Dynamics-based MCMC methods with Riemannian Manifolds

Jeegn Dani
jdani@purdue.edu

Abstract

Markov Chain Monte Carlo (MCMC) sampling methods are crucial for Bayesian Inference when obtaining the analytical closed form of a posterior is infeasible. They generate samples using the unnormalized posterior density and have asymptotical guarantees to converge to the posterior. In this paper, we survey popular dynamics-based MCMC methods like Metropolis adjusted Langevin Algorithm (MALA) and Hamilton Monte Carlo (HMC). We go through their derivation and discuss how they satisfy detailed balance and their asymptotical guarantees. These dynamics-based methods are not efficiently able to sample from high-dimensional target densities with strong correlations. These shortcomings were addressed by developing a general geometric framework for these methods based on Riemannian Manifolds. This framework provided a fully automated mechanism called Riemannian Manifold MCMC. After a brief overview of manifolds and how they relate to MCMC methods, we finally describe these adapted methods and provide a visual intuition of their advantages.

1 Introduction

Statistical Inference estimates a set of unknown parameters, given some observed data that may be contaminated by noise and possibly by other types of distortion and interferences. These estimates are typically obtained by solving multivariate optimization problems like maximum likelihood estimation (MLE) or maximum a posteriori (MAP) or utilizing a Bayesian framework. In the Bayesian framework, a prior distribution over the unknown parameters is formulated which gathers all the available information external to the observed data and models the relationship between observed data and unknown parameters using a probability distribution (likelihood). Thereafter using the Bayes theorem to obtain the posterior $p(\theta)$, which takes into account both the effect of prior and observed data in an optimal way. We obtain $p(\theta) = \frac{\tilde{p}(\theta)}{\int \tilde{p}(\theta) d\theta}, \theta \in \mathbb{R}^D$, wherein $\tilde{p}(\theta)$ is the unnormalized probability density function consisting of the product of the likelihood and the prior. For many statistical models, achieving the closed form of this posterior is intractable due to the multi-dimensional integration $\int \tilde{p}(\theta) d\theta$.

We can avoid those complicated integrals by simply relying on the unnormalized posterior probabilities $\tilde{p}(\theta)$ to get samples from the unknown posterior distribution. This is known as Bayesian Posterior Sampling. Monte Carlo Markov Chain (MCMC) is a feasible computational approach that can be used to generate samples from a desired invariant target stationary distribution (the posterior) by drawing samples from a proposal density, building then an ergodic Markov Chain whose stationary distribution is the desired distribution by accepting or rejecting those candidate samples as the new state of the chain. The most general algorithm defining a Markov Process with an invariant density $p(\theta)$ is the Metropolis-Hastings algorithm [20, 12], which is arguably one of the most successful and influential algorithms from the 20th century.

Preprint. Under review.

Algorithm 1 Metropolis-Hastings algorithm

Require: $p(\theta)$: probability density function to be sampled
Require: $q(\theta^* \mid \theta)$: proposal distribution
Require: N: number of iterations
Require: θ_0: initial state
  Set $\theta \leftarrow \theta_0$
  for $i \leftarrow 1$ to N **do**
    Generate θ^* from the proposal distribution: $\theta^* \sim q(\cdot \mid \theta)$
    Calculate acceptance ratio: $\alpha = \min\left\{1, \frac{\tilde{p}(\theta^*)q(\theta|\theta^*)}{\tilde{p}(\theta)q(\theta^*|\theta)}\right\}$
    Generate u from the uniform distribution on $[0, 1]$

    if $u < \alpha$ **then**
      Set $\theta \leftarrow \theta^*$
      Record θ
    end if
  end for

The Metropolis-Hastings algorithm 1proposes transitions from θ to θ^* with density $q(\theta^*|\theta)$, which are accepted with probability $min\{1, \frac{\tilde{p}(\theta^*)q(\theta|\theta^*)}{\tilde{p}(\theta)q(\theta^*|\theta)}\}$. This acceptance probability ensures that the Markov Chain is reversible with respect to the stationary target density $p(\theta)$ and satisfies detailed balance [25, 18]. The proposal mechanism $q(\theta^*|\theta)$ governs the behavior of the Markov Chain. Typically, simple choices for the proposal distribution are $\mathcal{N}(\theta, \lambda^2\Sigma)$, where Σ is often chosen in an attempt to match the correlation structure of $p(\theta)$, or some other symmetric distribution. In these scenarios, the Markov Chain takes the form of a random walk.

