SIMULATION OF A MARKOV CHAIN USING THE METROPLOLIS ALGORITHM

DOMINIC ESSUMAN

ABSTRACT. A good choice of the proposal distribution is crucial for the rapid convergence of the Metropolis algorithm. In this paper, the Cauchy C(0,1) distribution is used as the target distribution. Two proposed distributions were selected; the normal pdf with mean 0 and variance 2 and the t pdf with degree of freedom 20 We tend to find how the simulated Markov chain depend on the variance of the proposed distribution.

INTRODUCTION

Markov Chain Monte Carlo (MCMC) methods encompass a general framework of methods introduced by Metropolis et al. and Hastings for Monte Carlo integration. Many applications of Markov Chain Monte Carlo methods are problems that arise in Bayesian inference because in practice, the normalizing constant for posterior densities are often difficult to evaluate. Markov Chain Monte Carlo provides a method for integrating mathematically intractable equations that are difficult to compute by numerical methods, especially in higher dimensions.
The algorithm of Metropolis et al. (1953) is an important tool in statistical computation, especially in calculation of posterior distributions arising in Bayesian statistics. The Metropolis algorithm evaluates a (typically multivariate) target distribution $\pi()$ by generating a Markov chain whose stationary distribution is π. Practical implementations often suffer from slow mixing and therefore inefficient estimation, for at least two reasons: the jumps are too short and therefore simulation moves very slowly to the target distribution; or the jumps end up in low probability areas of the target density, causing the Markov chain to stand still most of the time. In practice, adaptive methods have been proposed in order to tune the choice of the proposal, matching some criteria under the invariant distribution (e.g. Kim, Shephard, and Chib (1998), Haario, Saksman, and Tamminen (1999), Laskey and Myers (2001), Andrieu and Robert (2001), and Atchade and Rosenthal (2003)). These criteria are usually defined based on theoretical optimality results, In general, the adaptive proposal Metropolis algorithms do not simulate exactly the target distribution: We shall for this algorithm where we use the cauchy distribution C(0,1) as the target distribution and the normal distribution with mean 0 and variance 2 and the t distribution with degree of freedom 20 as our selected proposed distributions

Date: December 4, 2019.
1991 *Mathematics Subject Classification.* STATISTICAL COMPUTING.

OBJECTIVE OF STUDY

The objective of this study is to simulate a Markov chain using the metropolis independence sampler algorithm to simulate a Markov chain where the Cauchy distribution C(0,1) is used as the target distribution. The normal distribution with mean 0 and variance 2 and the t distribution with degree of freedom 20 are selected as the proposed distribution.

ANALYSIS

Metropolis-Independence Sampler Algorithms. The independence sampler is another form of the metropolis Hastings sampler. The proposal distribution in the independence sampling algorithm does not depend on the previous value of the chain that is the random variable Y is independent of X_t
The Metropolis-algorithm generates the Markov chain $\{X_0, X_1, \ldots\}$ as follows:

- Choose a proposal distribution $g(\cdot|X_t)$ with the same support set as the target distribution.
- Generate X_0 from a distribution g.
- Repeat (until the chain has converged to a stationary distribution according to some criteria):
  - Generate Y from a distribution $g(\cdot|X_t)$.
  - Generate U from $Uniform(0,1)$.
  - If
    $$U \leq \frac{f(Y)g(X_t|Y)}{f(X_t)g(Y|X_t)}$$
    accept Y and set $X_{t+1} = Y$; otherwise set $X_{t+1} = X_t$.
  - Increment t.

Note in third step candidate point Y is accepted with probability

$$\alpha(X_t, Y) = \min\left(1, \frac{f(Y)g(X_t)}{f(X_t)g(Y)}\right) \tag{0.1}$$

so that it is only necessary to know the density of the target distribution f up to a constant Y.
The independence sampler is easy to implement and tends to work well when the proposed density is close match to the target density.

0.1. Implementing the algorithm using the normal distribution as the proposed distribution.

The algorithm is implemented in the R and the probability of acceptance was calculated to be 0.5580673.
To check whether the generated values are independent (necessary for a random sample), we computed the auto correlation for 100 adjacent values (or equivalently $10000/100 = 100$ lags). It is easier to just check plot the auto correlation values.

To plot the auto correlation, we use the function 'acf' in the '$VGAM$' package in R.

