

[80240603 Advanced Machine Learning, Fall, 2012]

Monte Carlo Methods

Jun Zhu

dcszj@mail.tsinghua.edu.cn

State Key Lab of Intelligent Technology & Systems
Tsinghua University

November 22, 2011

Monte Carlo Methods

- ◆ a class of **computational algorithms** that rely on repeated **random sampling** to compute their results.
- ◆ tend to be used when it is infeasible to compute an exact result with a **deterministic algorithm**
- ◆ was coined in the 1940s by **John von Neumann**, **Stanislaw Ulam** and **Nicholas Metropolis**

Games of Chance

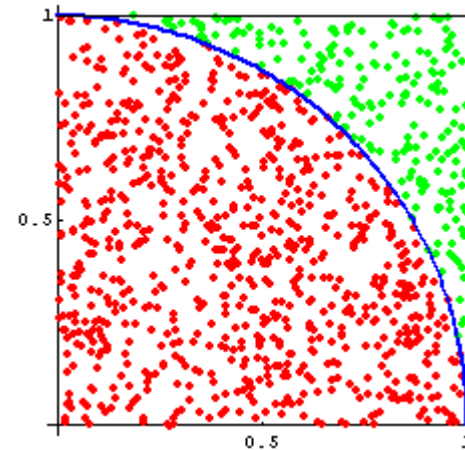


Monte Carlo Methods to Calculate Pi

◆ Computer Simulation

$$\hat{\pi} = 4 \times \frac{m}{N}$$

- N: # points inside the square
- m: # points inside the circle

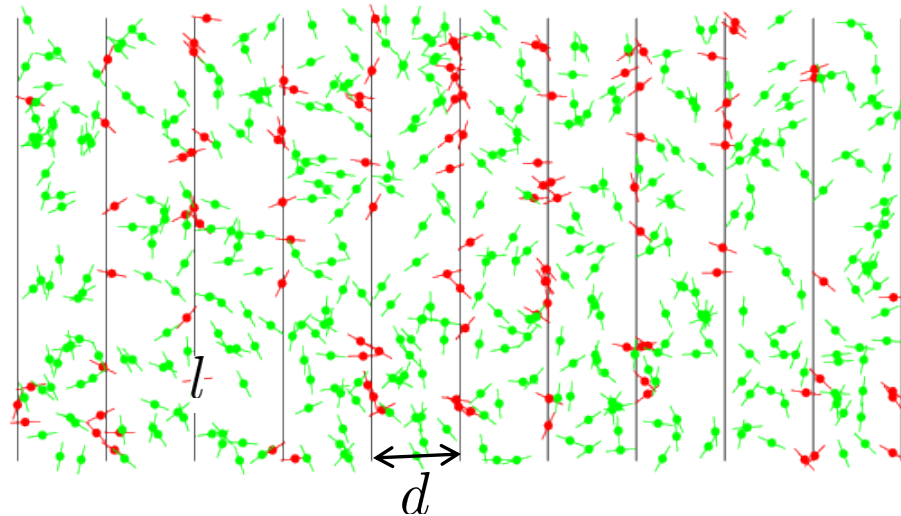


◆ Buffon's Needle Experiment

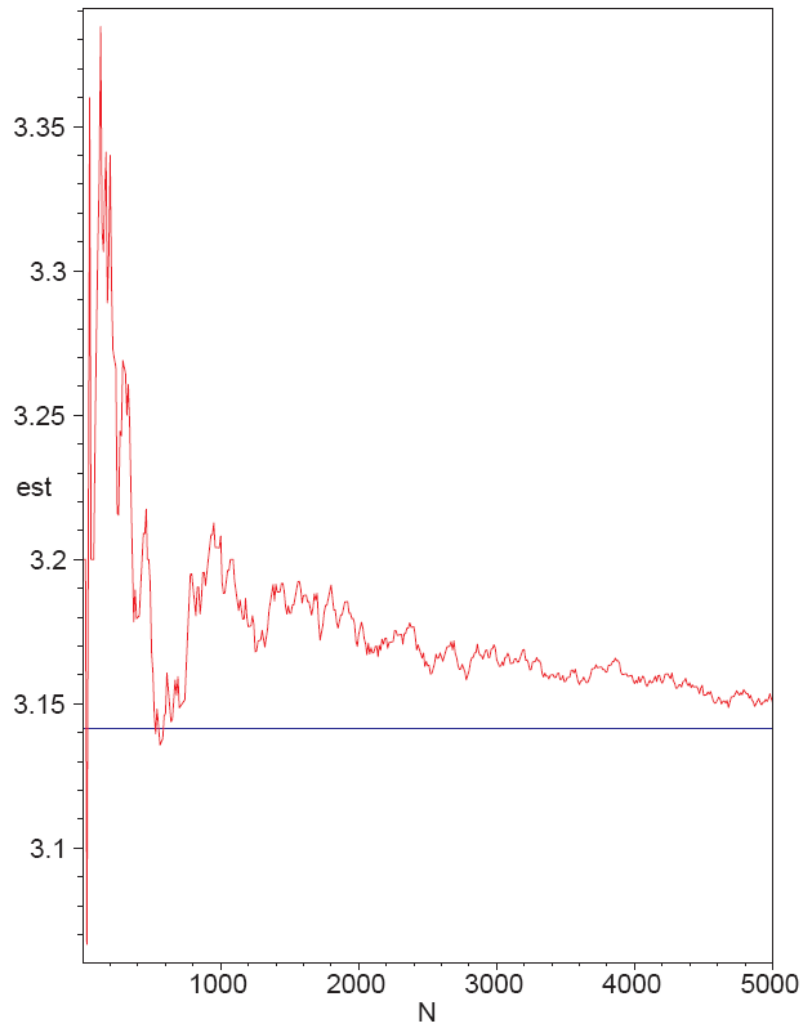
$$\hat{\pi} = \frac{2Nx}{m}$$

- m: # line crossings

$$x = \frac{l}{d}$$



Typical Outputs with Simulation



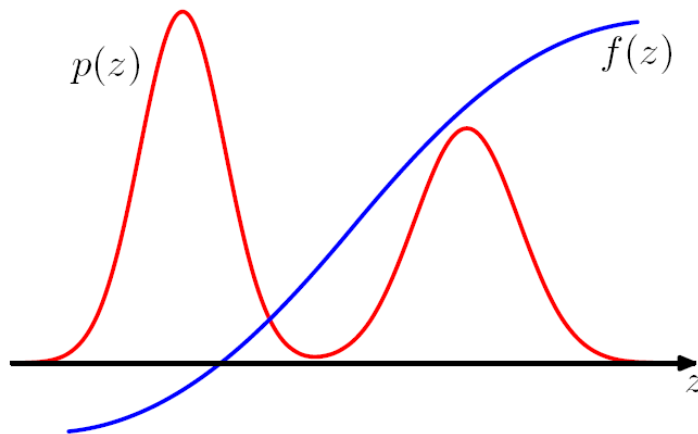
Problems to be Solved

◆ Sampling

- to generate a set of samples $\{\mathbf{z}_l\}_{l=1}^L$ from a given probability distribution $p(\mathbf{z})$
- the distribution is called **target distribution**
- can be from statistical physics or data modeling

◆ Integral

- To estimate expectations of functions under this distribution



$$\mathbb{E}[f] = \int f(\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

Use Sample to Estimate the Target Dist.