A critical issue for the good performance of the MH algorithm is the acceptance rate (AR), which depends on the variance of the proposal PDF $q(\theta^*|\theta)$ and should be neither too high nor too low. Our goal is to form a well-mixed chain that discovers the entire structure of $p(\theta)$ in a short time. On the one hand, a high variance typically leads to a low AR, thus implying that the MH algorithm gets stuck because most candidate samples are rejected. On the other hand, a low variance can easily lead to a high AR, as only local moves around previously accepted samples are proposed, but this can result in the MH algorithm failing to explore the target. In higher dimensional parameter space, the random walk proposal becomes inefficient, resulting in low rates of acceptance, poor mixing of the chain, and highly correlated samples as the proposal mechanism might not be able to visit a distant region of high density. As a consequence, our effective sample size (ESS) from the chain will be small [25, 18]. It has been shown that the optimal acceptance rate for a random walk proposal tends to be 0.234 as the dimension of parameter space tends to ∞ ($D_\theta \to \infty$) for a wide range of target distributions [30]. Theoretical research also shows if we employ the typical $\mathcal{N}(\theta, \lambda^2\Sigma)$ as the proposal distribution than optimal tuning of λ should be $\lambda^2 \propto D_\theta^{-1}$ [28]. Hence, we can say that algorithm converges in $\mathcal{O}(D_\theta)$.

Major research was done to design a good general-purpose proposal mechanism providing large proposal transitions that are accepted with high probability. Inspired by the Langevin equation that describes the motion of a particle under the influence of friction and random forces, the Metropolis Adjusted Langevin Algorithm (MALA) [26] was suggested. Likewise Hamiltonian Monte Carlo (HMC) [22] was proposed in statistical physics literature as a method of simulating the evolution of a dynamical system. It was based on a set of first-order, nonlinear, partial differential equations, known as Hamiltonian Equations. Later [22] applied it to the problem of statistical inference in the context of Monte Carlo Sampling. These methods will be referred to Dynamics-Based MCMCs.

Despite the potential gains obtained in Dynamics-Based MCMC sampling, the tuning of these MCMC methods remains a major issue, especially for challenging inference problems. To counter this issue Information Geometry [1]concepts in Bayesian Inference and MCMC methods were exploited by [11] to formulate a geometric framework for developing the Dynamics-Based MCMCs called Riemannian Manifold MCMCs.

This paper covers the details of Dynamics-Based MCMCs in section 2, then studies manifolds and the differential geometric concepts employed for Riemannian Manifolds MCMCs in section 3 and finally covers how to adapt Dynamics-Based MCMCs on Riemannian Manifolds in section 4.

2 Dynamics-Based MCMCs

In this section, we consider Dynamics-Based MCMCs, also known as gradient-based MCMCs as they use the gradient of the log-posterior $\nabla \log p(\theta)$, to improve the efficiency of the sampling procedures. The intuition is that, by using the gradient, we can form proposal distributions that allow for longer jumps without pushing the acceptance ratio of the method too low.

2.1 Metropolis Adjusted Langevin Algorithm (MALA)

MALA is based on a Langevin Diffusion, with stationary distribution $p(\theta)$, defined by the stochastic differential equation

$$d\boldsymbol{\theta}(\tau) = \mathbf{f}(\boldsymbol{\theta}(\tau))\, d\tau + d\mathbf{b}(\tau), \tag{1}$$

where $\mathbf{b}(\tau)$ is a D_θ-dimensional Brownian motion.

