Dean&Francis

ISSN 2959-6157

Theory of Markov Chain Monte Carlo method and Its Several Applications

Ao Xiong

Beijing Etown Academy, Beijing, China

burnsr3@nwfsc.edu

Abstract:

Markov Chain Monte Carlo (MCMC) methods represent a significant advancement in computational statistics, offering powerful tools for solving complex problems involving high-dimensional probability distributions. This paper provides a comprehensive overview of the theoretical foundations and practical applications of MCMC methods. This paper begins by discussing the fundamental principles of Markov chains and Monte Carlo simulations, highlighting how their combination facilitates the estimation and how it use for solving of complex integrals and optimization problems. The paper further explores various applications of MCMC in fields such as finance, computer science, and biology, including risk management, Bayesian inference, and genetic data analysis. Despite their extensive use, MCMC methods face challenges related to convergence and computational efficiency, which are addressed through ongoing advancements in algorithmic techniques and computational resources. This overview aims to elucidate the core principles and practical relevance of MCMC methods, offering insights into their applications and encouraging future research in this dynamic area.

Keywords: Markov Chain Monte Carlo, Bayesian Inference, Computational Statistics, Genetic Data Analysis.

1. Introduction

In this paper, the author wants to introduce about MCMC method and its application. Basic on this purpose the author will firstly talk about the background of Markov chain and Monte Carlo Methods which include three ideas: state machine, random process and Markov Chain. Then the author will introduce about Monte Carlo method. The author aspiration is to introduce Markov Chain Monte Carlo method (MCMC method) as clear as possible to people who might have no foundation of MCMC method. After the author talk about previous idea, the author will introduce three specific applications in different ground in order to makes reader to have a relatively comprehensive idea about what MCMC method can do and why they make sense to almost every ground. With the development of Markov Chain Monte Carlo (MCMC) algorithms, statistical computing and data analysis have transformed providing powerful tools to tackle complex problems in different disciplines. MCMC methods can be thought of as an extension to the Monte Carlo simulations, refined by incorpo-

rating properties associated with Markov chains. Markov chains are sequences of random variables where the future state depends only on now in real time, not on all events that came before. Monte Carlo methods use random sampling to solve problems that are not deterministic and often analytically intractable [1].

Naively, one may think these are two disjoint ideas: Markov chain methods and Monte Carlo techniques (algorithms), but in fact combining them for sampling from high-dimensional probability distributions is a beautiful engine. MCMC methods work great for computing integrals over complex spaces of probabilities and resolving problems based on optimization. Therefore, the ability to deal with high dimension spaces and estimate integrals makes MCMC an important tool for many applications in physics, finance, computation science or biology. In finance, MCMC methods are used to model risk and manage it, which is invaluable for applications like evaluating portfolios or pricing financial derivatives. MCMC flexibility enables the modeling of complex financial instruments and understanding uncertainties in market dynamics. This technique is important in computer science as MCMC has had a major impact on the development of machine learning, especially for Bayesian inference and to create probabilistic models. These strategies help to improve the posterior inference and algorithms update in deep learning settings.

Again, MCMC methods are incredibly common in biological and chemical research to model the pathological processes of more complex systems. Such closures can be found in both stochastic simulations, and inference methods like MCMC employed for instance to analyze genetic data or reconstruct evolutionary histories. They have been used for an easier prediction of chemical behaviors and molecular trans pages in chemistry. Although versatile, MCMC methods have difficulties with convergence and computational burden. Check that Markov at has mixed to the target distribution and computational burden for simulations. New methods and improved computational power seem to be incrementally solving these issues, opening new grounds for MCMC practices. This paper attempts to give an introductory account of the theory and practice of MCMC. This paper aims to spark further research in this fast-evolving field by probing the theoretical foundations and applications of these developments.

