Proceedings of the 3rd International Conference on Mathematical Physics and Computational Simulation
DOI: 10.54254/2753-8818/92/2025.22031

Brief Introduction to Markov Chain Monte Carlo and Its Algorithms

Qiaomu Liu

Irvine Valley College, Irvine, USA
qliu29@ivc.edu

Abstract: The Markov Chain Monte Carlo (MCMC) methods have become indispensable tools in modern statistical computation, enabling researchers to approximate complex probability distributions that are otherwise intractable. This paper focus on MCMC in Statistics and Probability area which is used to draw samples from a probability distribution. In order to introduce this algorithm in a relatively light and straightforward way, this paper breaks the content into two parts: Markov Chain and MCMC, and brings in stochastic process, Markov property, Ordinary Monte Carlo, and Monte Carlo Integration in succession. On this foundation, this paper formally introduces two algorithms of MCMC that are Metropolis-Hasting Algorithm and Gibbs Sampling, the most famous and significant algorithms of MCMC. With this approach, this paper concludes both the advantages and disadvantages of these two algorithms, laying the foundation of the further study and the deeper utilizations of MCMC and, hopefully, contributing to providing a certain degree of assistance for scholars.

Keywords: Markov Chain, Monte Carlo, Gibbs Sampling, Metropolis-Hasting Algorithms

1. Introduction

The Markov Chain Monte Carlo (MCMC) methods have become indispensable tools in modern statistical computation, enabling researchers to approximate complex probability distributions that are otherwise intractable. Since the pioneering work of Gelfand and Smith in 1990, MCMC techniques have revolutionized Bayesian inference and other fields requiring efficient sampling from high-dimensional distributions [1]. In order to generate dependent samples for statistical estimation and inference, MCMC essentially builds a Markov chain whose stationary distribution matches a target distribution of interest.

Despite their theoretical elegance, MCMC methods face practical challenges, particularly regarding convergence and efficiency. Assessing when an MCMC simulation produces representative samples of the target distribution is crucial for ensuring the reliability of the resulting estimates [1]. Moreover, as modern applications often involve high-dimensional parameter spaces and complex models, there is an increasing need for robust diagnostic tools and algorithms to verify convergence and improve sampling efficiency [2]. Meanwhile, traditional MCMC algorithms can be computationally expensive and often struggle with efficiency when applied to large-scale datasets [3]. To address these challenges, researchers have developed various MCMC algorithms designed to enhance mixing and explore target distributions more effectively. For instance, an ensemble MCMC approach introduces interactions among multiple chains (or walkers), allowing states to be duplicated and removed in a manner that mitigates metastability. This ensemble interaction results in rapid mixing and asymptotic convergence rates that are independent of the spectral gap of the underlying Markov chain [4]. Such innovations offer promising solutions for sampling from multimodal distributions and high-dimensional spaces. In addition to algorithmic advancements, the theoretical underpinnings of MCMC methods continue to evolve. Recent studies emphasize the importance of geometric ergodicity and the role of drift and minorization conditions in establishing the convergence of Markov chains to their stationary distributions [1]. These theoretical developments provide the foundation for the reliability of MCMC-based estimators and contribute to the broader understanding of Monte Carlo methodologies. The practical relevance of MCMC methods is further highlighted by their widespread adoption across scientific disciplines. Applications include Bayesian hierarchical models, predictive inference, and machine learning tasks, where MCMC simulations play a critical role in estimating posterior distributions and predictive quantities [5]. However, the efficiency and scalability of MCMC algorithms remain active areas of research, particularly as the complexity of data and models continues to increase.

This paper aims to provide an overview of MCMC, focusing on prominent algorithms that are, specifically, Metropolis-Hastings and Gibbs sampling. By examining the theoretical foundations and practical implementations of these methods, this paper seeks to highlight their strengths, limitations, and ongoing developments in the field of stochastic simulation. Besides, this paper also pursues to lay a solid foundation of MCMC in order to provide assistance for scholars.

2. Markov Chain and Basic Examples

Named after Andrey Markov, Markov, the famous Russian mathematician, Markov Chain is a stochastic process which satisfies the Markov Property, describing a series of potential occurrences with the probability of which doesn't rely on the past state, but only depends on the current state.