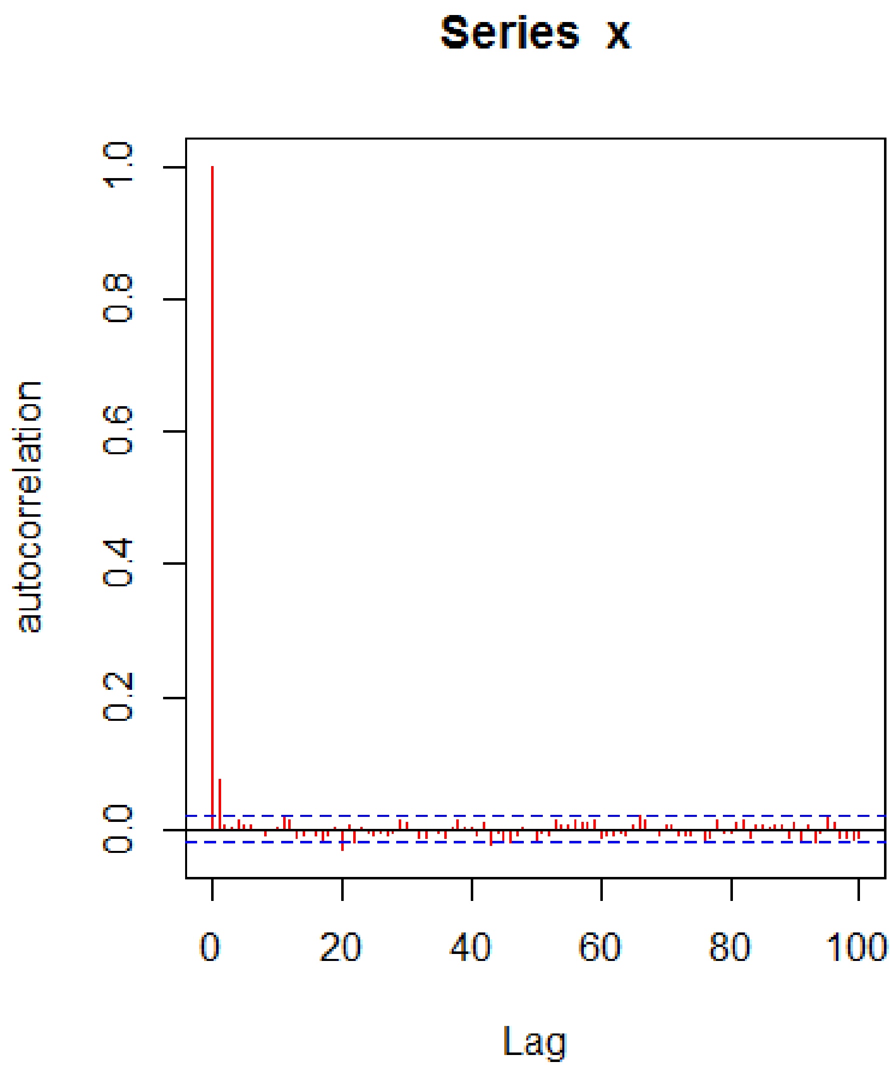

FIGURE 1. Autocorrelation plot

The highly correlated values in the first part of the chain tells us that the chain has not converged yet. We refer these early values as burning sample. The first burning sample from the auto correlation sample plot is discarded. From the auto correlation plot above, we see that the auto correlation started to hover around 0 after the $200th$ (or $2th$ lag) value. With this, we discard the first 200 values in the chain.

The last 9800 values can now be treated as the generated values from the Cauchy distribution.

