Probability, Approximate Inference, and Sampling

Shichang Zhang

UCLA

Slides adapted from Rob Hall, Eric Xing, Qirong Ho (CMU), Stefano Ermon, Yumeng Zhang (Stanford), and David Sontag (MIT)

Agenda

- Probability Review
- Approximate Inference
  - Monte Carlo and Importance Sampling
  - Markov Chain Monte Carlo (MCMC)
    - Theoretical Aspects of MCMC
  - Gibbs Sampling and Practical MCMC

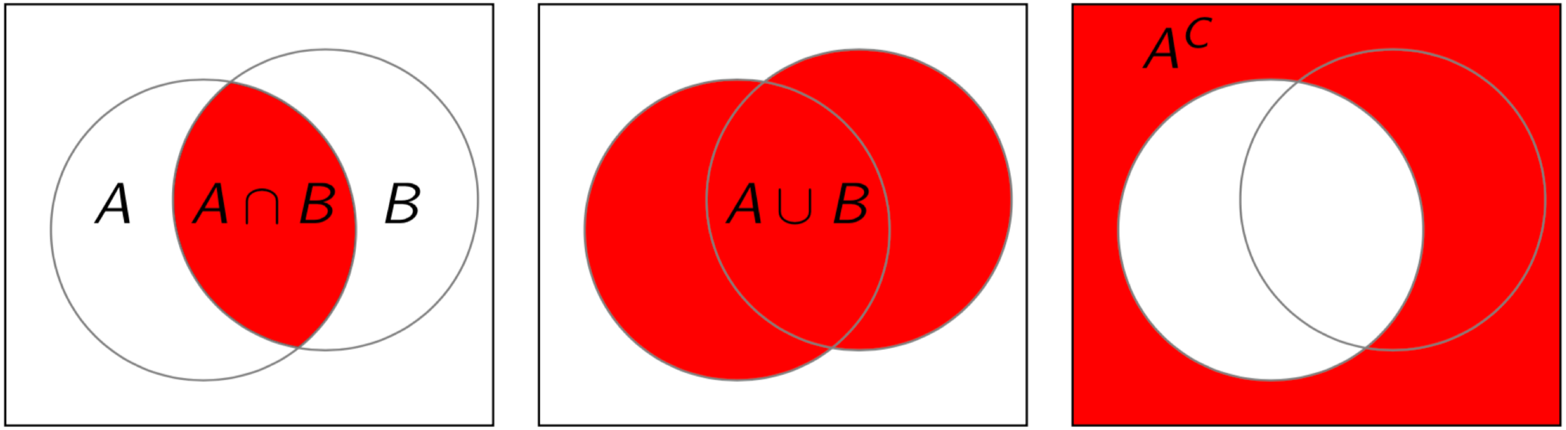
Agenda

- Probability Review
- Approximate Inference
  - Monte Carlo and Importance Sampling
  - Markov Chain Monte Carlo (MCMC)
    - Theoretical Aspects of MCMC
  - Gibbs Sampling and Practical MCMC

Sets

A *set* is just a collection of *elements* denoted e.g., $S = \{s_1, s_2, s_3\}$, $R = \{r : \text{some condition holds on } r\}$.

- **Intersection**: the elements that are in both sets: $A \cap B = \{x : x \in A \text{ and } x \in B\}$
- **Union**: the elements that are in either set, or both: $A \cup B = \{x : x \in A \text{ or } x \in B\}$
- **Complementation**: all the elements that aren't in the set: $A^C = \{x : x \notin A\}$.

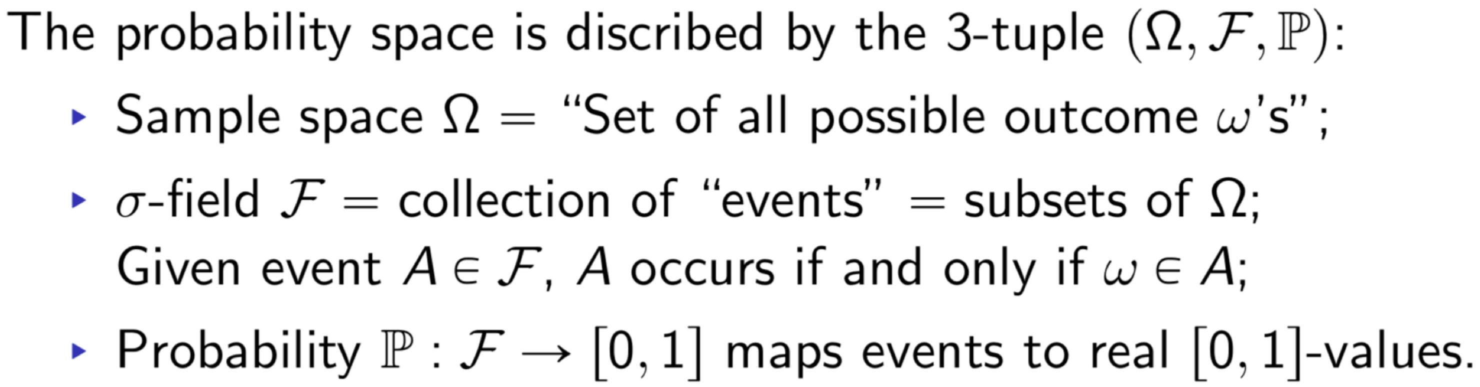


Sets

- A sequence of sets $A_1, A_2 \ldots$ is called **pairwise disjoint** or **mutually exclusive** if for all $i \neq j, A_i \cap A_j = \{\}$.
- If the sequence is pairwise disjoint and $\bigcup_{i=1}^{\infty} A_i = S$, then the sequence forms a **partition** of S.

What is Probability

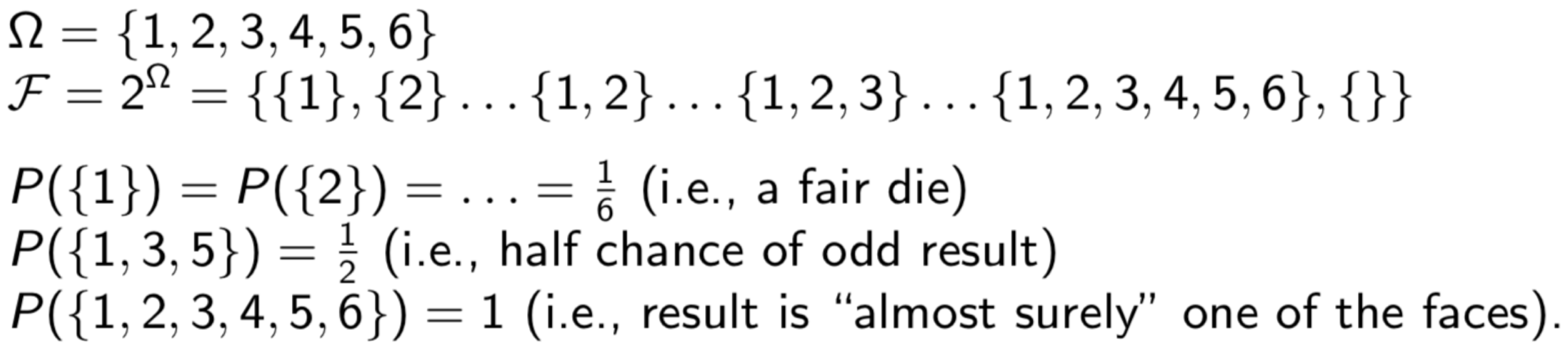
- When we talk about probability, we are actually assuming there is a probability space.



The probability space is discribed by the 3-tuple $(\Omega, \mathcal{F}, \mathbb{P})$:

- Sample space $\Omega =$ "Set of all possible outcome ω's";
- σ-field $\mathcal{F} =$ collection of "events" $=$ subsets of Ω;
  Given event $A \in \mathcal{F}$, A occurs if and only if $\omega \in A$;
- Probability $\mathbb{P} : \mathcal{F} \rightarrow [0, 1]$ maps events to real $[0, 1]$-values.

- Example of rolling a die



$\Omega = \{1, 2, 3, 4, 5, 6\}$
$\mathcal{F} = 2^{\Omega} = \{\{1\}, \{2\} \ldots \{1, 2\} \ldots \{1, 2, 3\} \ldots \{1, 2, 3, 4, 5, 6\}, \{\}\}$

$P(\{1\}) = P(\{2\}) = \ldots = \frac{1}{6}$ (i.e., a fair die)
$P(\{1, 3, 5\}) = \frac{1}{2}$ (i.e., half chance of odd result)
$P(\{1, 2, 3, 4, 5, 6\}) = 1$ (i.e., result is "almost surely" one of the faces).

Axioms of Probability

- Three axioms and corresponding

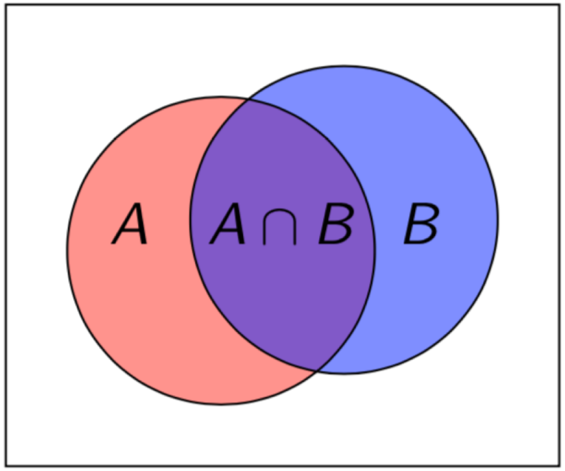
A set of conditions imposed on probability measures (due to Kolmogorov)

- $P(A) \geq 0, \forall A \in \mathcal{F}$
- $P(\Omega) = 1$
- $P(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i)$ where $\{A_i\}_{i=1}^{\infty} \in \mathcal{F}$ are pairwise disjoint.

These quickly lead to:

- $P(A^C) = 1 - P(A)$ (since $P(A) + P(A^C) = P(A \cup A^C) = P(\Omega) = 1$).
- $P(A) \leq 1$ (since $P(A^C) \geq 0$).
- $P(\{\}) = 0$ (since $P(\Omega) = 1$).

Conditional Probabilities



For events $A, B \in \mathcal{F}$ with $P(B) > 0$, we may write the **conditional probability of A given B**:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Interpretation: the outcome is definitely in B, so treat B as the entire sample space and find the probability that the outcome is also in A.

Independence

Two events A, B are called **independent** if $P(A \cap B) = P(A)P(B)$.

When $P(A) > 0$ this may be written $P(B|A) = P(B)$ (why?)

Two events A, B are called **conditionally independent given** C when $P(A \cap B|C) = P(A|C)P(B|C)$.

When $P(A) > 0$ we may write $P(B|A, C) = P(B|C)$

The difference is important. Later, we will need this to understand the Markov Chain.

Bayes' Rule

Using the chain rule we may see:

$$P(A|B)P(B) = P(A \cap B) = P(B|A)P(A)$$

Rearranging this yields **Bayes' rule**:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

Often this is written as:

$$P(B_i|A) = \frac{P(A|B_i)P(B_i)}{\sum_i P(A|B_i)P(B_i)}$$

Where B_i are a partition of Ω (note the bottom is just the law of total probability).