2.1. Stochastic Process

To introduce Markov Chain further, this paper will go over stochastic process at the first place. The stochastic process is defined as a collection of random variables in a probability space, where its index has interpretation of time. It involves one or more random variables, following certain probability laws under certain conditions. Generally speaking, stochastic process is used to describe systems or phenomena that cannot be precisely predicted by deterministic models.

Formally, a stochastic process can be a function:

$$\{X_t, t \in T\}, \tag{1}$$

where X_t is a random variable for each t and can take values in state space S, and t belongs to index set. If T is countable, this process can be defined as a discrete parameter process, especially when $T = \mathbb{N} = \{0,1 \dots\}$. Otherwise, if the T is not countable, the process is going to have a continuous parameter [6]. There are basically three key features of stochastic process needed to briefly introduce: index set (T) which can be either discrete or continuous; state space (S), the possible value of X_t, such as $S = \mathbb{R}$ or $S = \{0,1\}$; dependence structure, the way past values of X_t influence future values.

Through the utilization of stochastic processes, plenty of new processes have been inspired and eventually came to become significant part of them. For example, the Poisson Process, one of the most essential one among the stochastic processes, describing the number of times an event occurs within a fixed time interval where the occurrence of the event is independent and uniform. To be specific, the Poisson Process, denoted by $N(t)$, satisfies three key properties. First, the process starts at 0, representing no events occurred at initial time, and can be denoted as $N(0) = 0$. Second, it has independent increments, showing the number of events occur in disjoint time intervals are independent of each other. Third, event's number in any time interval $(t, t+s]$ follows a Poisson Distribution:

$$P(N(t+s)-N(t)=k)=\frac{e^{-\lambda s}(\lambda s)^k}{k!}, k=0,1,2\cdots, \tag{2}$$

where λ represents the rate at which events occur, and s represents the interval's length. In addition, Poisson process also follows memoryless property, meaning that the time between consecutive events, known as interarrival times, follows an exponential distribution with mean $\frac{1}{\lambda}$, making the Poisson process particularly useful in modeling random arrival times of customers in a queue, decay of radioactive particles, or network traffic.

2.2. Markov Property

Markov Property refers a stochastic process with memoryless property, indicating that the future evolution of the process does not depend on its past, but only its present state. Formally, a stochastic process $\{X_t, t \in T\}$ satisfies the Markov property if, for all times $t_0 < t_1 < \cdots < t_n < t$, the conditional probability distribution of X_t given the entire past history depends only on the most recent state X_n:

$$P(X_t | X_{t_n}, X_{t_{n-1}}, \dots, X_{t_1}, X_{t_0}) = P(X_t | X_{t_n}), \tag{3}$$

which indicates that the past states provide no additional information about the future state X_t given the present state X_{t_n}.

To further explain the Markov property, this paper will introduce a simple example that reflects the property well. Assume there is a weather model where weather is classified into three states: Sunny (S), Cloudy (C), and Rainy (R). In a Markovian weather model, one can assume that tomorrow's weather only depends on the weather of today, but never on the weather of previous days. If presenting this system with a transition probability matrix, where each row represents today's weather, and each column represents tomorrow's weather. There is a specific example, for instance, if today is sunny, the probability of tomorrow also being sunny might be 70%, the probability of it becoming cloudy 30%, and the probability of rain 20%. Regardless of the weather from previous days, these probabilities always remain the same. In this case, this example is obviously a utilization of Markov property. The probability of tomorrow's weather is shown in Table 1.

Table 1: The Probability of Tomorrow's Weather

Today- Tomorrow	Sunny (S)	Cloudy (C)	Rainy (R)
Sunny (S)	0.7	0.2	0.1
Cloudy (C)	0.3	0.4	0.3
Rainy (R)	0.2	0.5	0.3

2.3. Basic Examples of Markov Chain

As mentioned, Markov Chain is contained in stochastic process and it satisfies Markov property as well. Considering of brief introduction of stochastic process and Markov property have been discussed, this paper will dive into Markov chain further by talking about a few examples of it.

First, and perhaps the most famous example of Markov Chain is called Random Walk. Assume a Markov chain with state space given by integers $i = 0, \pm 1, \pm 2, \dots$, and for some number $0 < p < 1$,

$$P_{i,i+1} = p = 1 - P_{i,i-1}, i = 0, \pm 1, \cdots \tag{4}$$

This previous Markov chain is then referred to as a Random Walk, originally representing a person walking along a linear way who at each point of time, either steps to the right direction with probability p or to the left with probability $1-p$ [7].