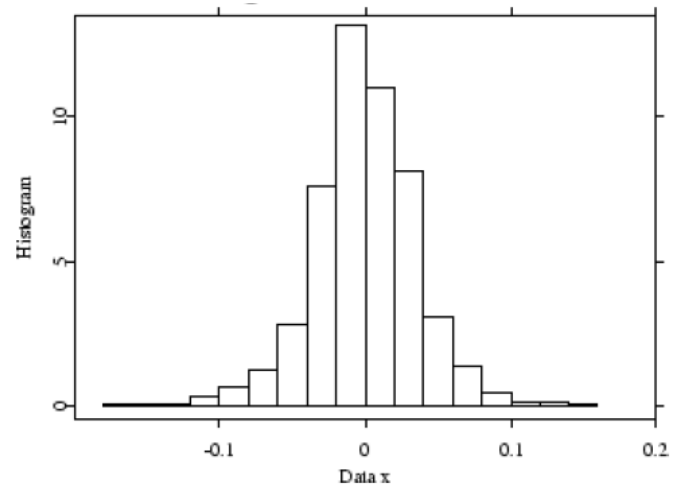
- ◆ Draw a set of **independent** samples (a **hard problem**)

$$\forall 1 \leq l \leq L, \mathbf{z}^{(l)} \sim p(\mathbf{z})$$

- ◆ Estimate the target distribution as **count frequency**

$$p(\mathbf{z}) \approx \frac{1}{L} \sum_{l=1}^L \delta_{\mathbf{z}, \mathbf{z}^{(l)}}$$

Histogram with Unique
Points as the Bins



Basic Procedure of Monte Carlo Methods

- ◆ Draw a set of **independent** samples

$$\forall 1 \leq l \leq L, \mathbf{z}^{(l)} \sim p(\mathbf{z})$$

- ◆ Approximate the expectation with

$$\hat{f} = \frac{1}{L} \sum_{l=1}^L f(\mathbf{z}^{(l)})$$

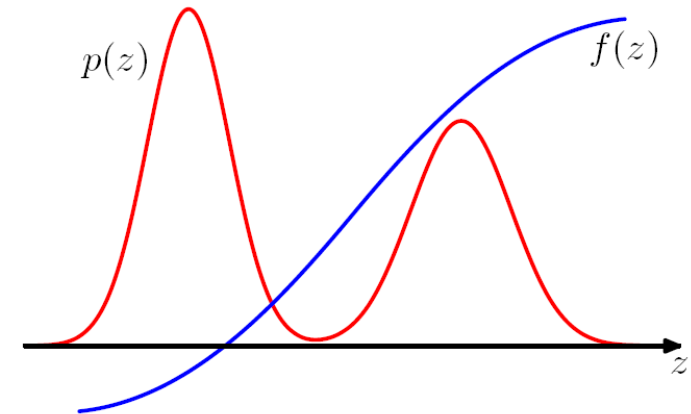
- where is the distribution p ?
- why this is good?

$$p(\mathbf{z}) \approx \frac{1}{L} \sum_{l=1}^L \delta_{\mathbf{z}, \mathbf{z}^{(l)}}$$

Histogram with Unique
Points as the Bins

$$\mathbb{E}[\hat{f}] = \mathbb{E}[f] \quad \text{var}[\hat{f}] = \frac{1}{L} \mathbb{E}[(f - \mathbb{E}[f])^2]$$

- Accuracy of estimator does not depend on dimensionality of \mathbf{z}
- High accuracy with few (10-20 independent) samples
- However, obtaining independent samples is often not easy!



Why Sampling is Hard?

◆ Assumption

- The target distribution can be evaluated, at least to within a multiplicative constant, i.e.,

$$p(\mathbf{z}) = p^*(\mathbf{z})/Z$$

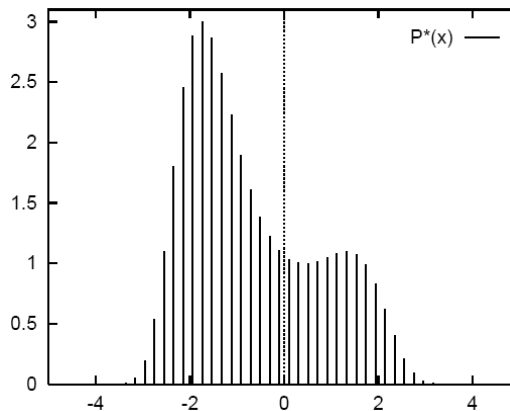
- where $p^*(\mathbf{z})$ can be evaluated

◆ Two difficulties

- Normalizing constant is typically unknown
- Drawing samples in high-dimensional space is challenging

A Simple Example

- ◆ Draw samples from a discrete distribution with a finite set of uniformly distributed points

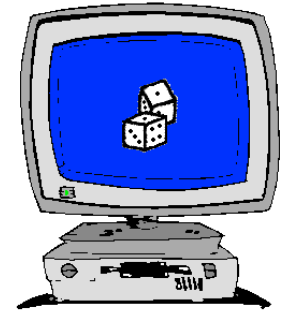


- ◆ We can compute the distribution via

$$Z = \sum_i p_i^* \quad p_i = p_i^* / Z$$

- ◆ ... then draw samples from the multinomial distribution
- ◆ **But, the cost grows exponentially with dimension!**

Basic Sampling Algorithms



- ◆ Strategies for generating samples from a given standard distribution, e.g., Gaussian
- ◆ Assume that we have a pseudo-random generator for *uniform distribution over $(0, 1)$*
- ◆ For standard distributions we can *transform* uniformly distributed samples into desired distributions

Basic Sampling Algorithms

- ◆ If z is uniformly distributed over $(0, 1)$, then $y = f(z)$ has the distribution

$$p(y) = p(z) \left| \frac{dz}{dy} \right|$$

- where $p(z) = 1$

- ◆ Normally, we know $p(y)$ and infer f . This can be done via

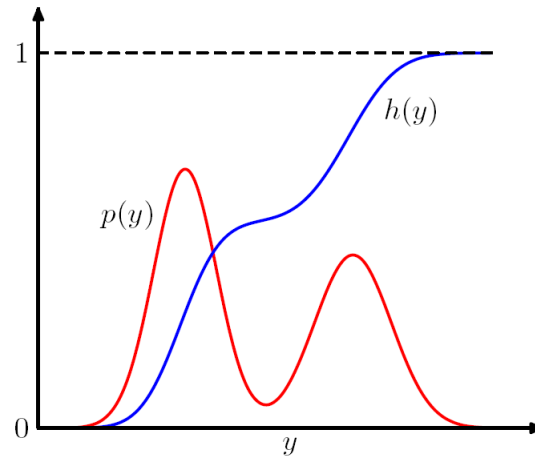
$$z = h(y) = \int_{-\infty}^y p(\hat{y}) d\hat{y}$$

$$y = h^{-1}(z)$$

- ◆ So we have to transform uniformly distributed random numbers
 - using a function which is the inverse of the indefinite integral of the distribution

Geometry of Transformation

- ◆ Generating non-uniform random variables



- ◆ $h(y)$ is indefinite integral of desired $p(y)$
- ◆ $z \sim \text{Uniform}(0, 1)$ is transformed using $y = h^{-1}(z)$
- ◆ Results in y being distributed as $p(y)$

Example #1

- ◆ How to get the **exponential distribution** from uniform variable?

$$p(y) = \lambda \exp(-\lambda y)$$

- ◆ Do the integral, we get

$$\begin{aligned} z = h(y) &= \int_{-\infty}^y p(\hat{y}) d\hat{y} = \int_{-\infty}^y \lambda \exp(-\lambda \hat{y}) d\hat{y} \\ &= 1 - \exp(-\lambda y) \end{aligned}$$

- ◆ Thus

$$y = h^{-1}(z) = -\frac{1}{\lambda} \ln(1 - z)$$

Example #2

- ◆ How to get the **standard normal distribution** from uniform variable?

$$p(y) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right)$$

- ◆ Do the integral, we get

$$\begin{aligned} z = h(y) &= \int_{-\infty}^y p(\hat{y}) d\hat{y} = \Phi(y) \\ &= \frac{1}{2} \left[1 + \operatorname{erf}\left(\frac{y}{\sqrt{2}}\right) \right] \end{aligned}$$

- ◆ Thus

$$y = h^{-1}(z)$$

No closed form!!

