STAT431/ASRM453 Spring 2024 Final Project Metropolis-Hastings sampling for Bayesian Statistics

Patrick Liu[1], Shoun Lo[1], and Rohan Narasimhan[1] and Anh Phung[1]

University of Illinois Urbana-Champaign
{pzl2,shounlo2,rohann4,anhnp2}@illinois.edu

Abstract. Final project paper of Patrick Liu, Shoun Lo, Rohan Narasimhan, Anh Phung for the class STAT431/ASRM453: Applied Bayesian Analysis Spring 2024 under Dr. Trevor Park. We choose to follow option 2 for this paper: a survey and implementation for a special topic of the Metropolis-Hastings algorithm.
In section 1, we aim to define the Metropolis-Hastings algorithm, explain the motivation and the background behind the algorithm and its applications in a Bayesian context. In section 2, we then will briefly explain Markov Chains and their properties, specifically their convergence and stationary distributions as it relates to the Metropolis-Hastings algorithm. Next, in section 3, we want to illustrate the Metropolis-Hastings algorithm through examples of the implementation of Metropolis-Hastings algorithm, including a real-world application of the algorithm. Finally, the appendix with our work distribution is included in section 4.

1 Introduction

1.1 Motivations for Monte Carlo methods

We start off by setting up a simple Bayesian's Statistics problem as follows:

- $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_p)$: parameters.
- $\mathbf{y}$: observed data.
- $\pi(\boldsymbol{\theta})$: prior distribution.
- $p(\boldsymbol{\theta}|\mathbf{y})$: posterior distribution.
- $p(\theta_i|\mathbf{y})$: marginal posterior distribution for θ_i.
- $f(\mathbf{y}|\theta)$: likelihood function of $\mathbf{y}$.

In many problems (in fact, most of the problems), our parameters $\boldsymbol{\theta}$ are assumed to be continuous. This results in many tasks involving integrations, such as:

- Computing marginal data density $m(\mathbf{y})$:

$$m(\mathbf{y}) = \int f(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}$$

- Computing posterior expectation $E[g(\boldsymbol{\theta}|\boldsymbol{y})]$:

$$E[g(\boldsymbol{\theta}|\boldsymbol{y})] = \int g(\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{y})d\boldsymbol{\theta}$$

- Computing marginal posterior distribution $p(\theta_j|\boldsymbol{y})$:

$$p(\theta_j|\boldsymbol{y}) = \int_{\theta_k:k\neq j} p(\boldsymbol{\theta}|\boldsymbol{y})d\boldsymbol{\theta}_{\boldsymbol{(-j)}}$$

- Compute posterior probability: $H_0 = \boldsymbol{\theta} \in \Theta_0$:

$$P(H_0|\boldsymbol{y}) = \int_{\Theta_0} p(\boldsymbol{\theta}|\boldsymbol{y})d\boldsymbol{\theta}$$

- And many more!

However, taking an integral is not always easy mathematically and computationally. Deterministic methods such as Numerical Integration suffer from the curse of high dimensionality - they perform significantly worse in a high dimensional settings with multiple parameters. Thus, Monte Carlo methods (those that use randomized sampling for approximation) are utilized in such cases.

1.2 Motivations for Metropolis-Hastings

Two of the most commonly used Monte Carlo methods for sampling in Bayesian statistics are Gibbs sampling and Metropolis-Hastings. With the idea of sample alternatively between the full conditional of parameters and update their values after each steps, Gibbs samplings are a powerful tools for many problems, especially those that have semi-conjugate distributions. Despite that, there are still cases that Gibbs samplings perform worse, or even unusable:

- Parameters have high posterior correlation.
- Posterior has multiple modes (offset from each other)
- There are no easy way to sample from the full conditional or no available type of semi-conjugacy is satisfactory.