2. Method and Theory

2.1 Background of Markov Chain and Monte Carlo Methods

Before delving into the concept of Markov chains, it is essential to introduce several foundational ideas to facilitate a comprehensive understanding.

The first concept is the state machine. A state machine is a mathematical framework that describes a limited set of states and the transitions or actions that occur between these states. In the context of a Markov chain, a state machine is employed to model the state of each variable.

The second concept is a random process. A random process is essentially a collection of random variables, often indexed by time. For example, let $N(t)$ denote the number of customers who have entered a bank from t=9 (when the bank opens at 9:00 am) until time t, on a particular day, where t is in the range [9,16]. In this case, time t is measured in hours, but it can take any real value between 9 and 16. The author assumes that $N(9)=0$, and $N(t)\in\{0,1,2,\ldots\}$ for all $t\in[9,16]$. Notice that for any given time t_1, the random variable $N(t?)$ is discrete. Thus, $N(t)$ is a discrete-valued random process. However, because t can take any continuous value between 9 and 16, $N(t)$ is classified as a continuous-time random process.

Following the introduction of the state machine and random process, this paper can now delve into the concept of the Markov chain [2]. A Markov chain is a mathematical model that characterizes a random process where the current state depends solely on the immediately preceding state, with no consideration of states prior to that. In simpler terms, a Markov chain is a random process that focuses exclusively on the present state, disregarding past states.

For example, imagine someone playing an adventure game that consists of various levels, each with different levels of difficulty and rewards. The player must make decisions at each level to maximize their total rewards. At any given level, they must evaluate the current state (such as the character's health, equipment, and position) and how their choices will influence future states and rewards (e.g., deciding whether to attack or retreat, or which item to select). By utilizing the Markov decision process, one can compute the expected return of each decision and determine the best course of action to maximize overall rewards. This approach aids individuals in making optimal decisions in uncertain scenarios.The Markov process can be described by a simple formula:

$$P\left(X_{(t+1)}=j \mid X_t=i,\ldots,X_0=i_0\right)=P\left(X_{(t+1)}=j \mid X_t=i\right) \quad (1)$$

Where X_t denotes the state at time t, and i and j represent two distinct states within the state space. The expression $P(X_{t+1}=j \mid X_t=i)$ refers to the probability of transi-

ISSN 2959-6157

tioning from state i to state j. Another key concept in the context of Markov processes is the Stationary Markov Process. Let P represent the state transition matrix of the Markov chain, and let π denote the steady-state probability distribution. Based on the stationary state equation, one can derive the following equation: $\pi = \pi P$. This indicates that the steady-state probability distribution is an eigenvector associated with an eigenvalue of 1. To find the steady-state probability distribution, one must solve for the eigenvector corresponding to this eigenvalue.

Markov chains can be fully represented by their initial states and probabilistic transition matrices. Let the vector $\pi(j)$ represent the probability that a random variable in the sequence takes exactly the J^{th} state, P represents the state transition matrix, and P_{ij} represents the probability of moving from the i to the J^{th} state. First conclude about stationary Markov process is:

$$\lim_{n\to\infty} P_{ij}^{n} = \pi(j) \tag{2}$$

Assuming the state space consists of K states (where K can be either finite or infinite), after n multiplications, the value of each element in the square matrix depends solely on the column number. Furthermore, all elements within the same column will share the same value..

$$\lim_{n\to\infty} P^{n} = \begin{pmatrix} \pi(1) & \pi(2) & \dots & \pi(K) \\ \pi(1) & \pi(2) & \dots & \pi(K) \\ \pi(1) & \pi(2) & \dots & \pi(K) \\ \pi(1) & \pi(2) & \dots & \pi(K) \end{pmatrix} \tag{3}$$

Or called:

$$\pi(j) = \sum_{i=0}^{K} \pi(i) P_{ij} \tag{4}$$

Therefore, easily can find when $n \to \infty$, π(j) went to a constant. Based on the previous explanation, the author identifies three key properties. Markov Property: The future state of the system depends solely on the current state, with no influence from any prior states. Time Homogeneity: The transition probabilities remain consistent over time, meaning the transition matrix does not change between different time points. Strong Markov Property: The distribution of the system's future states is completely determined by its present state, regardless of the duration spent in that state [3].

