An Introduction to Markov Chain Monte Carlo and Metropolis Hastings

Zhao Bob

June 2025

1 Introduction

Sampling from complex probability distributions is one of the most important aspects of modern study of statistics and probability, and effectively doing so remain an important challenge. The difficulty of such task emerges in many fields not limited to math, from simulating physical processes to unknown data-generating models in machine learning. However, many of these probability distributions that you see in recent times are analytically intractable, meaning we either cannot write down their exact form such a posterior distribution in Bayesian inference. Other times, they are computationally demanding with conventional methods such as rejection sampling. To address these difficulties, researchers developed the **Markov Chain Monte Carlo (MCMC)** as both a robust and a versatile set of techniques to sample from almost any distribution.

MCMC algorithms rely on the construction of **Markov Chain**s as its fundamental process. A Markov Chain is a stochastic process that describes a sequence of possible states that transform solely based on its current state. The transition probability describes the likelihood of transitioning from one state to another. When done right such chains will naturally spend proportionally more time in regions of higher probability over many iterations, effectively generating samples that approximate the desired distribution, or the stationary distribution.

Among MCMC algorithms, the **Metropolis-Hastings** (MH) algorithm is one the most fundamental. It was first introduced by Metropolis et al. in 1953 in the context of simulating particle systems in statistical mechanics, and was later generalized by Hastings in 1970. The core innovation of the MH algorithm lies in its use of a simple and powerful idea. It proposes a move to a new state, and to accept it with a carefully chosen probability that ensures the overall process converges to the correct stationary distribution. Even if the target distribution is only known up to a constant, the MH algorithm can still be used.

The MH algorithm also highlights a profound connection between stochastic processes and probability distributions. By carefully designing the transition kernel of a Markov chain to satisfy properties such as **detailed balance**, we can guarantee that the desired distribution is stationary. And if the chain

is also **irreducible** and **aperiodic**, we are further guaranteed that the chain will converge to this stationary distribution regardless of where it starts. This allows the MH algorithm to serve as a general purpose method for sampling from complex distributions.

In this paper, we provide a rigorous mathematical display of the Metropolis-Hastings algorithm. We begin by introducing the core concepts of Markov chains, including transition kernels, stationary distributions, and key structural properties like irreducibility and aperiodicity. We then define the Metropolis-Hastings transition kernel and prove that it satisfies the detailed balance condition. From there, we show that under mild assumptions, the MH algorithm yields a Markov chain that converges to the target distribution in total variation distance. We will then look into mixing time or how fast the process converges to the target distribution. Finally we will compare the MH algorithm to rejection sampling and introduce some of the variations of MH.

In doing so, we aim to build both an intuitive and theoretical understanding of why the MH algorithm works, what guarantees its convergence, and why it remains as one of the foundations of computational probability and Bayesian statistics to this day.

2 The Convergence of the Metropolis Hastings algorithm

To rigorously analyze the Metropolis-Hastings algorithm, we must first formalize the mathematical framework. In this section, we introduce key definitions and properties of Markov chains, including transition kernels, stationary distributions, and structural assumptions such as irreducibility and aperiodicity. These concepts provide the theoretical foundation necessary to construct and understand the behavior of MCMC algorithms.

In this section, we also formally prove that the Metropolis-Hastings (MH) algorithm converges to the target distribution under standard assumptions with our established foundational concepts of Markov Chains that the MH algorithm is constructed upon.

2.1 Preliminaries and Definitions

Definition 1. *(Markov Property) A stochastic process $(X_t)_{t\geq 0}$ with state space X satisfies the Markov property if,*

$$\forall\, t \in \mathbb{N}, \quad \forall\, x_0, \ldots, x_{t+1} \in X, \quad \mathbb{P}(X_{t+1} = x_{t+1} \mid X_0 = x_0, \ldots, X_t = x_t) = \mathbb{P}(X_{t+1} = x_{t+1} \mid X_t = x_t).$$

This property says that the future behavior of a Markov process only depends on the present state, not the sequence of states before it. In other words, once the current state X_t is known, the conditional probability of the next state X_{t+1} is independent from all of the previous states $X_0, X_1, \ldots, X_{t-1}$. This kind of

process is called a **stochastic process**, a system that changes over time in a way that involves randomness. Rather than following a fixed rule, its next state is determined by probabilities. The Markov property makes such stochastic processes particularly tractable, because it allows us to model complex random systems using only local transition rules between states. .

Definition 2. *(Transition Kernel) Additionally, let X be a countable state space, and let $P(x \to x')$ denote the transition kernel of a time-homogeneous Markov chain. Let $\pi\colon\ X \to [0,1]$ be a probability distribution on X.*

A transition kernel P satisfies:

$$\sum_{x'} P(x \to x') = 1, \quad P(x \to x') \geq 0, \quad \forall x, x' \in\ X.$$

This means that starting from any state x, the total probability of transtioning to some state x' in X is one, guaranteeing that the Markov Chain must produce a valid distribution over the next states.

Definition 3. *(Stationary Distribution) A distribution π is stationary with respect to the transition kernel P if:*

$$\sum_{x\in\ X} \pi(x)P(x \to x') = \pi(x') \quad \textit{for all } x' \in \mathcal{X}.$$

Meaning that if the probability distribution over the current Markov state is π, and you apply the transition matrix to the state again, the distribution doesn't change.

