Lecture 29
Jump Diffusion MCMC
ECEN 5283 Computer Vision

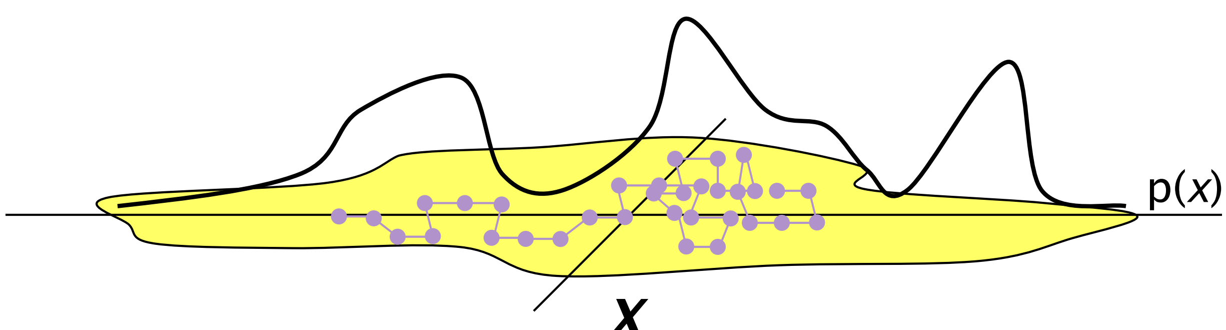
Dr. Guoliang Fan
School of Electrical and Computer Engineering
Oklahoma State University

Goal

- To review a few important sampling methods involved in Monte Carlo methods
- To present the jump diffusion MCMC (JD-MCMC) method for a disconnected parameter space.
- To apply JD-MCMC for object detection

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.
  - A new sample is accepted or rejected according to the ratio between the evaluation of the present sample over that of previous one.
  - The way that samples are evaluated determines the final sample distribution.

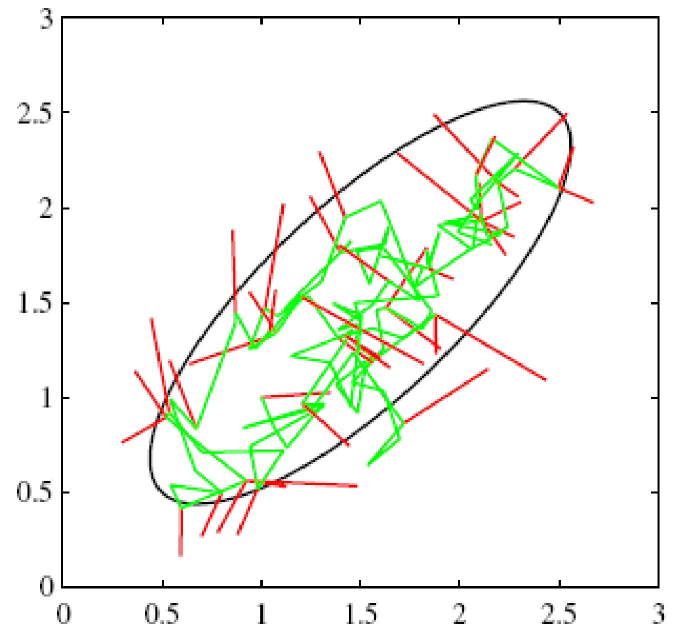


Metropolis Sampling Example

- 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.



