Lecture 28
MCMC Sampling
ECEN 5283 Computer Vision

Dr. Guoliang Fan
School of Electrical and Computer Engineering
Oklahoma State University

Goals

- To review two direct sampling methods involved in Monte Carlo methods

- To introduce the Markov chain for statistical modeling

- To study two Markov Chain Monte Carol (MCMC) sampling methods

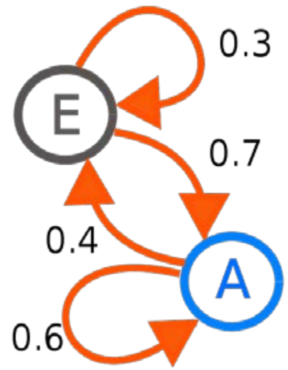
Sampling Methods

- Direct Sampling
  - *Importance sampling*
    - Step 1: Generate samples from a proposal Q(x).
    - Step 2: Weight each sample by the weight computed by $\omega_r \equiv \dfrac{P^*(x^{(r)})}{Q^*(x^{(r)})}$
    - Step 3: Use normalized weighted samples to represent the unknown density.
  - *Rejection sampling*
    - Step 1: Generate a sample from Q(x)
    - Step 2: Generate U from unit(0,1)
    - Step 3: Accept the sample if $U < \dfrac{P(x)}{cQ(x)}$
- Markov Chain Monte Carlo (MCMC) Sampling
  - Metropolis sampling
  - Gibbs sampling

Markov Chain

- A discrete Markov chain is a mathematical system that transits from one state to another in a chainlike manner.

$$\begin{bmatrix} 0.3 & 0.7 \\ 0.4 & 0.6 \end{bmatrix}$$

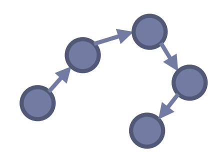


A simple two-state Markov chain.


Andreyevich Markov
Russia mathematician
(1856-1922) (Wikipedia)

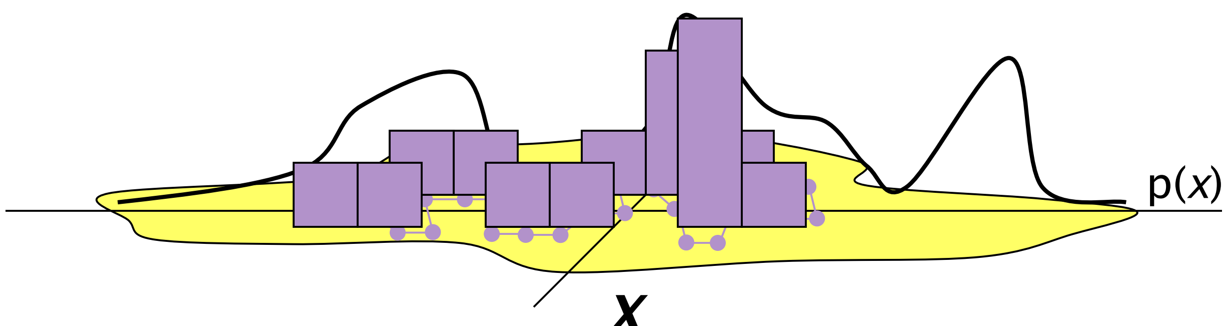
- A continuous Markov chain involves a continuous distribution (i.e., Gaussian) centered around the previous state from which the current state is drawn.

$$z^{(m+1)} \sim p(z \mid z^{(m)}) = N(z \mid z^{(m)}, \sigma^2)$$



Markov Chain Monte Carlo (MCMC)

- Suppose that it is hard to sample p(*x*) but that it is possible to "walk around" in *X* using only local state transitions
- **Insights:**
  - We can use a "random walk" to help us draw random samples from $p(x)$
  - What we need is a proposal function that allows us to move locally based on the previous sample.
  - Samples are accepted or rejected according to the evaluation result.
  - The way that samples are evaluated determines the final sample distribution.



Metropolis Sampling: Algorithm

- Given current state $x^{(\tau)}$, draw a new sample x^* according to symmetric proposal $q(x^{(A)} \mid x^{(B)}) = q(x^{(B)} \mid x^{(A)}) = q(\left| x^{(A)} - x^{(B)} \right|)$