The Fokker–Planck equation is a partial differential equation that describes the time evolution of the probability density function of a stochastic process. In particular, it provides a way to compute the probability density function of a stochastic process from the knowledge of its stochastic differential equation. So given the eqn(1), the Fokker-Plankc equation giving the probability density $p(\theta, \tau)$ of the diffusion state is

$$\frac{\partial p(\boldsymbol{\theta}, \tau)}{\partial \tau} = -\nabla \cdot [\mathbf{f}(\boldsymbol{\theta})\, p(\boldsymbol{\theta}, \tau)] + \frac{1}{2}\nabla^2 p(\boldsymbol{\theta}, \tau) \tag{2}$$

$\mathbf{f}(\boldsymbol{\theta})$ guides the diffusion process and is called the drift of the diffusion. We select the drift to be the gradient of the log posterior, so our proposals are guided toward high-density areas. Thereby, setting $\mathbf{f}(\boldsymbol{\theta}) = \frac{1}{2}\nabla \log p(\boldsymbol{\theta})$ results in the stationary solution of $\frac{\partial p(\boldsymbol{\theta},\tau)}{\partial \tau} = 0$ to be $p(\boldsymbol{\theta}, \tau) = \tilde{p}(\boldsymbol{\theta})$, which is the unnormalized posterior obtained by the product of likelihood and prior.

By starting at $\boldsymbol{\theta}^{(0)} \sim \tilde{p}(\boldsymbol{\theta})$ and solving the SDE eqn(1) for $\tau > 0$ we can generate more samples from $\tilde{p}(\boldsymbol{\theta})$, because the marginal distribution of the SDE solution $\boldsymbol{\theta}^{(t)}$ is $\tilde{p}(\theta)$ for all $t \geq 0$.

MALA 2 uses the SDE equation and the drift term as described above as the proposal distribution in an MH algorithm. Unfortunately, we cannot solve or simulate the SDE exactly. Hence, we typically approximate its solution using the Euler–Maruyama method [15] to get:

$$\boldsymbol{\theta}^{(\tau_{n+1})} \approx \boldsymbol{\theta}^{(\tau_n)} + \frac{\Delta\tau}{2}\nabla \log p(\boldsymbol{\theta}^{(\tau_n)}) + \sqrt{\Delta\tau}\, \mathbf{z}_n \tag{3}$$

where $\mathbf{z}_n \sim \boldsymbol{\mathcal{N}}(0, \mathbf{I})$ and $\Delta\tau = \tau_{n+1} - \tau_n$, also called the discretization step size. This results in $q(\boldsymbol{\theta}^*|\boldsymbol{\theta})$ being a Gaussian with mean $\boldsymbol{\theta} + \frac{\Delta\tau}{2}\nabla \log p(\boldsymbol{\theta})$ and variance $\Delta\tau\mathbf{I}$.

It has been concluded that the optimal asymptotic acceptance rate (when $D_\theta \to \infty$) is approximately 0.574 [27], compared to 0.234 of Random Walk Metropolis-Hastings. This increased acceptance rate is due to the incorporation of additional information about the target density into the sampling procedure through the use of SDE and particularly using log posterior as the drift term. Furthermore, [27] also shows that the proposal variance should scale to $D_\theta^{-1/3}$, and thus the algorithm's convergence is $\boldsymbol{\mathcal{O}}(D_\theta^{1/3})$.

