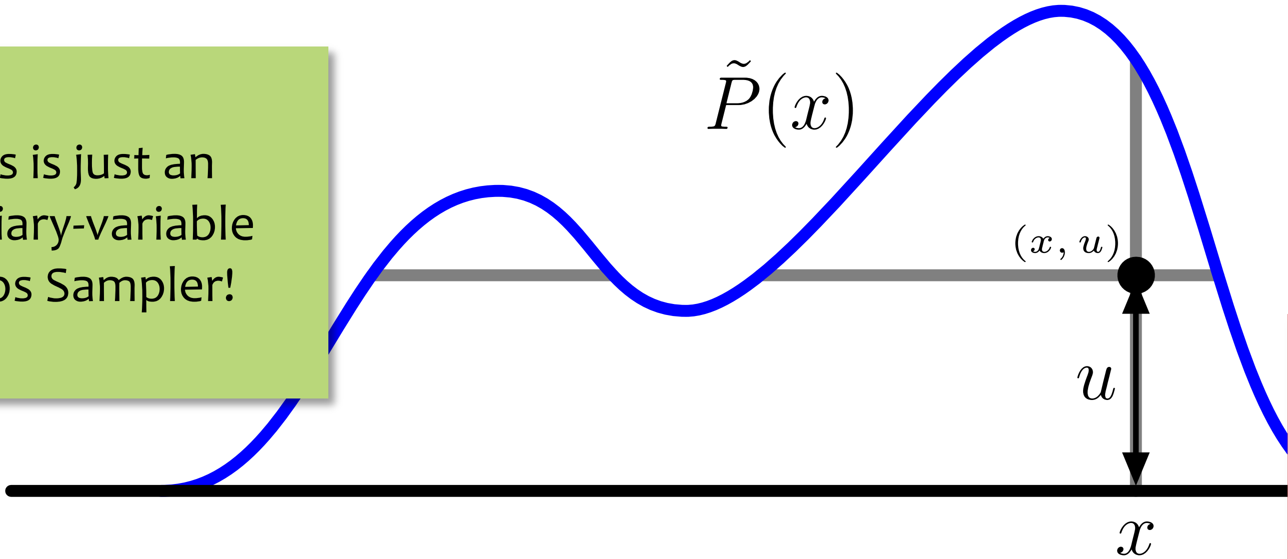
$\tilde{p}(x)$ 

  - 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!