$$x^* \sim q(x \mid x^{(\tau)})$$

- Then the candidate sample x^* is then accepted with probability

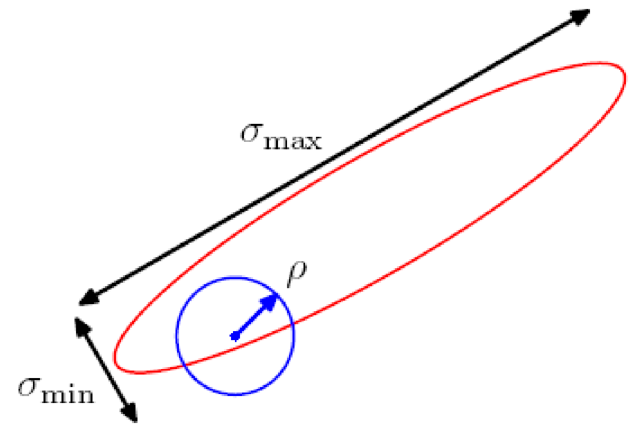
$$\min\left(1, \frac{p(x^*)}{p(x^{(\tau)})}\right)$$

  - If accepted, then $x^{(\tau+1)} = x^*$
  - Otherwise, x^* is discarded, $x^{(\tau+1)} = x^{(\tau)}$
  - Another candidate is drawn from distribution, and so on …

- Unlike rejection sampling, where rejected samples are discarded (wasted), the previous sample is included in the final sample list.

Metropolis Sampling: Proposals

- A common choice of the proposal is a Gaussian centered on the current state, leading to an important trade-off in determining the variance parameter of this distribution.
  - If the variance is too small, then the proportion of accepted transitions with be high, but progress through the state space takes the form of slow random walk leading to long correlation times.
  - It the variance is large, then the rejection rate will be high, leading to an inefficient sampling process.



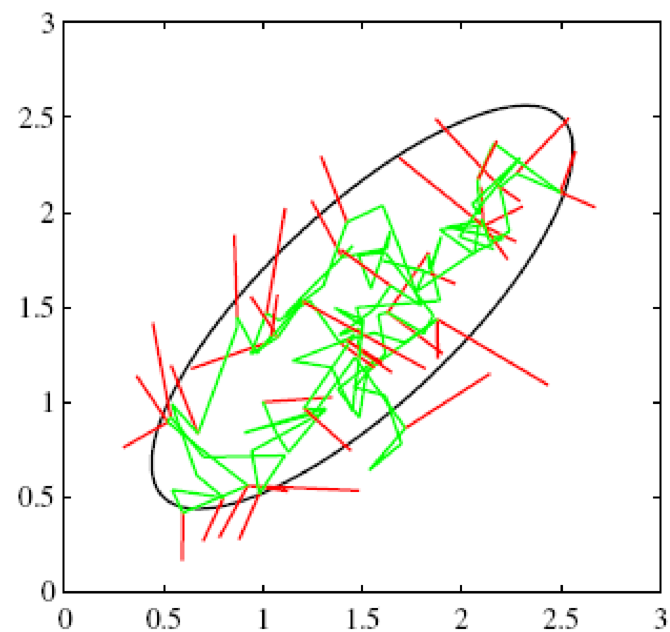
Schematic illustration of the use of an isotropic Gaussian proposal distribution (blue circle) to sample from a correlated multivariate Gaussian distribution (red ellipse) having very different standard deviations in different directions, using the Metropolis-Hastings algorithm. In order to keep the rejection rate low, the scale ρ of the proposal distribution should be on the order of the smallest standard deviation $\sigma_{\min}$, which leads to random walk behaviour in which the number of steps separating states that are approximately independent is of order $(\sigma_{\max}/\sigma_{\min})^2$ where $\sigma_{\max}$ is the largest standard deviation.

Metropolis Sampling: Example

- The general idea of the algorithm is to generate a series of samples that are correlated in a Markov chain and that can be used to match the unknown distribution $p(x)$.