Intuitively, MALA proposals comprise a deterministic shift towards a *local* mode of $p(\theta)$ combined with some random additive Gaussian noise, with a uniform variance $\Delta\tau$ for each component. The uniform variance across components makes the diffusion isotropic (equal in all directions) which will be inefficient for strongly correlated variables $\boldsymbol{\theta}$ with vastly differing variances forcing the step size to only accommodate the component with the smallest variance. This issue can be circumvented by employing a preconditioning matrix [29] $\mathbf{M}$ such that,

$$\boldsymbol{\theta}^{(\tau_{n+1})} \approx \boldsymbol{\theta}^{(\tau_n)} + \frac{\Delta\tau}{2}\mathbf{M}\nabla \log p(\boldsymbol{\theta}^{(\tau_n)}) + \sqrt{\Delta\tau\mathbf{M}}\, \mathbf{z}_n \tag{4}$$

This preconditioning matrix allows the sampling procedure to adapt to the local structure of parameter space better and leads to faster convergence. However, it was unclear how to define $\mathbf{M}$ in any

Algorithm 2 Metropolis adjusted Langevin algorithm (MALA)

Initialization: Choose an initial state $\boldsymbol{\theta}^{(\mathbf{0})}$, the discretization step $\Delta\tau$ and total iterations T
for $t \leftarrow 1$ to T **do**
  $\mathbf{z} \sim \boldsymbol{\mathcal{N}}(0, \mathbf{I})$
  $\mathbf{u} \sim \boldsymbol{\mathcal{U}}(0, \mathbf{I})$

  Generate $\boldsymbol{\theta}^*$ from the discretized SDE eqn(1)
  $\boldsymbol{\theta}^* \leftarrow \boldsymbol{\theta}^{(t-1)} + \frac{\Delta\tau}{2}\nabla \log p(\boldsymbol{\theta}^{(t-1)}) + \sqrt{\Delta\tau}\,\mathbf{z}_n$

  Calculate acceptance ratio:
  $\alpha = \min\left\{1, \frac{\tilde{p}(\boldsymbol{\theta}^*)\mathcal{N}(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}^* + \frac{\Delta\tau}{2}\nabla \log p(\boldsymbol{\theta}^*), \Delta\tau\mathbf{I})}{\tilde{p}(\boldsymbol{\theta}^{(t-1)})\mathcal{N}(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(t-1)} + \frac{\Delta\tau}{2}\nabla \log p(\boldsymbol{\theta}^{(t-1)}), \Delta\tau\mathbf{I})}\right\}$

  if $u < \alpha$ **then**
    Set $\boldsymbol{\theta}^t \leftarrow \boldsymbol{\theta}^*$
    Record $\boldsymbol{\theta}^t$
  end if
end for

systematic and principled manner until Mark Girolami and Ben Calderhead exploited geometric principles to define this preconditioning matrix, which we will see in Section (4)

2.2 Hamilton Dynamics

The Hamiltonian Monte Carlo (HMC) is grounded in a statistical physical simulation of a physical system to form the proposal distribution. Imagine a particle of unit mass in a potential field $f : \mathbb{R}^D \rightarrow \mathbb{R}$. If the particle has position $x \in \mathbb{R}^D$ and velocity $v \in \mathbb{R}^D$, its total energy is given by the function $H : \mathbb{R}^{2D} \rightarrow \mathbb{R}$ defined as [22]

$$H(x, v) = f(x) + \frac{1}{2}||v||^2 \tag{5}$$

This function is called the Hamiltonian of the particle. In the MCMC context, the x is $\boldsymbol{\theta}$ and as we did in MALA, $f(\boldsymbol{\theta}) = -\log p(\boldsymbol{\theta})$ the negative log posterior. For the v we introduce an auxiliary variable $\boldsymbol{\rho} \in \mathbb{R}^D$ with density $p(\boldsymbol{\rho}) = \mathcal{N}(\boldsymbol{\rho}|0, \mathbf{M})$. Physically, $\boldsymbol{\rho}$ is interpreted as the momentum variable, and the $\mathbf{M}$ is the covariance matrix or the Mass matrix. This gives us the following Hamiltonian

$$H(\boldsymbol{\theta}, \boldsymbol{\rho}) = -\log p(\boldsymbol{\theta}) + \frac{1}{2}\log(((2\pi)^D|\mathbf{M}|)) + \frac{1}{2}\boldsymbol{\rho}^T\mathbf{M}^{-1}\boldsymbol{\rho} \tag{6}$$