Example #2: Box-Muller for Gaussian

- ◆ Example of a bivariate Gaussian

$$p(y_1, y_2) = \left[\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y_1^2\right) \right] \left[\frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y_2^2\right) \right]$$

- ◆ Generate pairs of uniformly distributed random numbers

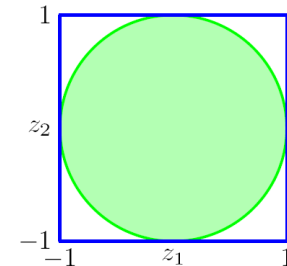
$$z_1, z_2 \sim \text{Uniform}(-1, 1)$$

- ◆ Discard each pair unless

$$z_1^2 + z_2^2 \leq 1$$

- ◆ Leads to uniform distribution of points inside unit circle with

$$p(z_1, z_2) = \frac{1}{\pi}$$



Example #2: Box-Muller for Gaussian

- ◆ Evaluate the two quantities

$$y_1 = z_1 \left(\frac{-2 \ln z_1}{r^2} \right)^{1/2} \quad y_2 = z_2 \left(\frac{-2 \ln z_2}{r^2} \right)^{1/2}$$

- where $r^2 = z_1^2 + z_2^2$

- ◆ Then, we have independent standard normal distribution

$$p(y_1, y_2) = \left[\frac{1}{\sqrt{2\pi}} \exp \left(-\frac{1}{2} y_1^2 \right) \right] \left[\frac{1}{\sqrt{2\pi}} \exp \left(-\frac{1}{2} y_2^2 \right) \right]$$

- ◆ How about non-zero means and non-standard variance?
- ◆ How about multivariate Gaussian?

Rejection Sampling

◆ Problems with transformation methods

- depend on ability to calculate and then invert indefinite integral
- feasible only for some standard distributions

◆ More general strategy is needed

- Rejection sampling and importance sampling are limited to univariate distributions
 - Although not applicable to complex problems, they are important components in more general strategies
- Allows sampling from complex distributions

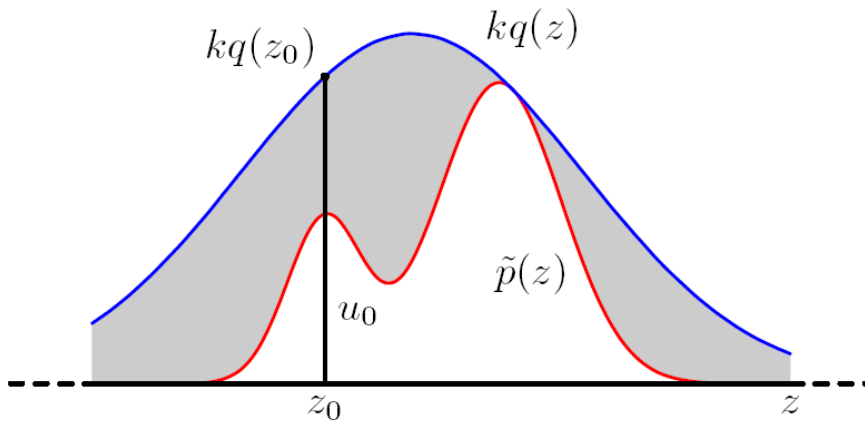
Rejection Sampling

- ◆ Wish to sample from distribution $p(z)$
- ◆ Suppose we are able to easily evaluate $p(z)$ *for any* given value of z
- ◆ Samples are drawn from simple distribution, called **proposal distribution** $q(z)$
- ◆ Introduce constant k whose value is such that $kq(z) \geq p(z)$ for all z
 - Called **comparison function**

Rejection Sampling

- ◆ Samples are drawn from simple distribution $q(z)$
- ◆ Rejected if they fall in grey area between $\tilde{p}(z)$ and $kq(z)$

$$p(z) = \frac{\tilde{p}(z)}{Z_p}$$



- ◆ Resulting samples are distributed according to $p(z)$ which is the normalized version of $\tilde{p}(z)$

How to determine if sample is in shaded region?

- ◆ Each step involves generating two random numbers

$$z_0 \sim q(z) \quad u_0 \sim \text{Uniform}(0, kq(z_0))$$

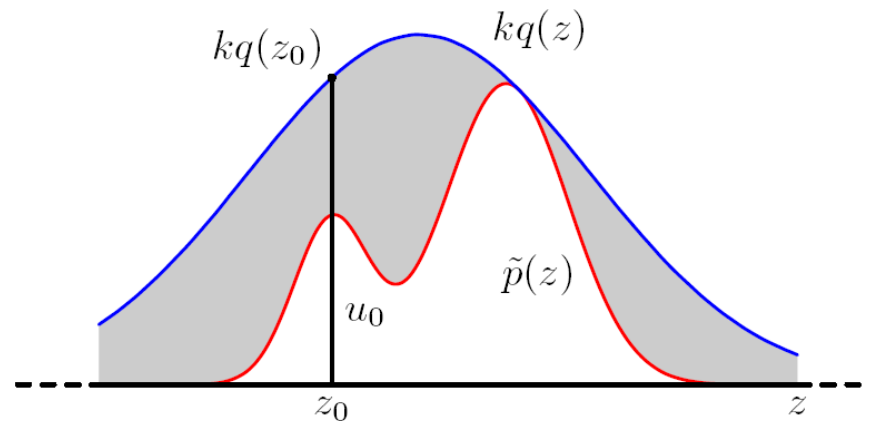
- ◆ This pair has uniform distribution under the curve of function $kq(z)$
- ◆ If $u_0 > p(z_0)$ the pair is rejected otherwise it is retained
- ◆ Remaining pairs have a uniform distribution under the curve of $p(z)$ and hence the corresponding z values are distributed according to $p(z)$ as desired
- ◆ **Proof?**

$$\hat{p}(z) = q(z) \times \frac{\tilde{p}(z)}{kq(z)} \propto \tilde{p}(z)$$

More on Rejection Sampling

- ◆ The probability that a sample will be accepted (**accept ratio**)

$$p(\text{accept}) = \int q(z) \times \frac{\tilde{p}(z)}{kq(z)} dz = \frac{1}{k} \int \tilde{p}(z) dz$$