Random Variables

- A random variable X is just a function: $X : \Omega \longrightarrow \mathbb{R}^d$

Intuitively, a random variable is a variable that takes on its values by chance. (Usually denoted by capital letters $X, Y, Z \dots$)

Discrete: Can be described by the **probability mass function** $\mathbb{P}(X = x_i) = p_i$ for $i = 1, 2, \dots$
e.g. Bernoulli, Binomial, Geometric, Poisson, etc.

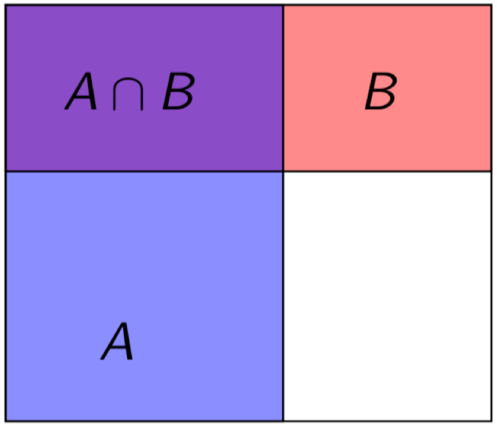
Continuous: Can be described by the **probability density function** $\mathbb{P}(a \leqslant X \leqslant b) = \int_a^b f(x)dx$.
e.g. Exponential, Normal, Beta, etc.

Singular: Can not be described by either. Not useful.

Nonetheless, a random variable can always be determined by its **cumulative distribution function** $F(x) = \mathbb{P}(X \leqslant x)$.

Joint Distributions

We may consider multiple functions of the same sample space, e.g., $X(\omega) = 1_A(\omega)$, $Y(\omega) = 1_B(\omega)$:



May represent the **joint distribution** as a table:

	X=0	X=1
Y=0	0.25	0.15
Y=1	0.35	0.25

We write the joint PMF or PDF as $f_{X,Y}(x, y)$

Independent Distributions

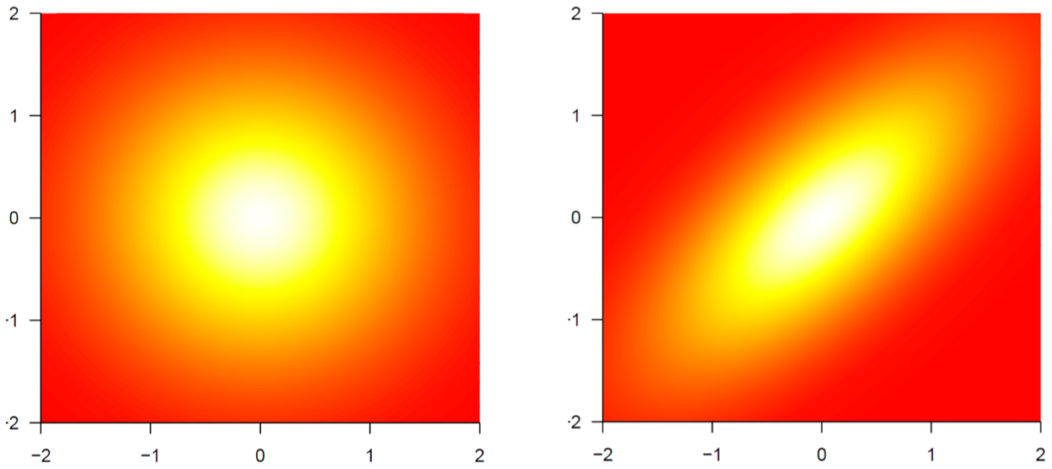
- We talked about independent events. Now we can extend the same idea to random variables

Two random variables are called **independent** when the joint PDF factorizes:

$$f_{X,Y}(x,y) = f_X(x)f_Y(y)$$

When RVs are independent and identically distributed this is usually abbreviated to “i.i.d.”

Relationship to independent events: X, Y ind. iff $\{\omega : X(\omega) \leq x\}, \{\omega : Y(\omega) \leq y\}$ are independent events for all x, y.



Marginalizing and Conditioning

- Given a joint distribution of more than one random variable, we can find the distribution of one random variable

$$P(X = x) = \sum_{y} P(X = x, Y = y) = \sum_{y} P(X = x | Y = y) P(Y = y)$$

- We can also find the distribution of one random variable conditioning on the other random variable

$$P(X = x | Y = y) = \frac{P(X = x, Y = y)}{P(Y = y)} = \frac{\text{joint pmf}}{\text{marginal pmf}}$$

Expectation and Variance

We may consider the **expectation** (or "mean") of a distribution:

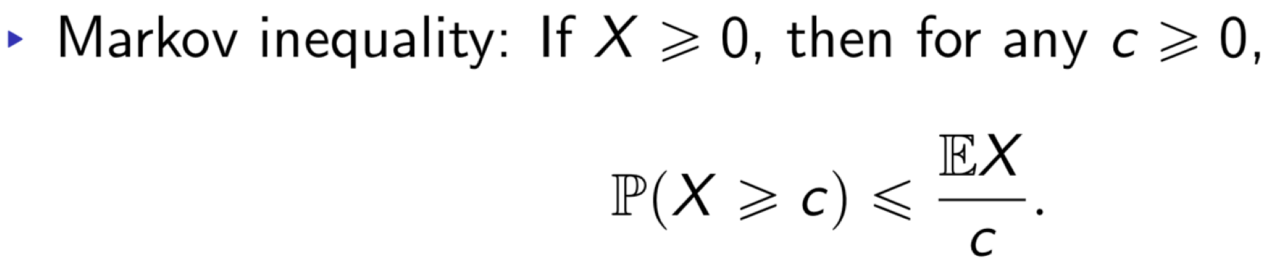
$$E(X) = \begin{cases} \sum_x x f_X(x) & \text{X is discrete} \\ \int_{-\infty}^{\infty} x f_X(x)\ dx & \text{X is continuous} \end{cases}$$

We may consider the **variance** of a distribution:

$$\mathrm{Var}(X) = E(X - EX)^2$$

This may give an idea of how "spread out" a distribution is.

Markov Inequality



- Markov inequality: If $X \geqslant 0$, then for any $c \geqslant 0$,

$$\mathbb{P}(X \geqslant c) \leqslant \frac{\mathbb{E}X}{c}.$$

- This inequality is telling us a random variable can’t be too different from its mean. Note: we know nothing about the distribution of X!

Law of Large Numbers (LLN)

- LLN describes the asymptotic behavior of the sample mean.

Recall our variable $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$.
We may wonder about its behavior as $n \to \infty$.

We had: $E\bar{X}_n = \mu, \mathrm{Var}(\bar{X}_n) = \frac{\sigma^2}{n}$.

Distribution appears to be "contracting:" as n increases, variance is going to 0.

The **weak law of large numbers**:

$$\lim_{n\to\infty} P(|\bar{X}_n - \mu| < \epsilon) = 1$$

In English: choose ϵ and a probability that $|\bar{X}_n - \mu| < \epsilon$, I can find you an n so your probability is achieved.

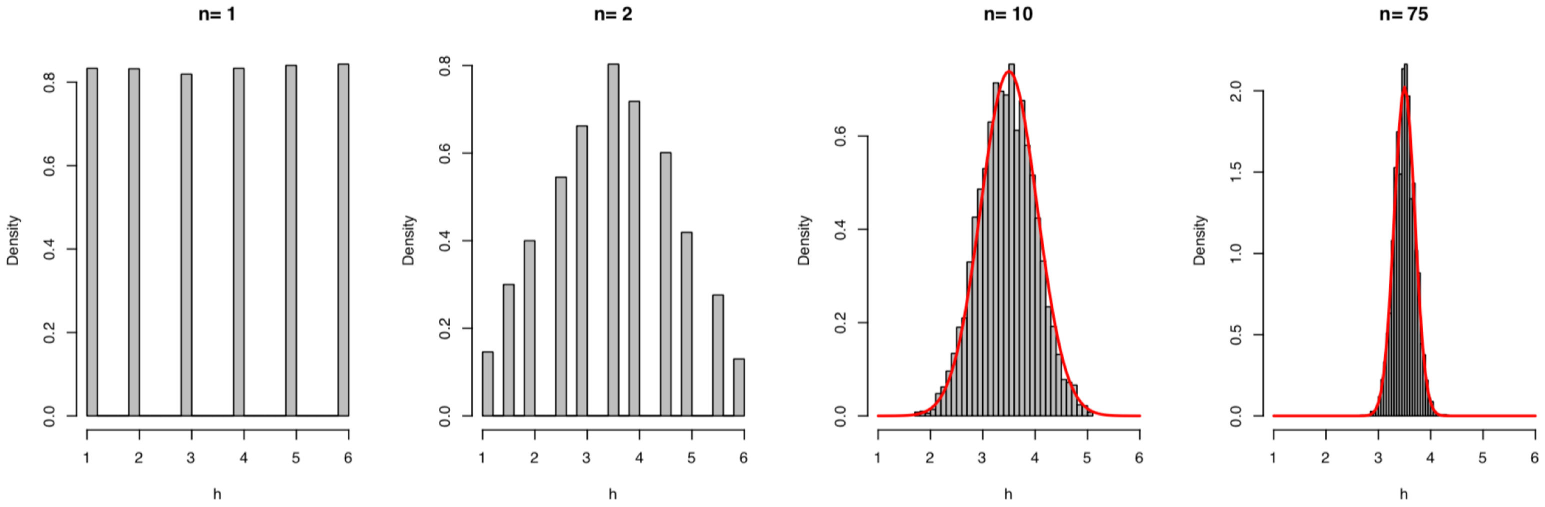
Central Limit Theorem (CLT)

- Similarly to LLN, CLT also describes the asymptotic behavior of the sample mean.

The distribution of $\bar{X}_n$ also converges weakly to a Gaussian,

$$\lim_{n\to\infty} \frac{\overline{X}_n - \mu}{\sigma/\sqrt{n}} \sim \mathcal{N}(0,1)$$

Simulated n dice rolls and took average, 5000 times:



LLN v.s. CLT

- How are these two different?

  Recall our variable $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ $\quad E\bar{X}_n = \mu, \mathrm{Var}(\bar{X}_n) = \frac{\sigma^2}{n}$

- As n goes to infinity
  - LLN: $P(|\bar{X}_n - \mu| < \epsilon) = 1$
  - CLT: $\frac{\overline{X}_n - \mu}{\sigma/\sqrt{n}} \sim \mathcal{N}(0, 1)$

- The converges rates are different!
- Another way to understand it is that we standardized the random variable first before taking n to infinity.

Markov Chains

- We will see it when we get to the MCMC part later.

Agenda

- Probability Review
- Approximate Inference ⇦
    - Monte Carlo and Importance Sampling
    - Markov Chain Monte Carlo (MCMC)
        - Theoretical Aspects of MCMC
    - Gibbs Sampling and Practical MCMC

Probabilistic Inference

- Many tasks actually boil down to inference tasks, and we can further reduce them to answering probability queries.
  - The notation we will use through out this talk
    - some random variables X, some evidence variables E (variables we have observed), all the unobserved variables Z = X – E.
  - Some questions we can ask
    - Marginal probability: what is P(E=e)?
    - Conditional/Posterior probability: what is P(X_i=x | E=e)?
- Examples:
  - All the classification problems can fit in to this framework, e.g. node classification on graphs, P(label of X | labels of neighbours of X)?
  - Language model: P(X3="mathematics" | X1="I", X2="like")?

Why Approximate Inference?

