STT 7020 Project Report: Exploring Markov Chain Monte Carlo Methods

Matt Piekenbrock

College of Engineering and Computer Science, Wright State University

Due date: April 20, 2016

1 Introduction

Markov Chain Monte Carlo methods are often used for analyzing a large, highly dimensional state space, with which there are two primary applications:

1. Approximating the expected value of a posterior distribution
2. Sampling values that approximately follow the stationary distribution of a Markov Chain

In both cases, the distribution being sampled is often a multi-dimensional integral. While constructing a Markov Chain and implementing one of several random walk algorithms isn't considered to be difficult, there a large area of research that exists exploring how to reduce the number of steps/computational cost required to converge to the stationary distribution of the Markov Chain within some level of error. When an MCMC sampling method converges in distribution[1] quickly, the chain is said to have *rapid mixing*. Mark Chain Monte Carlo methods that exhibit rapid mixing for many general Markov Chains are often used in computer science to approximate solutions to several problems, some of which I will outline in this report.

2 Definition

From Introduction to Probability Models (10^{th} edition)[12], Markov Chain Monte Carlo (MCMC) methods are defined as methods that are used for approximating the expected value of a difficult function to integrate. Specifically, if X is a discrete random vector with possible values x_j, $j \geq 1$, and the probability mass

[1]With convergence being determined with respect to some convergence criterion, such the Geweke diagnostic

function (PMF) of X is given by $P(X = x_j), j \geq 1$, Markov Chain Monte Carlo methods can be used to approximate:

$$\theta = E[h(X)] = \sum_{j=1}^{\infty} h(x_j)P(X = x_j)$$

for some function h. It is important to note, however, that MCMC methods are often used in Machine Learning and Computer Science classes as a method of using a Monte Carlo simulation to generate samples from a posterior distribution that are approximately representative of the stationary distribution, and thus the samples can be used to approximate the distribution.[7]

3 Implementation

I implemented the Gibbs sampler in R and proceeded in demonstrating how to implement the problem in example 4.40 from the textbook Introduction to Probability Models.[12] The problem is stated as follows:

”Suppose that we want to generate n uniformly distributed points in a circle with radius 1 centered at the origin, conditional on the event that no two points are within a distance d of each other, when the probability of this conditioning event is small. This can be accomplished by using the Gibbs sampler as follows. Start with any n points $x_1, \ldots, x_n$ in the circle that have the property that no two of them are within d of the other; then generate the value of I, equally likely to be any of the values $1, \ldots, n$. Then continually generate a random point in the circle until you obtain one that is not within d of any of the other $n - 1$ points excluding x_I. At this point, replace x_I by the generated point and then repeat the operation. After a large number of iterations of this algorithm, the set of n points will approximately have the desired distribution.”

I implemented the Gibbs sampler and ran a simulation in R to test out this problem. I arbitrarily choose an $n = 8$ and I varied d to observe the side effects. After running the Gibbs sampler for approximately 10,000 iterations, saving the samples that were generated that followed the rule of being at least d away from any other point in the $n = 8$ sample (and replacing x_I along the way), I ended up with the following:

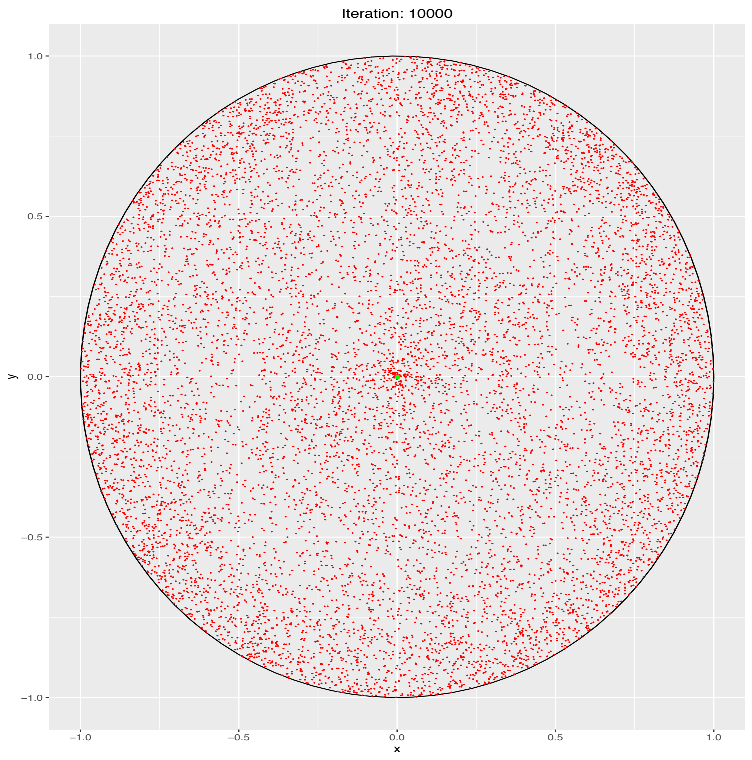

Figure 1: 10,000 iterations

I proceeded in varying the distance threshold d that each point needed to away from each other, and then again instead of discarding the points at each iteration, I recorded them. Although it's not mentioned in the text, to approximate the joint density function I used a kernel density estimator to both approximate and visualize the density in a easier way than by simply plotting several points. The results are as follows:

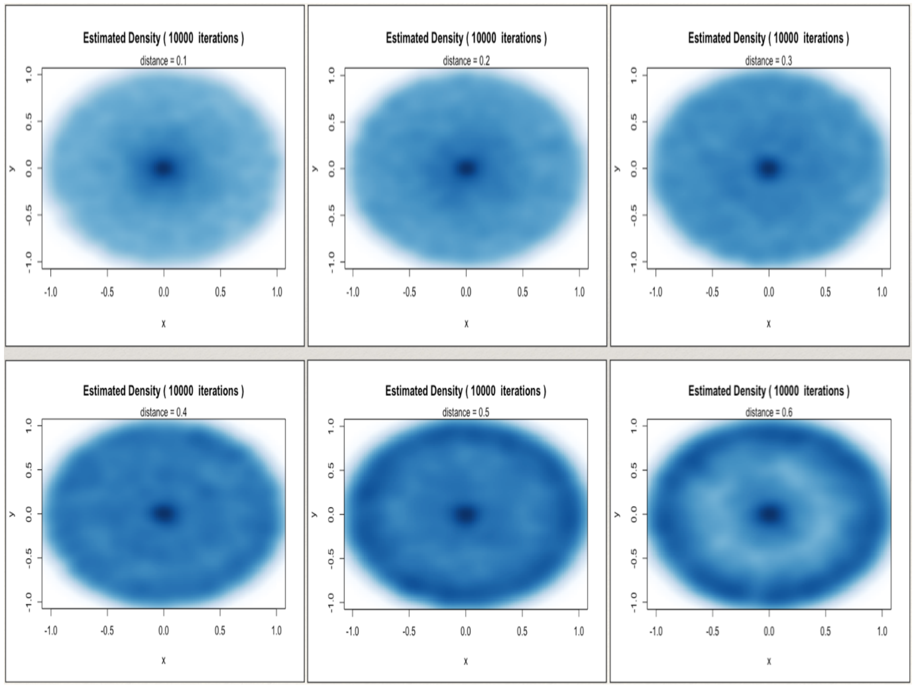

Figure 2: Approximated densities with Kernel Density Estimator

Notice that although points always accumulate towards the center of the circle, the restriction of d has a large effect on the approximated density. Specifically, as d approaches higher values, the restriction that all points must be at least d away from each other becomes more difficult (computationally) to simulate, as more randomly generated points fail to meet the distance criterion. As a result, the densities start to aggregate towards the outside of the circle, as the majority of the possible combinations where $n = 8$ points are at least d away from each other starts to be a small number proportionally for point combinations where points exist within the middle ring of the circle.