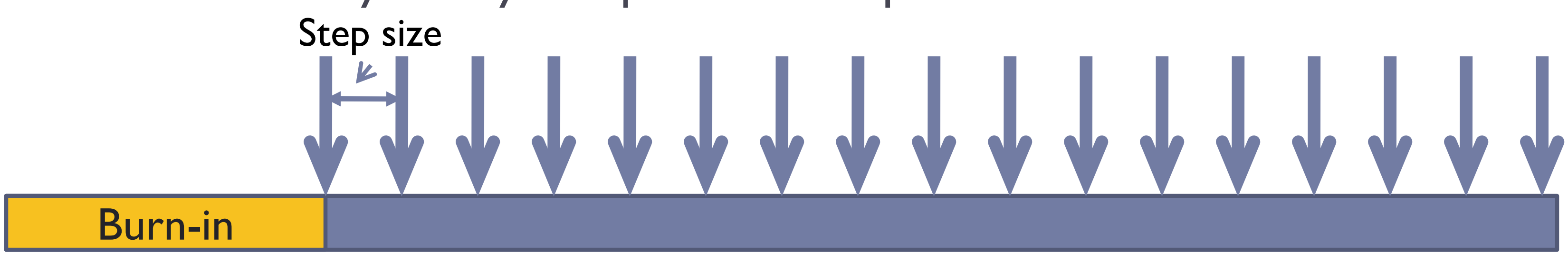
What ensures that final samples will follow the unknown distribution?

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

Additional Issues about MCMC

- Burn-in:
  - We start to collect samples that are generated after some iterations, making sure that MCMC sampling have a good starting point.

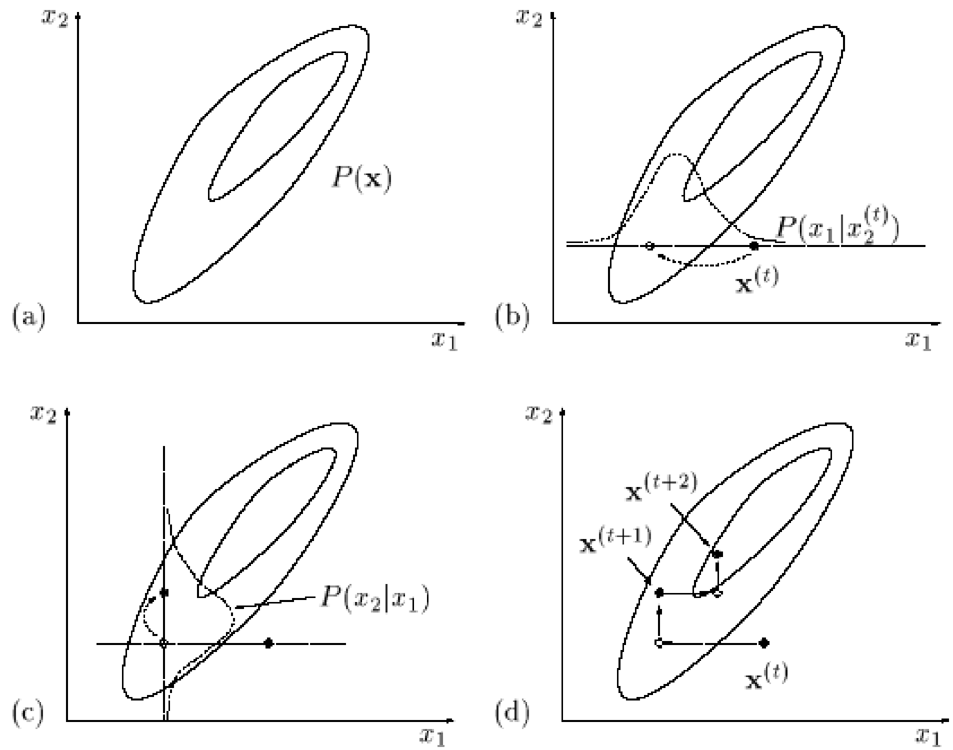
- Step size
  - Since MCMC neighboring samples could be high-correlated, a step size is needed to only nearly independent samples.



- Mean estimation
  - The final solution is the mean of samples picked under certain step size.