Definition 4. *(Detailed Balance) Let π be a probability distribution on a countable state space X, and let $P(x \to x')$ be the transition probabilities of a Markov chain. We say that π satisfies detailed balance with respect to P if:*

$$\pi(x)P(x \to x') = \pi(x')P(x' \to x) \quad \textit{for all } x, x' \in\ X.$$

This condition implies that for every pair of states x and x', the probability of flowing from state x to x' is exactly balanced by the flow from x' back to x. This is a stronger condition than stationarity and guarantees that the transitions between states are symmetric. If a distribution π satisfies detailed balance with respect to P, then π is stationary for P.

Lemma 1 (Detailed Balance implies Stationarity)**.** *Let $P(x \to x')$ be the transition kernel of a Markov chain over state space X, and let π be a distribution on X. If π satisfies the detailed balance condition*

$$\pi(x)P(x \to x') = \pi(x')P(x' \to x) \quad \textit{for all } x, x' \in X,$$

then π is a stationary distribution for P, i.e.,

$$\sum_{x\in X} \pi(x)P(x \to x') = \pi(x') \quad \textit{for all } x' \in X.$$

Proof. If we assume that detailed balance holds:

$$\pi(x)P(x \to x') = \pi(x')P(x' \to x) \quad \text{for all } x, x' \in X.$$

Now with any state, $x' \in X$. We compute

$$\begin{aligned}\sum_{x \in X} \pi(x)P(x \to x') &= \sum_{x \in X} \pi(x')P(x' \to x) \\ &= \pi(x') \sum_{x \in X} P(x' \to x) \\ &= \pi(x') \cdot 1 \\ &= \pi(x').\end{aligned}$$

This uses the assumption of detailed balance to substitute $\pi(x)P(x \to x')$ with $\pi(x')P(x' \to x)$, and the fact that the transition probabilities from x' sum to 1

$$\sum_{x \in X} P(x' \to x) = 1.$$

□

Therefore, π satisfies the stationarity condition.

Detailed balance is often used because it simplifies the verifying of stationarity, and at the same time, it also implies that the chain will converge to the distribution π . While not necessary for stationarity, it is a sufficient condition and is commonly employed in the design and foundation of sampling algorithms such as Metropolis-Hastings.

Definition 5. *(Irreducibility) A Markov chain with state space X and transition kernel P is said to be irreducible if for any two states $x, x' \in X$, there exists a positive integer $t \in \mathbb{N}$ such that*

$$P^t(x \to x') > 0.$$

This means that it is possible to reach any state x' from any other state x in a finite number of steps, with positive probability. Irreducibility guarantees that the Markov chain is not broken up into isolated subsets of states, and all parts of the state space are connected.

Definition 6. *(Aperiodicity) A state $x \in X$ is said to have period d if*

$$d = \gcd\{t \in \mathbb{N} : P^t(x \to x) > 0\}.$$

The period (d) of a state x is the greatest common divisor of all time steps t such that the chain can return to x in exactly t steps. A Markov chain is aperiodic if every state has period 1. Intuitively, this means that the chain does not get trapped in predictable cycles of fixed length. Instead, it has the flexibility to return to a state at irregular, non-multiplicative intervals. In many

practical Markov chains, aperiodicity is ensured by a nonzero self-transition probability, that is, $P(x \to x) > 0$. This makes it possible to stay in place with some probability, preventing rigid periodic behavior.

Irreducibility ensures that the chain can explore the entire state space, while aperiodicity ensures that it does not get stuck in cyclical behavior. When combined with the existence of a stationary distribution, these two conditions are sufficient to guarantee convergence of the chain to that distribution, regardless of the starting state.

2.2 Example: A Simple Markov Chain

To better illustrate the definitions introduced above, consider the following Markov chain with a finite state space $X = \{A, B, C\}$. The transition probabilities between states are given by the matrix:

$$P = \begin{bmatrix} 0.2 & 0.5 & 0.3 \\ 0.4 & 0.2 & 0.4 \\ 0.3 & 0.3 & 0.4 \end{bmatrix}$$

This matrix defines a transition kernel $P(x \to x')$, where the entry in row x, column x' represents the probability of transitioning from state x to state x'. Each row sums to 1, satisfying the condition for a valid transition kernel.

This Markov chain is:

- **Irreducible:** From any state, there is a nonzero probability of reaching any other state in a finite number of steps.
- **Aperiodic:** Each state has a nonzero probability of remaining in the same state (i.e., $P(x \to x) > 0$), ensuring the chain is not stuck in fixed-length cycles.
- **Finite:** The state space is finite with only 3 elements.

A graphical representation of the Markov chain is shown in Figure 1, with arrows denoting transitions and edge labels indicating probabilities.

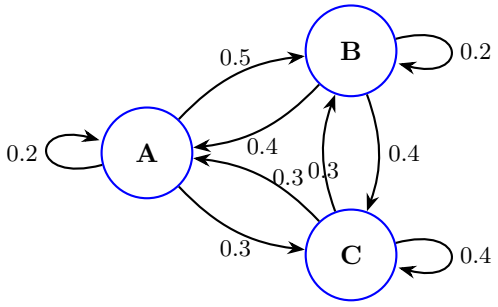


Figure 1: A 3-state Markov chain with labeled transition probabilities.