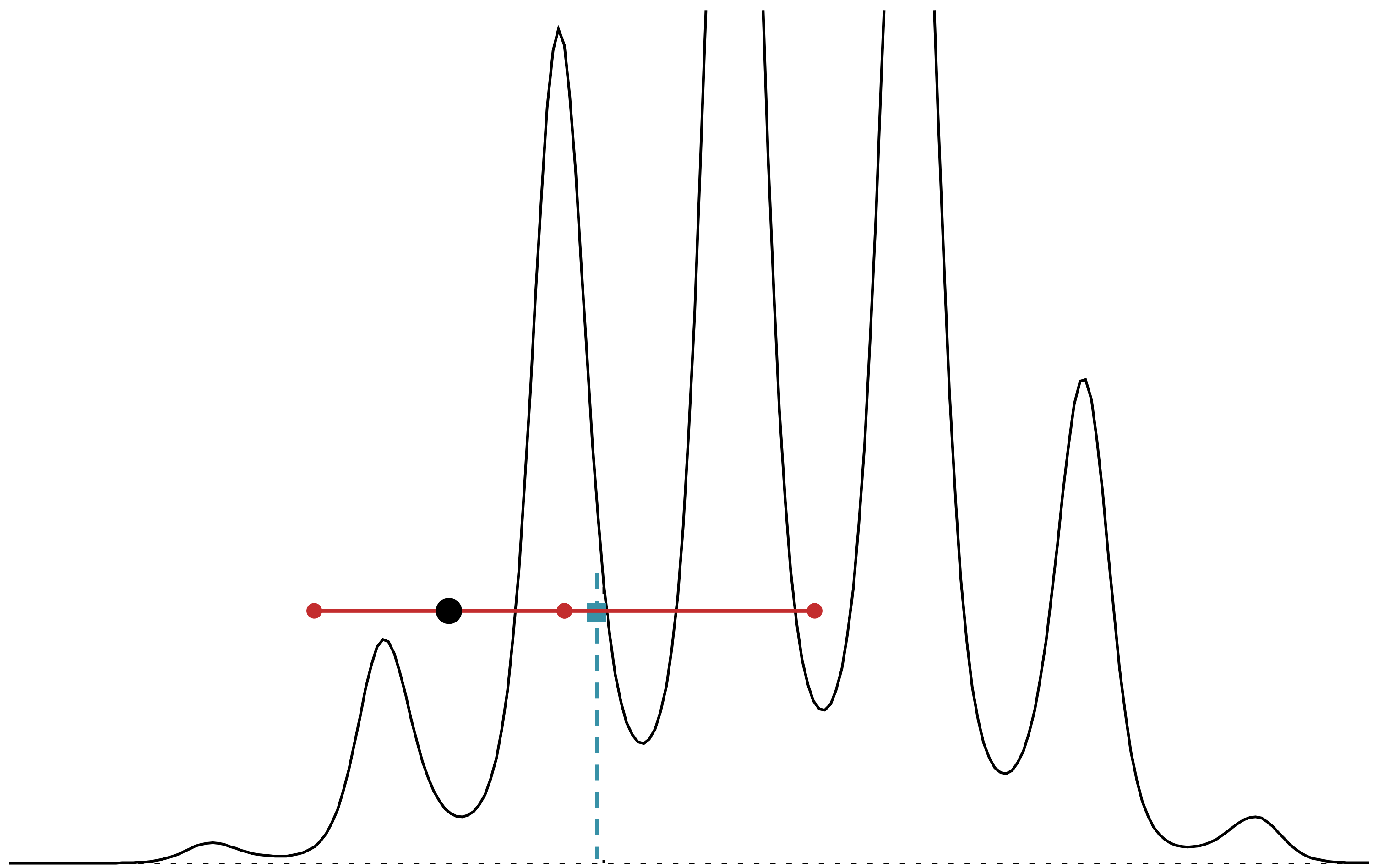

Problem: Sampling from the conditional *p(x | u)* might be infeasible.

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