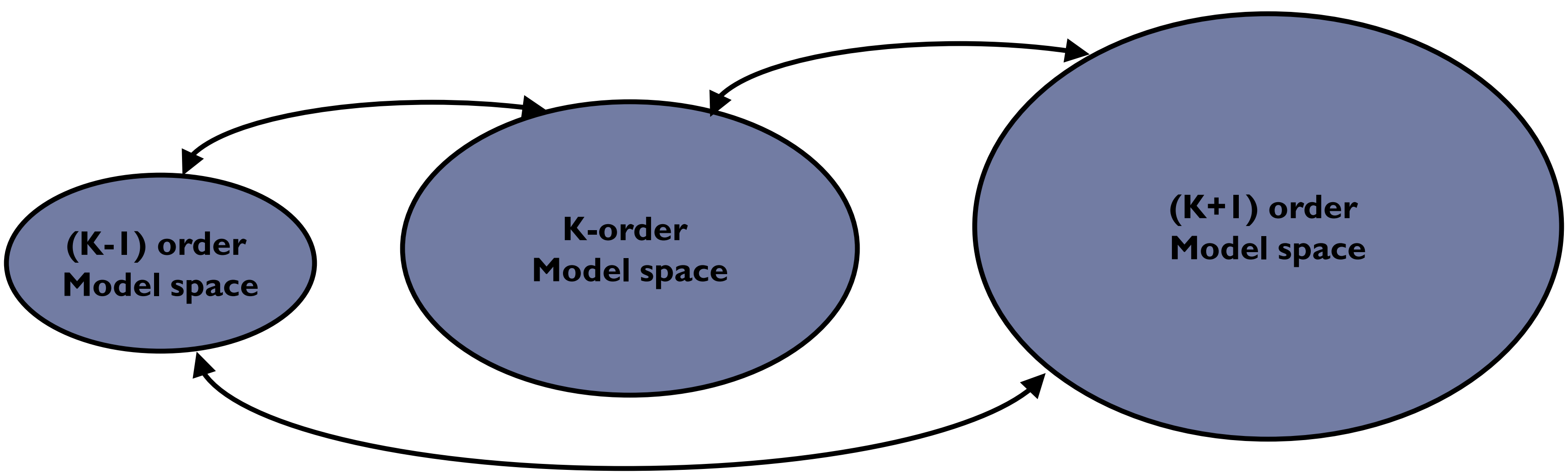
Gibbs Sampling

- It is an algorithm to generate a sequence of samples from the joint probability distribution of two or more random variables. The purpose is to approximate the joint distribution of multiple variables.