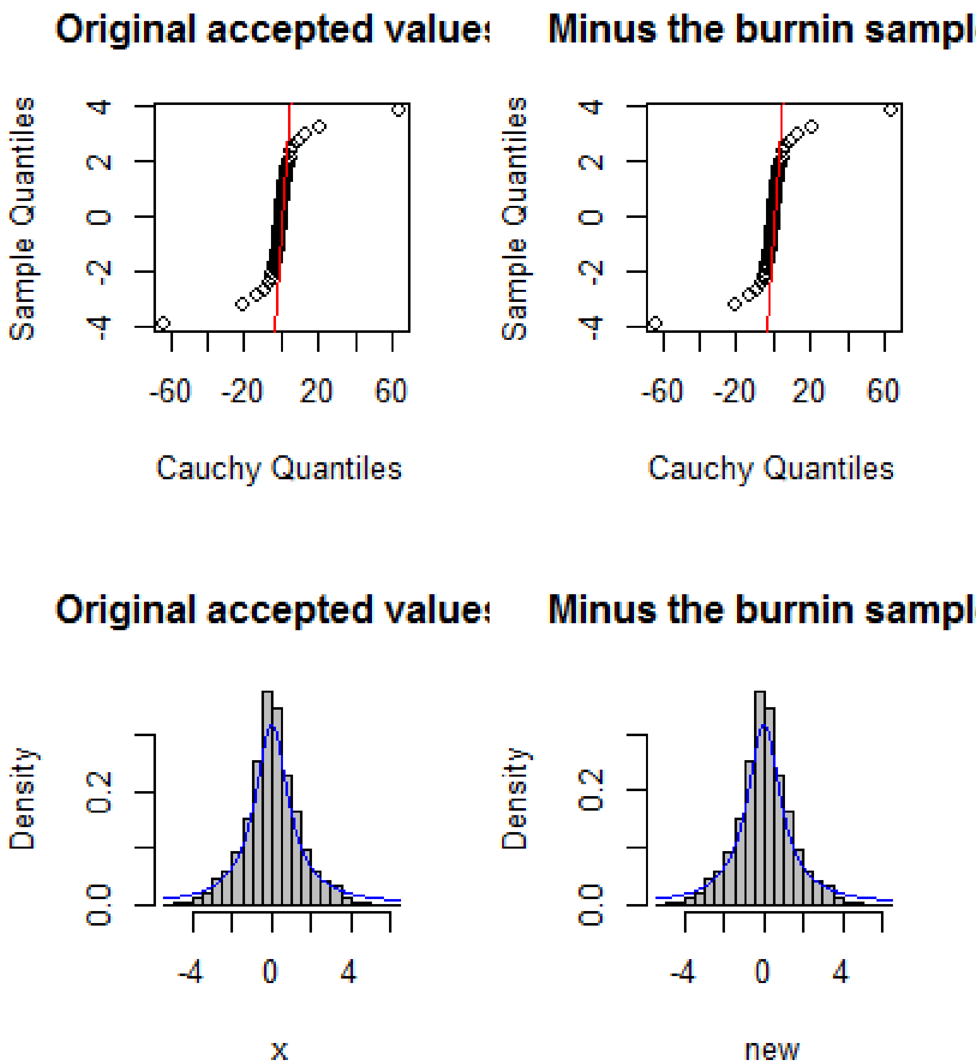

FIGURE 2. qqplot and histogram

The histogram of the generated sample after discarding the burning sample is shown in figure 2. The mean of the remaining sample is 0.005525871.

0.2. **Implementing the algorithm using the normal distribution as the proposed distribution.** Implementing the same algorithm with the t distribution with degree of freedom 20 the probability of acceptance was calculated to be 0.8824849.
The auto correlation plot is shown in figure 3

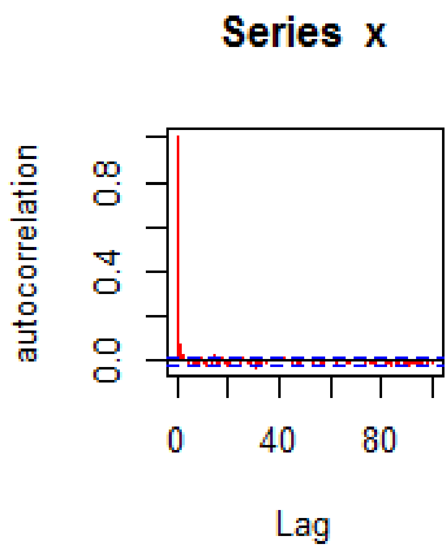

FIGURE 3. Autocorrelation plot

The first burning sample from the auto correlation sample plot is discarded. From the auto correlation plot above, we see that the auto correlation started to hover around 0 after the $100th$ (or $1th$ lag) value. With this, we discard the first 100 values in the chain.
The last 9900 values can now be treated as the generated values from the Cauchy distribution.

