Markov Chain Monte Carlo

Neil Mascarenhas, Sydney Zhong

January 16, 2024

1 Introduction

1.1 Aim of Proposal

In this project, our primary objective is to delve deep into the realm of Markov Chain Monte Carlo (MCMC). We aim to not only understand the mathematical underpinnings of MCMC but also to gain practical insights through the implementation of one of the algorithms widely employed in real-world scenarios.

We will start with a rigorous examination of the theoretical foundations, and then employ an MCMC algorithm to model a complex distribution. This involves not only understanding the fundamental concepts of Markov Chains but also navigating the intricacies of Monte Carlo methods. We will discuss the nuanced landscape of random sampling, scrutinizing both the pitfalls and the benefits inherent in the MCMC methodology.

Our aim is to provide a comprehensive and nuanced understanding of Markov Chain Monte Carlo, offering valuable insights into its theoretical foundations, practical applications, and the broader implications of its utilization in random sampling methodologies.

1.2 Brief Overview of Markov Chain Monte Carlo (MCMC)

MCMC is a method of random sampling that offers a fast alternative to real-life random experimentation. This approach gains its efficiency by harnessing the power of pseudo-random number generators, enabling the rapid generation of collections of samples from diverse probability distributions. This strategic utilization of pseudo-random number generators not only accelerates the sampling process but also ensures a robust and reproducible methodology for drawing samples from complex probability distributions. The versatility of MCMC becomes particularly evident in its application across disciplines characterized by intricate or challenging probability distributions. It finds a niche in fields where conducting real-life experiments and collecting data is a time-intensive endeavor, offering a computational shortcut that proves invaluable in scenarios where efficiency and expeditious results are paramount. As a consequence, MCMC emerges as a go-to tool for researchers and practitioners grappling with intricate models or situations where traditional empirical approaches might be impractical. In this project, we aim to dissect the inner workings of MCMC and its practical applications in domains marked by complex probability distributions and time constraints. Through this exploration, we seek to unravel the layers of sophistication that MCMC brings to the realm of random sampling, elucidating its significance in diverse scientific and computational landscapes.

1.3 Role of MCMC

MCMC plays a crucial role in various fields because it allows people to sample from complex and high-dimensional probability distributions when direct sampling or computation is infeasible. It is used for parameter estimation, model fitting, optimization etc.

1.4 Scope in different applications

MCMC is applicable to a wide range of areas. For example, in astrophysics, MCMC can be used to estimate cosmological parameters related to cosmic microwave background radiation, supernovae data, and dark energy; in computer science, it can be used to solve classic optimization problems such as traveling salesman problem (TSP) and 0-1 knapsack problem and widely used in natural language processing (NLP) for tasks like language modeling[1]. In finance and economics, MCMC is used for tasks such as modeling financial time series data, estimating parameters in economic models, and assessing risks.

1.5 Metropolis-Hastings Algorithm Overview

The Metropolis-Hastings is one of the most common algorithms in MCMC. It can help us sample from an unknown probability distribution with probability density $p(x)$, given we know a function $f(x)$ that is proportional to the desired probability density function $p(x)$

Here is an overview of the Metropolis-Hastings algorithm:

1. Take an easier distribution to sample from, i.e. the normal distribution
2. Propose a next candidate sample based on the current sample
3. Accept or reject the candidate

1.6 Potential Pitfalls

The burn-in period denotes the temporal span required for the Markov chain to attain convergence towards the target distribution. Conventional industry protocol involves excising the burn-in period during the analysis of the target distribution. The duration of the burn-in period is predominantly contingent upon the morphology of the distribution, the selection of the initial point, and the standard deviation or shape of the distribution from which sampling occurs. In instances where the distribution is bimodal

or exhibits intricate morphology, the risk of becoming ensnared in a local optimum is heightened. This is attributable to the preponderance of the curve's area surrounding the local optimum, thereby increasing the likelihood of sampling from that region when employing a uniform distribution.

As the parameter t tends towards infinity, convergence to the target distribution is assured. However, the convergence process is protracted when confronted with numerous local optima. The choice of the initial state further influences both the burn-in period and the time required for convergence. If the initial state resides in one of the distribution's tail ends, and a normal distribution is employed for sampling, the transition toward the distribution's center is delayed. This delay results from the necessity to sample from the distant tails of the normal distribution and the extreme boundaries of the uniform distribution for acceptance. Consequently, the duration for the chain to converge to the target distribution is significantly affected. The optimal initial condition is situated within the distribution's density majority.