2.3 The Metropolis Hastings Algorithm

The definitions and preliminaries form the foundation for constructing valid Markov chains. Concepts like the Markov property, transition kernels, irreducibility, aperiodicity, and stationary distributions are not just theoretical and irrelevant to the MH algorithm, they are the requirements for designing markov chains that converge to a desired long-term behavior. MCMC)methods use this structure in reverse: rather than analyzing an existing process, MCMC constructs a Markov chain whose stationary distribution is a specific target distribution π from which we want to sample. In other words, MCMC "reverse engineers" a transition process so that, over time, the samples it generates follow π, even if π is only known up to a normalizing constant. One of the most widely used MCMC algorithm, the Metropolis-Hastings algorithm, defines a transition kernel using a proposal distribution and an acceptance probability in such a way that the resulting chain satisfies detailed balance with respect to π, guaranteeing that π is stationary. If the chain is also irreducible and aperiodic, the Markov chain will converge to π regardless of the starting point; that is called ergodicity, which the paper will later touch on.

We will start by defining the key components of the Metropolis Hastings Algorithms.

Definition 7. *(Proposal Distribution) Let $q(x \to x')$ be a proposal distribution satisfying:*

$$\sum_{x'} q(x \to x') = 1, \quad \forall x \in X.$$

Definition 8. *Given a target distribution $\pi \colon X \to [0,1]$, the Metropolis-Hastings transition kernel $P(x \to x')$ is defined as:*

$$P(x \to x') = \begin{cases} q(x \to x') \cdot \alpha(x, x') & \text{if } x \neq x', \\ 1 - \sum_{x'' \neq x} q(x \to x'') \cdot \alpha(x, x'') & \text{if } x = x', \end{cases}$$

where the acceptance probability is defined as,

$$\alpha(x, x') = \min\left(1, \frac{\pi(x')q(x' \to x)}{\pi(x)q(x \to x')}\right).$$

This construction defines how the Markov chain evolves under the Metropolis-Hastings algorithm. The proposal distribution $q(x \to x')$ suggests a possible next state x' given the current state x, and the chain accepts this proposed move with probability $\alpha(x, x')$. If $x' \neq x$, the transition probability is given by the product $q(x \to x') \cdot \alpha(x, x')$. However, with some probability, the proposed move is rejected, and the chain remains at the current state. The case $x = x'$ ensures that the total transition probability out of x still sums to 1. This is handled by subtracting the total probability of moving to any other state $x'' \neq x$. This construction guarantees that the resulting kernel $P(x \to x')$ defines a valid Markov chain that keeps π as its stationary distribution.

This construction guarantees that the chain is a valid Markov chain and ensures reversibility with respect to π, provided that $q(x \to x') > 0 \Rightarrow q(x' \to x) > 0$ and $\pi(x) > 0$ for all $x \in X$.

Lemma 2. *The transition kernel P defined by the Metropolis-Hastings algorithm satisfies the detailed balance condition:*

$$\pi(x)P(x \to x') = \pi(x')P(x' \to x).$$

Proof. We prove it by looking at the two cases $x \neq x'$ and $x = x'$.

Case 1: $x \neq x'$. Then,

$$\begin{aligned}\pi(x)P(x \to x') &= \pi(x)\, q(x \to x')\, \alpha(x, x') \\ &= \pi(x)\, q(x \to x') \cdot \min\left(1, \frac{\pi(x')\, q(x' \to x)}{\pi(x)\, q(x \to x')}\right).\end{aligned}$$

There are two sub-cases: - If the ratio inside min is less than 1, then:

$$\begin{aligned}\pi(x)P(x \to x') &= \pi(x)\, q(x \to x') \cdot \frac{\pi(x')\, q(x' \to x)}{\pi(x)\, q(x \to x')} \\ &= \pi(x')\, q(x' \to x) \\ &= \pi(x')P(x' \to x).\end{aligned}$$

- If the ratio is greater than or equal to 1, then:

$$\begin{aligned}\pi(x)P(x \to x') &= \pi(x)\, q(x \to x'), \\ \pi(x')P(x' \to x) &= \pi(x')\, q(x' \to x) \cdot \frac{\pi(x)\, q(x \to x')}{\pi(x')\, q(x' \to x)} \\ &= \pi(x)\, q(x \to x').\end{aligned}$$

In both sub-cases, the equality holds.

Case 2: $x = x'$. Then,

$$\pi(x)P(x \to x) = \pi(x)\left[1 - \sum_{y \neq x} q(x \to y)\alpha(x, y)\right],$$

and

$$\sum_{y \neq x} \pi(x)q(x \to y)\alpha(x, y) = \sum_{y \neq x} \pi(y)q(y \to x)\alpha(y, x),$$

so the flow into and out of x is balanced, which maintains the detailed balance condition.

□

Having proved that the Metropolis-Hastings transition kernel satisfies the detailed balance condition with respect to the target distribution π, it follows that π is a stationary distribution of the Markov chain defined by this kernel.

However, stationarity alone does not guarantee that the chain will converge to π from any initial state. To ensure convergence, we must consider the dynamical properties of the chain, more specifically, whether it is irreducible and aperiodic. These properties, when satisfied, imply that the chain is ergodic, which in turn ensures that the distribution of the chain after many steps approaches the unique stationary distribution π, regardless of the starting distribution. We now formalize this convergence behavior and state the conditions under which it holds.

Definition 9. *The **total variation distance** between two distributions μ and ν on a countable space X is defined as:*

$$\|\mu - \nu\|_{TV} := \frac{1}{2} \sum_{x \in X} |\mu(x) - \nu(x)|.$$

Theorem 1. *Let P be the transition kernel defined by the Metropolis-Hastings algorithm on a finite state space X, with target distribution π. If P is irreducible and aperiodic, then P is ergodic and satisfies:*

$$\lim_{t \to \infty} \|\mu P^t - \pi\|_{\mathrm{TV}} = 0,$$

for any initial distribution μ.