- For real world problems with many random variable, doing exact inference is computationally intractable.
- Approximation is useful:
  - Suppose the ground truth is P(Z=z | E=e)=0.29292, and the approximate inference yields P(Z=z | E=e) = 0.3. This might be good enough for many applications.

Approximate Inference

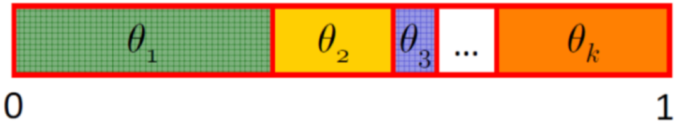
- Two main families of approximate inference algorithms:
  - Variational algorithms
  - Monte-Carlo sampling methods
- The basic idea of sampling method is to approximate a probability distribution using a small number of states that are “representative” of the entire probability distribution

Agenda

- Probability Review
- Approximate Inference
  - Monte Carlo and Importance Sampling
  - Markov Chain Monte Carlo (MCMC)
    - Theoretical Aspects of MCMC
  - Gibbs Sampling and Practical MCMC

How to generate a sample?

- Given a set of variables $X = \{X_1, \ldots, X_n\}$, a sample $x = (x_1, \ldots, x_n)$ is an an assignment to all variables (also called an instantiation or a state)
- How to randomly generate a sample/state according to probabilities assigned by $P(x)$?
- Algorithm to draw a sample from a *univariate* distribution $P(X)$. A sample is just an assignment to X. Domain of $X = \{a^0, \ldots, a^{k-1}\}$
  1. Divide a real line $[0, 1]$ into k intervals such that the width θ_j of the j-th interval is equal to $P(X = a^j)$
  2. Draw a random number $r \in [0, 1]$
  3. Determine the region j in which r lies. Output a^j



Monte Carlo Estimation

1. **Express the quantity of interest as the expected value of a random variable.**

$$E_{x\sim P}[g(x)] = \sum_{x} g(x)P(x)$$

2. Generate T samples $\mathbf{x}^1, \ldots, \mathbf{x}^T$ from the distribution P with respect to which the expectation was taken.
3. Estimate the expected value from the samples using:

$$\hat{g}(\mathbf{x}^1, \cdots, \mathbf{x}^T) \triangleq \frac{1}{T}\sum_{t=1}^{T} g(\mathbf{x}^t)$$

where $\mathbf{x}^1, \ldots, \mathbf{x}^T$ are independent samples from P. Note: $\hat{g}$ is a random variable. Why?

Properties of the Monte Carlo

- **Unbiased:**

$$E_P[\hat{g}] = E_P[g(x)]$$

- **Convergence:** By law of large numbers

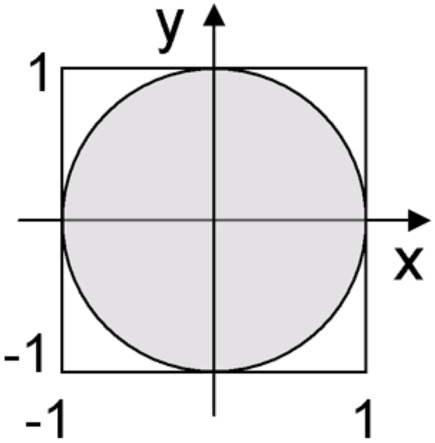
$$\hat{g} = \frac{1}{T}\sum_{t=1}^{T} g(x^t) \rightarrow E_P[g(x)] \text{ for } T \rightarrow \infty$$

- **Variance:**

$$V_P[\hat{g}] = V_P\left[\frac{1}{T}\sum_{t=1}^{T} g(x^t)\right] = \frac{V_P[g(x)]}{T}$$

Thus, variance of the estimator can be reduced by increasing the number of samples. We have no control over the numerator when P is given. How quickly does the estimate converge to the true expectation?

Rejection Sampling



- Suppose you want to sample points uniformly within the circle
- You have access to a uniform random generator in $[-1, 1]$
- Sample $x \sim \mathcal{U}[-1, 1]$
- Sample $y \sim \mathcal{U}[-1, 1]$
- If $x^2 + y^2 \leq 1$, accept the sample. Otherwise reject it and try again.

Rejection Sampling

- Express $P(E=e)$ as an expectation:

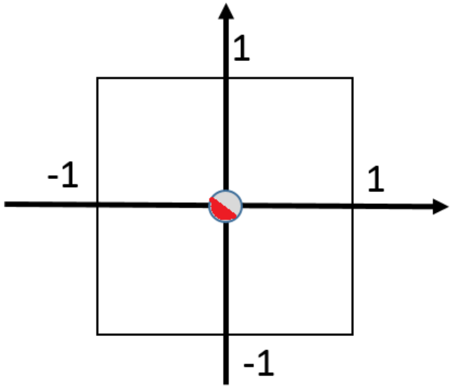
$$P(E=e) \quad = \quad \sum_x \delta_e(x)P(x) = E_P[\delta_e(x)]$$

  where $\delta_e(x)$ is an indicator function which is 1 if x is consistent with the evidence $E=e$ and 0 otherwise.
- Generate samples
- Monte Carlo estimate $\hat{g}(x_1,\cdots,x_T) = \frac{1}{T}\sum_{t=1}^{T} g(x^t)$:

$$\hat{P}(E=e) = \frac{\text{Number of samples that have } E=e}{\text{Total number of samples}}$$

- Issues: If $P(E=e)$ is very small (e.g., 10^{-55}), nearly all samples will be rejected.
- Note: even if $P(E=e)$ is extremely small, $p(X=x \mid E=e) = p(X=x, E=e)/p(E=e)$ can be large.

Failure Case



- Suppose you want to sample points uniformly within the circle
- You have access to a uniform random generator in $[-1, 1]$
- Sample $x \sim \mathcal{U}[-1, 1]$, sample $y \sim \mathcal{U}[-1, 1]$
- If (x, y) is in the circle, accept the sample. Otherwise reject it and try again.
- Can be extremely inefficient if the circle is small
- A conditional probability is like the ratio between the red vs. gray circle areas. Can we sample directly inside the gray circle?

Importance Sampling

- Idea: evidence variables are fixed, so let's just sample over non-evidence ones
- Idea: use a **proposal distribution** over non-evidence variables $Q(Z = X \setminus E)$ that we **can efficiently sample from** and such that $P(Z = z, E = e) > 0 \Rightarrow Q(Z = z) > 0$. Express $P(E = e)$ as follows:

$$\begin{aligned} P(E = e) &= \sum_z P(Z = z, E = e) \\ &= \sum_z P(Z = z, E = e) \frac{Q(Z = z)}{Q(Z = z)} \\ &= E_Q \left[\frac{P(Z = z, E = e)}{Q(Z = z)} \right] = E_Q[w(z)] \end{aligned}$$

- Generate samples from Q and estimate $P(E = e)$ using the following Monte Carlo estimate:

$$\hat{P}(E = e) = \frac{1}{T} \sum_{t=1}^{T} \frac{P(Z = z^t, E = e)}{Q(Z = z^t)} = \frac{1}{T} \sum_{t=1}^{T} w(z^t)$$

where $(z^1, \ldots, z^T)$ are sampled from Q.

Ideal Proposal Distribution

- For optimum performance, the proposal distribution Q should be as close as possible to $P(Z|E=e)$.
  - When $Q = P(Z|E=e)$, the weight of every sample is $P(E=e)$!

$$\begin{aligned} w(z^t) = \frac{P(Z=z^t, E=e)}{Q(Z=z^t)} &= \frac{P(Z=z^t, E=e)}{P(Z=z^t|E=e)} \\ &= \frac{P(Z=z^t, E=e)P(E=e)}{P(Z=z^t, E=e)} \\ &= P(E=e) \end{aligned}$$

  - Weight does not depend on z^t
  - One sample would be sufficient!

Issue of Importance Sampling

- (Un-normalized) IS is not suitable for estimating $P(X_i = x_i | E = e)$.
- One option: Estimate the numerator and denominator by IS.

$$\hat{P}(X_i = x_i | E = e) = \frac{\hat{P}(X_i = x_i, E = e)}{\hat{P}(E = e)}$$

- This ratio estimate can be inaccurate because errors in the numerator and denominator may be cumulative.
  - For example, if the numerator is an under-estimate and the denominator is an over-estimate.

Normalized Importance Sampling

- Partition the variables into evidence E and non-evidence Z
- Given an indicator function $\delta_{x_i}(z)$ (which is 1 if z is consistent with $X_i = x_i$ and 0 otherwise), we can write $P(X_i = x_i|E = e)$ as:

$$P(X_i = x_i|E = e) \quad = \quad \frac{P(X_i = x_i, E = e)}{P(E = e)} = \frac{\sum_z \delta_{x_i}(z) P(Z = z, E = e)}{\sum_z P(Z = z, E = e)}$$

- Now we can use the same Q and **same samples** from it to estimate both the numerator and the denominator.

$$\hat{P}(X_i = x_i|E = e) = \frac{\frac{1}{T}\sum_{t=1}^{T} \delta_{x_i}(z^t) w(z^t)}{\frac{1}{T}\sum_{t=1}^{T} w(z^t)}$$

Agenda

- Probability Review
- Approximate Inference
  - Monte Carlo and Importance Sampling
  - Markov Chain Monte Carlo (MCMC)
    - Theoretical Aspects of MCMC
  - Gibbs Sampling and Practical MCMC

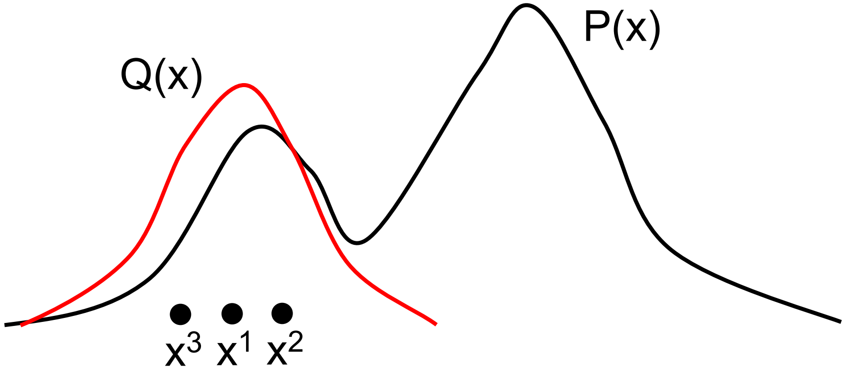
Limitations of IS

- Does 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?