In scenarios where multiple modes exist, executing the algorithm iteratively, equal to the number of modes, with each initial state corresponding to a mode may be sensible. However, in cases of complex distributions, this approach becomes impractical. In such situations, testing various initial conditions and subsequently comparing the outcomes serves as a suitable surrogate.

2 Theoretical Background of MCMC

2.1 Definition and Framework

Markov Chain Monte Carlo (MCMC) derives its name from two concepts in mathematics — Markov Chains and Monte Carlo simulation.

A Markov Chain is a mathematical process that undergoes transitions from one state to another, where the future state only depends on the current state and not on any of the previous states.

Monte Carlo simulation is a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results and estimate statistical properties. Markov Chain Monte Carlo (MCMC) is a particular type of Monte Carlo simulation, it incorporates the Markov Chain property where the next sample depends on the current sample. MCMC is a method for sampling from a probability distribution by constructing a Markov chain that has the desired distribution as its equilibrium distribution.

2.2 Definition of Sampling

An experiment is defined as a procedure that can be repeated that has a well-defined set of results. An outcome is the result of one of these experiments. You can generate a probability distribution for a phenomenon using collections of these outcomes and assign probabilities to events. Sampling is generating outcomes using the probability distribution or the phenomenon as the defining rule for generation. An example of sampling in real life would be flipping a fair 6-sided die and writing down the outcome. The probability distribution is $\frac{1}{6}$ for each potential outcome of 1 through 6. In real life, conducting these experiments can be time-intensive and resource-intensive. This is where sampling comes into play. When you have a model you believe accurately represents a natural phenomenon, you likely have a need to have a collection of outcomes that mirrors this distribution so you can conduct analysis on it or understand the phenomenon better without actually conducting experiments in real life. You can generate this collection of outcomes computationally with the use of pseudo-random number generators and the modeled probability distribution of the phenomenon. The combination of these is the basis of Monte Carlo sampling methods.

We defined Markov chain, stochastic matrix, the Markov property, and periodic/aperiodic (recurrent/transient), so we will not redefine this in the paper. The Markov Chain aspect of MCMC is simply that the transition matrix converges to the target distribution, which means that if you record each state, after some time you will be sampling from the target distribution. We prove this below.

2.3 Relevant Theorems and Definitions

Citing from Rosenthal [3], a sufficient condition for MCMC to work is asymptotic convergence to π, the stationary distribution.

Theorem 1. On a finite state space χ, if a time-homogeneous Markov chain is irreducible and aperiodic, then it has a unique stationary distribution π, to which it will converge in distribution as $n \to \infty$.

Definition 2.1 (Irreducible)**.**

$$\forall x, y \in \chi, \exists n \in \mathbb{N} \; s.t. P^n(x, \{y\}) = P(X_n \in \{y\} | X_0 = x) > 0$$

The above definition of irreducible does not make sense in the continuous case because of the fact that this is a countable set space, so Rosenthal [3] discusses a weaker condition that also assures asymptotic convergence called $\phi - irreducible$

Definition 2.2 (ϕ-irreducible)**.**

$$\exists \text{ non-zero } \sigma\text{-finite measure } \phi \text{ on } \chi \; s.t.$$

$$\forall \text{ measurable } A \subseteq X \text{ with } \phi(A) > 0 \; \cap \; \forall x \in \chi, \exists n \in \mathbb{N} \; s.t. \; P^n(x, A) > 0$$

For reference, a σ-finite measure is a countable union of measurable sets with finite measure. Then, Rosenthal states the theorem below, which proves that there exists a continuous analog for asymptotic convergence to the target distribution as long as there is a stationary distribution, which we are no longer guaranteed in the continuous case as our state space is not countable.

Theorem 2. If a Markov chain on a general countably-generated state space χ is ϕ-irreducible and aperiodic, and posesses a stationary distribution, then asymptotic convergence still holds, and

$$\lim_{n\to\infty} \sup_{A\subseteq\chi} |P^n(x, A) - \pi(A)| = 0, \pi\text{-}a.e.x \in \chi$$

This means that if we can prove that the Markov chain in MCMC is irreducible in either sense and aperiodic, and posesses a stationary distribution, it is guaranteed to converge to our target distribution.