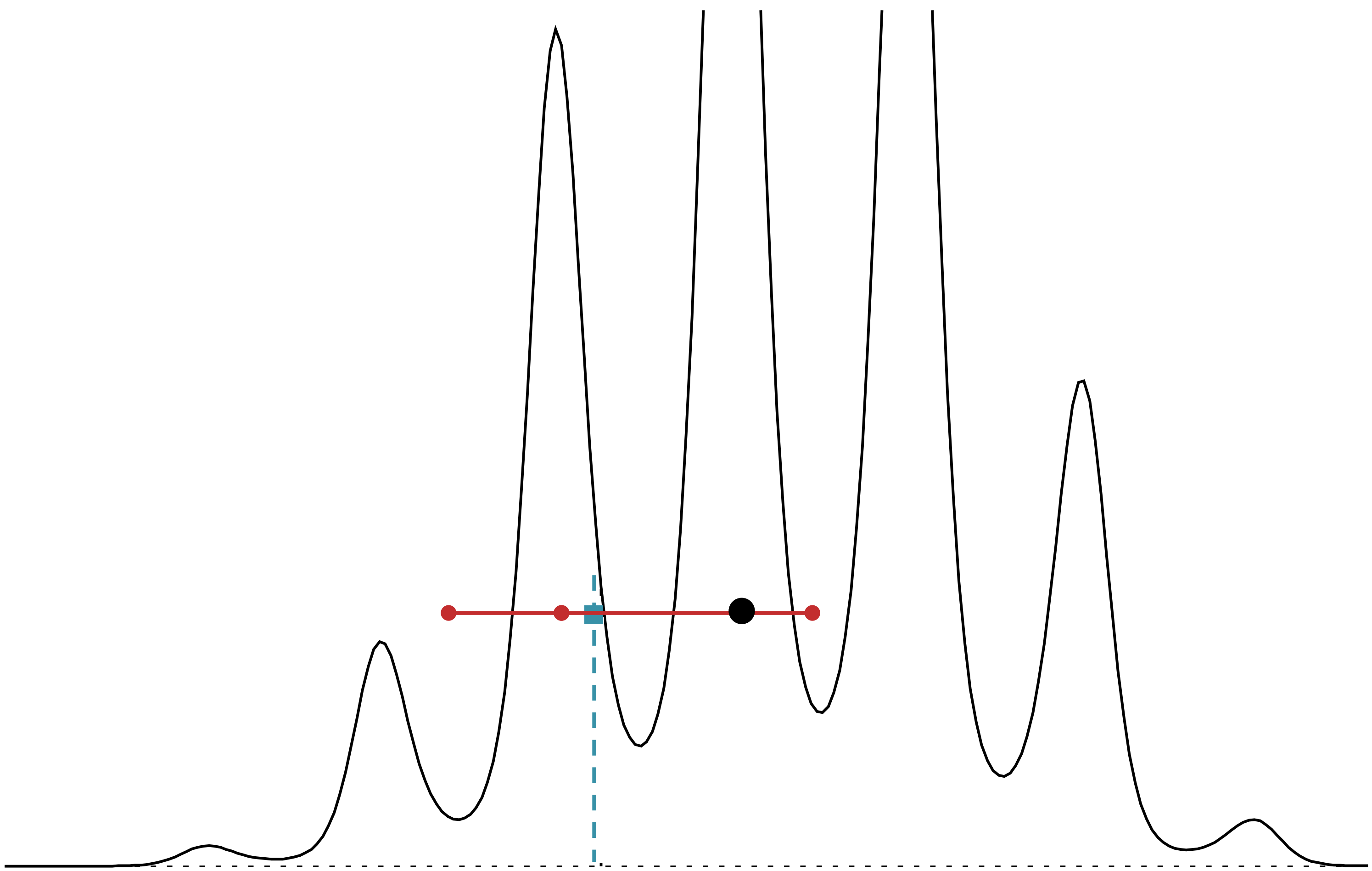
$$p(x|u) \propto \begin{cases} 1 & \tilde{P}(x) \geq u \\ 0 & \text{otherwise} \end{cases} = \text{“Uniform on the slice”}$$