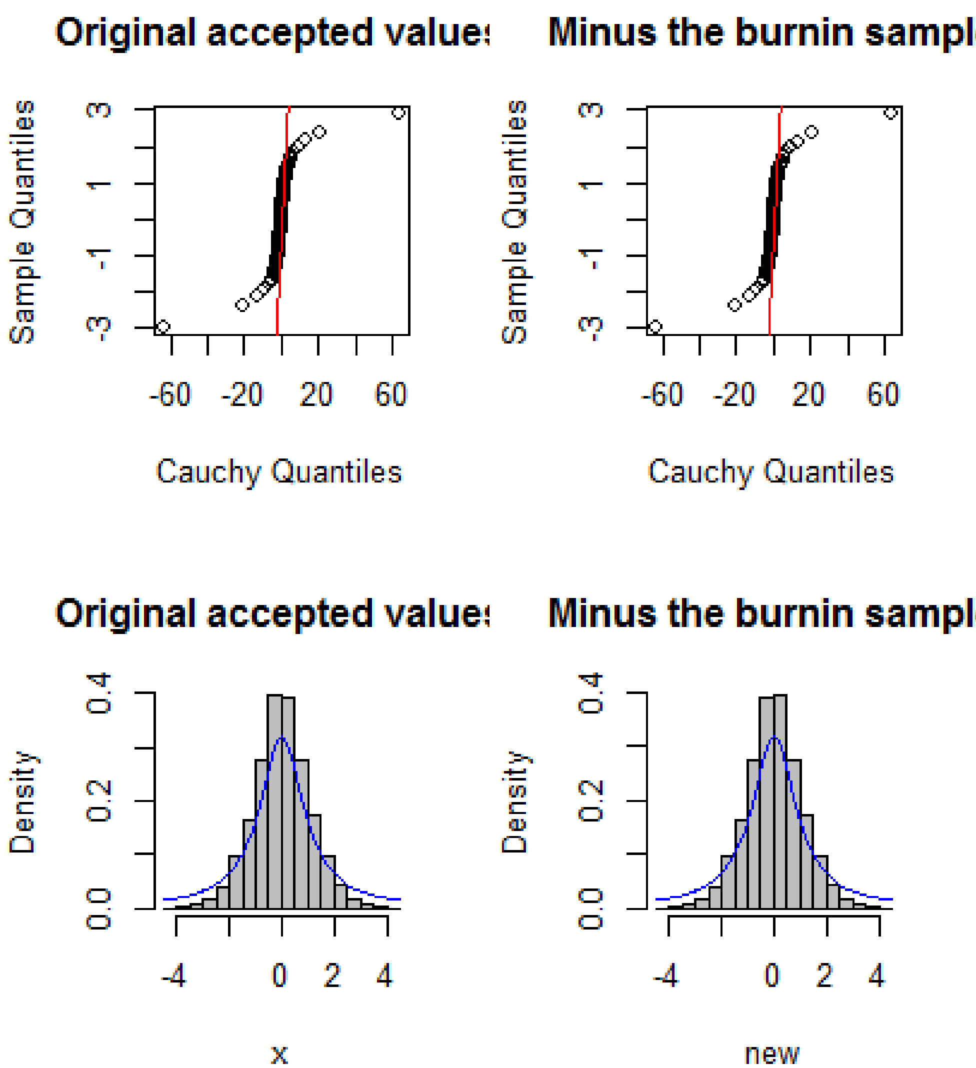

FIGURE 4. qqplot and histogram

The histogram of the generated sample after discarding the burning sample is shown in figure 4. The mean of the remaining sample is 0.002184433.

0.3. **Effect of Variance of Sampler Function.** For comparison, we repeated the simulation with different scale values of the proposed distribution. Figure 5 shows the difference in variations