2.4 Ideas and Intuitions

Let $f(x)$ be a function that is proportional to the target probability density function $p(x)$

i.e. $p(x) = f(x)/NC$ (normalized constant)

We start by choosing an arbitrary point x_t to be our first sampling point and choosing an arbitrary proposal probability distribution $g(x)$, which can be the normal distribution, we propose a candidate for our next sample x_{t+1}, which depends on x_t, we will need to define a rule on how do we accept or reject the candidate:

We need to show the probability going from x_t to x_{t+1} follows in the detailed balance condition, for any state a and b.

In the Metropolis-Hastings algorithm, the acceptance probability from a to b is defined by the expression:

$$\alpha(b|a) = \min\left(1, \frac{f(b)g(a|b)}{f(a)g(b|a)}\right)$$

With f being a pdf that is proportional to the target distribution p, and g being the proposal probability distribution.

The rationale behind this probability is to favor the acceptance of a candidate if it is in a more probable region. Here, $f(a)$ represents the probability of being at position a, and $g(b|a)$ is the conditional probability of transitioning from a to b. The same principles

apply to $f(b)$ and $g(a|b)$, representing the probability of being at b and transitioning from b to a, respectively. In the discrete case, $g(b|a)$ is well defined for any b and a. However, this probability is 0 in the continuous case, so choosing any given points a and b in the continuous case does not make sense because the probability of choosing any given point is 0. This is remedied by the typical remedies for probabilities in the continuous case, by substituting integrals for sums and by substituting densities in a neighborhood for equality.

If the probability of transitioning from a to b is more likely than the reverse direction, i.e., $\frac{f(b)g(a|b)}{f(a)g(b|a)} > 1$, then $\alpha(b|a) = 1$. In this case, it is desirable to move to b. Conversely, if $\frac{f(b)g(a|b)}{f(a)g(b|a)} < 1$, $\alpha(b|a)$ takes the value of $\frac{f(b)g(a|b)}{f(a)g(b|a)}$. This implies that there is still some probability of moving to b from a, allowing for the exploration of other regions.

The acceptance probability serves a dual purpose: it facilitates the sampling of probable data by favoring moves to more likely regions, and it also permits exploration of less probable regions, ensuring a comprehensive traversal of the entire domain when working with a substantial set of samples.

Furthermore, the acceptance probability reflects the properties of a Markov Chain, whether we accept the next candidate only depends on our current state.

2.5 Transition Matrix and Proprties in Metropolis-Hastings

The transition matrix under Metropolis-Hastings according to the Robert Davies [2] lecture is defined to be

$$P = \begin{bmatrix} s_{x_1,y_1} & \cdots & s_{x_n,y_1} \\ \vdots & \ddots & \vdots \\ s_{x_1,y_n} & \cdots & s_{x_n,y_n} \end{bmatrix}$$

where

$$s_{x_j,y_i} = g(y_i|x_j)\alpha(y_i|x_j) + \mathbb{I}\{y_i = x_j\}(1 - \sum_{i}^{n} g(y_i|x_j)\alpha(y_i|x_j))$$

This only makes sense in the discrete case, but as shown above there exists a continuous analog.

Recall that in a transition matrix P, each entry $P_{x,y}$ represents the probability of transitioning from x to y. The probability of transitioning from state x to state y is simply the probability given we are at state x, we sample at state y and we accept our sample, which is reflected in the first part of the above definition. In the special case of not moving to any other state, i.e. from x to x, we will need to add the probability of rejection which is represented by the second part of the above definition.
Before defining properties, a sanity check is necessary to ensure this is a valid transition matrix. For this, each column's values should sum to 1. Fix state space χ. Fix $x_j \in \chi$. Want to show

$$\sum_i (g(y_i|x_j)\alpha(y_i|x_j) + \mathbb{I}\{y_i = x_j\}(1 - \sum_k^n g(y_k|x_j)\alpha(y_k|x_j))) = 1$$

The proof is as follows:

$$\sum_i g(y_i|x_j)\alpha(y_i|x_j) + \mathbb{I}\{y_i = x_j\}(1 - \sum_k^n g(y_k|x_j)\alpha(y_k|x_j)) =$$

$$\sum_i g(y_i|x_j)\alpha(y_i|x_j) + \sum_i \mathbb{I}\{y_i = x_j\}(1 - \sum_k^n g(y_k|x_j)\alpha(y_k|x_j))$$

$$\sum_i \mathbb{I}\{y_i = x_j\}(1 - \sum_k^n g(y_k|x_j)\alpha(y_k|x_j)) = 0 + (1 - \sum_k^n g(y_k|x_j)\alpha(y_k|x_j))$$

$$\sum_i g(y_i|x_j)\alpha(y_i|x_j) + 1 - \sum_k^n g(y_k|x_j)\alpha(y_k|x_j) = \sum_i g(y_i|x_j)\alpha(y_i|x_j) + 1 - \sum_i^n g(y_i|x_j)\alpha(y_i|x_j) = 1$$

Because x_j was arbitrary, every column adds to 1.