The next topic is Monte Carlo Methods. Monte Carlo methods are computational techniques that estimate values through a large number of random samples to analyze a system. They are highly versatile and effective, yet relatively simple to implement. For many complex problems, Monte Carlo methods are often the most straightforward approach and, in some cases, the only practical solution. For instance, consider a square with a circle inscribed within it, where the circle is tangent to the square. The ratio of their areas is $\pi/4$. By randomly generating 10,000 points (i.e., coordinate pairs (x, y)) inside the square and calculating their distance from the center, one can determine whether each point lies inside the circle. Assuming the points are evenly distributed, the proportion of points inside the circle should approximate $\pi/4$, and multiplying this ratio by 4 provides an estimate of π. This example encapsulates the essence of the Monte Carlo method.

The next concept to discuss is Markov Chain Monte Carlo (MCMC) Integration [4]. The Bayesian inference tells that the observed variables and parameters in the model are all random variables. So, sample $x = (x_1, \dots x_n)$ and parameter θ The joint distribution can be expressed as:

$$f_{x,\theta}(x,\theta) = f_{x|\theta}(x_1, \dots x_n)\pi(\theta) \tag{5}$$

And based on Bayes axiom, the expectation of $g(\theta)$ can be express as:

$$E\left[g(\theta \mid x)\right] = \int g(\theta) f_{\theta|x}(\theta \mid x) d\theta = \frac{\int g(\theta) f_{\theta|x}(x)\pi(\theta) d\theta}{\int f_{\theta|x}(x)\pi(\theta) d\theta} \tag{6}$$

The integration $E[g(\theta \mid x)] = \int g(\theta) f\theta \mid x(\theta \mid x) d\theta$ has its Monte Carlo estimation sample mean:

$$\bar{g} = \frac{1}{m}\sum_{i=1}^{m} g(x_i) \tag{7}$$

where $x_1, \dots, x_m$ is the sample that sampling from $f_{\theta|x}(\theta \mid x)$. When $x_1, \dots, x_m$ independent, based on the law of large numbers tells that as the sample size n approaches infinity, $\bar{g}$ converges to $E[g(\theta \mid x)]$.

2.2 Markov Chain Monte Carlo Methods

In some problems, sampling from distribution function $f_{\theta|x}(\theta \mid x)$ can be really hard, here is This is where the Markov Chain Monte Carlo (MCMC) algorithm comes into play. Another key aspect of the Monte Carlo Method (MCM) is the Metropolis-Hastings (MH) Algorithm, which encompasses four primary sampling techniques: Metropolis sampling, Gibbs sampling, independent sampling, and random walk sampling. The core idea behind the MCMC method is to construct an appropriate Markov chain, with the algorithm primarily focused on utilizing a given Markov chain for sampling purposes. Markov chain $\{X_t \mid t = 0,1,2\dots\}$ were given a state that X_t current-

ly experienced, produce next state X_{t+1}. MH algorithm [5] constructed by following steps.

1. constructed appropriate proposal distribution $g(\cdot|X_t)$

2. produce Y from $g(\cdot|X_t)$

3. if Y are accepted, $X_{t+1}=Y$. Else $X_{t+1}=X_t$

Another concept is that MH sampler. A MH sampling produce Markov chain by following step:

1. constructed appropriate proposal distribution $g(\cdot|X_t)$.

2. Produce X_0 from a distribution g.

3. Repeat until Markov chain reach it equilibrium state.

The probability of acceptance is $\alpha(X_t,Y)=min(1,\frac{f(Y)g(X_t|Y)}{f(X_t)g(Y|X_t)})$. In metropolis algorisms, proposal distribution is symmetric. So, its acceptance probability is $\alpha(X_t,Y)=min(1,\frac{f(Y)}{f(X_t)})$.