Another significant example of Markov chain is gambling model. Assume a context of gambling, a gambler's fortune can be modeled as a Markov chain with absorbing states, representing a finite-state random walk. At each round of play, the gambler either wins \$1 with probability p or loses \$1 with probability $1-p$. Assume this gambler continue playing until reaching either of these two absorbing states: one ultimately goes broke where one’s fortune = \$0, or one successfully earned oneself a fortune of $\$N$. Then, with transition probability, this gambler’s fortune can be represented as a Markov chain:

$$P_{i,i+1} = p = 1 - P_{i,i-1}, i = 1,2,\cdots,N-1, \tag{5}$$

$$P_{00} = P_{NN} = 1, \tag{6}$$

where the states 0 and N are absorbing, meaning that once the gambler reaches these values, they remain there permanently [7].

In summary, Markov Chain is an essential type of stochastic process that follows Markov property, indicating that the future state of the system only depends on its present state, not its past. Covering from simple example like random walk to structured model such as the gambler’s ruin problem, Markov chains provide a powerful framework for understanding probabilistic behavior in dynamic systems. With this foundational understanding in place, people can now explore the practical application of Markov chains.

3. Markov Chain Monte Carlo

Having established the fundamental content of Markov chain, including the stochastic process and Markov property, this paper is going to introduce the specific application of Markov chain in this section. One notable technique is Markov Chain Monte Carlo, or MCMC in short, which combines Markov chain and Monte Carlo simulation and is used to approximate complex probability distributions through random sampling.

3.1. Ordinary Monte Carlo

Since this paper has covered the basic content of Markov chain, to further introduce Markov Chain Monte Carlo, it is essential to talk about Ordinary Monte Carlo (OMC) first.

OMC is a fundamental computational method used to estimate expectations through random sampling, particularly when direct analytical solutions are impractical. It operates by generating independent and identically distributed (IID) samples from a given probability distribution, and using these samples to approximate numerical results, such as integrals or expectations. Assume there is an expectation to be calculated:

$$\mu = E\{g(X)\}. \tag{7}$$

However, when it comes to be too difficult to compute analytically, it can be approximated using Monte Carlo sampling. Suppose $X_1, X_2, \cdots$ IID can be simulated sharing the same distribution with X. Define:

$$\hat{\mu}_n = \frac{1}{n}\sum_{i=1}^{n} g(X_i), \tag{8}$$

According to LLN, the law of large numbers, as $n \to \infty$, the Monte Carlo estimate $\hat{\mu}_n$ converges to the true expectation μ. Furthermore, using the central limit theorem (CLT), for sufficiently large n, the estimator follows a normal distribution:

$$\hat{\mu}_n \approx Normal\left(\mu, \frac{\sigma^2}{\mu}\right), \tag{9}$$

where $\sigma^2 = var\{g(X)\}$. Then, the variance of the Monte Carlo estimator can be approximated by:

$$\hat{\sigma}_n{}^2 = \frac{1}{n}\sum_{i=1}^{n}(g(X_i) - \hat{\mu}_n)^2, \tag{10}$$

which determines the Monte Carlo standard error (MCSE), given by:

$$MSCE = \frac{\hat{\sigma}_n}{\sqrt{n}}, \tag{11}$$

quantifying the uncertainty in the Monte Carlo estimate [8]. Notably, OMC follows the square root law, indicating that improving accuracy requires a quadratic increase in sample size. Despite this computational challenge, OMC remains a widely used approach in numerical integration, probability estimation, and scientific simulations, laying the groundwork for more sophisticated methods such as MCMC.

3.2. Monte Carlo Integration

As mentioned, OMC provides a fundamental framework for random sampling and expectation approximation which makes it a powerful tool for problem-solving where direct analytical methods are infeasible. However, when Monte Carlo methods are used to approximate definite integrals, especially in cases where standard numerical techniques become inefficient, especially in high-dimensional settings. This leads to the Monte Carlo integration, an extension of OMC, applying random sampling to numerical integration problems.

Suppose a complex integral:

$$\int_a^b h(x)dx. \tag{12}$$

Over the interval (a, b), if decomposing $h(x)$ into the production of a function $f(x)$ and a pdf, probability density function, $p(x)$, it comes to be an expectation of $f(x)$ over $p(x)$:

$$\int_a^b h(x)dx = \int_a^b f(x)p(x)dx = E_{p(x)}[f(x)]. \tag{13}$$

As a result, if randomly extracting a large number $x_1, \cdots, x_n$of random variables from $p(x)$, there then is:

$$\int_a^b h(x)dx = E_{p(x)}[f(x)] \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i), \tag{14}$$

which is called Monte Carlo integration [9].