Proof. Since the state space X is finite and P satisfies irreducibility and aperiodicity, standard results from Markov chain theory ensure that P is ergodic. Additionally, the Metropolis-Hastings kernel satisfies detailed balance with respect to π, so π is a stationary distribution of the chain. Ergodicity combined with stationarity guarantees convergence of the distribution μP^t to π in total variation distance. □

This means that regardless of where the Markov chain starts, after many iterations, the samples it produces will resemble draws from the target distribution π. This convergence is what makes MCMC a powerful tool for sampling from complex, high-dimensional, or unnormalized distributions.

2.4 An Example of the Metropolis-Hastings Algorithm

To illustrate how the Metropolis-Hastings (MH) algorithm operates in practice, we now are going to walk through a concrete example where the target distribution π is known up to a constant, and we show how the algorithm constructs a Markov chain to approximate samples from it. This will help demonstrate step by step how MH works and solidify the abstract definitions in the previous section.

Target Distribution

Let the target distribution $\pi(x)$ be proportional to a function that is not normalized:

$$\pi(x) \propto f(x) = e^{-x^4+3x^2}, \quad x \in \mathbb{R}.$$

This is a bimodal distribution with modes around $x = \pm 1.2$, but we cannot directly sample from $\pi(x)$ because it lacks a closed-form normalization constant.

This means that while we know the unnormalized shape of the distribution, we do not know the constant that scales it to become a valid probability distribution. A closed-form normalization constant refers to an explicit expression for the integral of the unnormalized function over its entire domain, which would allow us to compute the exact probability density function. In our case, the target distribution is given only up to a proportional function:

$$\pi(x) \propto f(x) = e^{-x^4+3x^2},$$

but computing the integral

$$Z = \int_{-\infty}^{\infty} e^{-x^4+3x^2}\, dx$$

is analytically intractable. Without this constant Z, we cannot normalize $f(x)$ to obtain $\pi(x) = \frac{1}{Z} f(x) = 1$. This makes standard sampling methods, such as inverse transform sampling or rejection sampling, difficult to use directly. The Metropolis-Hastings algorithm overcomes this issue by relying only on the ratio $\frac{\pi(x')}{\pi(x)}$, where the unknown constant Z cancels out.

Although the normalization constant $Z = \int_{-\infty}^{\infty} e^{-x^4+3x^2}\, dx$ exists and can be approximated numerically, it does not have a closed-form expression. That is, we cannot write down the exact value of Z using standard mathematical functions . In many real-world cases, especially in high-dimensional problems, this makes direct sampling from such functions intractable. This is where MCMC methods like Metropolis-Hastings are useful; they are able to sample from such distributions without requiring knowledge of the normalization constant.

Proposal Distribution

We now choose a symmetric proposal distribution $q(x \to x')$, specifically a Gaussian random walk:

$$x' = x + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2).$$

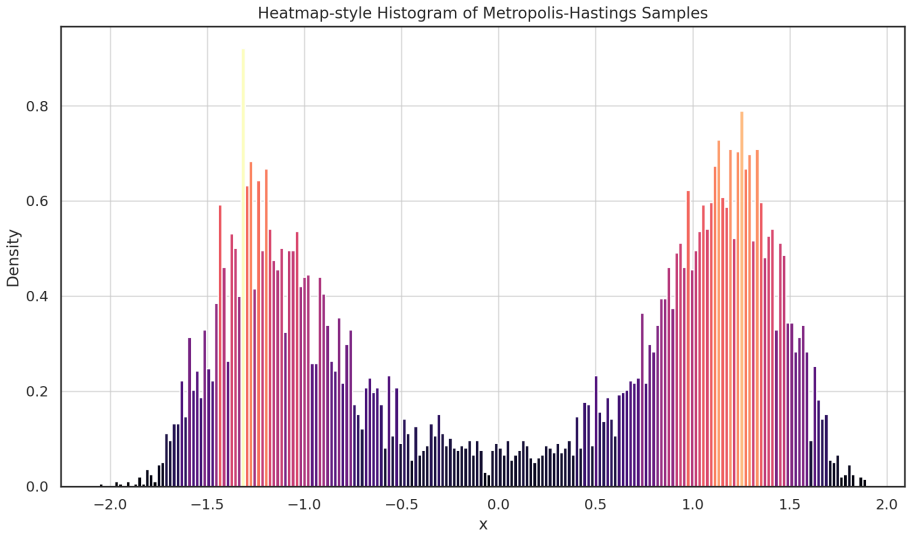
This means $q(x \to x') = q(x' \to x)$, so the proposal is symmetric. The choice of σ affects how far the walker can jump in one step and impacts mixing behavior, which we will discuss later in the paper.

Algorithm Steps

Lets denote the current state of the chain as x_t. At each iteration:

1. Propose a new state: $x' = x_t + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$.
2. Compute the acceptance probability:

$$\alpha(x_t, x') = \min\left(1, \frac{f(x')}{f(x_t)}\right) = \min\left(1, \frac{e^{-x'^4+3x'^2}}{e^{-x_t^4+3x_t^2}}\right).$$



3. Now, we accept or reject: Generate $u \sim \text{Uniform}(0, 1)$. If $u < \alpha(x_t, x')$, then set $x_{t+1} = x'$. Otherwise, set $x_{t+1} = x_t$.