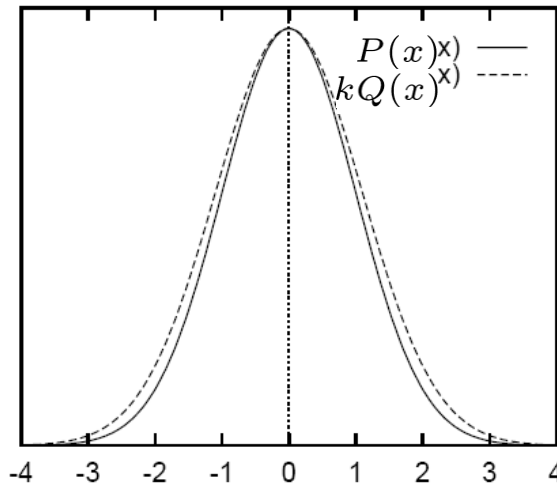


- ◆ To have high accept ratio, k should be as small as possible
 - ... but it needs to satisfy

$$kq(z) \geq \tilde{p}(z) \quad \forall z$$

Curse of Dimensionality

- ◆ Consider two univariate Gaussian distributions



$$P(x) \sim \mathcal{N}(0, \sigma_p^2)$$

$$Q(x) \sim \mathcal{N}(0, \sigma_q^2)$$

$$\sigma_q > \sigma_p$$

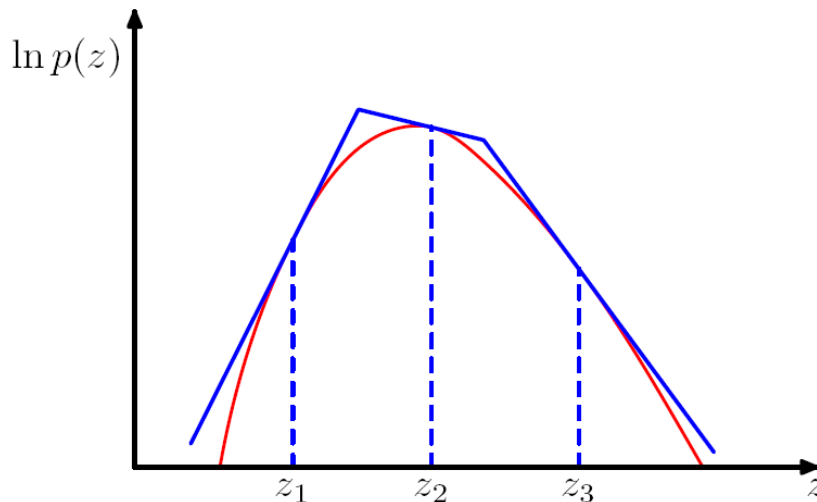
- ◆ What is k ?

- At the origin, we have $k \frac{1}{\sqrt{2\pi}\sigma_q} = \frac{1}{\sqrt{2\pi}\sigma_p}$, so $k = \frac{\sigma_q}{\sigma_p}$
- How about in 1000 dimensions?

$$k = \left(\frac{\sigma_q}{\sigma_p}\right)^{1000} \approx 20,000 \text{ if } \sigma_q = 1.01\sigma_p$$

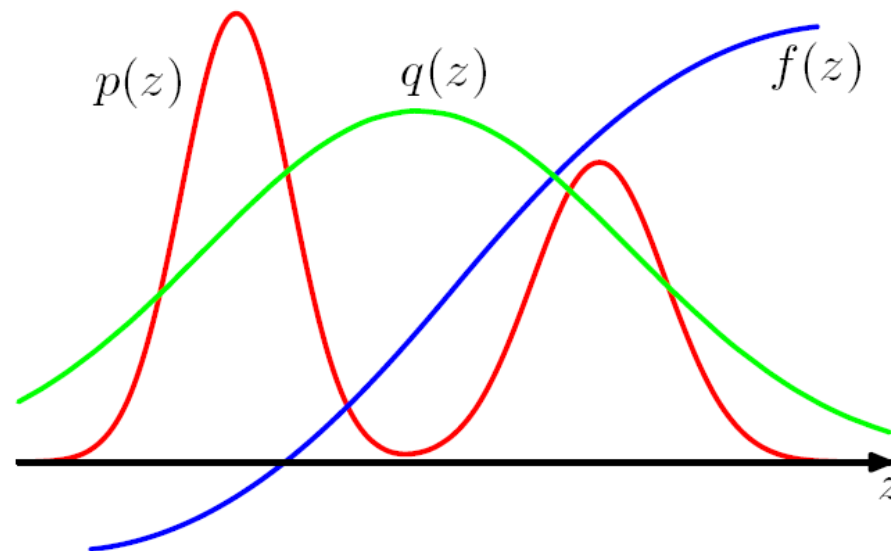
Adaptive Rejection Sampling

- ◆ When difficult to find suitable analytic distribution
- ◆ Straight-forward when $p(z)$ is log concave
 - When $\ln p(z)$ has derivatives that are non-increasing functions of z
 - Function $\ln p(z)$ and gradients are evaluated at set of grid points
 - Intersections are used to construct envelope \rightarrow a sequence of linear functions



Importance Sampling

- ◆ **Evaluating expectation** of $f(z)$ with respect to distribution $p(z)$ from which it is difficult to draw samples directly
- ◆ Samples $\{z^{(l)}\}$ are drawn from simpler distribution $q(z)$
- ◆ Terms in summation are weighted by ratios $\frac{p(z^{(l)})}{q(z^{(l)})}$



Importance Sampling

- ◆ The expectation can be computed as

$$\mathbb{E}[f] = \int f(\mathbf{z})p(\mathbf{z})d\mathbf{z} = \int f(\mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z})}q(\mathbf{z})d\mathbf{z}$$

- ◆ Use Monte Carlo methods

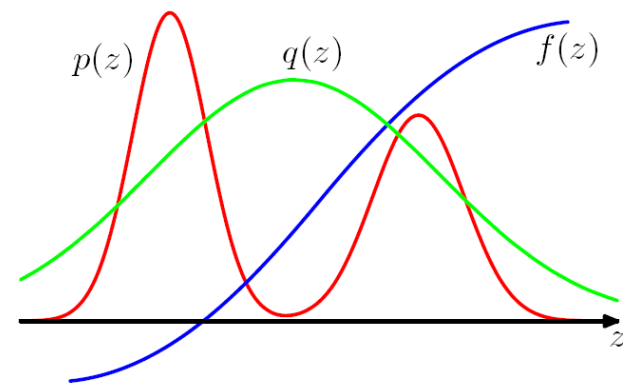
$$\mathbb{E}[f] \approx \frac{1}{L} \sum_{l=1}^L r_l f(\mathbf{z}^{(l)})$$

- where the importance weights are

$$r_l = \frac{p(\mathbf{z}^{(l)})}{q(\mathbf{z}^{(l)})}$$

- and the samples are

$$\mathbf{z}^{(l)} \sim q(\mathbf{z})$$



Importance Sampling

◆ For **unnormalized distributions**

$$p(\mathbf{z}) = \frac{\tilde{p}(\mathbf{z})}{Z_p}, \quad q(\mathbf{z}) = \frac{\tilde{q}(\mathbf{z})}{Z_q}$$

◆ We have $\mathbb{E}[f] = \int f(\mathbf{z})p(\mathbf{z})d\mathbf{z} = \frac{Z_q}{Z_p} \int f(\mathbf{z})\frac{\tilde{p}(\mathbf{z})}{\tilde{q}(\mathbf{z})}q(\mathbf{z})d\mathbf{z}$

$$\mathbb{E}[f] \approx \frac{Z_q}{Z_p} \frac{1}{L} \sum_{l=1}^L \tilde{r}_l f(\mathbf{z}^{(l)}) \quad \text{where } \tilde{r}_l = \frac{\tilde{p}(\mathbf{z}^{(l)})}{\tilde{q}(\mathbf{z}^{(l)})}$$