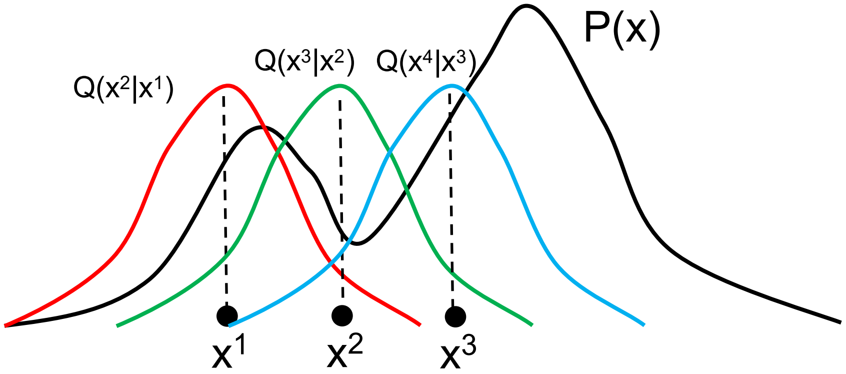
Markov Chain Monte Carlo

- 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 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)

Metropolis-Hastings 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 $x^*$ ~ $Q(x^*|x)$   // draw from proposal
    • sample $u$ ~ 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
3. Take samples from P(x): Reset t=0, for t=1:$N$
    • $x$(t+1) ← Draw sample ($x$(t))
4. Monte Carlo Estimation using these N final samples
```

Function
Draw sample (x(t))

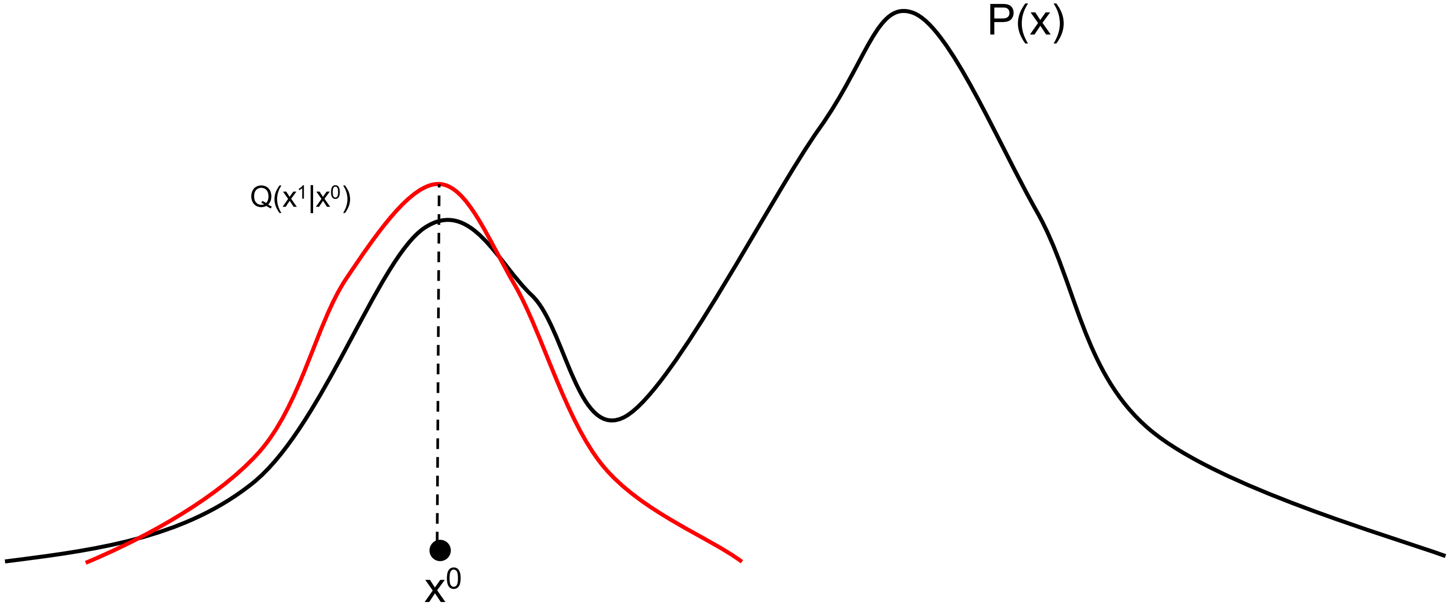
$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

The MH Algorithm

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

Initialize $x^{(0)}$

…

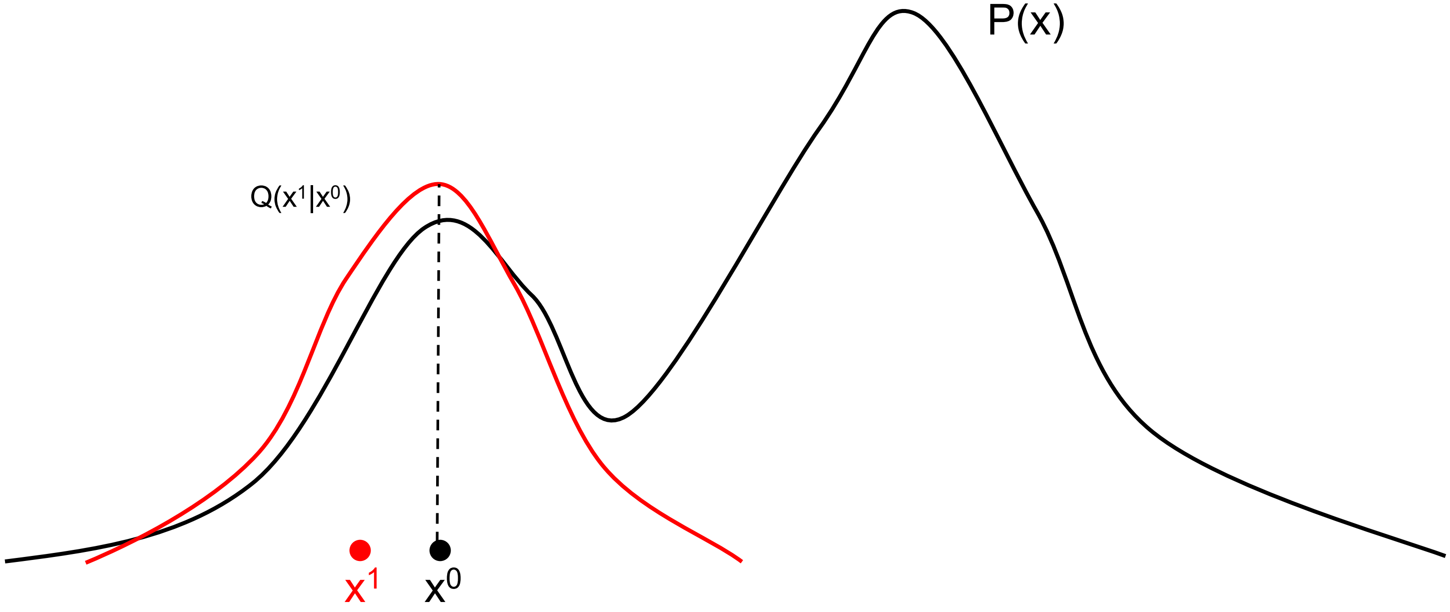


$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

The MH Algorithm

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

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

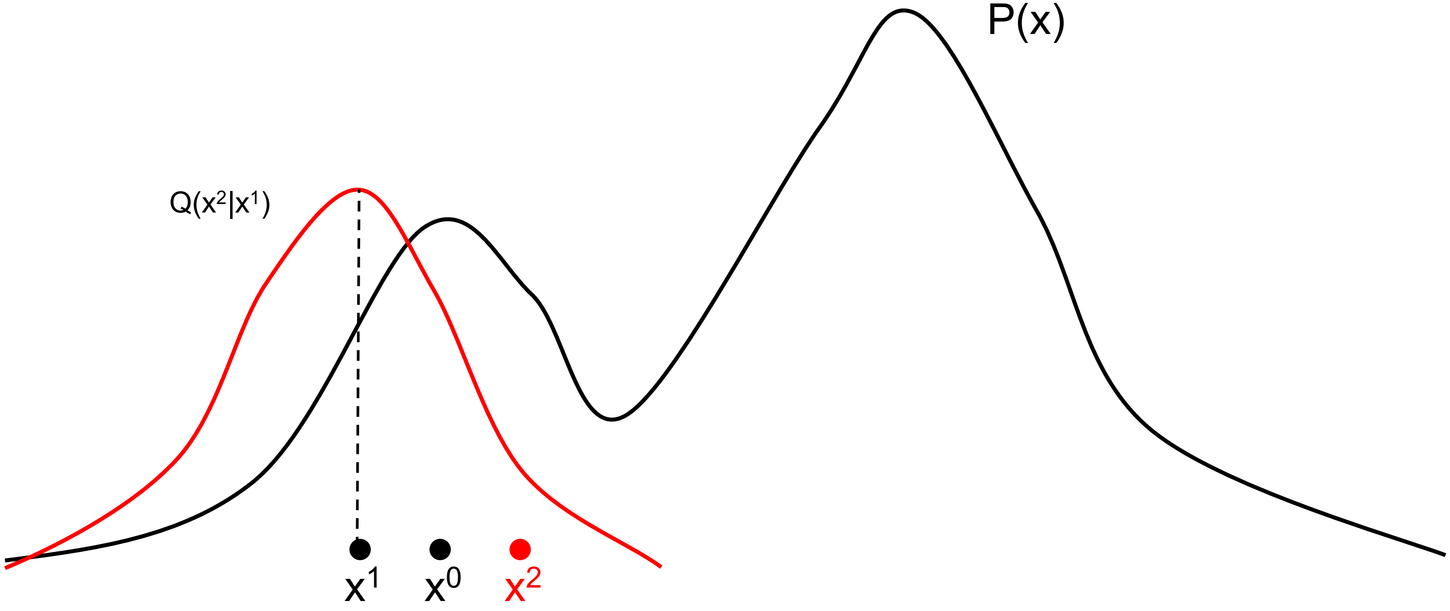


$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

The MH Algorithm

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

Initialize $x^{(0)}$
Draw, accept x^1
Draw, accept x^2

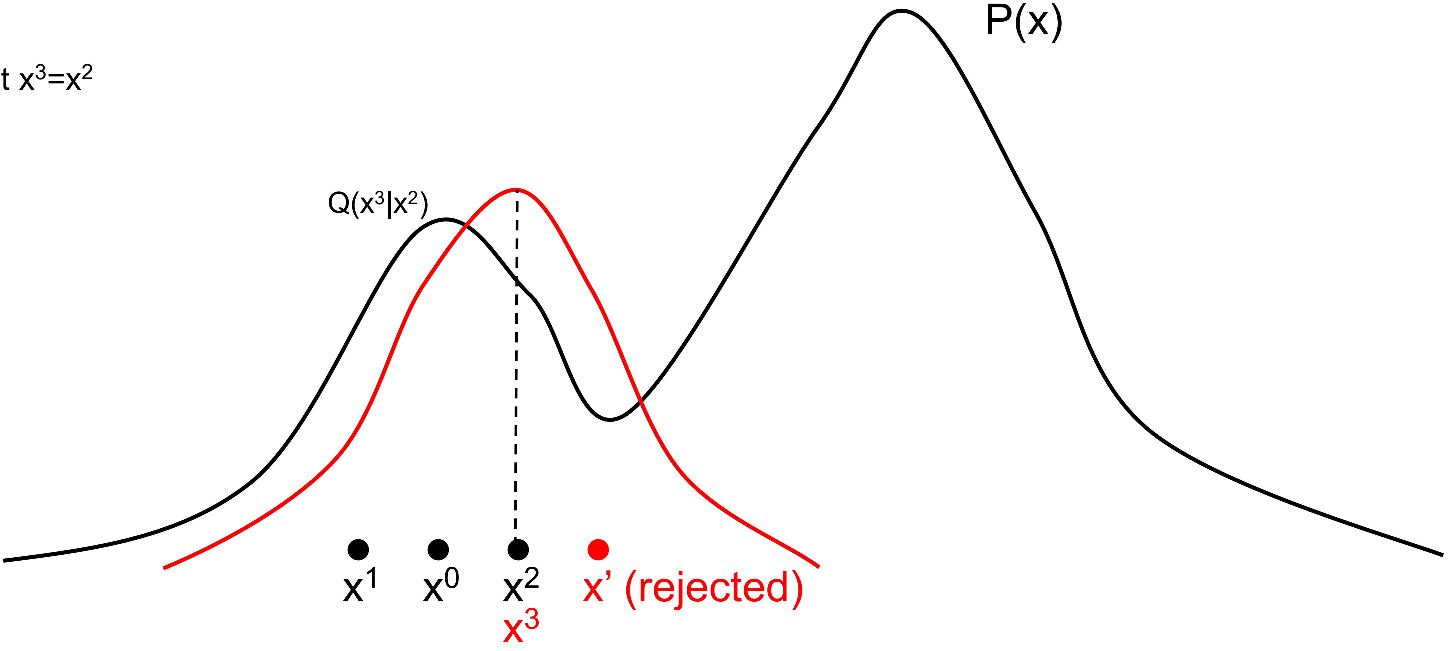


$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

The MH Algorithm

- Example:
  - Let Q(x’|x) be a Gaussian centered on x (it is symmetric)
  - 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 $x^3=x^2$



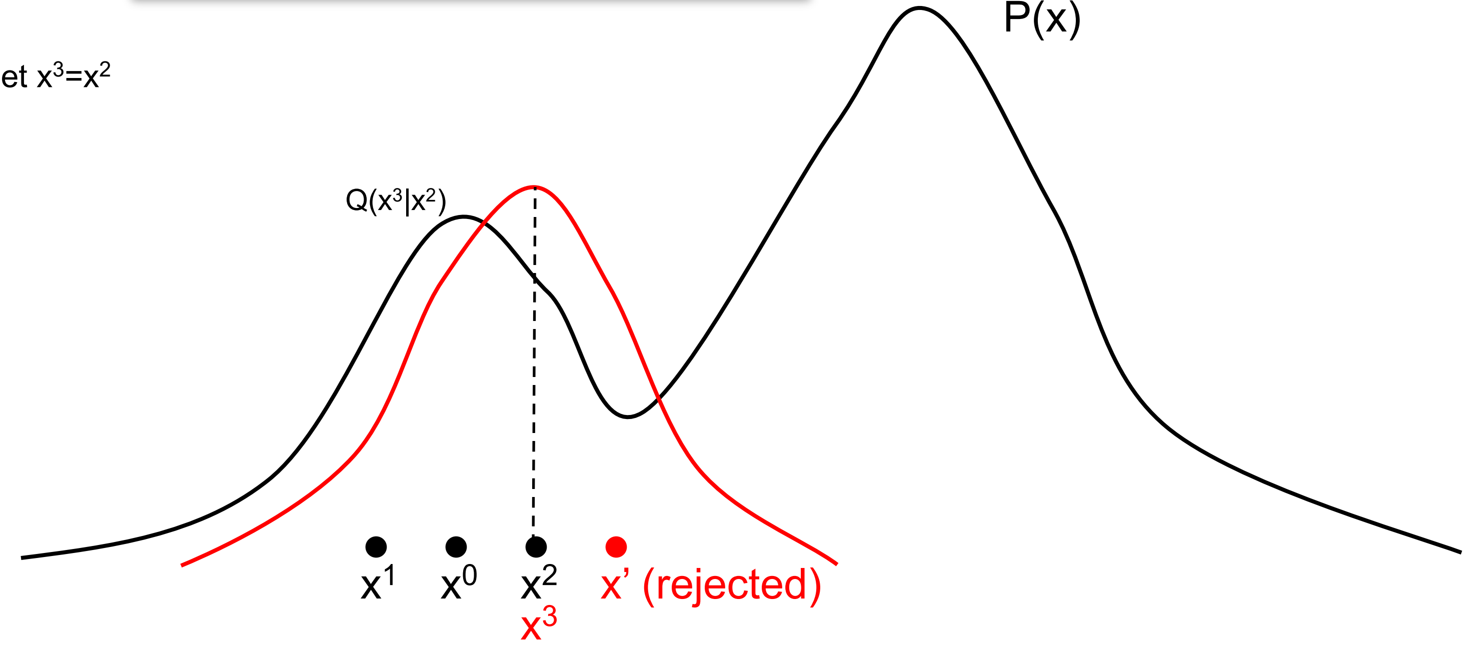
$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

The MH Algorithm

- Example:
  - Let Q(x'|x) be a Gaussian centered on x (it is symmetric)
  - 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 $P(x')/P(x^2)$ is very small, hence $A(x'|x^2)$ is close to zero!

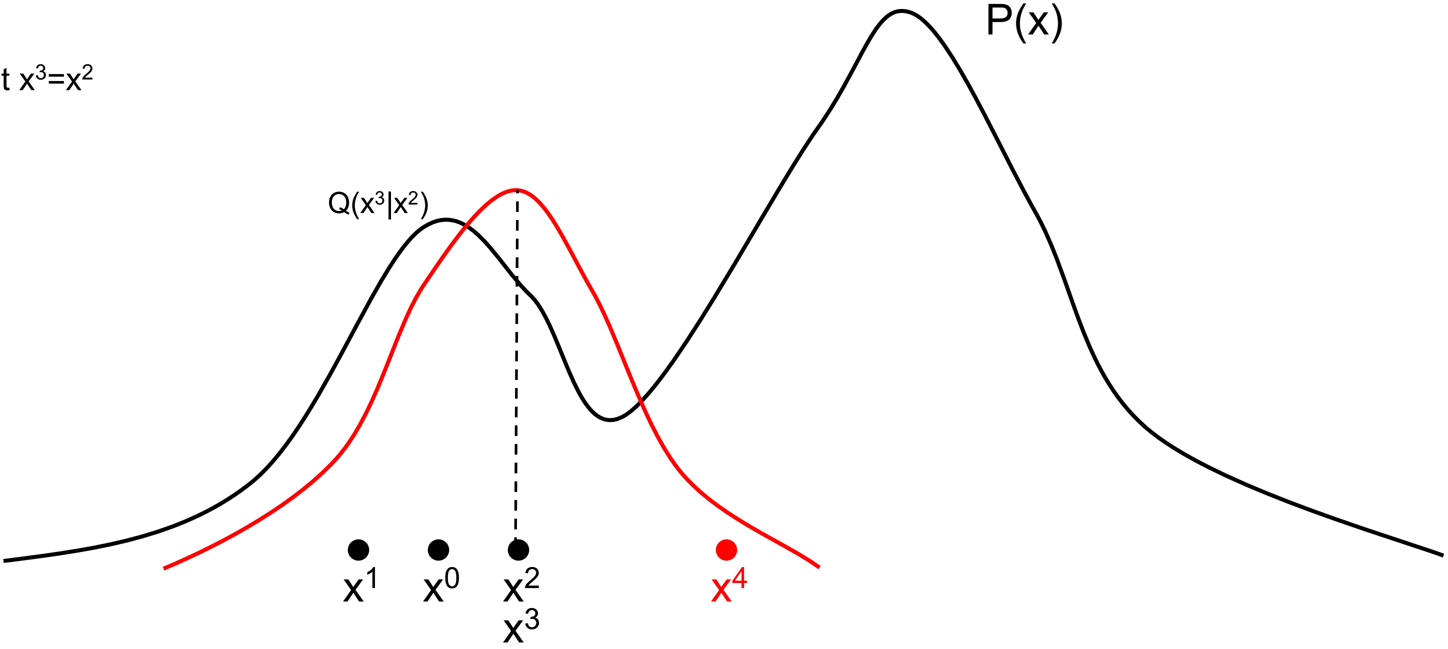


$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

The MH Algorithm

- Example:
  - Let Q(x’|x) be a Gaussian centered on x (it is symmetric)
  - 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

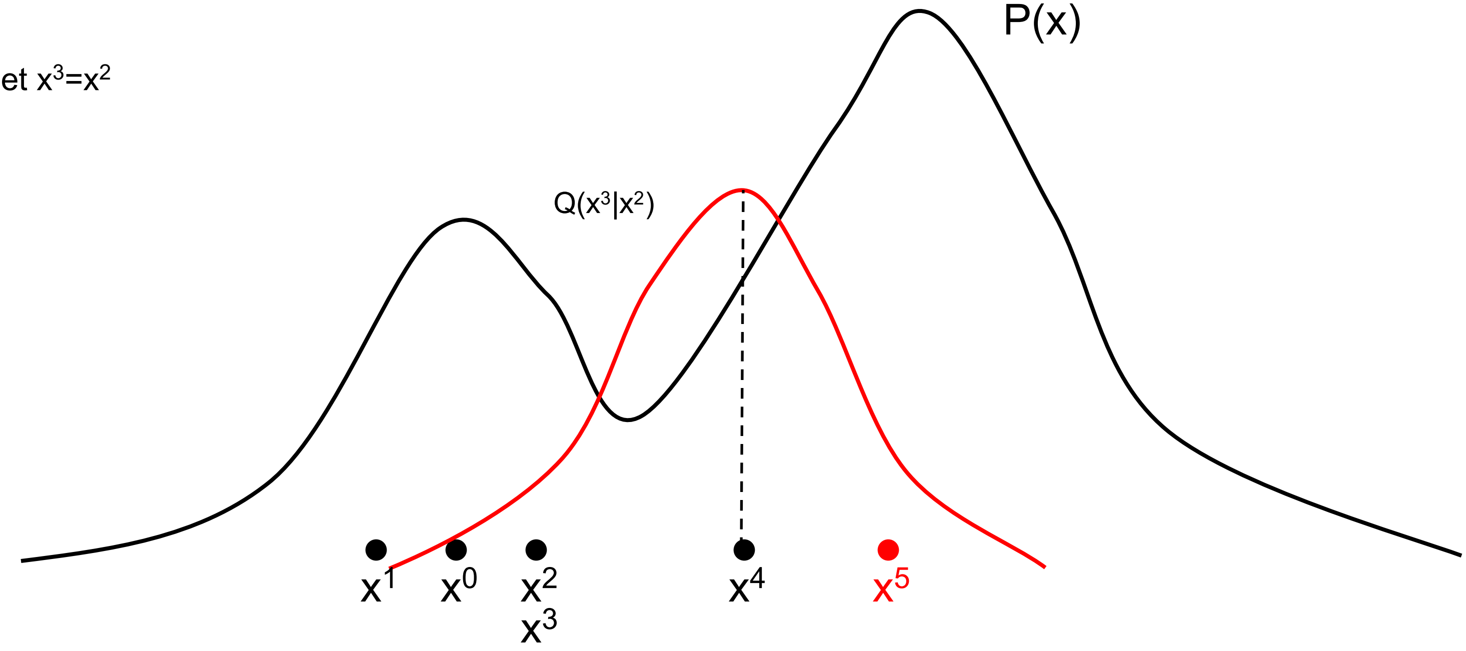


$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

The MH Algorithm

- Example:
  - Let Q(x'|x) be a Gaussian centered on x (it is symmetric)