This matrix can obviously only be constructed in the discrete case, because g$(y_k|x_j)$ is only non-zero in the discrete case.

In order for this to have a unique stationary distribution, at some point in time every point needs to be reachable in the state space. This is why you have to choose a proposal distribution with the same range as the target distribution for this algorithm to work. This is obvious even without theory because if your proposal can't reach a spot that your target can, it will miss some of the density of the target distribution. The

chain is time-homogenous because at each step the target distribution remains the same and there is no time-dependent part of the algorithm. This chain is obviously aperiodic in the continuous case because the likelihood of ever returning to a single state is 0. In the discrete case it is aperiodic because we define our matrix with a distribution in mind that has non-zero probability of every outcome. Because we can move from x to x and from x to y with non-zero probability for every x and y in the state space at every step, it is aperiodic because there's no loops. Again by how we define the matrix, every entry is non-zero, so it is irreducible in the discrete case. The definition of probability density function for continuous random variables guarantees that every value in the state space must have a non-zero probability for every measure around a point, and with a proposal distribution that covers the entire range, every point is reachable and all entries of this theoretical infinite dimensional matrix has non-zero entry. In the discrete case, this is enough to prove it has a unique stationary distribution. All that is left to prove in the discrete case is that the stationary distribution is in fact our target distribution $p(x)$. For the continuous case with an infinite state space, it is possible there are multiple stationary distributions, but as shown below $p(x)$ is at least one of them by definition of reversible, and by Theorem 2 it will converge to it as n approaches infinity. From the Miranda Holmes-Cerfon [4] lecture,

Definition 2.3 (Reversible (Detailed Balance Equation))**.** Let $X_0, X_1, ...$ be a Markov chain with a stationary distribution π. The chain is said to be reversible with respect to π or to satisfy detailed balance with respect to π if

$$P_{ij}\pi_i = P_{ji}\pi_j \forall i, j \in S$$

Now, in order to prove that in the discrete case, the unique stationary distribution of the Metropolis-Hastings MCMC algorithm Markov Chain is $p(x)$ and at least one of the stationary distributions in the continuous case, it will suffice to prove it is reversible.

Fix arbitrary $p(x)$ and take $P_{x,y}$ as defined above. For $x \neq y$,

$$\begin{aligned} P_{x,y}p(x) &= g(y|x)\alpha(y|x)p(x) \\ &= g(y|x) \cdot min\{1, \frac{p(y)g(x|y)}{p(x)g(y|x)}\}p(x) \\ &= min\{g(y|x)p(x), g(x|y)p(y)\} \\ &= g(x|y) \cdot min\{\frac{p(x)g(y|x)}{p(y)g(x|y)}, 1\}p(y) \\ &= g(x|y)\alpha(x|y)p(y) \\ &= P_{y,x}p(y) \end{aligned}$$

Trivially, for $x = y$, $P_{x,x}p(x) = P_{x,x}p(x)$. Thus, we have proved that P is reversible with respect to p, thus p is the stationary distribution.[2]

Therefore, because this Markov chain is time-homogenous, aperiodic, irreducible, and has a stationary distribution, in the discrete case our stationary distribution is unique and can only be $p(x)$. Because we did not assume whether or not $p(x)$ was discrete or continuous, we can conclude that in the continuous case $p(x)$ is at least one of the stationary distributions. We can conclude in both the discrete and continuous case that the chain will asymptotically converge to $p(x)$, and so we know that as n approaches infinity, if we record each state along the Markov chain we will be sampling from $p(x)$, the target distribution.