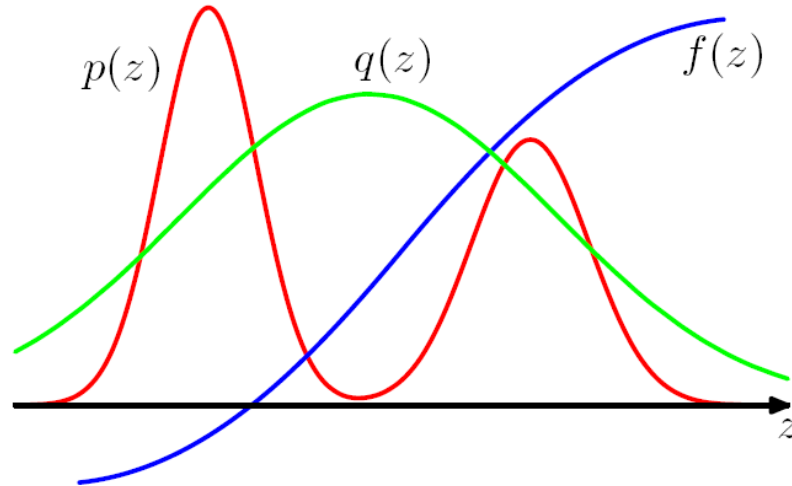
◆ The ratio $\frac{Z_p}{Z_q} = \frac{1}{Z_q} \int \tilde{p}(\mathbf{z})d\mathbf{z} = \int \frac{\tilde{p}(\mathbf{z})}{\tilde{q}(\mathbf{z})}q(\mathbf{z})d\mathbf{z} \approx \frac{1}{L} \sum_{l=1}^L \tilde{r}_l$

◆ Then, the expectation is

$$\mathbb{E}[f] \approx \sum_{l=1}^L w_l f(\mathbf{z}^{(l)}), \quad \text{where } w_l = \frac{\tilde{r}_l}{\sum_m \tilde{r}_m}$$

Problems with Importance Sampling

- ◆ As with Rejection sampling, the performance depends crucially on how well the proposal matches the target



- a lot of wastes in the areas where $p(z)f(z)$ is small
- more serious in high dimensional spaces

Summary so far ...

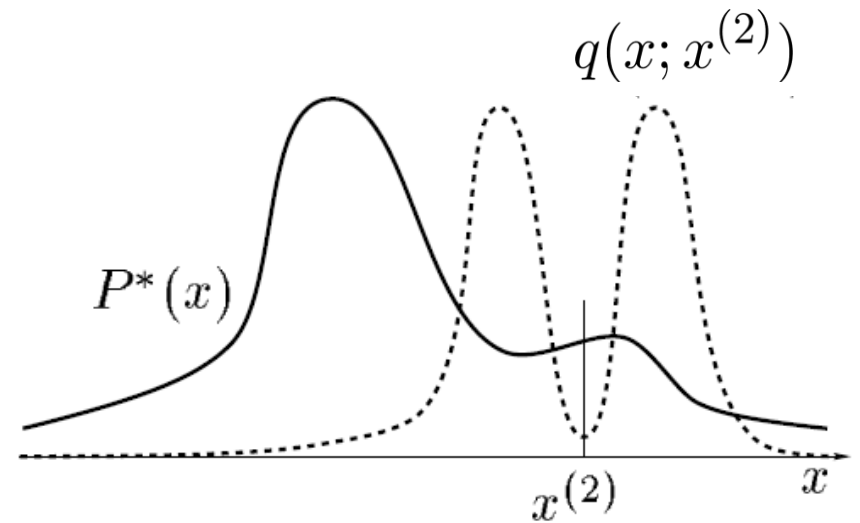
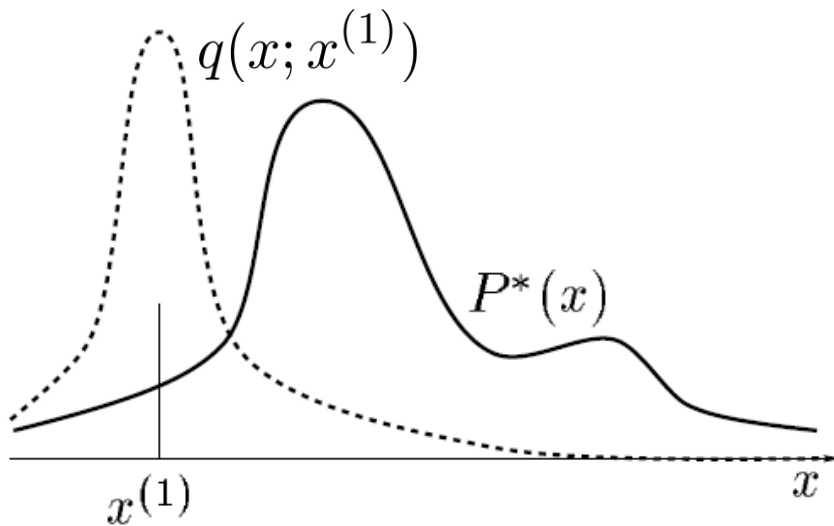
- ◆ Monte Carlo methods use **samples** to estimate **expectations**
- ◆ **Rejection sampling** and **importance sampling** are useful when no closed-form transformation is available or is hard
- ◆ But they can be inefficient in high-dimensional spaces
 - **only works well when the proposal approximate the target well**

Markov Chain Monte Carlo (MCMC)

- ◆ As with rejection and importance sampling, it samples from a **proposal** distribution
- ◆ But, it maintains a record of \mathbf{z}^T , and the **proposal distribution depends on current state** $q(\mathbf{z}|\mathbf{z}^T)$
- ◆ It's not necessary for the proposal to look at all similar to the target
- ◆ The sequence $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots$ forms a **Markov chain**
- ◆ Configurable components:
 - Proposal distribution
 - Accept strategy

Geometry of MCMC

- ◆ Proposal depends on current state
- ◆ Not necessarily similar to the target
- ◆ Can evaluate the un-normalized target



Metropolis Algorithm

- ◆ Proposal distribution is symmetric

$$q(\mathbf{z}|\mathbf{z}') = q(\mathbf{z}'|\mathbf{z})$$

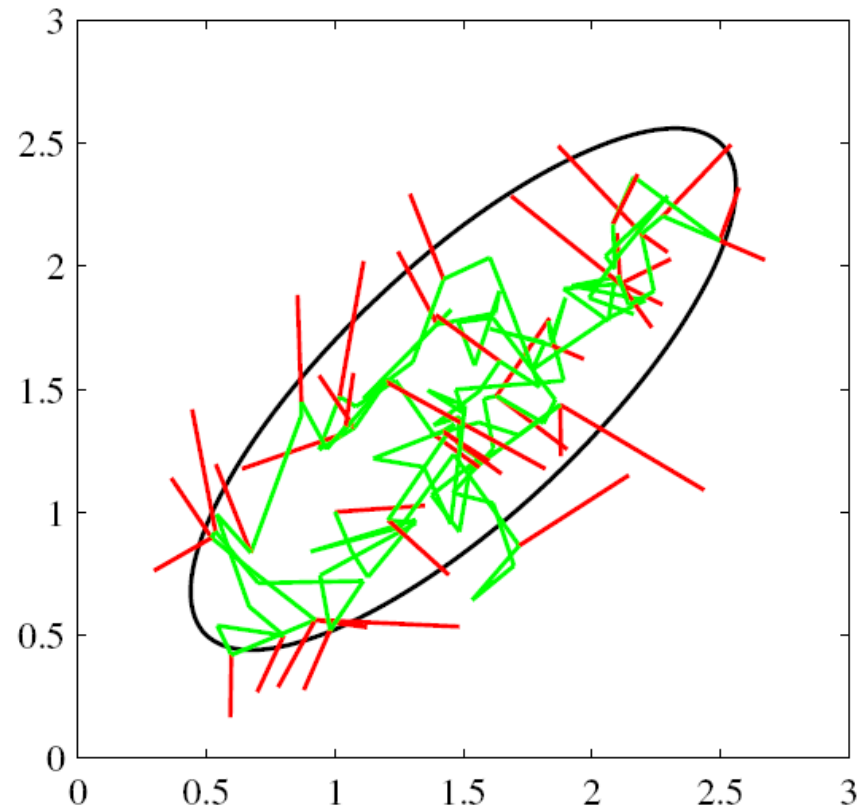
- ◆ The candidate sample \mathbf{z}^* is accepted with probability

$$A(\mathbf{z}^*, \mathbf{z}^{(\tau)}) = \min\left(1, \frac{\tilde{p}(\mathbf{z}^*)}{\tilde{p}(\mathbf{z}^{(\tau)})}\right)$$