4. Repeat the process for the desired number of iterations.

Visualizing the Chain

Suppose we start at $x_0 = 0$, and run the chain for 10,000 steps. After discarding the first few thousand samples as *burn-in* (which is standard practice to allow the chain to reach stationarity), the remaining samples can be used to estimate properties of π, such as the mean, variance, or the shape of the distribution.

Figure 2.4 shows a histogram of the samples after burn-in. We can see that the samples concentrate around $x = -1.2$ and $x = +1.2$, which correspond to the modes of $\pi(x) \propto e^{-x^4+3x^2}$. This shows that the Metropolis-Hastings algorithm correctly samples from the target distribution.

Discussion

This simple example illustrates several key strengths of the MH algorithm:

- It does not require the normalizing constant of the target distribution. Only ratios $\pi(x')/\pi(x)$ are needed, and that makes it ideal for Bayesian applications where the posterior is known only up to a constant.

- Even in complex or multimodal distributions like the one above, MH is able to produce accurate samples, provided the proposal distribution is chosen well.

- The symmetry of the proposal distribution simplifies the acceptance ratio, but the algorithm still works with asymmetric proposals as long as the ratio $\frac{\pi(x')q(x'\to x)}{\pi(x)q(x\to x')}$ is used.

In later sections, we will analyze how the choice of proposal affects mixing time and convergence, and we will discuss how examples like this extend to high-dimensional settings common in machine learning and statistical inference.

3 Mixing Time in Metropolis-Hastings

Although the convergence of the Metropolis-Hastings algorithm to the stationary distribution π is guaranteed under irreducibility and aperiodicity, these results are asymptotic: they describe the behavior of the chain as $t \to \infty$. In real applications, we are always limited to a finite number of steps, so a natural question arises: how quickly does the Markov chain constructed by Metropolis-Hastings approach stationarity?

This question is answered by the notion of *mixing time*, which measures how many steps are required before the distribution of the chain is close to π. Formally, the mixing time $t_{\text{mix}}(\varepsilon)$ is defined as

$$t_{\text{mix}}(\varepsilon) = \min\left\{ t \in \mathbb{N} \mid \max_{x \in X} \left\| P^t(x, \cdot) - \pi \right\|_{\text{TV}} \leq \varepsilon \right\},$$

where $\|P^t(x,\cdot) - \pi\|_{\text{TV}}$ denotes the total variation distance.

In the context of Metropolis-Hastings, the mixing time depends critically on the design of the proposal distribution $q(x \to x')$ and the structure of the target distribution π. This section examines how these two components interact to influence the rate of convergence.

Step Size vs. Acceptance Rate Trade-off

Consider a symmetric Gaussian random walk proposal:

$$x' = x + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2).$$

As σ increases, the chain can explore the state space through bigger steps, but the acceptance rate decreases due to more frequent proposals in low-probability regions. When σ is too large, most proposals are rejected and the chain stagnates. Conversely, when σ is too small, the chain accepts frequently but moves slowly, resulting in poor exploration.

This trade-off between step size and acceptance rate is very important to the efficiency of MH. A good mixing time requires a proposal distribution that balances local exploration with sufficient movement to reach distant regions of high probability.

Multimodal Distributions and Trapping

Mixing time increases significantly when the target distribution π is multimodal. In such cases, MH chains often become trapped in a single mode or high frequency area, especially when the proposal distribution does not allow for transitions across low-density regions. Since the acceptance probability is low for proposals that jump across modes, the chain may take an exponentially long time to transition between them.

Let π be a distribution with two modes separated by a large region of near-zero probability. If the proposal distribution only supports local moves, the probability of jumping from one mode to the other is exponentially small in the distance between them. This causes the chain to remain in a single mode for many iterations, drastically increasing mixing time.

3.1 Example: Mixing Time in a Bimodal Distribution

To illustrate how proposal design affects mixing time in Metropolis-Hastings, we revisit the target distribution

$$\pi(x) \propto f(x) = e^{-x^4+3x^2},$$

a smooth, bimodal distribution with peaks near $x \approx \pm 1.2$. Although the distribution is symmetric, its non-convexity poses a challenge for Metropolis-Hastings chains with naive proposals, which often struggle to transition between modes. This makes it an ideal case study for analyzing how mixing time is influenced by proposal scale.

We consider a Gaussian random walk proposal of the form

$$x' = x + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2),$$

and examine the behavior of the chain under different values of σ. The corresponding proposal density is symmetric:

$$q(x \to x') = q(x' \to x),$$

which simplifies the acceptance probability to

$$\alpha(x, x') = \min\left(1, \frac{f(x')}{f(x)}\right).$$

Local Behavior of the Acceptance Ratio

To understand how the acceptance probability behaves with varying σ, we consider the logarithmic ratio:

$$\log\left(\frac{f(x+\epsilon)}{f(x)}\right) = -[(x+\epsilon)^4 - x^4] + 3[(x+\epsilon)^2 - x^2].$$

Using Taylor expansions around small ϵ, we get:

$$(x+\epsilon)^2 - x^2 = 2x\epsilon + \epsilon^2,$$
$$(x+\epsilon)^4 - x^4 = 4x^3\epsilon + 6x^2\epsilon^2 + \mathcal{O}(\epsilon^3).$$

Therefore, for small ϵ, the log-acceptance ratio behaves like

$$\log\left(\frac{f(x+\epsilon)}{f(x)}\right) \approx -4x^3\epsilon - 6x^2\epsilon^2 + 6x\epsilon + 3\epsilon^2.$$

This suggests that large proposals (large ϵ) are penalized heavily, especially in regions far from the modes. This confirms that the acceptance rate declines sharply as σ increases.