Figure adapted from MacKay Ch. 29

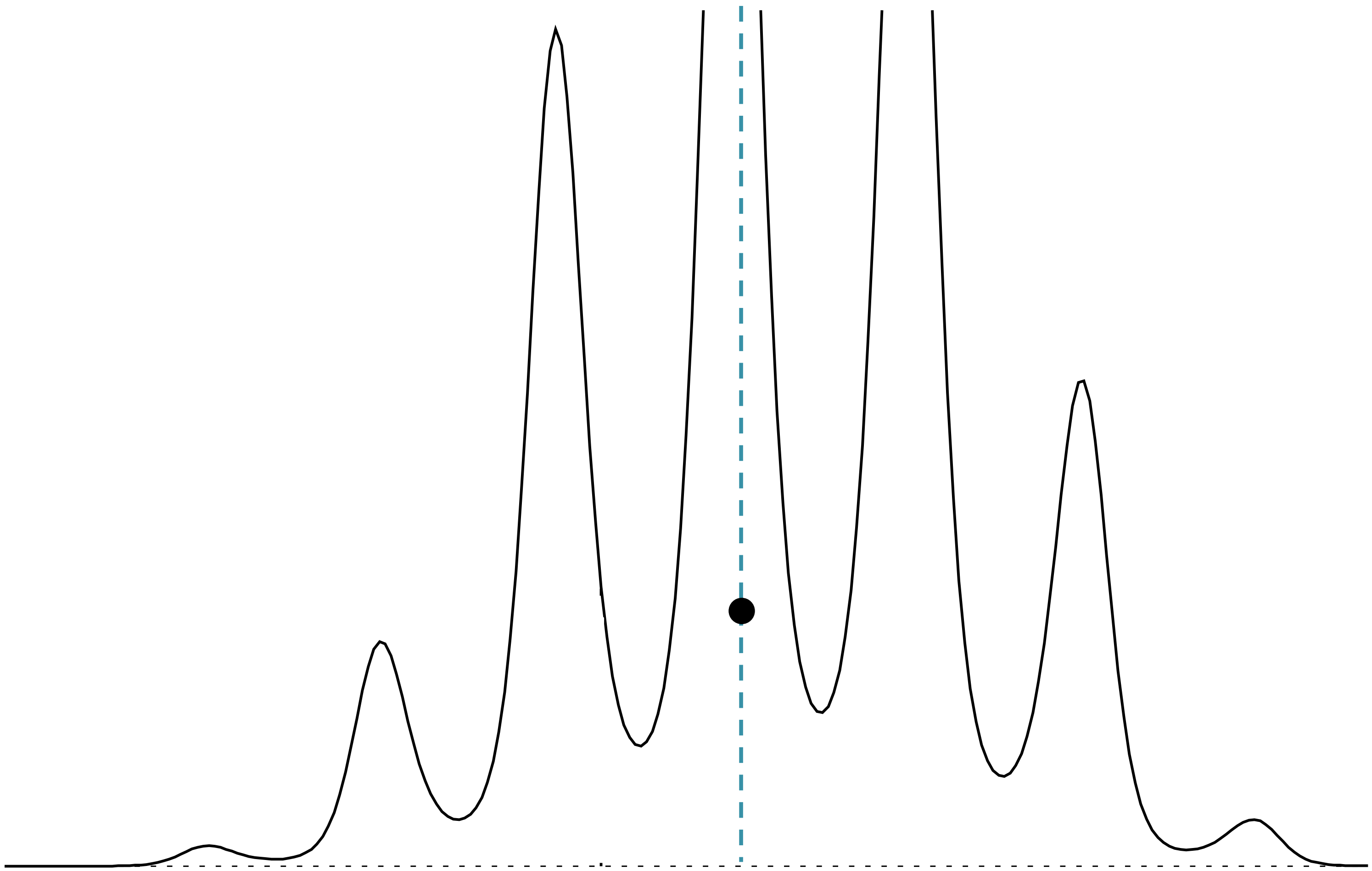
# Slice Sampling



# Slice Sampling



# 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 $(u^{(t)}, x^{(t)})$.

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

**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)

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

**Parameters to tune:**

1. Step size, $\epsilon$
2. Number of iterations, $L$

**Leapfrog Algorithm:**

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

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

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

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

Figure from Neal (2011)