3. Result and Application

3.1 Application of MCMC Method in Biology

In gene expression analysis, the MCMC method is often used to fit a mixed model to identify genes that are differentially expressed between different groups (e.g., healthy controls and disease case groups). Specifically, the MCMC algorithm performs the following steps:

1. Initialize parameters: Assign initial values to the parameters of the mixed model (e.g., mean variance and mix ratio)

2. Calculate a posteriori distribution: Use a Markov chain to extract samples from a Posteriori distribution The Posteriori distribution is a joint distribution of model parameters, given the observed data

3. Update parameters: Update model parameters based on the value of the current sample in the Markov chain

4. Repeat steps 2-3: Repeat steps 2-3 until the Markov chain reaches a stationary state, where the parameter estimates no longer change significantly with the number of iterations.

Once the MCMC algorithm converges, the model parameters can be used to identify differentially expressed genes. Specifically, the researchers can calculate the posterior probability of each gene, representing the likelihood that it is differentially expressed between different groups The MCMC method, which has a high posterior probability that genes are considered differentially expressed, has several advantages in gene expression analysis: it can handle complex experimental designs, such as data with multiple sample groups and covariates; it can fit non-normal distributions, a common feature of gene expression data; and it can provide uncertainty estimates of model parameters.

Similar to gene expression, biological network analysis also uses the same logic. In biological network analysis, MCMC methods are often used to infer gene regulatory networks [6], where nodes represent genes and edges represent regulatory interactions. Specifically, the MCMC algorithm performs the following steps:

1. Initializes the network: Specifies initial values for nodes and edges in the network.

2. Calculate a posterior distribution: Use Markov chain to extract samples from a posterior distribution. A posteriori distribution is the joint distribution of network structure and parameters, given the observed data (e.g., gene expression data).

3. Update the network: Update the network structure or parameters based on the value of the current sample in the Markov chain. This can be done by adding or removing nodes or edges, or updating edge weights.

4. Repeat steps 2-3: Repeat steps 2-3 until the Markov chain reaches a stationary state, i.e. the network structure and parameter estimate no longer change significantly with the number of iterations.

Once the MCMC algorithm converges, the network structure and parameters can be used to infer gene regulatory interactions. Specifically, the researchers can identify edges with a high posterior probability, indicating that they represent real regulatory interactions. The MCMC method has several advantages in biological network analysis. It can handle complex data types, such as gene expression data and protein interaction data. It can fit nonlinear models, which is a common feature of gene regulatory networks. It can provide uncertainty estimation of network structure and parameters. One concrete example is the use of MCMCS to infer gene regulatory networks that regulate cell cycles in yeast. The researchers used gene expression data from microarray experiments, in which the expression levels of genes in yeast cells were measured at different time points After the convergence of the MCMC algorithm, the researchers were able to identify gene regulatory interactions with high posterior probabilities and infer the gene regulatory networks that regulate the yeast cell cycle [7].

3.2 Application of MCMC method in Finance and Economics

Firstly, the author needs to explain the idea of GARCH model (Generalized autoregressive conditional heteroscedasticity model). Conditional heteroscedasticity is an econometric model used to capture financial time series