3.3. Algorithms of Markov Chain Monte Carlo

After explaining ordinary Monte Carlo and Monte Carlo integration, this paper has established a preliminary foundation for understanding MCMC. Now, it is time to dive into MCMC and its algorithms.

As mentioned in former sections, MCMC methods construct a Markov chain at the very first place. By carefully designing the transition probabilities of the Markov chain, MCMC ensures the chian's stationary distribution matches its target distribution $\pi(x)$, meaning that the samples generated by the chain can be used for Monte Carlo estimation after a sufficient number of iterations [10]. Compared to Monte Carlo methods, which diminishes its effectiveness in high-dimensional spaces since independent sampling becomes computationally expensive, MCMC methods address this limitation by introducing Markovian dependence in the sampling process, enabling more efficient exploration of complex probability distributions. Unlike ordinary Monte Carlo, which generates independent samples, MCMC constructs a Markov chain that, over time, converges to the desired distribution. This transition from traditional Monte Carlo to MCMC further leads to specialized MCMC algorithms, such as the Metropolis-Hastings algorithm and Gibbs Sampling, which optimize the sampling process to make statistical inference possible to be done even in high-dimensional problems.

3.4. Metropolis-Hasting Algorithm

First, this paper is going to introduce Metropolis-Hasting algorithm, one of the most widely used MCMC methods for sampling from a probability distribution when direct sampling is difficult. The key advantage of the MH algorithm is that it only requires evaluating $\pi(x)$ up to a proportionality constant, making it particularly useful for Bayesian inference, where computing normalizing constants is often infeasible.

The Metropolis-Hasting algorithm follows the following steps: Firstly, choose an arbitrary starting point X_0 to initialize. Secondly, propose a candidate sample, indicating that it is going to generate a new candidate state X^* from a proposal distribution $Q(X^*|X_t)$, where X_t represents the current state. Thirdly, compute the acceptance probability by defining acceptance ratio:

$$a = \min\left(1, \frac{\pi(X^*)Q(X_t|X^*)}{\pi(X_t)Q(X^*|X_t)}\right), \tag{15}$$

that adjusts for potential asymmetries in the proposal distribution. Then, it generates a uniform random number $u \sim \text{Uniform}(0,1)$. If $u \leq a$, it will accept the candidate and set $X_{t+1} = X^*$. Otherwise, it will reject the candidate and set $X_{t+1} = X_t$. At last, the algorithm is going to continue iterating for a sufficiently long period, discarding initial "burn-in" samples to ensure proper convergence. In a special case where $Q(X_t|X^*) = Q(X^*|X_t)$, the acceptance ratio will simply to:

$$a = \min\left(1, \frac{\pi(X^*)}{\pi(X_t)}\right), \tag{16}$$

that is the classical Metropolis algorithm, first introduced in the 1953 paper by Metropolis et al. It is widely used in physics and Bayesian statistics [11].

The Metropolis-Hastings algorithm satisfies detailed balance, ensuring that the Markov chain converges to the desired target distribution $\pi(x)$. Over time, the chain explores the entire state space and samples are drawn according to $\pi(x)$, making them suitable for Monte Carlo estimation [12].

3.5. Gibbs Sampling

Gibbs Sampling is a powerful MCMC method widely used in Bayesian statistics and computational simulations. As a special case of the Metropolis-Hastings algorithm, where samples are iteratively drawn from conditional distributions of each variable while keeping others fixed, it is particularly useful for generating random variables from complex joint distributions when direct sampling is infeasible. Given a joint probability distribution $f(X_1, X_2, \cdots, X_p)$, the goal is to get a sample from the marginal distribution of a particular variable.

Instead of sampling from the complex joint distribution, Gibbs Sampling decomposes the problem into a sequence of easier sampling steps:

- Start with an initial guess $({X_1}^{(0)}, {X_2}^{(0)}, \cdots, {X_p}^{(0)})$.
- Update each variable sequentially by drawing from its conditional distribution, which means that ${X_1}^{(t+1)} \sim f\left(X_1 \middle| {X_2}^{(t)}, {X_3}^{(t)}, \cdots, {X_p}^{(t)}\right)$, ${X_2}^{(t+1)} \sim f\left(X_2 \middle| {X_1}^{(t+1)}, {X_3}^{(t)}, \cdots, {X_p}^{(t)}\right)$, as well as ${X_p}^{(t+1)} \sim f\left(X_p \middle| {X_1}^{(t+1)}, {X_2}^{(t+1)}, \cdots, {X_{p-1}}^{(t+1)}\right)$.
- Repeat this process for a sufficiently large number of iterations until convergence [13].