# Background: Hamiltonian Dynamics

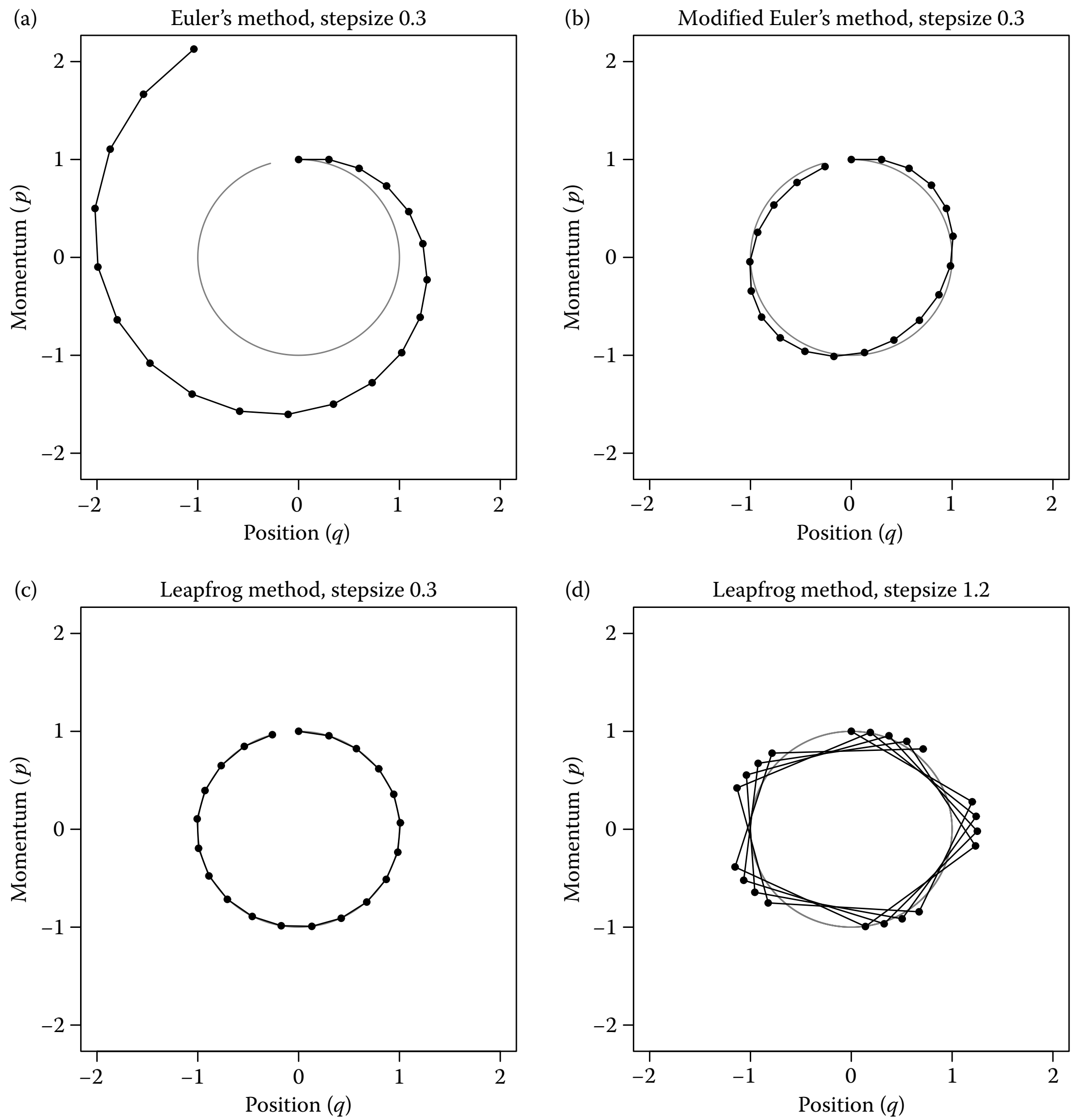


Figure from Neal (2011)