$$(x^{(1)}, x^{(2)}, \ldots, x^{(\tau)}) \xrightarrow{\tau \to \infty} p(x)$$

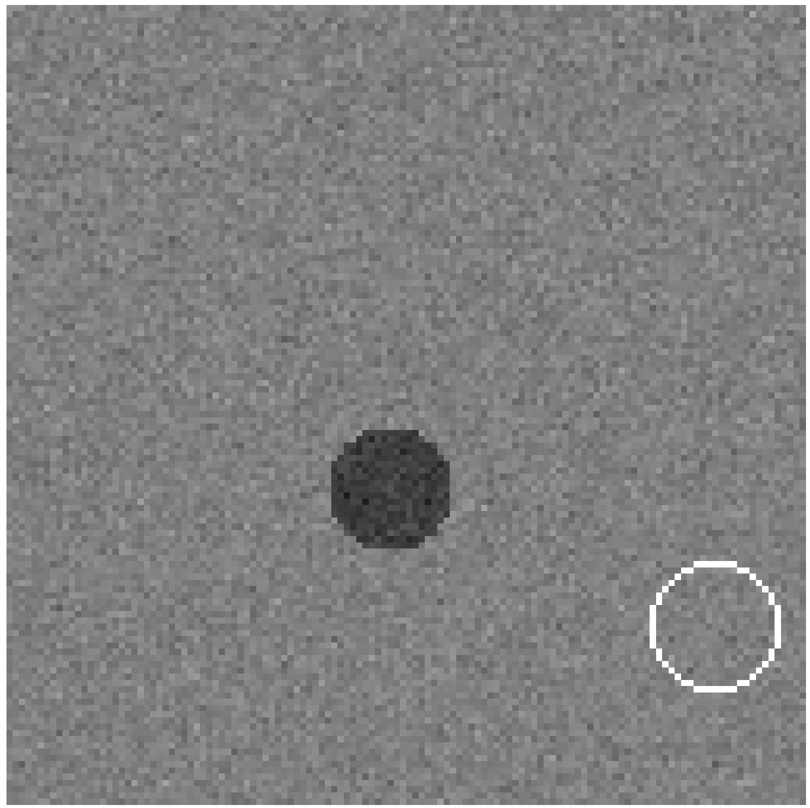
Figure 11.9 A simple illustration using Metropolis algorithm to sample from a Gaussian distribution whose one standard-deviation contour is shown by the ellipse. The proposal distribution is an isotropic Gaussian distribution whose standard deviation is 0.2. Steps that are accepted are shown as green lines, and rejected steps are shown in red. A total of 150 candidate samples are generated, of which 43 are rejected.



What ensures that final samples will follow the unknown distribution?

Pattern Recognition and Machine Learning, Christopher M. Bishop, Springer, page 539.

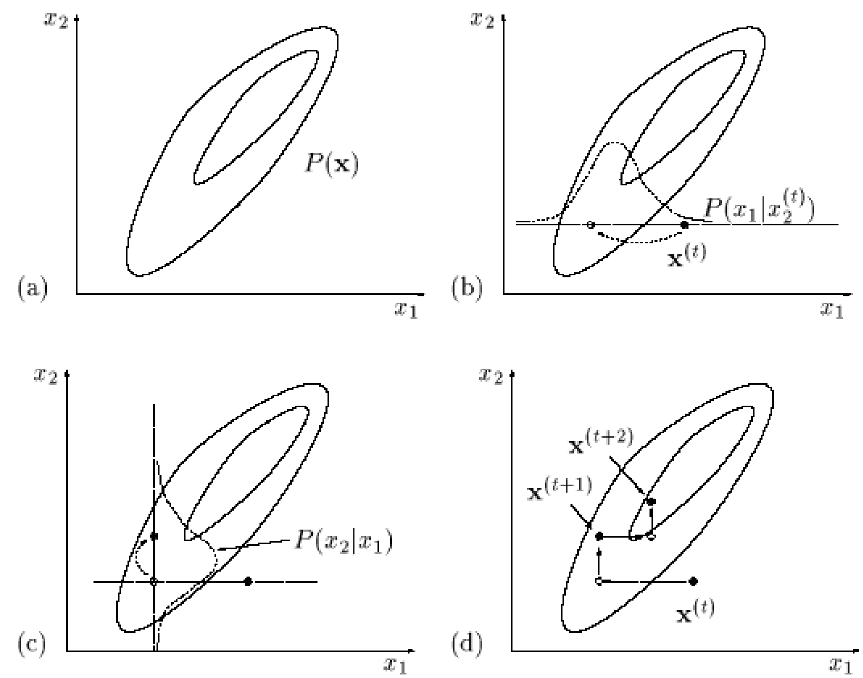
Metropolis Sampling Example



Why we need to accept a move with a probability?

Gibbs Sampling

- What is Gibbs sampling?
  - It is an algorithm to generate a sequence of samples from the joint probability distribution of two or more random variables.



- Why do we need Gibbs sampling?
  - The purpose is to approximate the joint distribution of multiple variables by sampling each variable individually and sequentially.

Gibbs Sampling Algorithm

Gibbs Sampling

1. Initialize $\{z_i : i = 1, \ldots, M\}$ — Number of random variables
2. For $\tau = 1, \ldots, T$: — Number of MCMC steps
   - Sample $z_1^{(\tau+1)} \sim p(z_1 | z_2^{(\tau)}, z_3^{(\tau)}, \ldots, z_M^{(\tau)})$.
   - Sample $z_2^{(\tau+1)} \sim p(z_2 | z_1^{(\tau+1)}, z_3^{(\tau)}, \ldots, z_M^{(\tau)})$.

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

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

Gibbs Sampling Example