The code that I used to generate these circles, along with the gibbs sampler and comments describing the functionality of the program, is included with this report.

4 Convergence Diagnostics

There are a number of techniques for improving or altering the basic Metropolis Hastings or Gibbs sampling techniques. Some examples include reversible-jump MCMC, modeling the Markov Chain as a birth-and-death process, or even using a Double Metropolis Hasting algorithm with a alternative algorithm for accepting candidate states[1][3][9]. Regardless of the sampler that is used, however, it is often a necessity to use some type of diagnostic test of convergence to give a degree confidence or confirmation of convergence in the MCMC chain's representation of the stationary distribution[8]. Of course, the only way of ensuring your chain has converged would be to simply sample every possible value in the support, but this is often computationally infeasible in practice. In the following sections, I outline some of the convergence diagnostics that exist for Markov Chain Monte Carlo methods.

4.1 Geweke Diagnostic

The Geweke Diagnostic[5] is a easy-to-implement diagnostic for a single chain to approximate how well the chain is mixing. To use the Geweke Diagnostic, the means of two non-overlapping proportions of the Markov Chain are compared using a difference of means test to assess whether or not they came form the same distribution. The computation of Geweke's diagnostic, given a Markov Chain $G(j)$ is defined as:

$$\hat{g}_p^A = p_A^{-1} \sum_{j=1}^{p_A} G(j)$$

for chain A, and:

$$\hat{g}_p^B = p_B^{-1} \sum_{j=p^*}^{p} G(j) \text{ where } p^* = p - p_B + 1$$

for chain B. Geweke then goes on the define $\hat{S}_G^A(0)$ and $\hat{S}_G^B(0)$ as the spectral density estimates[5] for $G(j), j = 1, \ldots, p_A$ and $G(j), j = p^*, \ldots, p$ respectively. Assuming the ratios of $\frac{p_a}{p}$ and $\frac{p_b}{p}$ are fixed and $\frac{(p_A+p_B)}{p} < 1$, then as $p \to \infty$, Geweke's convergence diagnostic is defined as:

$$(\bar{g}_p^A - \bar{g}_p^B)/[p_A^{-1}\hat{S}_G^B(0) + p_B^{-1}\hat{S}_G^B(0)]$$

In many implementations in R, Geweke's diagnostic compares the first 10% of the Markov Chain samples with the last 50% of the samples for an individual chain.

4.2 Raftery and Lewis Diagnostic

Rather than diagnosing the convergence of the MCMC chain, this diagnostic is used in determining the number of iterations (+burn-in iterations) needed to estimate a quantile of the stationary distribution with some specified degree of accuracy. It is defined as follows: For some posterior quantile of interest q, some acceptable tolerance of accuracy of being in this quantile r, and some probability s, of the MCMC chain being within this quantile $(q \pm r)$, a pilot Markov Chain must be run for n_{min} iterations, where:

$$n_{min} = [\Phi^{-1}(\frac{s+1}{2})\frac{\sqrt{q(1-q)}}{r}]^2$$

Where Φ^{-1} is the inverse of a standard normal CDF.

After running a pilot chain for n_{min} iterations, it is possible to compute the number of burnin iterations M and the number of Markov Chain iterations N needed for the MCMC chain to estimate the quantile of interest as mentioned above. The calculation of M and N is omitted here, but can be found in the original paper [11]. It's important to note that, as with every diagnostic check, there are a few draw backs to the Raftery and Lewis diagnostic, specifically:

1. This diagnostic only provides information on a specific quantile, not on the convergence of the chain as a whole[2]
2. There is a dependence factor $I = \frac{N}{n_{min}}$ that quantifies the proportional increase in the number of iterations estimated from serial dependence, and thus there are specific cases when I is high that indicates the chain will likely not converge given the specific MCMC parametrization