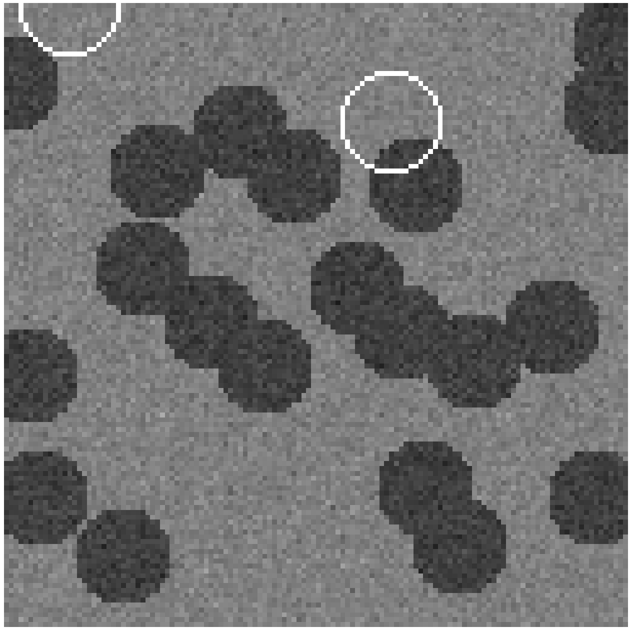


Jump-Diffusion MCMC

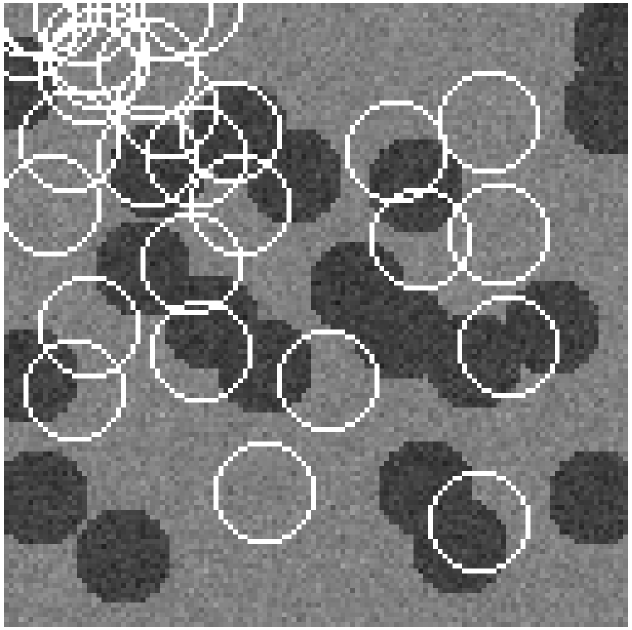
- Jump-diffusion provide a mixed mechanism to draw samples from a disconnected state space where both discrete and continuous state variables exist.
  - Jump contributes in sampling over the parameter size.
  - Diffusion contributes in sampling over the parameter values.



JD-MCMC for Object Detection



$$k_0 = 2$$

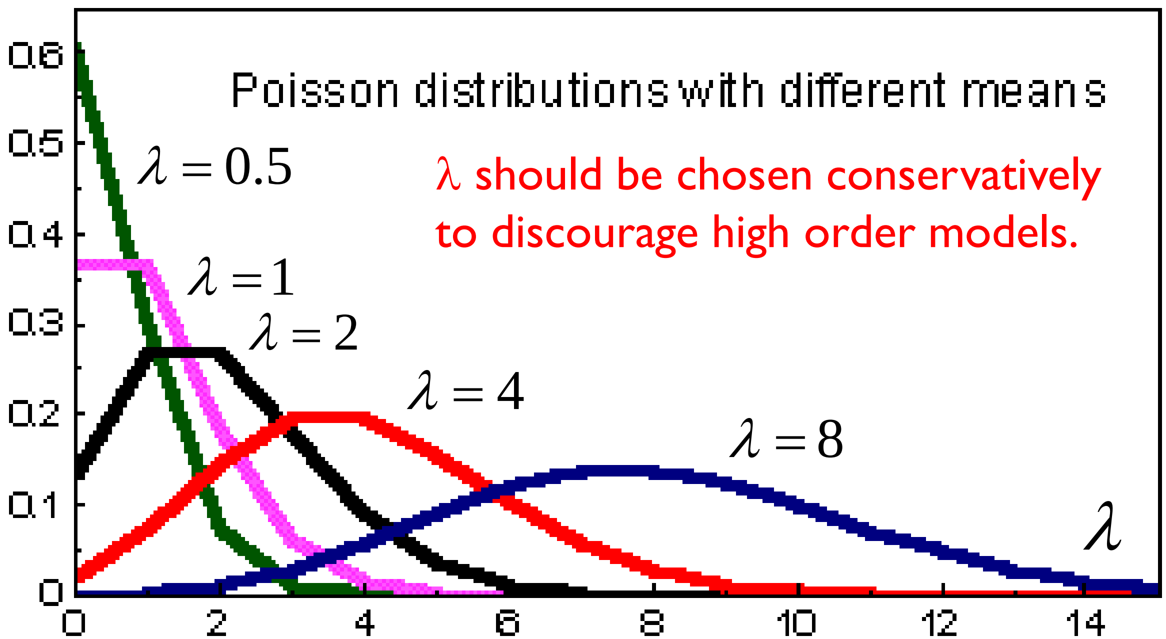


$$k_0 = 30$$

Model Order Estimation

- If we assume a prior probability of models of different orders as a Poisson distribution as

$$P(K^* = k) = \frac{\lambda^k}{e^{\lambda} k!} \qquad \text{mean } \lambda \quad \text{variance } \lambda$$