Expected Squared Jump Distance

To quantify both acceptance and exploration in MH, we define the *expected squared jump distance (ESJD)*:

$$\text{ESJD} = \mathbb{E}[(x_{t+1} - x_t)^2] = \mathbb{E}[\epsilon^2 \cdot \alpha(x, x+\epsilon)],$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is the proposal step.

This definition captures a trade-off between two goals: making large moves to explore the space (via ϵ^2) and maintaining a high acceptance rate (via $\alpha(x, x+\epsilon)$). If σ is too small, the chain only makes small moves and mixes slowly. If σ is too large, the acceptance probability drops and most proposals are rejected. The product $\epsilon^2 \cdot \alpha$ represents how far the chain actually moves on average, not just how far it tries to move. This makes ESJD a useful tool for choosing the right proposal variance σ^2 to ensure efficient sampling.

Empirical Switching Time Between Modes

To empirically demonstrate the impact on mixing, let us define the *first switch time* T_{switch} as the number of steps it takes for the chain to visit both modes. For the purposes of this example, we define mode neighborhoods as:

$$\mathcal{M}_{\pm} = \{x \in \mathbb{R} \mid |x \mp 1.2| < 0.3\}.$$

We simulate the MH algorithm for 10,000 steps from $x_0 = 0$ using three values of σ, and record T_{switch} as the iteration index when the chain first visits both $\mathcal{M}_+$ and $\mathcal{M}_-$.

Proposal Std. Dev. σ	Acceptance Rate (Approx.)	T_{switch}
0.1	≈ 0.97	> 4000
1.0	≈ 0.45	≈ 500
3.0	< 0.05	> 7000

As the table shows, extremely small or large σ values yield high switching times, either due to slow movement (small σ) or low acceptance (large σ). Only the intermediate value achieves good exploration within reasonable time, demonstrating the importance of a good mixing time.

4 Why Metropolis-Hastings Outperforms Other Sampling Methods

Having seen the flexibility and robustness of the Metropolis-Hastings algorithm, it is worth comparing it to a more classical method: *rejection sampling*. Although both algorithms can, in principle, be used to generate samples from a target distribution, their practical applicability and scalability differ dramatically.

4.1 Comparing MH to Rejection Sampling

Rejection sampling is conceptually simple and produces independent samples, but its usefulness is severely constrained. In contrast, Metropolis-Hastings sacrifices independence to gain generality and tractability, especially in cases where the target distribution is known only up to a normalizing constant or has complex structure.

Rejection Sampling: Setup and Limitations

Suppose the target distribution $\pi(x)$ is known up to a constant, i.e.,

$$\pi(x) = \frac{f(x)}{Z}, \quad \text{where } Z = \int f(x)\,dx \text{ is unknown.}$$

Rejection sampling requires a proposal distribution $q(x)$ and a constant $c \geq 1$ such that

$$f(x) \leq c \cdot q(x) \quad \text{for all } x.$$

Then, the procedure is:

1. Sample $x \sim q(x)$ and $u \sim \text{Uniform}(0, 1)$.
2. Accept x if $u \leq \frac{f(x)}{c \cdot q(x)}$; otherwise, reject and repeat.

While it mathematically makes sense, the main issue lies in the requirement that the envelope constant c must bound the target-to-proposal ratio globally. For most nontrivial or high-dimensional distributions, satisfying this condition makes c extremely large, which in turn makes the acceptance rate $1/c$ very small.

For example, in even high dimensional settings, the regions of high probability under $f(x)$ occupy a negligible volume relative to any realistic $q(x)$. As a result, the most of the proposals are rejected. This inefficiency renders rejection sampling effectively useless in such cases.

Metropolis-Hastings: Local Adaptivity and Scalability

The Metropolis-Hastings algorithm avoids this global bounding requirement entirely. Instead of needing a constant c that dominates $f(x)/q(x)$ everywhere, MH only requires evaluating the local ratio

$$\frac{\pi(x')q(x' \to x)}{\pi(x)q(x \to x')},$$

at each proposal step. This local comparison enables MH to operate in settings where rejection sampling would fail outright, such as when the normalizing constant of π is unknown, or when the target has sharp peaks or complex multimodal structure.

Moreover, the MH framework adapts naturally to high dimensions, particularly when the proposal distribution is tuned to reflect the geometry of the target. For example, using isotropic Gaussian proposals or gradient-informed dynamics allows the chain to navigate complicated probability distributions and shapes without requiring any global envelope.

While rejection sampling has historical importance and remains theoretically appealing for its simplicity and independence, it is fundamentally limited by its reliance on global bounds and envelope functions. These requirements are often unrealistic in practical problems, especially in high-dimensional or irregular distributions.

The Metropolis Hastings algorithm, on the other hand, is not only more general, but also practically superior. Its ability to operate with only unnormalized densities, make local decisions, and adapt to complex target structures makes it a better method in almost all every way. In this light, rejection sampling is best viewed as a predecessor to Metropolis-Hastings, good in theory but rarely usable in practice.