- The acceptance can be done by
 - draw a random $u \sim \text{Uniform}(0, 1)$
 - accepting the sample if $A(\mathbf{z}^*, \mathbf{z}^{(\tau)}) > u$
- ◆ If sample is accepted, set $\mathbf{z}^{(\tau+1)} = \mathbf{z}^*$; otherwise $\mathbf{z}^{(\tau+1)} = \mathbf{z}^{(\tau)}$
- ◆ **Note:** $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots$ is not a set of independent samples

Geometry of Metropolis Algorithm

- ◆ Sample from Gaussian distribution with the proposal being an isotropic Gaussian with std 0.2.
- ◆ **Green:** accepted steps; **Red:** rejected steps



Properties of Markov Chains

- ◆ $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}, \dots, \mathbf{z}^{(M)}$ is a *first-order Markov chain* if conditional independence property holds

$$p(\mathbf{z}^{(m+1)} | \mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}) = p(\mathbf{z}^{(m+1)} | \mathbf{z}^{(m)})$$

- ◆ *Transition probabilities* $T_m(\mathbf{z}^{(m)}, \mathbf{z}^{(m+1)}) \triangleq p(\mathbf{z}^{(m+1)} | \mathbf{z}^{(m)})$

- ◆ If T_m are the same for all m , it is *Homogeneous* Markov chain

- ◆ $p^*(\mathbf{z})$ satisfies the *detailed balance* if

$$p^*(\mathbf{z})T(\mathbf{z}, \mathbf{z}') = p^*(\mathbf{z}')T(\mathbf{z}', \mathbf{z})$$

- ◆ If $p^*(\mathbf{z})$ satisfies the detailed balance, then it's *invariant (stationary)*

$$p^*(\mathbf{z}) = \sum_{\mathbf{z}'} T(\mathbf{z}', \mathbf{z})p^*(\mathbf{z}')$$

- ◆ A chain is *ergodic* if it converges to the invariant distribution, irrespective of the initial distribution

Metropolis-Hasting Algorithm

- ◆ A generalization of the Metropolis algorithm to the case where the proposal distribution is no longer symmetric
- ◆ Draw sample $\mathbf{z}^* \sim q_k(\mathbf{z}|\mathbf{z}^{(\tau)})$ and accept it with probability

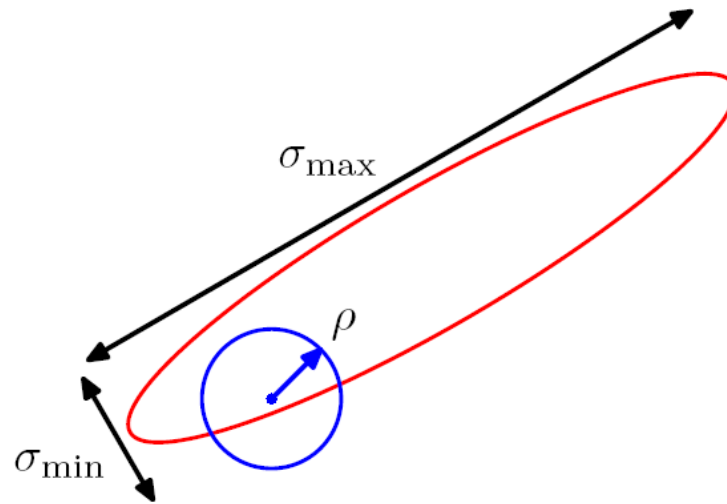
$$A_k(\mathbf{z}^*, \mathbf{z}^{(\tau)}) = \min \left(1, \frac{\tilde{p}(\mathbf{z}^*)q_k(\mathbf{z}^{(\tau)}|\mathbf{z}^*)}{\tilde{p}(\mathbf{z}^{(\tau)})q_k(\mathbf{z}^*|\mathbf{z}^{(\tau)})} \right)$$

- ◆ We can show $p(\mathbf{z})$ is an invariant distribution of MC defined by MH algorithm, by showing the detailed balance

$$\begin{aligned} p(\mathbf{z})q_k(\mathbf{z}'|\mathbf{z})A_k(\mathbf{z}', \mathbf{z}) &= \min \left(p(\mathbf{z})q_k(\mathbf{z}'|\mathbf{z}), p(\mathbf{z}')q_k(\mathbf{z}|\mathbf{z}') \right) \\ &= \min \left(p(\mathbf{z}')q_k(\mathbf{z}|\mathbf{z}'), p(\mathbf{z})q_k(\mathbf{z}'|\mathbf{z}) \right) \\ &= p(\mathbf{z}')q_k(\mathbf{z}|\mathbf{z}') \min \left(1, \frac{p(\mathbf{z})q_k(\mathbf{z}'|\mathbf{z})}{p(\mathbf{z}')q_k(\mathbf{z}|\mathbf{z}')} \right) \\ &= p(\mathbf{z}')q_k(\mathbf{z}|\mathbf{z}')A_k(\mathbf{z}, \mathbf{z}') \end{aligned}$$

Issues with Proposal Distribution

◆ **Proposal:** isotropic Gaussian (blue) centered at current state



- ❑ Small ρ leads to high accept rate, but progress through the state space takes a long time due to **random walk**
- ❑ Large ρ leads to high rejection rate
- ❑ Roughly best choice: $\rho \approx \sigma_{\min}$

Gibbs Sampling

◆ A special case of Metropolis-Hastings algorithm

◆ Consider the distribution $p(\mathbf{z}) = p(z_1, \dots, z_M)$

◆ Gibbs sampling performs the follows

□ Initialize $\{z_i : i = 1, \dots, M\}$

□ For $\tau = 1, \dots, T$

• Sample $z_1^{(\tau+1)} \sim p(z_1 | z_2^{(\tau)}, z_3^{(\tau)}, \dots, z_M^{(\tau)})$