ISSN 2959-6157

data. Conditional heteroscedasticity refers to the phenomenon that the difference of time series varies with time. The GARCH model, proposed by Engle (1982), captures conditional heteroscedasticity by modeling conditional variance as a function of past error terms and conditional variance The GARCH model, called the GARCH (1,1) model, is shown below:

$$r_t = \mu + \sigma_t \times \epsilon_t, \sigma_t^2 = \omega + \alpha \times \sigma_{t-1}^2 + \beta \times r_{t-1}^2 \quad (8)$$

where r_t is the yield of time t. μ is the mean of the yield of time t. σ_t is the conditional volatility of time t. ϵ_t is an independent uniformly distributed standard normal random variable. ω, α and β are the model parameters. The first part of the equation is a simple autoregressive model where the yield r_t consists of its mean μ and a random error term $\sigma_t \times \epsilon_t$ The second part of the equation describes the evolution of the conditional variance σ_t^2 which consists of three parts: the constant term ω, which represents the long run mean of the conditional variance $\alpha \times \sigma_{t-1}^2$, representing the dependence of the conditional variance on the previous period conditional variance α is called the autoregressive coefficient $\beta \times r_{t-1}^2$, representing the dependence of the conditional variance on the square value of the previous period yield β is called the ARCH coefficient (autoregressive conditional heteroscedasticity). For an example, the parameters of the GARCH (1,1) model [8] were estimated using MCMC method. Consider the price of a financial asset to follow GARCH (1,1). In order to estimate the parameters of this model using the MCMC method, one can specify the following prior distributions: $\mu\ Normal(0,1)$, $\omega\ Inverse-Gamma(\alpha,\beta)$, $\alpha\ Beta(a_1,b_1)$, and $\beta\ Beta(a_2,b_2)$. Here, $Normal(0,1)$ is a normal distribution with mean 0 and variance 1. $Inverse-gamma(\alpha,\beta)$ is an Inverse Gamma distribution with shape parameter α and scale parameter β. $Beta(a,b)$ is a beta distribution with shape parameter a and b.

People then use the Metropolis-Hastings algorithm to extract samples from the posterior distribution. A posteriori distribution is the joint distribution of model parameters, given the observed data (for example, the historical rate of return of an asset). Mathematically, the Metropolis-Hastings algorithm is as follows: 1. Initialize the values of the parameters $\mu, \omega, \alpha,$ and β. 2.Generate the candidate parameter μ' from the normal distribution $N(\mu, \sigma^2)$. 3.Generate the candidate parameter ω' from the Inverse Gamma distribution $inverse-gamma(\alpha,\beta)$. 4. Generate the candidate parameter α' from the Beta distribution $Beta(a_1,b_1)$. 5. Generate the candidate parameter β' from the Beta distribution $Beta(a_2,b_2)$. 6. Calculate the probability ratio of the posterior distribution at the candidate parameters:

$$r = \frac{p(\mu', \omega', \alpha', \beta' \mid y)}{p(\mu, \omega, \alpha, \beta \mid y)} \quad (9)$$

where y is the observed data (historical yield). S 7. If $r > 1$, the candidate parameter is accepted as the new parameter value. 8. If $r \leqslant 1$, the candidate parameters are accepted with probability r. 9. Repeat steps 2-8 until the Markov chain reaches a stable state. When the MCMC algorithm converges [6], the model parameters can be used to estimate the conditional distribution of asset returns. In other word, one can calculate the posterior predictive distribution of the future return of an asset, given past price information.

3.3 Application of MCMC Method in Physics

Before the author talks about the specific application in physics, some idea is crucial to understand the application later. The first idea is that cosmic microwave background radiation (CMB). The CMB is faint radiation left over from the Big Bang that carries important information about the early conditions and evolution of the universe. The MCMC method is used to fit CMB data to estimate cosmological parameters such as the Hubble Constant, matter density, and dark energy density. The second idea is that galactic rotation curve. Galactic rotation curve is a way to shows the relationship between the rotation speed of stars in the Milky Way and their distance from the galactic center.