4.2 Failure Example of Rejection Sampling on a Bimodal Distribution

To demonstrate the limitations of rejection sampling in practice, consider again the unnormalized bimodal distribution

$$\pi(x) \propto f(x) = e^{-x^4+3x^2},$$

defined over $x \in \mathbb{R}$. The function $f(x)$ has two sharp peaks near $x \approx \pm 1.2$ and decays rapidly toward zero outside a small interval. While Metropolis-Hastings can easily sample from this distribution using only the ratio $f(x')/f(x)$, rejection sampling requires a global bound of the form

$$f(x) \leq c \cdot q(x), \quad \forall x \in \mathbb{R},$$

for some proposal distribution $q(x)$ and constant c.

Suppose we choose a natural proposal, a standard normal distribution $q(x) = \mathcal{N}(0, 1)$, which has its peak at zero. However, unlike $f(x)$, which is bimodal,

the Gaussian proposal places its highest probability mass in a region where $f(x)$ is very small. Consequently, to ensure that $f(x) \leq c \cdot q(x)$ for all x, we must choose

$$c \geq \sup_{x \in \mathbb{R}} \frac{f(x)}{q(x)}.$$

Computing this ratio numerically reveals that $f(x)/q(x)$ attains its maximum not at $x = 0$, but near the modes of f, where $q(x)$ is exponentially small. For instance, at $x = \pm 1.2$, we find:

$$\frac{f(1.2)}{q(1.2)} \approx \frac{e^{-1.2^4+3(1.2)^2}}{\frac{1}{\sqrt{2\pi}}e^{-1.2^2/2}} \approx \frac{e^{1.5552}}{e^{-0.72}} \approx e^{2.275} \approx 9.73.$$

This implies that the minimum valid rejection constant is at least $c \approx 9.73$, giving an expected acceptance rate of less than $\frac{1}{10}$.

But what make it even worse is that this is only for dimension $d = 1$. In higher dimensions, the discrepancy between the support of f and q becomes much more severe. Since the volume of high-probability regions concentrates near the mode(s) and shrinks exponentially, the required constant c increases rapidly with dimension — and the acceptance rate decays correspondingly.

Thus, rejection sampling with a standard Gaussian proposal is already highly inefficient for this simple one-dimensional bimodal target. The situation deteriorates rapidly in higher dimensions or with more complex targets.

5 Advanced Variants of Metropolis-Hastings

The basic Metropolis-Hastings algorithm is versatile, but it often performs not too well in high-dimensional settings or when the target distribution has strong ridges or multiple separated modes. Over the years, several powerful variants have been developed to address these issues by incorporating additional information about the shape of the target or by dynamically adapting the proposal distribution during sampling. In this section, we present three such advanced techniques: the Adaptive Metropolis algorithm, the Metropolis-Adjusted Langevin Algorithm (MALA), and Hamiltonian Monte Carlo (HMC).

5.1 Adaptive Metropolis-Hastings

In the standard MH algorithm, the proposal distribution $q(x \to x')$, such as a Gaussian random walk, is fixed for the entire duration of the sampling process. However, the efficiency of sampling is highly sensitive to the scale and orientation of the proposal. The Adaptive Metropolis (AM) algorithm addresses this by allowing the proposal distribution to evolve over time based on the spread of the previously accepted samples.

Let $x_1, \ldots, x_t \in \mathbb{R}^d$ be the history of the chain up to time t. The AM algorithm uses a Gaussian proposal of the form

$$x' \sim \mathcal{N}\left(x_t, s_t^2 \Sigma_t\right),$$

where Σ_t is the empirical covariance matrix of the chain's history:

$$\Sigma_t = \mathrm{Cov}(x_1, \ldots, x_t) + \epsilon I,$$

with a small constant $\epsilon > 0$ to ensure positive definiteness. The scaling parameter s_t can be tuned or held constant.

Although this adaptation breaks the Markov property, it has been shown under some conditions that the AM algorithm retains ergodicity and converges to the correct stationary distribution. This method is especially effective in moderate to high dimensions where simple proposals fail to capture the geometry of the target distribution.

5.2 Metropolis-Adjusted Langevin Algorithm (MALA)

The Metropolis-Adjusted Langevin Algorithm (MALA) improves upon random walk proposals by using gradient information from the target distribution to guide the proposal toward regions of higher probability. The proposal is based on a discretization of the Langevin diffusion process:

$$dX_t = \frac{1}{2}\nabla \log \pi(X_t)\, dt + dW_t,$$

where W_t is a standard Brownian motion. The corresponding Euler-Maruyama discretization yields the proposal:

$$x' = x + \frac{\delta^2}{2}\nabla \log \pi(x) + \delta \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I).$$

This proposal is then accepted or rejected using the standard Metropolis-Hastings rule, where the proposal density is no longer symmetric and must be accounted for in the acceptance ratio:

$$\alpha(x, x') = \min\left(1, \frac{\pi(x')q(x' \to x)}{\pi(x)q(x \to x')}\right).$$

MALA significantly improves mixing in smooth, log-concave targets and has been shown to achieve optimal scaling in high-dimensional Gaussian targets when the step size δ is chosen appropriately (typically $\delta \propto d^{-1/6}$).

5.3 Hamiltonian Monte Carlo (HMC)

Hamiltonian Monte Carlo is one of the most powerful extensions of the MH framework. It introduces an physics inspired variable $p \in \mathbb{R}^d$ and simulates Hamiltonian dynamics to generate distant proposals that lie along high-probability trajectories.

