EE 621: Markov Chains and Queueing Systems
MCMC Application

Divyam Bapna
160070038

Spring 2019

1 Introduction

I have done two applications of Markov Chains Monte-Carlo, both of them use Metropolis-Hastings Markov Chain Monte Carlo for posterior probability calculation for acceptance-rejection. One application is for generating random variables of Inverse-Gamma distribution, other one is denoising of an image.

2 Inverse-Gamma Distribution

The markov chains is generated using a normal distribution with variance 4 as

$$X_0 \sim \mathcal{N}(0,4)$$
$$X_k \sim \mathcal{N}(x_{k-1},4); k \geq 1$$

The acceptance probability is found using the density function as $min\{1, \frac{f(x_k)}{f(x_{k-1})}\}$.
x_{k-1} is the k^{th} accepted sample, starting from x_0.
Comparing this posterior probability function with a uniformly generated random variable between [0,1] we do the sampling of the inverse-gamma random variable if the posterior probability is more than the uniform random variable.
The inverse gamma distribution has a density function f(x) as

$$f(x) = \frac{\beta^{\alpha}(1/x^{\alpha})exp(-beta/x)}{\Gamma(\alpha)}$$

The sampled inverse-gamma distribution random variable for parameters $\alpha = 2$ and $\beta = 1$ is shown in the below normalized histogram.

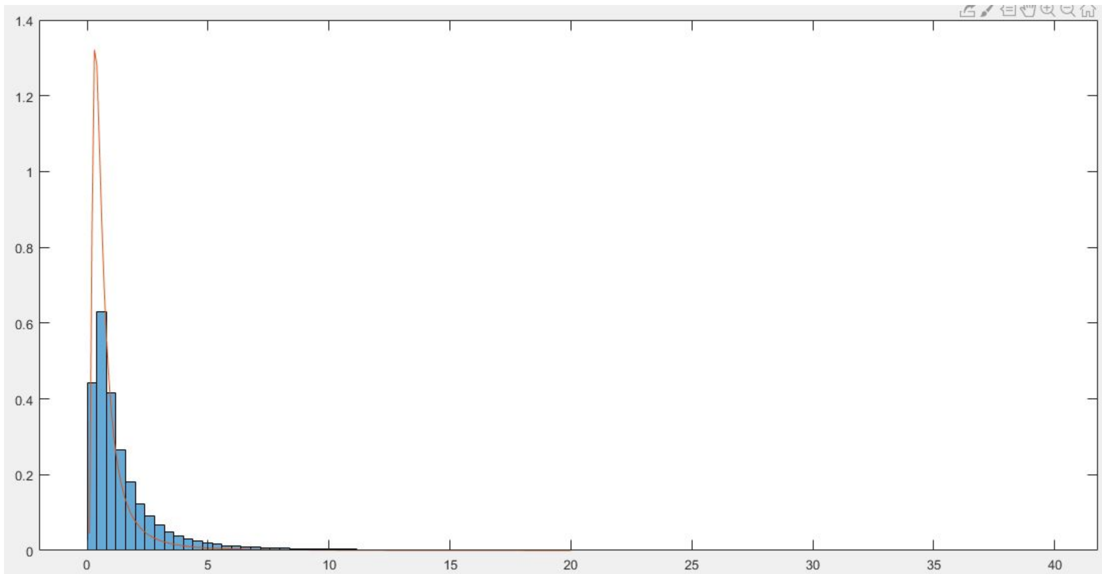

Figure 1: Sampled Random Variable vs the density function

3 Image Denoising

In this, I took a binary image of Lenna then randomly flipped some pixels with a certain probability π ($= 0.15$ here). This operation gives the below result.

Figure 2: Original vs Noisy Image

Here we compute the posterior probability of flipping a pixel selected at random. The probability is a function of the image that we start with, and to a small neighbourhood(3x3) from the last modified image around the current pixel location which is to be flipped or not.
We start with just the noisy image (X) and the image we get in each iteration is Z_k, where Z_0 = X.
The prosterior probability is given as

$$\frac{p(Z_k|X,\beta,\gamma)}{p(Z_{k-1}|X,\beta,\gamma)} = exp(-2\gamma X(i,j)Z_{k-1}(i,j) - 2\beta Z_{k-1}(i,j)(\sum_{x=1}^{3}\sum_{y=1}^{3} Z_{k-1}(i-2+x, j-2+y) - Z_{k-1}(i,j)))$$

Here β (= 1) indicates how much similarity of the center pixel, we consider with the neighbouring pixels. And $\gamma = 0.5log(\frac{1-\pi}{\pi})$.
A pixel location is randomly chosen and for that pixel posterior probability, if flipped, is computed and compared with a randomly generated number between [0,1].
The final result obtained is given below compared with the noisy image.

Figure 3: Noisy vs Obtained Image