4.3 Gelman and Rubin Diagnostic

The Gelman and Rubin diagnostic[4] is a diagnostic that depends on multiple chains, so in order to utilize it, you must run independent, asynchronous MCMC chains that eahc do n draws. After the initial burnin period, you calculate the within-chain variance:

$$W = \frac{1}{m}\sum_{j=1}^{m} s_j^2$$

where

$$s_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}(\theta_{ij} - \bar{\theta_j})^2$$

s_j^2 is defined as the variance of the j^{th} chain and W is the mean of the variances of each chain. The between chain variance for the Gelman and Rubin diagnostic is defined as:

$$B = \frac{n}{m-1}\sum_{j=1}^{m}(\bar{\theta_j} - \bar{\bar{\theta}})^2$$

where

$$\bar{\bar{\theta}} = \frac{1}{m}\sum_{j=1} m\bar{\theta}_j$$

The variance of the chain averages are multiplied by the number of draws from the posterior n that each chain takes Bearing both the between and the within chain variance, an estimate of the stationary distribution can be computed using the following weighted average between W and B:

$$\hat{Var}(\theta) = (1 - \frac{1}{n})W + \frac{1}{n}B$$

The final component of the Gelman and Rubin Diagnostic is the Potential Scale Reduction Factor. This measure is defined as the following:

$$\hat{R} = \sqrt{\frac{V\hat{a}r(\theta)}{W}}$$

This is used on each parameter used to run the chain. From this formula, since the goal is to try to see some stability between the MCMC chains, it's apparent that the lower the $\hat{R}$ value is, the more likely it is that the simulation itself is mixing well. It's also apparent that, theoretically (unless the proposal variance-covariance is too high or small), each potential scale reduction factor (PSRF) will become smaller over time.

It's important to note that this diagnostic is often used to measure the mixing of mcmc chains, as it's available *CODA*[10] and STATNET[6], which are popular R frameworks that are utilized to implement or diagnose MCMC chains as a means of approximating a solution to other problems such as, for example, approximating model parameters of exponential random graph models.

References

[1] SP Brooks, Px Giudici, and A Philippe. Nonparametric convergence assessment for mcmc model selection. *Journal of Computational and Graphical Statistics*, 12(1):1–22, 2003.

[2] SP Brooks and GO Roberts. Miscellanea. on quantile estimation and markov chain monte carlo convergence. *Biometrika*, 86(3):710–717, 1999.

[3] Olivier Cappé, Christian P Robert, and Tobias Rydén. Reversible jump, birth-and-death and more general continuous time markov chain monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(3):679–700, 2003.

[4] Andrew Gelman and Donald B Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, pages 457–472, 1992.

[5] J Geweke. Evaluating the accuracy of sampling0based approaches to the calculation of posterior moments, bayesian statistics 4, ed. *JM Bernardo, JO Berger, AP David, and AFM Smith*, 1690193, 1992.

[6] Mark S. Handcock, David R. Hunter, Carter T. Butts, Steven M. Goodreau, and Martina Morris. *statnet: Software tools for the Statistical Modeling of Network Data.* Seattle, WA, 2003.

[7] Lukas Kroc. Introduction to markov chain monte carlo.

[8] Patrick Lam. Lecture notes in convergence diagnostics.

[9] Faming Liang. A double metropolis–hastings sampler for spatial models with intractable normalizing constants. *Journal of Statistical Computation and Simulation*, 80(9):1007–1022, 2010.

[10] Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. Coda: Convergence diagnosis and output analysis for mcmc. *R news*, 6(1):7–11, 2006.

[11] Adrian E Raftery and Steven M Lewis. [practical markov chain monte carlo]: comment: one long run with diagnostics: implementation strategies for markov chain monte carlo. *Statistical science*, 7(4):493–497, 1992.

[12] Sheldon M Ross. *Introduction to probability models.* Academic press, 2014.