  - 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



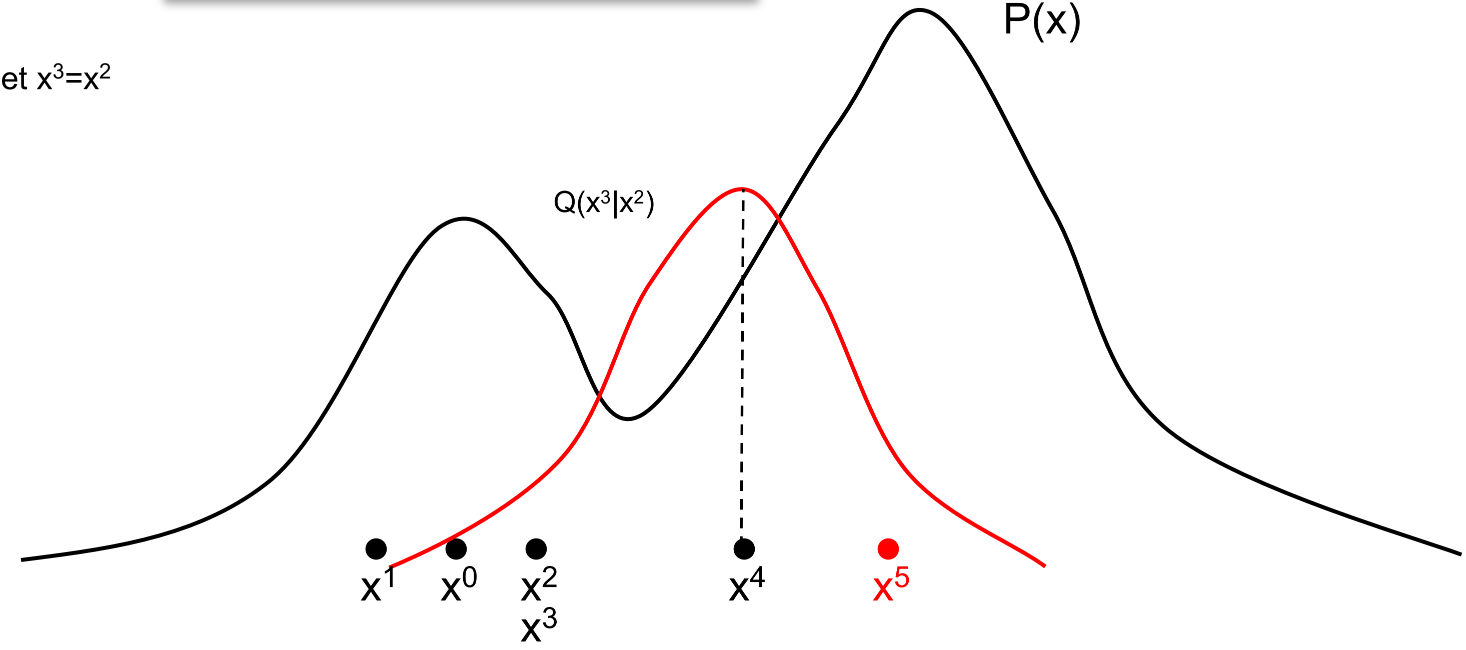
$$A(x'|x) = \min\left(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}\right)$$

The MH Algorithm

- Example:
  - Let Q(x’|x) be a Gaussian centered on x (it is symmetric)
  - 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)!



Agenda

- Probability Review
- Approximate Inference
  - Monte Carlo and Importance Sampling
  - Markov Chain Monte Carlo (MCMC)
    - Theoretical Aspects of MCMC
  - Gibbs Sampling and Practical MCMC

Theoretical Aspects of MCMC

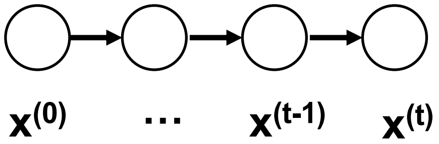
- The MH algorithm has a “burn-in”/“warm-up” period. We throw away all the samples we get from this period. Why?

- 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 are good, general-purpose, proposal distributions?

Markov Chains

- A Markov Chain is a sequence of random variables $x^{(1)},x^{(2)},\ldots,x^{(t)}$ with the Markov Property

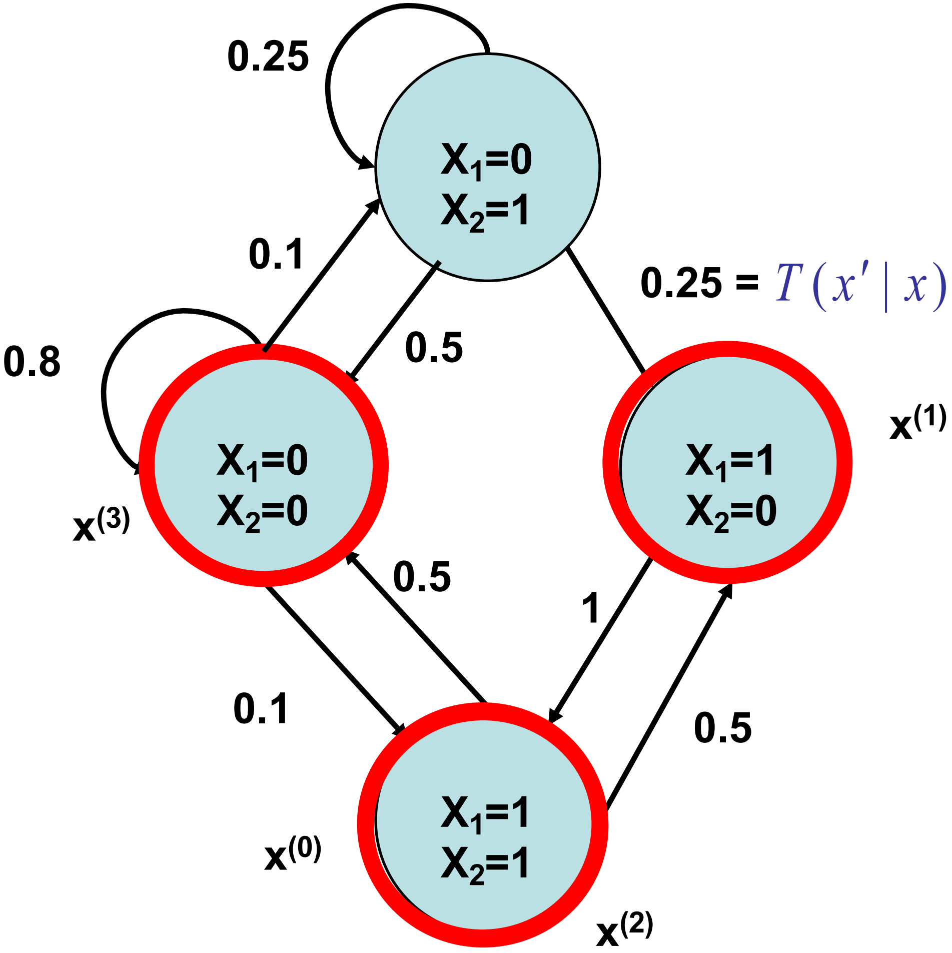


$$P(x^{(t)}=x\,|\,x^{(1)},\ldots,x^{(t-1)})=P(x^{(t)}=x\,|\,x^{(t-1)})$$

  - $P(x^{(t)}=x\,|\,x^{(t-1)})$ is known as the transition kernel (just a matrix for discrete random variables)
  - The whole process is completely determined by the transition kernel and the initial state. The next state depends only on the preceding state
  - Note: the random variable $x^{(i)}$ can be vectors
    - We define $x^{(t)}$ to be the t-th sample of all variables in our model

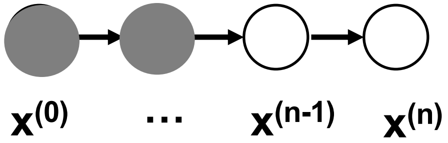
- We study homogeneous Markov Chains, in which the transition kernel $P(x^{(t)}=x'\,|\,x^{(t-1)}=x)$ is fixed with time
  - To emphasize this, we will call the kernel $T(x'\,|\,x)$, where x is the previous state and x' is the next state

Markov Chains



0.25 = $T(x' \mid x)$ = T (x' = (1,0) | x=(0,1))

Markov Chain Sampling = simulating the dynamics of a Markov Chain



Randomly pick an outgoing edge (sample $x^{(1)}$ given $x^{(0)}$ =(1,1))

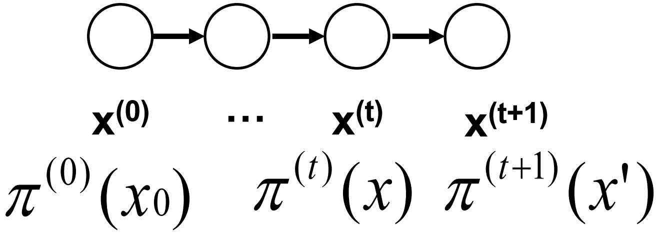
Initialize the simulation in one state (or randomly) $x^{(0)}$

Markov Chain Concepts

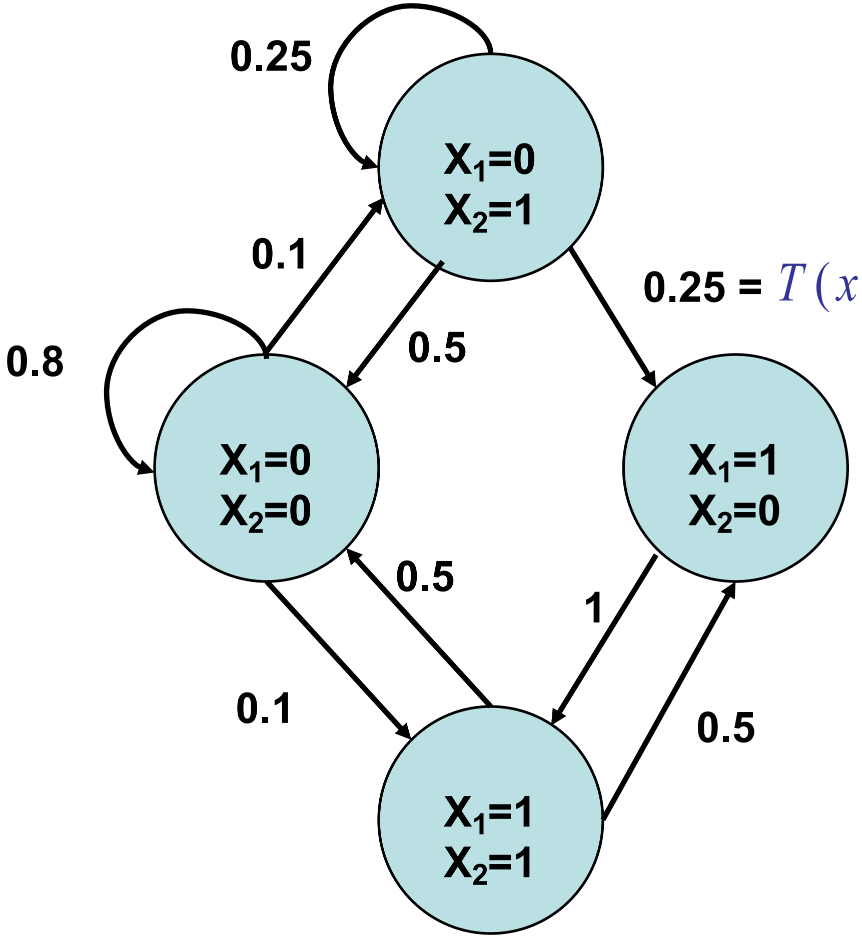
- To understand MCs, we need to define a few concepts:
  - Probability distributions over states: $\pi^{(t)}(x)$ is a distribution over the state of the system x, at time t
    - When dealing with MCs, we don't think of the system as being in one state, but as having a distribution over states
    - Here 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 transit the entire distribution:

$$\pi^{(t+1)}(x') = \sum_x \pi^{(t)}(x) T(x' \mid x)$$

    - At time t, state x has probability mass $\pi^{(t)}(x)$. The transition probability redistributes this mass to other states x'.

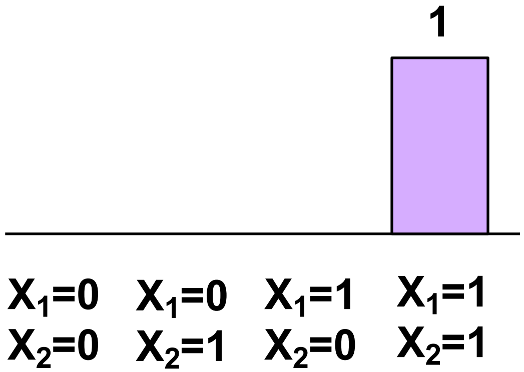


Markov Chains



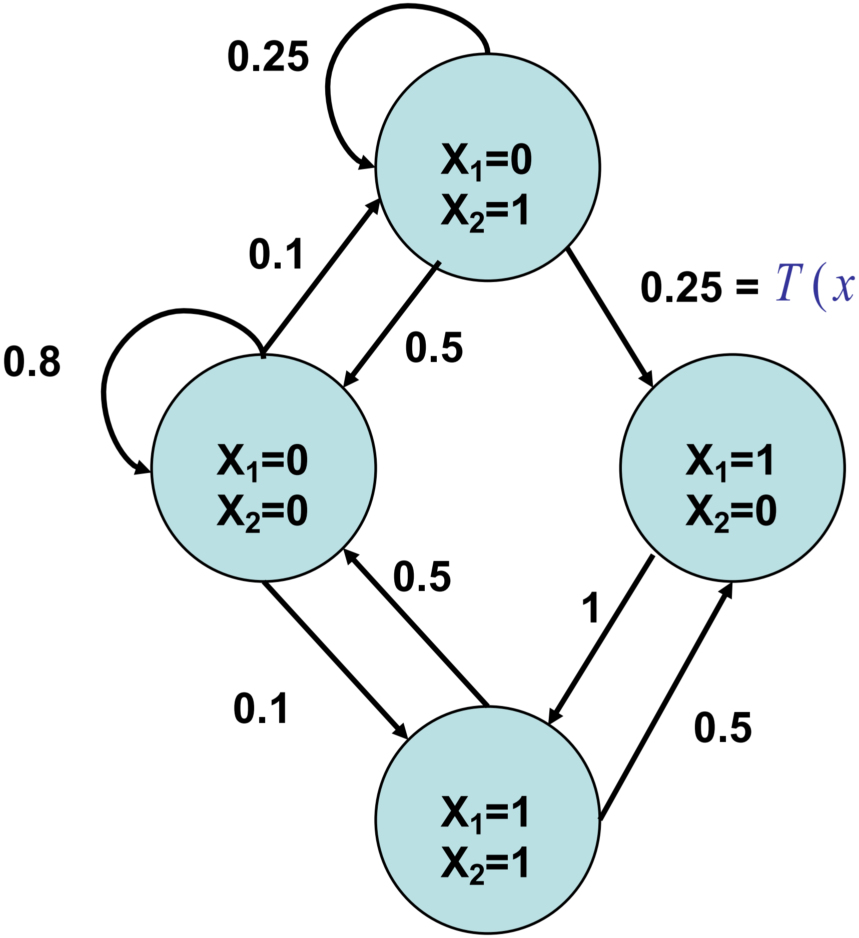
0.25 = $T(x' \mid x)$ = T (x’ = (1,0) | x=(0,1))

$\pi^{(0)}(x)$

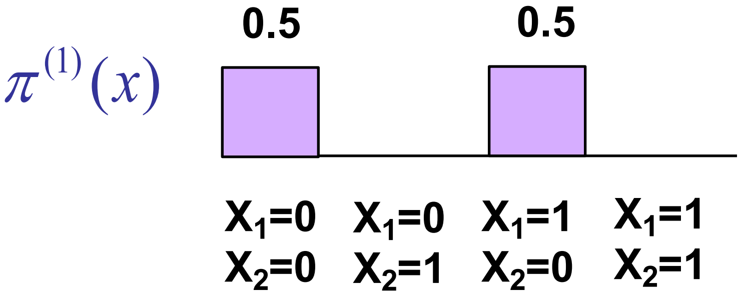


Initialize the simulation in one state $x^{(0)}$

Markov Chains

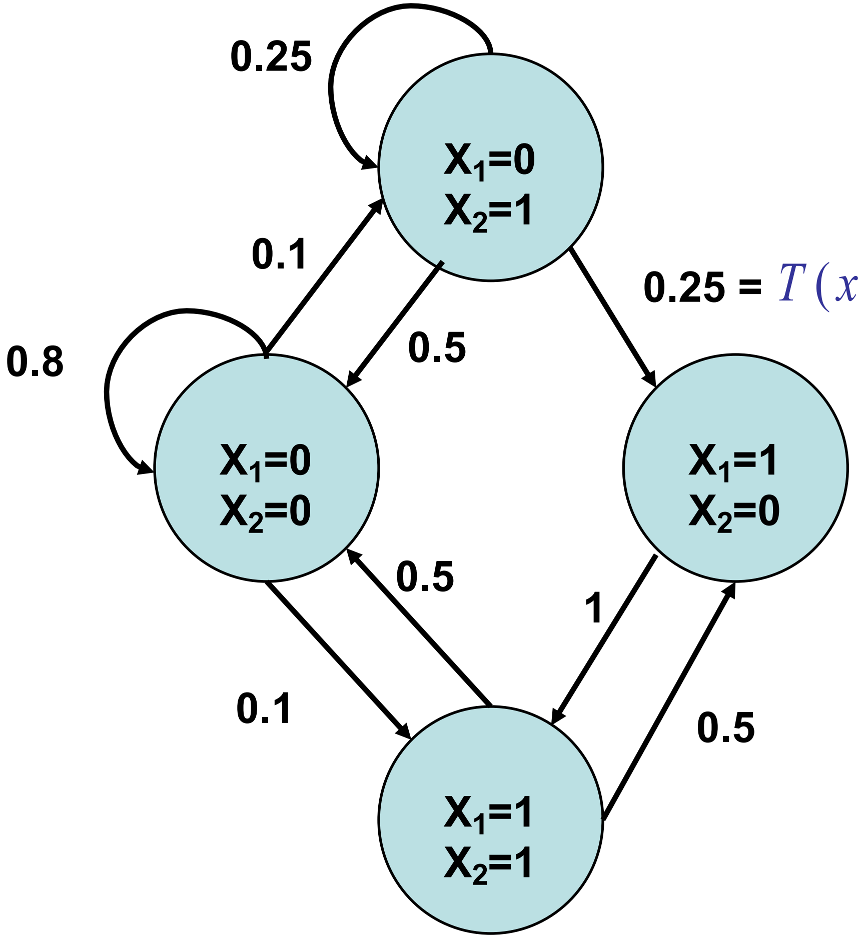


0.25 = $T(x' \mid x)$ = T (x' = (1,0) | x=(0,1))



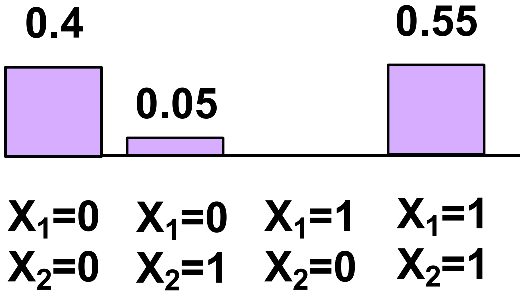
Initialize the simulation in one state $x^{(0)}$

Markov Chains



0.25 = $T(x' \mid x)$ = T (x' = (1,0) | x=(0,1))

$\pi^{(2)}(x)$



stationary if it does not change

Initialize the simulation in one state $x^{(0)}$

Stationary Distribution

- $\pi(x)$ is stationary if it does not change under the transition kernel $T(x' \mid x)$

$$\pi(x') = \sum_x \pi(x) T(x' \mid x) \quad \text{for all x'}$$

- A MC is reversible if there exists a distribution $\pi(x)$ such that the detailed balance condition is satisfied:

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

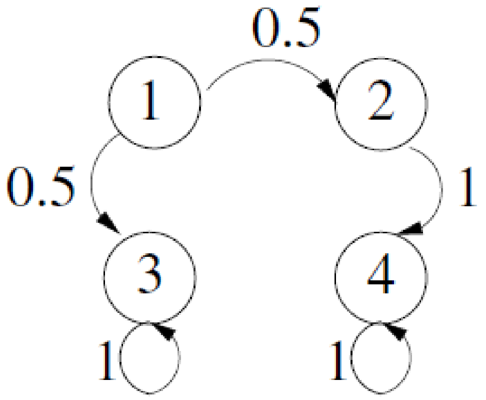
  - This is saying under the distribution $\pi(x)$, the probability of x'→x is the same as x→x'

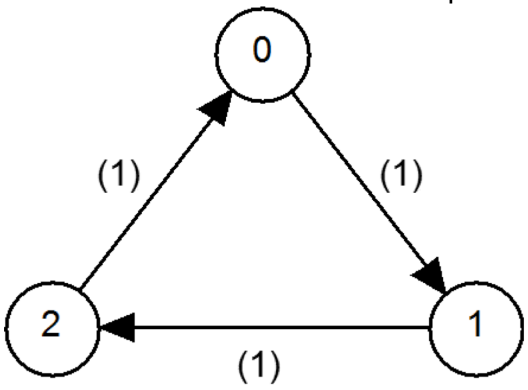
- Theorem: $\pi(x)$ is a stationary distribution of the MC if it is reversible

Properties of Markov Chains

- 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
  - This property only depends on the transition kernel, not the initial state
- Aperiodic: an MC is aperiodic if you can return to any state ***i*** at any time
  - If there exists n such that for all $n' \geq n$, $\Pr(x^{(n')} = i \mid x^{(0)} = i) > 0$
- Ergodic (or regular): an MC is ergodic if it is irreducible and aperiodic