⋮

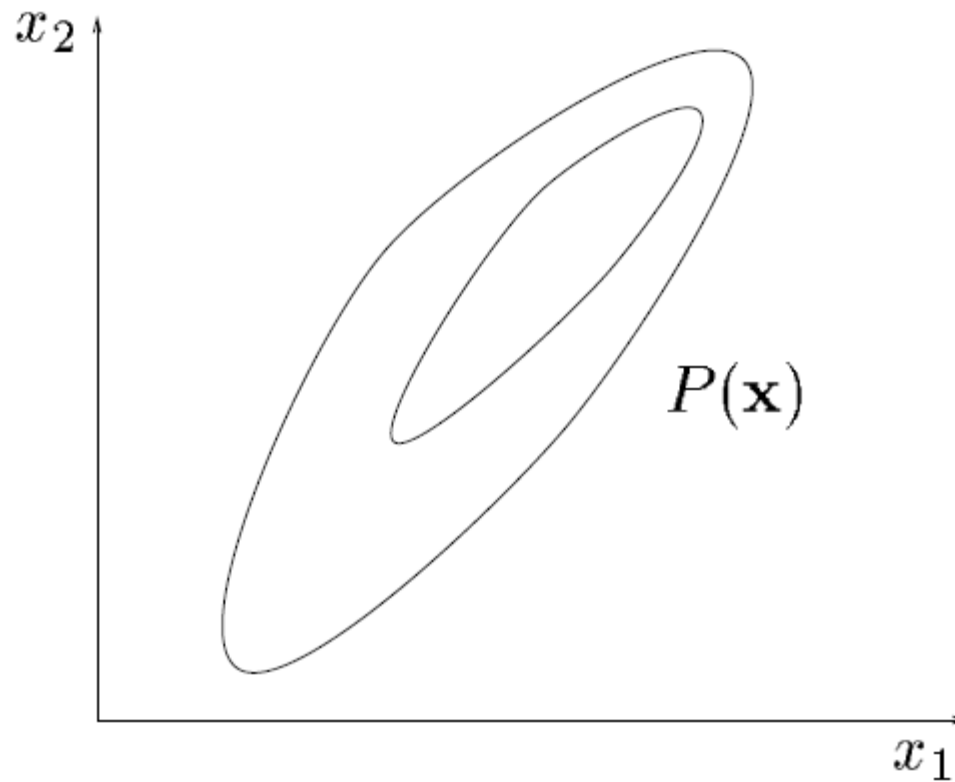
• Sample $z_j^{(\tau+1)} \sim p(z_j | z_1^{(\tau+1)}, \dots, z_{j-1}^{(\tau+1)}, z_{j+1}^{(\tau)}, \dots, z_M^{(\tau)})$

⋮

• Sample $z_M^{(\tau+1)} \sim p(z_M | z_1^{(\tau+1)}, z_2^{(\tau+1)}, \dots, z_{M-1}^{(\tau+1)})$

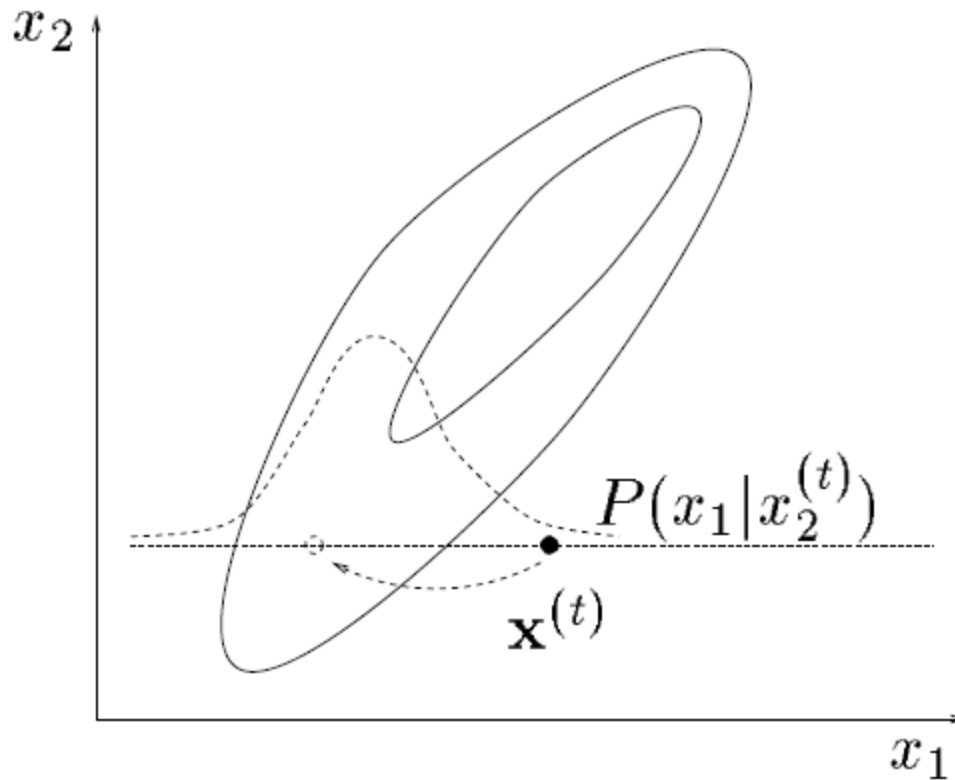
Geometry of Gibbs Sampling

- ◆ The target distribution in 2 dimensional space



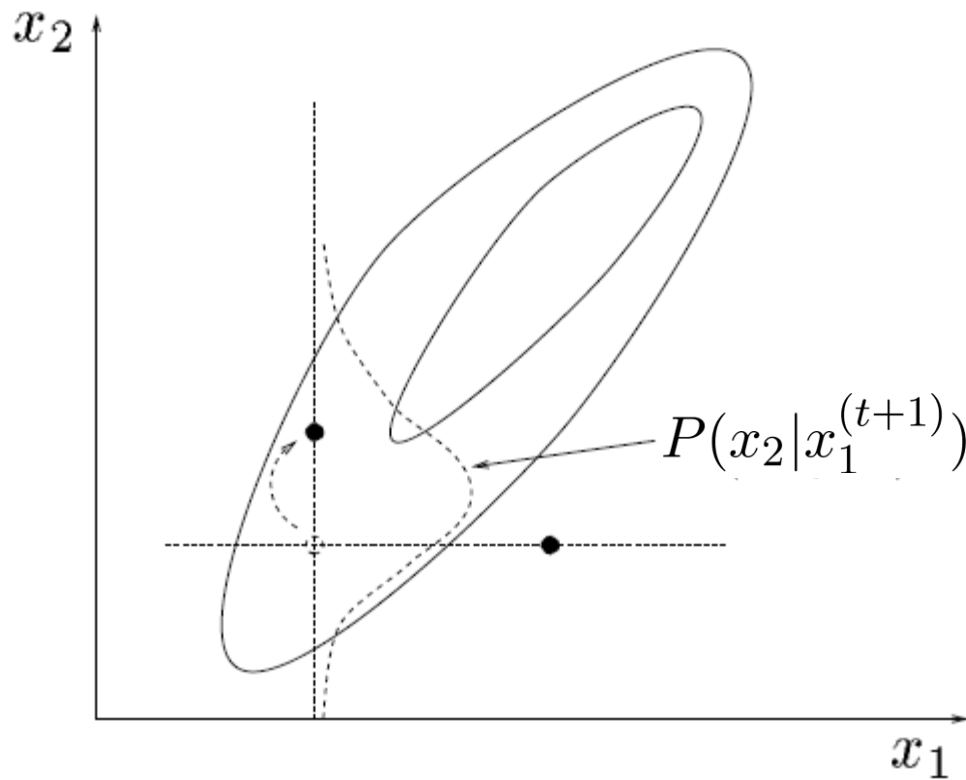
Geometry of Gibbs Sampling

- ◆ Starting from a state $\mathbf{x}^{(t)}$, $x_1^{(t+1)}$ is sampled from $P(x_1|x_2^{(t)})$