The first two cases might slow down the Gibbs sampling, but the third might make Gibbs sampling impossible to be implemented. An example for this is can be seen as follows, with β's being the parameters, and Y is the data:

$$Y_i|\boldsymbol{\beta} \sim \text{Poisson}(e^{\beta_1+\beta_2 i})$$

which means:

$$f(\boldsymbol{Y}|\boldsymbol{\beta}) \propto \prod e^{(e^{\beta_1+\beta_2 i})} e^{Y_i(\beta_1+\beta_2 i)}$$

The parameters appear in layers of exponential, and there's no well-known distribution one can use to capture this formula. Thus, both posterior and full conditional will most likely not belong to a well-known distribution, and thus not clear how to efficiently sample from them both!

Thus, Metropolis-Hastings to the rescue!

1.3 Background and History

The Metropolis-Hastings algorithm is a general term to refer towards a family of Markov chain Monte Carlo methods that are used for generating samples from probability distributions that are difficult to sample directly from (Chib & Greenberg, 1995,[1]). This was initially called the Metropolis Algorithm, named in part by Nicholas Metropolis and his paper in 1953 [3], and was intended to sample any distribution function regardless of the complexity, and was later generalized by W.K. Hastings in his paper in 1970 [4].

The main difference of Metropolis-Hastings compared to other MCMC methods , such as Gibbs sampling, is the fact that it's an acceptance-rejection methods (Ahrens and Dieter, 1972, [5]). That is, instead of directly sampling from the exact full conditional distribution, it draws from another candidate distributions of $\boldsymbol{\theta}$, followed by accepting or rejecting it (more details in the next section). This helps avoid costly sampling from a complex distribution of the full conditional, and instead a computationally easier chosen candidate distributions.

2 The Metropolis-Hastings algorithm

2.1 Algorithm and pseudo-code

The Metropolis-Hastings algorithm is structured as follows:

- Pick a family of candidate distribution $\boldsymbol{q} = (q_1, q_2, \ldots, q_p)$. This is the distributions we'll be sampling from instead of the full conditional in Gibbs.
- Choose a probability of success R. The formula will be shown in the pseudocode as it's complexity, more explanation by then.
- For probability R, "accepts" the newly sampled value, and update the value. Otherwise, do nothing and keep the value unchanged.

The pseudocode can be found as Algorithm 1 in the next page.

2.2 Correctness and rate of convergence

In the same paper in 1953, Metropolis et al. [3] managed to prove that Metropolis-Hastings preserve the stationary distribution π similarly to other MCMC methods, under the condition that the chain is irreducible. Here, a Markov chain is irreducible if from any state, it can reach any other states after finite iterations.

In our case of Metropolis-Hastings, our choice of q must let the chain constructed by them be irreducible. An easy sufficient condition for this is just, q is positive at all point! Notice that this means, one can choose any arbitrary q that is easy to sample that is positive anywhere (for example, Normal, Gamma, etc.), and arrive at the correct desired sampled distribution p!

However, our choice of q still does matter, as it decides the convergence rate of our algorithm p. More details of this can be found in the paper by Meyn (an UofI professor at the time) and Tweedie in 1994 [6], or by Mengersen and Tweedie in

Algorithm 1 Metropolis-Hastings

```
Initialize $\boldsymbol{\theta}$: $\boldsymbol{\theta_0} = (\theta_0^{(0)}, \theta_1^{(0)}, \ldots, \theta_p^{(0)})$
Choose candidate distributions: $\boldsymbol{p} = (q_1, q_2, \ldots, q_p)$
for $s = 1, \ldots, S$ do
    for $j = 1, \ldots, p$ do
        sample $\theta_j^* \sim q_j(\theta_j | \theta_j^{(s-1)})$
        set $\boldsymbol{\theta^*} = \left(\theta_1^{(s)}, \ldots, \theta_{j-1}^{(s)}, \theta_j^*, \theta_{j+1}^{(s-1)}, \ldots, \theta_p^{(s-1)}\right)$
        set $R = \frac{f\left(\boldsymbol{Y}|\boldsymbol{\theta^*}\right)}{f\left(\boldsymbol{Y}|\boldsymbol{\theta^{(s-1)}}\right)} \frac{\pi\left(\boldsymbol{\theta^*}\right)}{\pi\left(\boldsymbol{\theta}^{(s-1)}\right)} \frac{q_j\left(\theta_j^{(s-1)}|\theta_j^*\right)}{q_j\left(\theta_j^*|\theta_j^{(s-1)}\right)}$
        sample $U \sim \text{Uniform}(0, 1)$
        if $U < R$ then
            $\theta_j^{(s)} = \theta_j^*$
        else
            $\theta_j^{(s)} = \theta_j^{(s-1)}$
        end if
    end for
end for
```