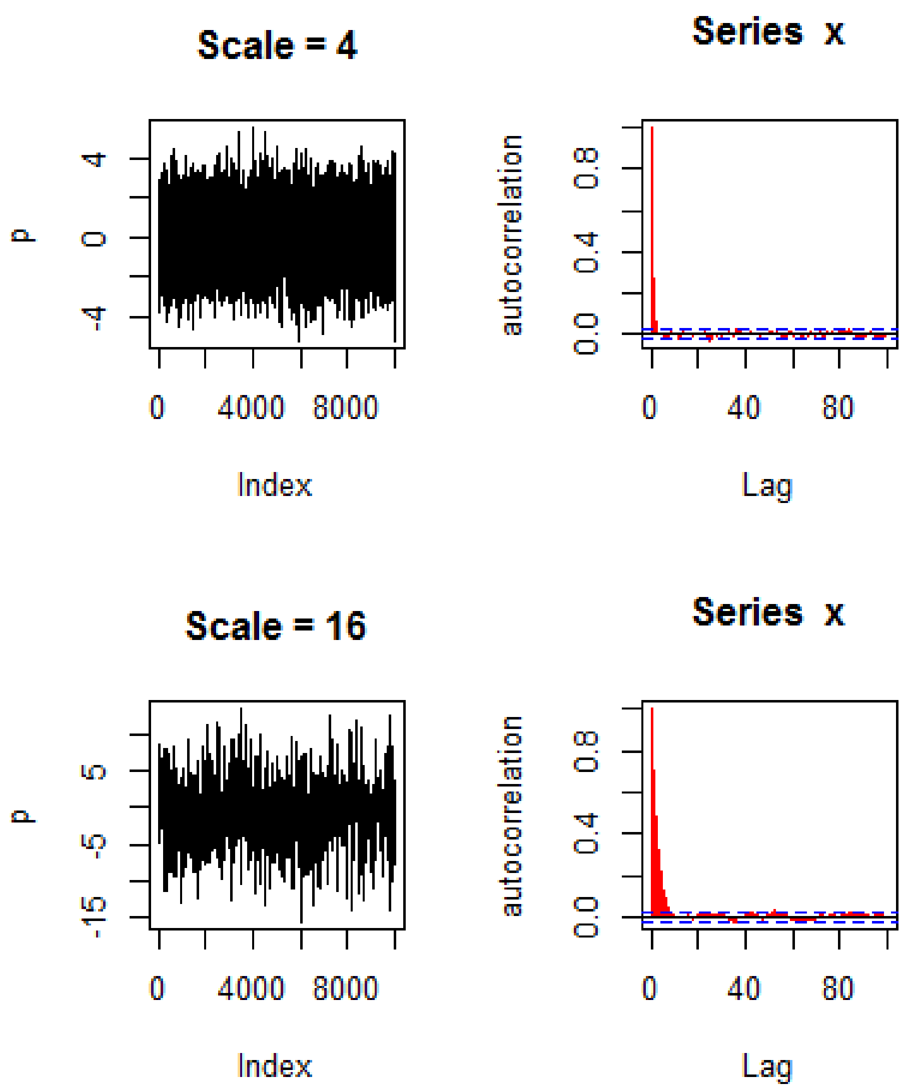

FIGURE 5. Autocorrelation plot

DISCUSSION AND CONCLUSION

The Cauchy distribution c(0,1) was used as the the target distribution to simulate a Markov chain using the metropolis independence sampler algorithm with normal distribution with mean 0 and variance 2 and the t distribution with degree of freedom 20 used as the proposed distribution
The scale parameter was varied to monitor the acceptance rate and the variance. The variance of the sampler is inversely related to the auto correlation and directly related to the count of rejected points. With a small variance in the sampler function the series has high auto correlation but large acceptance rates. Thus the entire sample space in a short amount of runs. With a large variance in the sampler function the series is has low auto correlation but low acceptance rates. This means it takes many runs to sample from the whole sample space.

1. LIMITATIONS

The choice of variance in the sampler function is highly important. If the variance is quite large then the acceptance rate is low so the algorithm converges slowly. On the other hand if it is too small, the the algorithm moves slowly throughout the sample space and converges slowly. The solution to this problem is outside of the scope of this project, is to specify a loss function (perhaps deviance from target distribution), and find the optimal variance that minimizes this loss function with optimization techniques like stochastic gradient descent or adaptive learning rates. A rigorous treatment of the latter can be found by Haario, Heikki; Saksman, Eero; Tamminen, Johanna. An adaptive Metropolis algorithm. Bernoulli 7(2001), $no.2, 223242$.

References

[1] Rizzo, M. L. (2019). Statistical Computing with R, Second Ed. Boca Raton, FL: Chapman Hall/CRC Press. ISBN-10: 9781466553323

[2] Applied Multivariate Statistical Analysis (6th Edition), Pearson Prentice Hall, 2007, by Richard Johnson, Dean W. Wichern.

[3] ANDERSON T.W. An introduction to Multivariate Statistical Analysis (3rd edition). New york : John Wiley 2003.

[4] Bacon Shone ,J, and W.K Fung." A New Graphical Method for Detecting Single and Multiple Outliers in Univariate nad Multivariate data", Applied Statistics,36no. 2 (1987), 153-162

[5] Barlett, M.S "Multivariate Analysis" Journal of the Royal Statistical Society Supplement(b),9 (1947),176-197

[6] Johnson R.A and G.K Bhattacharyya Statistics Principles and methods (5th ed). New York: John Wiley, 2005

Southern Illinois University Edwardsville