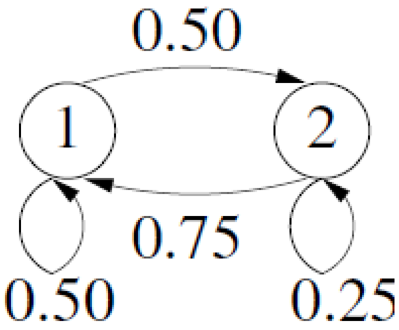
Examples





Reducible.
Limiting distribution depends on initial condition

Irreducible, periodic (each state visited every 3 iterations)
Limiting distribution does not exist



Irreducible, aperiodic.
Unique limiting distribution
P(x) = [0.6, 0.4]

Stationary Distribution

- Ergodicity 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

Why Does MH 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 MH is reversible, i.e. stationary distribution exists:
  - 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:

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

Why Does MH Work?

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

- Now suppose A(x'|x) < 1 and A(x|x') = 1. We have

$$A(x'|x) = \frac{P(x')Q(x\mid x')}{P(x)Q(x'\mid x)}$$

$$P(x)Q(x'\mid x)A(x'|x) = P(x')Q(x\mid x')$$

$$P(x)Q(x'\mid x)A(x'|x) = P(x')Q(x\mid x')A(x\mid x')$$

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

- 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
  - If ergodic (irreducible & aperiodic), MH algorithm eventually converges to the true distribution

Why Does MH Work?

- Theorem: If a Markov chain is **ergodic** and **reversible** with respect to P(x), then P(x) is its unique stationary distribution. The chain converges to the stationary distribution regardless of where it begins.

- The *mixing time*, or how long it takes to **reach** something close the stationary distribution, can’t be guaranteed.

Agenda

- Probability Review
- Approximate Inference
  - Monte Carlo and Importance Sampling
  - Markov Chain Monte Carlo (MCMC)
    - Theoretical Aspects of MCMC
  - Gibbs Sampling and Practical MCMC

Gibbs Sampling

- Gibbs Sampling is a special case of the MH algorithm
- Gibbs Sampling samples each random variable one at a time. Therefore, it has reasonable computation and memory requirements

Gibbs Sampling Algorithm

- Suppose the model contains variables $x_1,\ldots,x_n$
- Initialize starting values for $x_1,\ldots,x_n$
- 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 $x \sim P(x_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$, 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 immediately use its new value for sampling other variables x_j