1995 [7]. The borderline idea is that under good choices of q, the convergence rate of the algorithm is geometrically fast (Theorem 1.3 in Mengersen and Tweedie paper):

$$||P^n(x, \cdot) - \pi|| \leq (1 - \delta)^{\lceil n/m \rceil}$$

This can be understood that the difference between the state of the distribution after n iterations and the stationary desired distribution shrinks geometrically with the increase of n.

In our opinion, this results matter less about implying an implicit way for one to choose a candidate distribution q, but rather giving a proof that their exists some choices of q that make the rate of convergence really good, meaning less burn-in needed. This is because performing the math to find q is much more complicated than trial and errors with multiple easily sampled and well-known q, especially when the power of computation is growing rapidly as of right now!

3 Implementation of Metropolis-Hastings

3.1 An Illustration of the M-H Algorithm: The Gumbel Distribution

An illustration of the Metropolis-Hastings algorithm would be sampling from the Gumbel distribution to estimate the parameters, as conducted by a group of researchers in 2015 [8]. The Gumbel distribution is useful in modeling the distribution of extreme levels, either maximums or minimums, which are relevant in. for example, predicting extreme natural disasters such as floods and earthquakes. The distribution has two parameters: the location μ and the scale σ. The distribution function for the Gumbel distribution is:

$$G(z) = \exp\left\{-\exp\left[-\left(\frac{z-\mu}{\sigma}\right)\right]\right\}. \quad -\infty < \mu < \infty, \quad \sigma > 0$$

They found the conditional posterior distribution to be extremely difficult to ascertain any recognizable patterns. Thus, Metropolis-Hastings was employed instead of Gibbs sampling for posterior inference. The generated series was found to give good estimations of the values of the parameters. To utilize the Metropolis-Hastings algorithm, they selected initial values (μ_0, σ_0), with the criteria to update θ to be:

$$\theta^{(j+1)} = \begin{cases} \theta^* & \text{if } u < p, \\ \theta^{(j)} & \text{otherwise.} \end{cases}$$

where θ^* is the candidate, $u \sim \text{Unif}(0,1)$ and p is the acceptance probability:

$$p = \min\left\{1, \frac{\pi(u^*|\sigma^j)}{\pi(u^j|\sigma^j)}\right\}$$

Then, first 500 iterations were burned, given that iterations before do not converge to stable values. Given that main advantage of using Metropolis-Hastings is that it will work for any arbitrary distribution, they estimated parameters as such:

Table 1. Parameter values for Metropolis-Hastings

Parameter	**Value**
μ	100.100
σ	9.9515

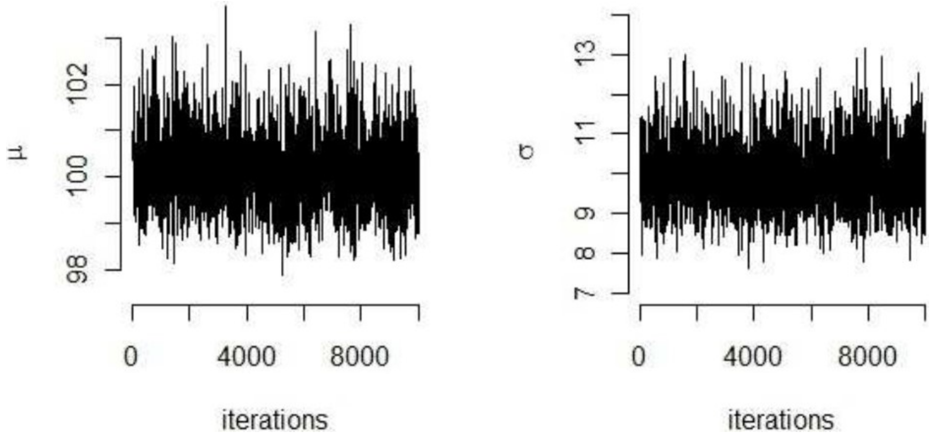

Fig. 1. Trace plots of M-H for Gumbel Fit for μ, and σ

However, they found that Metropolis-Hastings was extremely slow and inefficient, requiring a large number of iterations as seen with the trace plots. This prompted them to expand upon the Metropolis-Hastings algorithm to be more efficient while maintaining the advantages, which was the main purpose of their research.

3.2 Another example of the M-H algorithm using artificial data

Let's consider a simple example where we are trying to estimate the mean μ of a Normal distribution with known variance σ^2.

We'll use a conjugate prior for μ, which is also a Normal distribution with mean 0 and a large variance:

$$\mu|\sigma^2 \sim N(0, \tau^2)$$

The likelihood function for our data y given μ and σ^2 is the Normal distribution:

$$y|\mu, \sigma^2 \sim N(\mu, \sigma^2)$$

Using Bayes' theorem, we can derive the posterior distribution for μ given the data y.

$$p(\mu|y) \propto p(y|\mu) \cdot p(\mu)$$

Since both the likelihood and the prior are Normal distributions, the posterior will also be a Normal distribution. Since both the likelihood and the prior are Normal distributions, the posterior will also be a Normal distribution.

The Metropolis-Hastings algorithm can be used to generate samples from the posterior distribution when it's difficult to do so directly[[1]].

In the implementation, μ is initialized to 0 and the first 500 iterations are burned. A new value for μ, which is drawn from a Normal distribution centered at the current value of μ is proposed. The acceptance ratio, which is the ratio of the posterior densities at the proposed and current points. This ratio can be written as

$$p = \min\left\{1, \frac{p(\mu^*|y)}{p(\mu^{(current)}|y)}\right\}$$

If a random number from a Uniform(0, 1) distribution is less than the acceptance ratio, the proposed value is accepted and the current value is updated to the proposed value.

The mean of the samples after the burn-in was 5.0165. The 1000 data points was artificially generated from normal distribution with a mean of 5 and standard deviation of 1.

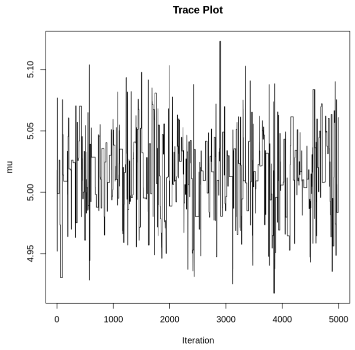

Fig. 2. Trace plot of M-H for Normal Fit for μ

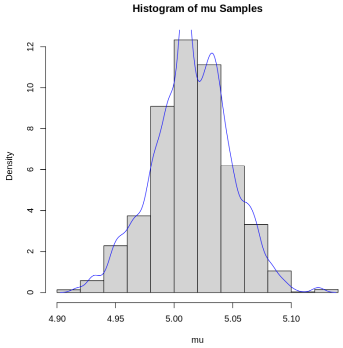

Fig. 3. Histogram of M-H samples for μ

The trace plot show that the samples are mixing well and not getting stuck in any particular region of the parameter space. Additionally, the estimated mean of the posterior distribution are stable over iterations.