3 Implementation and Examples

3.1 Implementation

The algorithm for Metropolis-Hastings is as follows:

Algorithm 2: Sampling from Distribution

Data:
Number of samples (*num_sample*),
Initial state (*initial_state*),
the target distribution we want to sample from (*target_dis*),
an arbitrary probability distribution we use for sampling the next sample (*Proposal*)
Result: List of samples
start $\leftarrow$ *initial_state*;
samples $\leftarrow$ [*start*];
for *i in range(num_sample)* **do**
    current_state $\leftarrow$ *samples*[-1];
    next_state $\leftarrow$ *proposal*(*current_state*);
    $A \leftarrow \frac{target\_dis(next\_state)*proposal(next\_state—current\_state)}{target\_dis(current\_state)*proposal(current\_state\ —\ next\_state)}$;
    if *random.uniform(0, 1)* $< A$ **then**
        samples.append(*next_state*);
    else
        samples.append(*current_state*);
return *samples*

If you want to check out our code in Python, here is the link: https://github.com/Sydney1004/mcmc

If we choose a symmetric probability distribution as our proposal, such as the normal distribution, the conditional proposal probabilities are equal, i.e., $g(b|a) = g(b|a)$. Consequently, the acceptance probability is simplified to

$$\alpha(b|a) = \min\left(1, \frac{f(b)}{f(a)}\right)$$

This is called the Metropolis algorithm dropping the Hastings.

3.2 Example in Discrete Case

We will do a simple example here to illustrate this process. Consider a biased coin with probability 0.7, i.e. $\mathbb{P}(H) = 0.7$, and we use the probability density function of a fair coin as our proposal. For simplicity, in this case our $p(x)$ is our $f(x)$ divided by constant 1. It is important to note that this works for any $f(x)$ proportional to $p(x)$. Our target distribution is $\begin{bmatrix} 0.7 \\ \\ 0.3 \end{bmatrix}$.
We first consider each entry in our transition matrix:

$$\begin{aligned}
P_{H,H} &= g(H|H)\alpha(H|H) + \rho(H) \\
&= \frac{1}{2} \cdot min\{1, \frac{f(H)g(H|H)}{f(H)g(H|H)}\} + 1 - (g(H|H)\alpha(H|H) + g(T|H)\alpha(T|H)) \\
&= \frac{1}{2} \cdot 1 + 1 - (\frac{1}{2} + \frac{1}{2} \cdot \frac{3}{7}) \\
&= \frac{11}{14} \\
P_{H,T} &= g(T|H)\alpha(T|X) \\
&= \frac{1}{2} \cdot min\{1, \frac{f(T)g(H|T)}{f(H)g(T|H)}\} \\
&= \frac{1}{2} \cdot \frac{3}{7} \\
&= \frac{3}{14}
\end{aligned}$$

$$
\begin{aligned}
P_{T,T} &= g(T|T)\alpha(T|T) + \rho(T) \\
&= \frac{1}{2} \cdot min\{1, \frac{f(T)g(T|T)}{f(T)g(T|T)}\} + 1 - (g(T|T)\alpha(T|T) + g(H|T)\alpha(H|T)) \\
&= \frac{1}{2} + 1 - (\frac{1}{2} + \frac{1}{2}) \\
&= \frac{1}{2} \\
P_{T,H} &= g(H|T)\alpha(H|T) \\
&= \frac{1}{2} \cdot min\{1, \frac{f(H)g(H|T)}{f(T)g(T|H)}\} \\
&= \frac{1}{2} \cdot min\{1, \frac{7}{3}\} \\
&= \frac{1}{2}
\end{aligned}
$$

Now we have our transition matrix P,

$$
\mathbf{P} = \begin{bmatrix} \frac{11}{14} & \frac{1}{2} \\ \frac{3}{14} & \frac{1}{2} \end{bmatrix}
$$

As you can easily verify each of its columns sums to 1. We can start with a random initial state, say H, which can be represented as $[1, 0]^T$

$$
\begin{bmatrix} \frac{11}{14} & \frac{1}{2} \\ \frac{3}{14} & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{11}{14} \\ \frac{3}{14} \end{bmatrix} \approx \begin{bmatrix} 0.786 \\ 0.215 \end{bmatrix}
$$

$$
\begin{bmatrix} \frac{11}{14} & \frac{1}{2} \\ \frac{3}{14} & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} \frac{11}{14} \\ \frac{3}{14} \end{bmatrix} = \begin{bmatrix} \frac{71}{98} \\ \frac{27}{98} \end{bmatrix} \approx \begin{bmatrix} 0.724 \\ 0.276 \end{bmatrix}
$$

$$
\cdots
$$

$$
\begin{bmatrix} \frac{11}{14} & \frac{1}{2} \\ \frac{3}{14} & \frac{1}{2} \end{bmatrix}^3 \cdot \begin{bmatrix} \frac{71}{98} \\ \frac{27}{98} \end{bmatrix} = \begin{bmatrix} \frac{23549}{33614} \\ \frac{10065}{33614} \end{bmatrix} \approx \begin{bmatrix} 0.700 \\ 0.300 \end{bmatrix}
$$

As we can see here, we soon reach the target distribution.