The new objective function

- Given a prior probability of model order *k* and an observed image Y, the solution of object detection (i.e., Θ: object locations) is represented by the joint posterior probability density as:

$$\boxed{p(\Theta, k \mid Y)} \propto \underbrace{p(Y \mid \Theta)}_{\text{data likelihood given model }\Theta} \underbrace{P(k)}_{\text{Prior for model order k}}$$

- See detailed derivation from Slide 13 of Lecture 25.

Application to Object Detection

- There are two kinds of parameters
  - The number of objects, *k*,
  - The location of each object

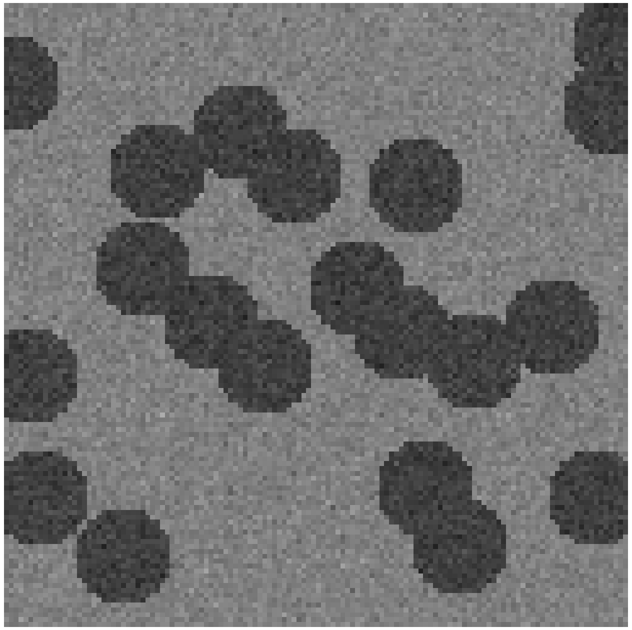
$$\Theta_k = \{(x_i, y_i) \mid i = 1, \ldots, k\}$$

- Two probabilistic functions
  - The prior probability of model order

$$P(K^* = k) = \frac{\lambda^k}{e^{\lambda} k!}$$

  - The likelihood of object detection given the order number

$$p(Y \mid \Theta_k)$$



Jump Diffusion MCMC Algorithm

- Initialize locations of k hypothesized objects and the maximum order K_{max}.
- for i=1:N
  - Draw a sample a~U(0,1)
  - If a<0.33 and k>1 (jump by -1)
    - k=k-1;
    - MCMC Gibbs sampling
    - Accept or reject by Metropolis Sampling
  - else if a<0.66 and k<Kmax (jump by +1)
    - k=k+1;
    - MCMC Gibbs sampling
    - Accept or reject by Metropolis Sampling
  - else (no jump)
    - MCMC diffusion (Gibbs sampling)
    - Accept or reject by Metropolis Sampling
- End
- Select samples after M iterations (burn-in);
- Obtain a set of samples with certain step size.
- Compute the mean estimate of object number and the location of each one.

$$p(\Theta, k \mid Y) \propto p(Y \mid \Theta) P(k)$$

PDF of interest used for evaluation

$$\alpha = \min\left(\frac{p(Y \mid \Theta^{(B)}) P(k^{(B)})}{p(Y \mid \Theta^{(A)}) P(k^{(A)})}, 1 \right)$$

(acceptance probability for jump)

$$\beta = \left(\frac{p(Y \mid \Theta^{(B)})}{p(Y \mid \Theta^{(A)})}, 1 \right)$$

(acceptance probability for no jump)