One example is using MCMC to fit CMB data to estimate the Hubble constant, a parameter that describes the expansion rate of the universe. Planck Full-ℓ, Planck $\ell \leqslant 800$, Planck $\ell > 800$, WMAP, ACTPol, and SPT. The author restricts the MCMC to TT power spectra in order to compare with Percival et al [9]. To operate comparison with Fisher matrix predictions one only used multipoles where the likelihood for the power spectrum Cℓ's is approximately Gaussian, and for this reason the author removed $2 \leqslant \ell < 30$ from the Planck Full ℓ, Planck ℓ ≤ 800, and WMAP analyses. For Planck this has only a minor effect on the power-law exponent. Are shown in following Table 1. Table 1 has shown the output of H_0 and Ω_m. The specific use of MCMC is use to calculate the real value interval.

Table 1. Output of H_0 and Ω_m for different experiments.

Experiment	$H_0\left(\frac{km}{s\times Mpc}\right)$	$\Omega_m(h^2)$
PlankFullℓ	67.4 ± 1.0	0.143 ± 0.002
WMAP	70.2 ± 2.2	0.136 ± 0.005
Planck$\ell\leq800$	70.1 ± 1.9	0.137 ± 0.003
Planck$\ell>800$	65.0 ± 1.5	0.149 ± 0.003
ACTPol	74.3 ± 5.7	0.140 ± 0.011
SPT	74.5 ± 3.4	0.129 ± 0.007

4. Conclusion

To wrap-up, Markov Chain Monte Carlo (MCMC) has proven to be an invaluable tool in contemporary computer-intensive statistics that allows a posterior distribution for complex problems to be achieved both rapidly and comprehensively. The MCMC theory, rooted in the fundamentals of Markov chain and Monte Carlo methods permits to estimate high dimensional integrals solve the optimization issue efficiently. The author has seen the considerable implications of MCMC methods in this paper on finance, where it helps improve risk management and portfolio optimization; computer science with Bayesian inference and machine learning as well biology from genetic research to molecular simulations. Though MCMC methods are highly flexible and find wide application, that does not mean they don't have their limitations. Hurdles for continual improvement include convergence, computational cost and the quality of sampling algorithms. Efficiently processing these challenges lies at the heart of improving MCMC methods, primarily by leveraging algorithmic innovations and advancements in computational technologies. The key components of further studies will be the rate at which MCMC converges, reducing computational costs and exploring uncharted territories in cutting-edge areas. Better understanding of these areas can significantly advance MCMC methods and confirm their position as one of the pillars in modern statistical inference and computational modeling. This overview highlights the need for further optimization and implementation of MCMC methods, to enhance our understanding in this area and pave way for more extensive research growth.

References

[1] Karras, C., Karras, A., Avlonitis, M., & Sioutas, S. An overview of mcmc methods: From theory to applications. In IFIP international conference on artificial intelligence applications and innovations (pp. 319-332). Cham: Springer International Publishing, 2012.

[2] Hammersley, J. Monte carlo methods. Springer Science & Business Media, 2013.

[3] Brooks, Stephen. Markov Chain Monte Carlo Method and Its Application. Journal of the Royal Statistical Society: Series D (The Statistician), 1998, 47(1): 69–100.

[4] Kable, Joshua A., et al. Quantifying the CMB Degeneracy between the Matter Density and Hubble Constant in Current Experiments. The Astrophysical Journal, 2019, 871(1): 77.

[5] Chernozhukov, V., & Hong, H. An MCMC approach to classical estimation. Journal of econometrics, 2003, 115(2), 293-346.

[6] Bengtsson, C. Applications of Bayesian Econometrics to Financial Economics, Lund University, 2006.

[7] Johannes, M., & Polson, N. MCMC methods for continuous-time financial econometrics. In Handbook of financial econometrics: Applications. Elsevier, 2010.

[8] Zimeras, S., and F. Gerogiakodis. Bayesian Models for Medical Image Biology Using Monte Carlo Markov Chains Techniques. Mathematical and Computer Modelling, 2005, 42(7–8): 759–68.

[9] Muller, S., et al. A Precise and Accurate Determination of the Cosmic Microwave Background Temperature at z = 0.89. Astronomy & Astrophysics, 2013, 551: A109.