Under mild conditions, Gibbs Sampling converges to the target distribution as the number of iterations increases. The Markov Chain defined by the sequence of sampled values has a unique stationary distribution, which is the desired joint distribution. The rate of convergence depends on factors such as dependencies between variables and the choice of starting values [14].

4. Conclusion

In conclusion, this paper goes over the basic MCMC and its two algorithms, Metropolis-Hasting algorithm and Gibbs sampling by covering stochastic process, Markov property, Markov Chain, Ordinary Monte Carlo, and Monte Carlo integration. Using this approach, from the easier to the more advanced, this paper briefly introduces MCMC, together with its two significant algorithms, laying a basic but meaningful foundation in probability and statistics area for further study of MCMC. In addition, the way this paper used to break MCMC into several sections shows an impact on clear introduction and demonstration, and, hopefully, has the slight but positive influence on education and teaching. Despite these significances, this paper is presented as an elementary introduction of MCMC that stops at the algorithms section, without stepping further to bring in any concrete application of MCMC due to author's limited ability and energy. This will encourage author's further study of MCMC. As for MCMC itself, reflecting on the practical implications, MCMC methods have become indispensable across a wide range of scientific disciplines, including Bayesian modeling, machine learning, and computational biology. Despite their success, challenges such as ensuring convergence and improving computational efficiency remain important areas for future research. Addressing these issues will enhance the scalability and reliability of MCMC methods, making them even more useful for increasingly complex models and large datasets. In this case, in future work, further exploration of advanced MCMC techniques and their applications in high-performance computing and big data analysis will be essential to keep pace with the demands of modern data science.

References

[1] Jones, G. L., & Qin, Q. (2022). Markov chain Monte Carlo in practice. Annual Review of Statistics and Its Application, 9(1), 557-578.

[2] Krüger, F., Lerch, S., Thorarinsdottir, T., & Gneiting, T. (2021). Predictive inference based on Markov chain Monte Carlo output. International Statistical Review, 89(2), 274-301.

[3] Nemeth, C., & Fearnhead, P. (2021). Stochastic gradient markov chain monte carlo. Journal of the American Statistical Association, 116(533), 433-450.

[4] Lindsey, M., Weare, J., & Zhang, A. (2022). Ensemble Markov chain Monte Carlo with teleporting walkers. SIAM/ASA Journal on Uncertainty Quantification, 10(3), 860-885.

[5] Harrington, S. M., Wishingrad, V., & Thomson, R. C. (2021). Properties of Markov chain Monte Carlo performance across many empirical alignments. Molecular Biology and Evolution, 38(4), 1627-1640.

[6] Cinlar, E. (2013). Introduction to stochastic processes. Courier Corporation.

[7] Ross, Sheldon M. Introduction to Probability Models. 12th ed., Academic Press, an Imprint of Elsevier, 2019. 195

[8] Geyer, C. J. (2011). Introduction to Markov Chain Monte Carlo. Handbook of Markov Chain Monte Carlo, 20116022(45), 22.

[9] Carlo, C. M. (2004). Markov Chain Monte Carlo and Gibbs sampling. Lec. notes for EEB, 581(540), 3.

[10] Brooks, S. (1998). Markov chain Monte Carlo method and its application. Journal of the royal statistical society: series D (the Statistician), 47(1), 69-100.

[11] Meng, Qingyi. (2012). Briefly discuss the application of Markov chain Monte Carlo in practice. Journal of educational institute of Jinlin Province Vol.28 (12),120-121.

[12] Robert, C. P., Casella, G., Robert, C. P., & Casella, G. (2004). The metropolis—hastings algorithm. Monte Carlo statistical methods, 267-320.

[13] Rouchka, E. C. (1997). A brief overview of Gibbs sampling. Bioinformatics Technical Report Series, No TR-ULBL-2008-02.

[14] Casella, G., & George, E. I. (1992). Explaining the Gibbs sampler. The American Statistician, 46(3), 167-174.