3.3 Modeling a Complex Continuous Distribution

The log-normal distribution is commonly used in finance to model the distribution of stock prices, returns, and other financial variables. This distribution is derived from the normal distribution but is applied to the logarithm of the variable of interest. If X follows a log-normal distribution, then $ln(X)$ follows a normal distribution.

Here are a few reasons why the log-normal distribution is used in finance. Many financial variables, such as stock prices, exhibit multiplicative, rather than additive, behavior. The product of a series of random multiplicative factors leads to a log-normal distribution. This is because the logarithm of a product is the sum of the logarithms. Another reason is that stock prices and other financial variables cannot be negative. The log-normal distribution, being defined over positive real numbers, naturally accommodates this characteristic. The log-normal distribution is positively skewed, which aligns with the observed skewness in many financial time series. This means that extreme positive events (large price increases) are more probable than extreme negative events. In the context of financial returns, which are often expressed as the percentage change in price, the log-normal distribution is used because it ensures that returns are additive over time. If returns were modeled with a normal distribution, compounding would not be correctly captured.

The log-normal distribution is also a fundamental assumption in the Black-Scholes-Merton model, which is widely used for pricing European-style options. The model assumes that stock prices follow a geometric Brownian motion, resulting in a log-normal distribution for future stock prices.

It's important to note that while the log-normal distribution is a useful approximation in many cases, financial markets are complex, and real-world data may not perfectly adhere to this distribution. Other distributions and models may be considered, especially for extreme events and tail risk, where empirical distributions or alternative models (like the Cauchy distribution) might be more appropriate.

The probability density function (pdf) of a log-normal distribution is given by:

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right)$$

The expectation (mean) of a log-normal distribution is:

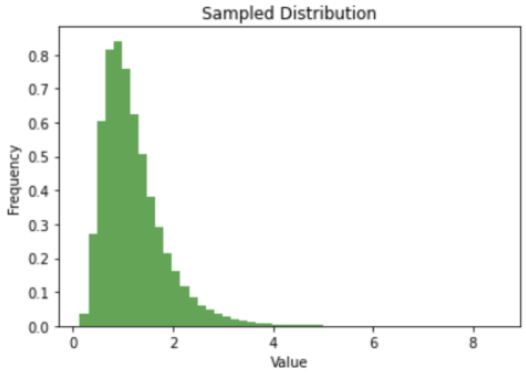
$$E[X] = e^{\mu + \frac{\sigma^2}{2}}$$

The variance of a log-normal distribution is:

$$\mathrm{Var}[X] = (e^{\sigma^2} - 1)e^{2\mu+\sigma^2}$$

μ and σ are the expected value and standard deviation of the random variable $ln(X)$, which is normally distributed. In our example we tested the parameters $\mu = 0$ and standard deviation = 0.5. The expected value of X should be 1.133 and the sample standard deviation should be 0.603. As you can see in below, the sample mean we achieved was 1.200, and the sample standard deviation was 0.63, which is remarkably close considering we only used 100,000 samples.

```
mean: 1.2003587836032739, sd: 0.6275007253094965
```



References

[1] "CS 109: Probability for Computer Scientists, Summer 2021." CS 109, Summer 2021, web.stanford.edu/class/archive/cs/cs109/cs109.1218/. Accessed 7 Dec. 2023.

[2] Davies, Robert. "Simulation - Lectures 7 - MCMC: Metropolis Hastings." www.stats.ox.ac.uk/rdavies/teaching/PartASSP/2020/lectures_latest/simulation_lecture7.pdf. Accessed 7 Dec. 2023.

[3] Rosenthal, J.S. (2009). Markov Chain Monte Carlo Algorithms: Theory and Practice. In: L' Ecuyer, P., Owen, A. (eds) Monte Carlo and Quasi-Monte Carlo Methods 2008. Springer, Berlin, Heidelberg. Accessed 7 Dec. 2023. https://doi.org/10.1007/978-3-642-04107-5_9

[4] Homes-Cerfon, Miranda. "Lecture 3: Markov Chains (II): Detailed Balance, and Markov Chain Monte Carlo (MCMC)." Applied Stochastic Analysis, Spring 2022. Accessed 7 Dec. 2023. https://personal.math.ubc.ca/~holmescerfon/teaching/asa22/handout-Lecture3_2022.pdf