Gibbs Sampling is MH

- The GS proposal distribution is

$$Q(x_i', \mathbf{x}_{-i} \mid x_i, \mathbf{x}_{-i}) = P(x_i' \mid \mathbf{x}_{-i})$$

($\mathbf{x}_{-i}$ denotes all variables except x_i)

- Applying Metropolis-Hastings with this proposal, we obtain:

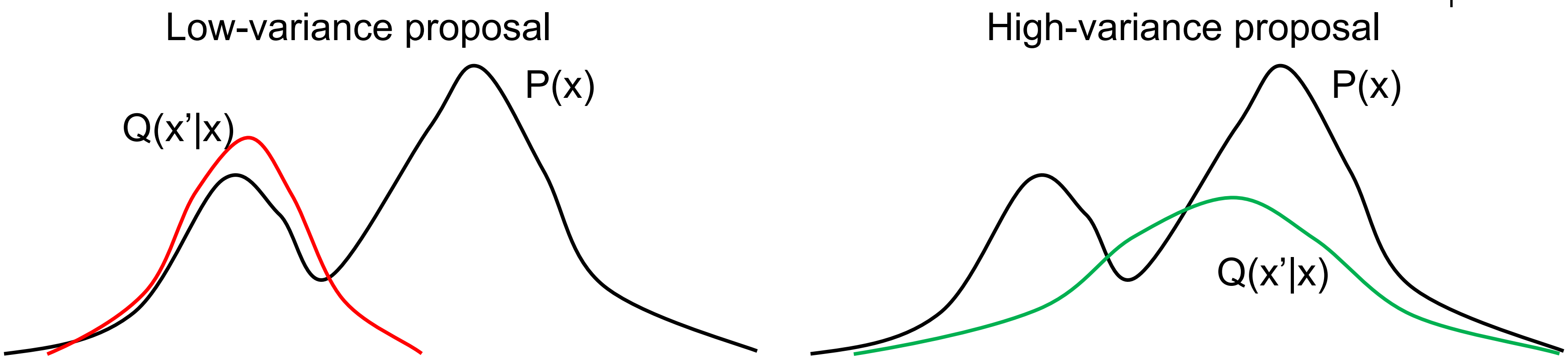
$$\begin{aligned} A(x_i', \mathbf{x}_{-i} \mid x_i, \mathbf{x}_{-i}) &= \min\left(1, \frac{P(x_i', \mathbf{x}_{-i}) Q(x_i, \mathbf{x}_{-i} \mid x_i', \mathbf{x}_{-i})}{P(x_i, \mathbf{x}_{-i}) Q(x_i', \mathbf{x}_{-i} \mid x_i, \mathbf{x}_{-i})}\right) \\ &= \min\left(1, \frac{P(x_i', \mathbf{x}_{-i})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

GS is simply MH with a proposal that is always accepted

Practical Aspects of MCMC

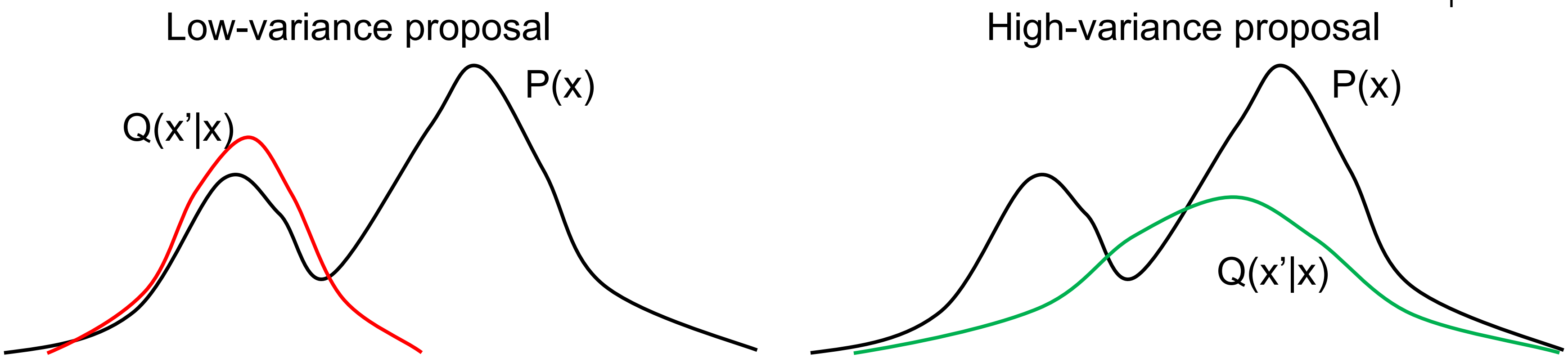
- How do we know if our proposal $Q(x'|x)$ is good or not?
  - Monitor the acceptance rate
  - Plot the autocorrelation function

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



Low-variance proposal

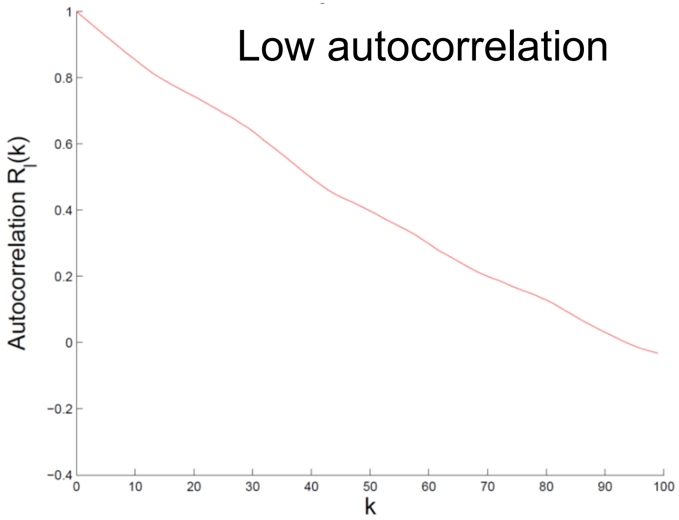
High-variance proposal

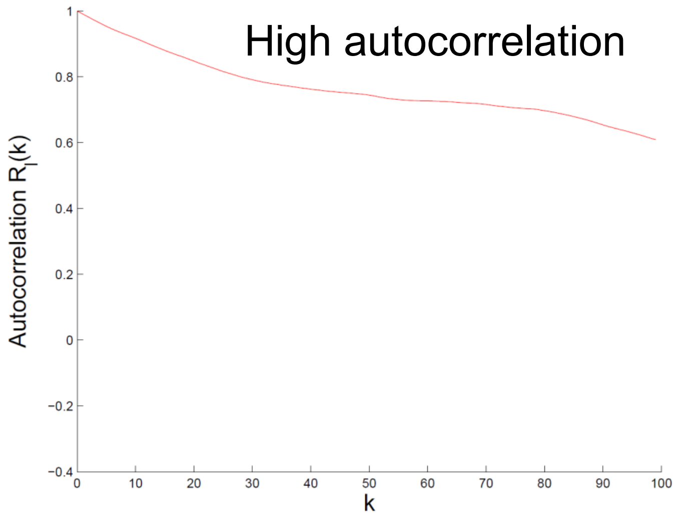
- Acceptance rate is the fraction of samples that MH accepts.
  - General guideline: proposals should have ~0.5 acceptance rate [1]
- Gaussian special case:
  - If both P(x) and Q(x'|x) are Gaussian, the optimal acceptance rate is ~0.45 for D=1 dimension and approaches ~0.23 as D tends to infinity [2]

[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:

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

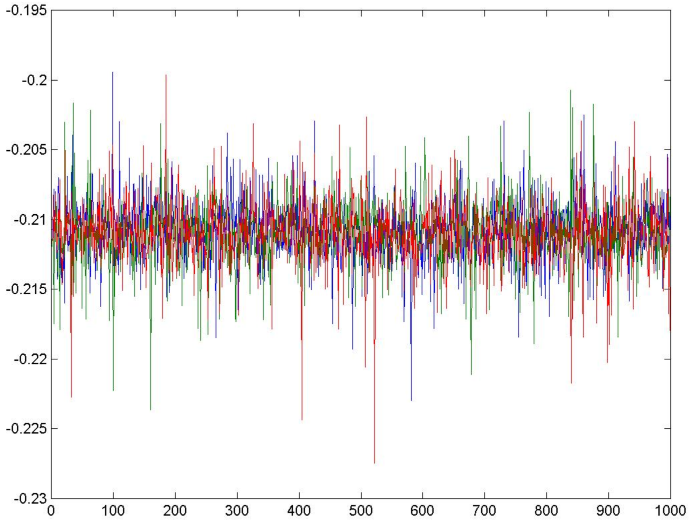
  - High autocorrelation leads to smaller effective sample size!
  - We want proposals Q(x’|x) with low autocorrelation

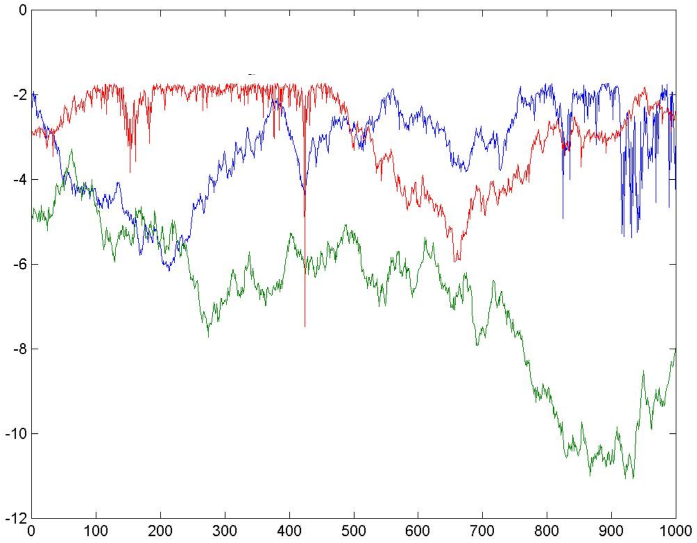
Practical Aspects of MCMC

- 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

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
- In practice, we usually start with multiple chains

Summary

- 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) is not easy

- Gibbs sampling sets the proposal Q(x'|x) to the conditional distribution P(x'|x)
  - Acceptance rate is 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

Thank you!

Q & A