Geometry of Gibbs Sampling

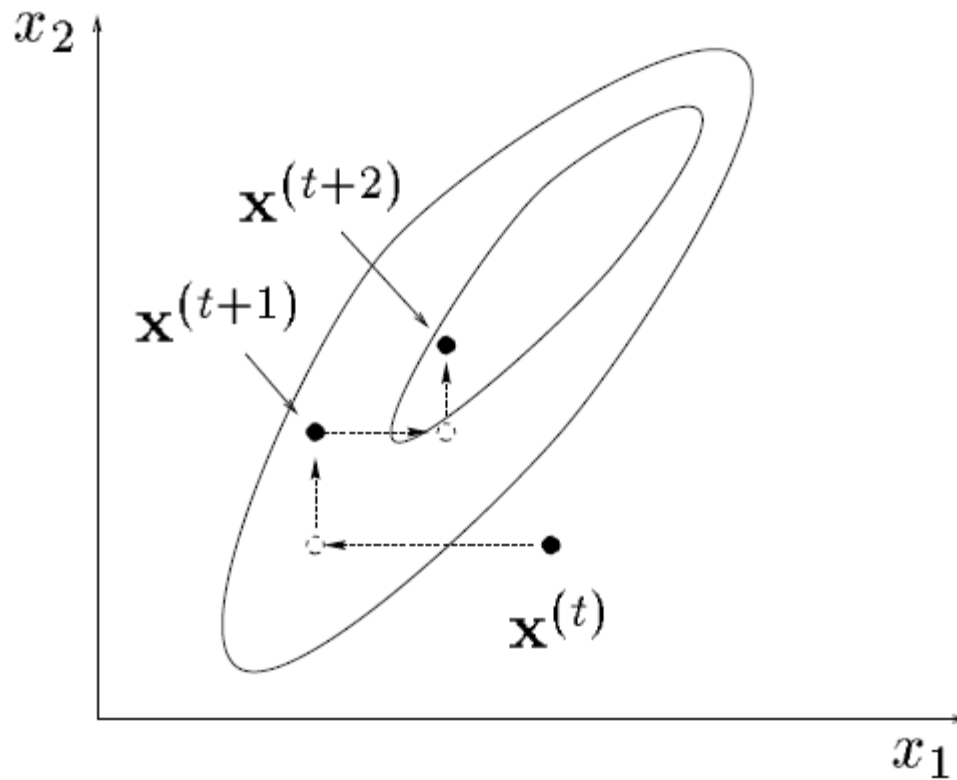
- ◆ A sample is drawn from $P(x_2|x_1^{(t+1)})$



this finishes one single iteration.

Geometry of Gibbs Sampling

◆ After a few iterations



Gibbs Sampling

- ◆ How to show Gibbs sampling samples from $p(\mathbf{z})$?
 - show that $p(\mathbf{z})$ is an invariant distribution at each sample steps
 - The marginal $p(\mathbf{z}_{-i})$ is invariant as \mathbf{z}_{-i} is unchanged
 - Also, the conditional $p(z_i|\mathbf{z}_{-i})$ is correct
 - Thus, the joint distribution $p(z_i|\mathbf{z}_{-i})p(\mathbf{z}_{-i})$ is invariant at each step
 - the Markov chain is ergodic
 - A sufficient condition is that none of the conditional distributions be anywhere zero
 - If the requirement is not satisfied (some conditionals have zeros), ergodicity must be proven explicitly

Gibbs Sampling

- ◆ a special case of Metropolis-Hastings algorithm
- ◆ Consider a MH sampling step involving variable z_k in which other variables \mathbf{z}_{-k} remain fixed
- ◆ The transition probability is

$$q_k(\mathbf{z}^*|\mathbf{z}) = p(z_k^*|\mathbf{z}_{-k})$$

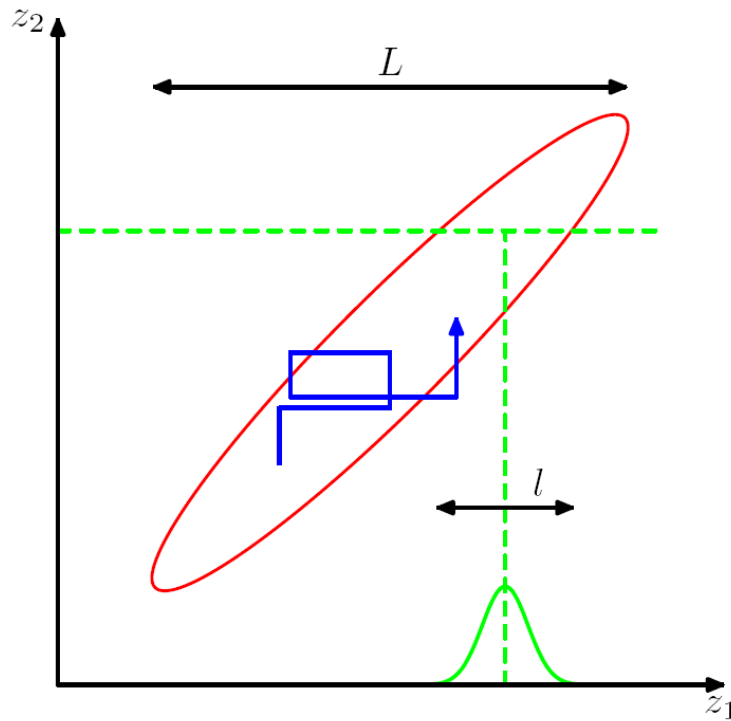
- ◆ Note that $\mathbf{z}_{-k}^* = \mathbf{z}_{-k}$ and $p(\mathbf{z}) = p(z_k|\mathbf{z}_{-k})p(\mathbf{z}_{-k})$
- ◆ Then, the MH acceptance probability is

$$A(\mathbf{z}^*, \mathbf{z}) = \frac{p(\mathbf{z}^*)q_k(\mathbf{z}|\mathbf{z}^*)}{p(\mathbf{z})q_k(\mathbf{z}^*|\mathbf{z})} = \frac{p(z_k^*|\mathbf{z}_{-k}^*)p(\mathbf{z}_{-k}^*)p(z_k|\mathbf{z}_{-k}^*)}{p(z_k|\mathbf{z}_{-k})p(\mathbf{z}_{-k})p(z_k^*|\mathbf{z}_{-k})} = 1$$

- always accepted!

Behavior of Gibbs Sampling

- ◆ **Correlated Gaussian:** marginal distributions of width L and conditional distributions of width l



Summary

- ◆ Monte Carlo methods are power tools that allow one to implement any distribution in the form

$$p(\mathbf{x}) = p^*(\mathbf{x})/Z$$

- ◆ Monte Carlo methods can answer virtually any query related to by putting the query in the form

$$\int f(\mathbf{x})p(\mathbf{x})d\mathbf{x} \approx \frac{1}{L} \sum_l f(\mathbf{x}^{(l)})$$

- ◆ In high-dimensional problems the only satisfactory methods are those based Markov chain Monte Carlo: Metropolis-Hastings and Gibbs sampling
- ◆ Simple Metropolis and Gibbs sampling algorithms, although widely used, may suffer from slow random walk. More sophisticated algorithms are needed.

Sampling and EM Algorithm

- ◆ General procedure of the EM algorithm

- **E-step**: compute the expected complete-data log-likelihood

$$Q(\theta, \theta^{\text{old}}) = \int p(\mathbf{Z}|\mathbf{X}, \theta^{\text{old}}) \ln p(\mathbf{X}, \mathbf{Z}|\theta) d\mathbf{Z}$$

- **M-step**: update model parameters

$$\theta^{\text{new}} = \arg \max_{\theta} Q(\theta, \theta^{\text{old}})$$

- ◆ Sampling methods can be applied to approximate the integral in E-step

- called **Monte Carlo EM** algorithm

$$Q(\theta, \theta^{\text{old}}) \approx \frac{1}{L} \sum_{l=1}^L \ln p(\mathbf{X}, \mathbf{Z}^{(l)}|\theta)$$

References

- ◆ Chap. 11 of Pattern Recognition and Machine Learning, Bishop, 2006
- ◆ Introduction to Monte Carlo Methods. D.J.C. MacKay, 1998