Let the target distribution be $\pi(x)$, and define the potential energy as $U(x) = -\log \pi(x)$. Introduce a kinetic energy $K(p) = \frac{1}{2}p^T M^{-1} p$, typically with $p \sim \mathcal{N}(0, M)$ for some mass matrix M. The Hamiltonian is:

$$H(x, p) = U(x) + K(p).$$

To propose a new state, simulate Hamiltonian dynamics using the equations:

$$\frac{dx}{dt} = \frac{\partial H}{\partial p} = M^{-1}p,$$
$$\frac{dp}{dt} = -\frac{\partial H}{\partial x} = -\nabla U(x),$$

via a symplectic integrator over a fixed number of steps L with step size ϵ. The resulting proposal (x', p') is accepted with probability

$$\alpha = \min\left(1, e^{-H(x',p')+H(x,p)}\right),$$

ensuring detailed balance.

HMC proposals follow long, deterministic trajectories that maintain high acceptance rates even in high dimensions. The method avoids the random walk behavior of standard MH and dramatically reduces autocorrelation in the samples. However, it requires gradient evaluations and careful tuning of the integrator parameters (ϵ, L) and mass matrix M.

6 Limitations of Metropolis-Hastings

Despite its generality and theoretical convergence guarantees, the Metropolis-Hastings algorithm exhibits several fundamental limitations that hinder its practical performance in some highly difficult modern applications. These limitations arise primarily from its local proposal mechanism, its inefficiency in high dimensions, and its sensitivity to the geometry of the target distribution. However the variations of the MH algorithm does account for these difficulties.

One of the most well-documented weaknesses of the original MH algorithm is its poor scaling behavior in high-dimensional settings. Consider a d-dimensional isotropic Gaussian target distribution:

$$\pi(x) \propto \exp\left(-\frac{1}{2}\|x\|^2\right), \quad x \in \mathbb{R}^d,$$

and suppose we use a Gaussian random walk proposal of the form:

$$x' = x + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I).$$

To maintain a nontrivial acceptance rate as $d \to \infty$, the proposal variance σ^2 must scale inversely with dimension:

$$\sigma^2 \propto \frac{1}{d}.$$

This result arises from the fact that the acceptance probability depends on the ratio:

$$\alpha(x, x') = \min\left(1, \exp\left(-\frac{1}{2}(\|x'\|^2 - \|x\|^2)\right)\right).$$

In high dimensions, $\|x\|^2 \approx d$ concentrates sharply due to the law of large numbers, and small changes in x lead to exponentially small changes in the target density. As a result, even modest proposals ϵ lead to significant drops in probability mass, causing the acceptance rate to collapse unless ϵ is extremely small. Consequently, the chain takes tiny steps and mixes slowly, requiring an increasingly large number of steps to explore the space.

Moreover, Metropolis-Hastings doesnt' do too well when the target distribution is multimodal. Consider a distribution with two well-separated modes. The probability of transitioning between them using a local proposal is exponentially small in the distance between the modes. If the chain starts in one mode, it may take thousands or millions of iterations before it escapes, effectively leading to biased samples. For example, for two modes centered at $x = \pm a$, and a local proposal of scale σ, the probability of proposing a point near the second mode is approximately:

$$\mathbb{P}(|x' \mp a| < \delta) \approx \exp\left(-\frac{(2a)^2}{2\sigma^2}\right),$$

which decays rapidly as a increases or σ decreases.

In addition, the basic MH algorithm does not leverage the geometry of the target. When the probability mass of π is concentrated along a narrow, curved manifold, simple proposals perform poorly, as most proposals fall outside the support of the distribution. This results in high rejection rates and extremely slow mixing. Without gradient or curvature information, MH cannot align itself with the geometry of π, leading to inefficient exploration.

Finally, the performance of MH is highly sensitive to the tuning of the proposal distribution. A step size that works well in one region of the space may be suboptimal in another. In adaptive or heterogeneous distributions, a fixed proposal kernel is naturally not the best.

In summary, although Metropolis-Hastings remains a foundational tool in computational statistics, it is inherently limited by its local, geometry-agnostic nature. Its effectiveness deteriorates rapidly in high dimensions, near sharp modes, or when the target distribution exhibits complex structure. These weaknesses motivate the development of adaptive, gradient-based, and Hamiltonian extensions that we have explored in the previous section.

7 Frontiers and Open Questions

Although the Metropolis-Hastings algorithm is well-established, it continues to be a subject of active research, particularly in high-dimensional inference, computational geometry, and machine learning. One major frontier lies in understanding and improving the *mixing time* of MH algorithms in complex geometries. Precise bounds on convergence rates remain difficult to obtain, especially for adaptive or non-reversible variants.

Another area of exploration is the design of *geometry-aware* proposals that use curvature and manifold structure, including Riemannian versions of MALA

and HMC. These methods are designed to sample efficiently from complex posteriors that have uneven scaling or tight constraints, like those found in Bayesian neural networks and hierarchical models.

There is also growing interest in *non-reversible MCMC*, which breaks detailed balance to accelerate convergence. Algorithms that use momentum, bounces, or piecewise-deterministic dynamics can improve asymptotic efficiency and have shown promise in both theory and practice.

8 Conclusion

The Metropolis-Hastings algorithm provides a powerful and general framework for sampling from complex distributions. Its ability to handle unnormalized targets and flexible proposal mechanisms makes it a cornerstone of modern computational statistics. However, as this paper has shown, its performance is highly sensitive to proposal design and suffers in high-dimensional or multimodal settings. Continued research into adaptive, gradient-based, and non-reversible variants promises to further enhance the reach and efficiency of MCMC in solving contemporary probabilistic problems.