3.3 A real world application of the M-H Algorithm

A useful situation where the M-H algorithm can be applied is in forward diffusion. An example is trying to model the motion of a pollen particle constantly being buffeted by air. In this case [10], we model the particle as a simple harmonic oscillator, with energy $E = \frac{x^2}{2}$.

As the position of the particle only depends on its previous position we use a Markov Chain Monte Carlo (MCMC) to arrive at our expected distribution (which in this case is $e^{-E(x)}$). In order to pick the particle's next position, we sample a guassian distribution with standard deviation $\sqrt{\delta}$, where δ represents the jump in position from x to x'.

Using the M-H algorithm we find that if we choose a move from x to x' with probability $T(x \to x')$, then the acceptance probability is given by

$$A(x \to x') = \frac{\pi(x')T(x' \to x)}{\pi(x)T(x \to x')}$$

For our case with a Gaussian having standard deviation $\sqrt{\delta}$,

$$T(x' \to x) = T(x \to x') = \exp\left(-\frac{(x - x')^2}{2\delta}\right)$$

And so

$$A(x \to x') = \exp[-(E(x') - E(x))]$$

As mentioned in earlier sections, if a random number from a Uniform(0, 1) distribution is less than the acceptance probability, we accept x' and update the original position.

We implemented this process, iterating over 10000 steps and burnt the first 1000 steps. We then plotted the distribution as a histogram and placed the theoretical curve on top, where the desired distribution is:

$$y = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$$

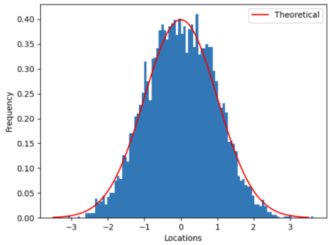

Fig. 4. Histogram of Forward Diffusion Using M-H

The figure above shows that we can use the M-H algorithm along with MCMC to correctly model forward diffusion.

4 Appendix

Work distribution:

- Anh Phung: Working on the project proposal. Survey and Write-up the Section 1 and 2.
- Shoun Lo: Survey on the history and background of M-H. Illustrated the M-H on the Gumbel Distribution in a Bayesian context.
- Patrick Liu: Implemented M-H algorithm and showed another application of the M-H algorithm.
- Rohan Narasimhan: Worked on the project proposal, editing and application of M-H algorithm in a diffusion model.

References

1. Chib, S.,& Greenberg, E. (1995). Understanding the Metropolis-Hastings Algorithm. The American Statistician, 49(4), 327–335. https://doi.org/10.2307/2684568
2. Robert, C. P., Casella, G., Robert, C. P., & Casella, G. (2004). The metropolis—hastings algorithm. Monte Carlo statistical methods, 267-320.
3. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). "Equations of state calculations by fast computing machines." J. Chem. Phys., 21(6): 1087–1092.
4. Hastings, W. (1970). "Monte Carlo sampling methods using Markov chains and their application." Biometrika, 57: 97–109.
5. Ahrens, J. H., & Dieter, U. (1972). Computer methods for sampling from the exponential and normal distributions. Communications of the ACM, 15(10), 873-882.
6. Meyn, S. P., & Tweedie, R. L. (1994). Computable bounds for geometric convergence rates of Markov chains. The Annals of Applied Probability, 981-1011.
7. Mengersen, K. L., & Tweedie, R. L. (1996). Rates of convergence of the Hastings and Metropolis algorithms. The annals of Statistics, 24(1), 101-121.

8. Mohd Amin, N.A., Adam, M.B., & Ibrahim, N.A. (2015). Bayesian Inference using Multiple-try Metropolis Hastings Scheme for the Efficiency of Estimating Gumbel Distribution Parameters. Matematika, 31, 25–36.
9. Reich, B. J., & Ghosh, S. K. (2019). Bayesian statistical methods. Chapman and Hall/CRC.
10. Clark, B.(2023). Generative Diffusion Models. Computing in Physics. https://illinois-compphys.github.io/ComputationalPhysics/ML/Diffusion.html