# 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)

# Hamiltonian Monte Carlo

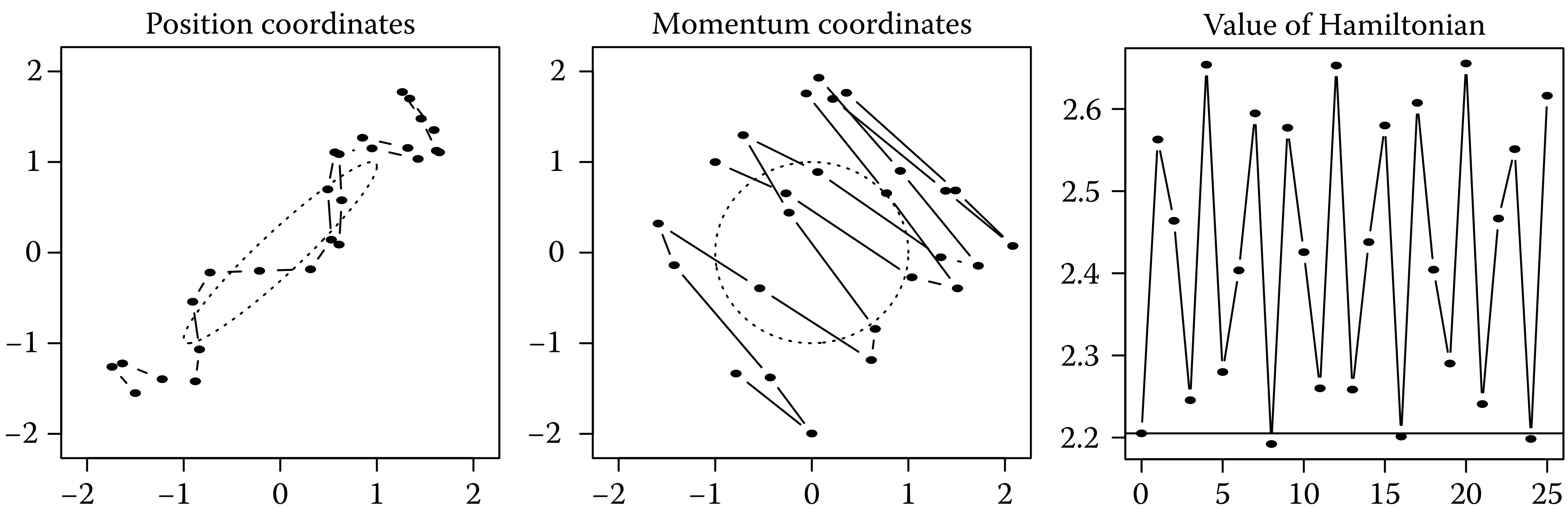
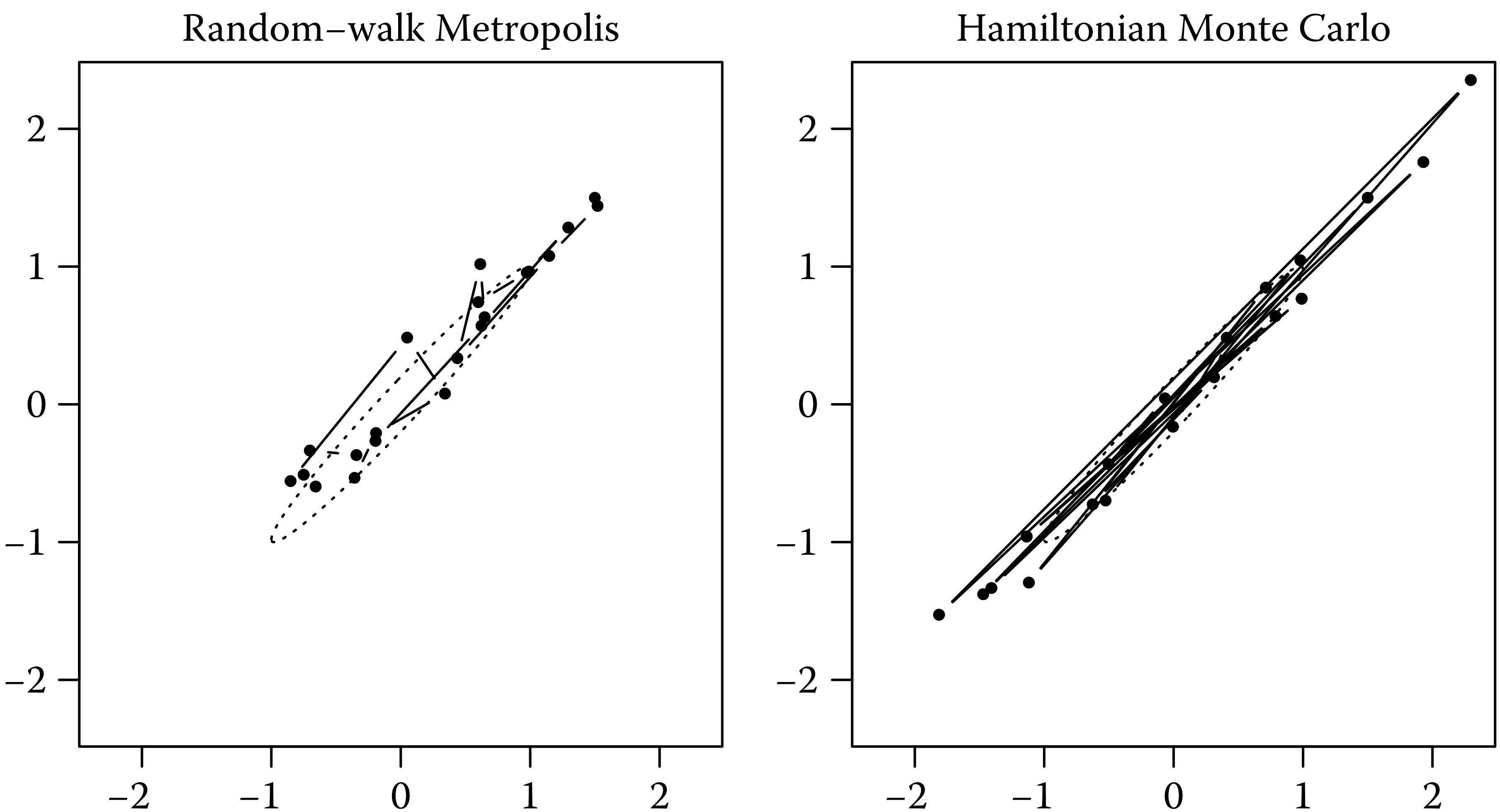


Figure from Neal (2011)

# M-H vs. HMC



# 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