Given our choice of $f(\boldsymbol{\theta})$, $p(\boldsymbol{\theta}) \propto \exp(-f(\boldsymbol{\theta}))$ and distribution of the particles $p(\boldsymbol{\theta}, \boldsymbol{\rho})$ is given by

$$\begin{aligned} p(\boldsymbol{\theta}, \boldsymbol{\rho}) &= p(\boldsymbol{\theta})p(\boldsymbol{\rho}) \\ &= p(\boldsymbol{\theta})\mathcal{N}(\boldsymbol{\rho}|0, \mathbf{M}) \\ &= \frac{1}{Z}\exp(-f(\boldsymbol{\theta}))\exp(-\frac{1}{2}\log(((2\pi)^D|\mathbf{M}|)) - \frac{1}{2}\boldsymbol{\rho}^T\boldsymbol{\rho})) \\ &= \frac{1}{Z}\exp(-H(\boldsymbol{\theta}, \boldsymbol{\rho})) \end{aligned} \tag{7}$$

The Hamiltonian equations for the dynamics of the particle in fictitious time τ are given by

$$\frac{d\boldsymbol{\theta}}{d\tau} = \nabla_{\boldsymbol{\rho}}H = \mathbf{M}^{-1}\boldsymbol{\rho}, \quad \frac{d\boldsymbol{\rho}}{d\tau} = -\nabla_{\boldsymbol{\theta}}H = \nabla \log p(\boldsymbol{\theta}). \tag{8}$$

The HMC algorithm constructs the proposal distribution by simulating trajectories from the above Hamiltonian Equations. Given starting configurations $\boldsymbol{\theta}_0$ and $\boldsymbol{\rho}_0$, these trajectories can be simulated by solving eqns(8) and getting the "Hamiltonian Flow" (ϕ), $(\boldsymbol{\theta}_\tau, \boldsymbol{\rho}_\tau) = \phi((\boldsymbol{\theta}_0, \boldsymbol{\rho}_0))$. We note that these "flows" should follow the following properties to form an ergodic Markov Chain whose stationary marginal density is $p(\boldsymbol{\theta})$:

1. Preservation of total energy, i.e. $H(\boldsymbol{\theta}_\tau, \boldsymbol{\rho}_\tau) = H(\boldsymbol{\theta}_0, \boldsymbol{\rho}_0)$
2. Preservation of volume element, i.e. $d\boldsymbol{\theta}_\tau d\boldsymbol{\rho}_\tau = d\boldsymbol{\theta}_0 d\boldsymbol{\rho}_0$

3. Time Reversible, $\phi(\boldsymbol{\theta}_\tau, -\boldsymbol{\rho}_\tau) = (\boldsymbol{\theta}_0, \boldsymbol{\rho}_0)$

However for practical applications of interest the differential eqns(8) cannot be solved analytically and numerical methods to simulate trajectories are required. According to [16] the Leapfrog integrator will fully satisfy preservation of volume and time reversibility, and approximately satisfy energy conservation. The Leapfrog integrator was initially applied by [10] and thereafter in various statistical applications like [18] and [22]. One step of the Leapfrog method for the Hamiltonian equations starting from τ with step size $\Delta\tau$ is given as

$$\widetilde{\boldsymbol{\rho}}^{(\tau+\Delta\tau/2)} = \widetilde{\boldsymbol{\rho}}^{(\tau)} + \frac{\Delta\tau}{2}\nabla \log p\left(\widetilde{\boldsymbol{\theta}}^{(\tau)}\right) \tag{9}$$

$$\widetilde{\boldsymbol{\theta}}^{(\tau+\Delta\tau)} = \widetilde{\boldsymbol{\theta}}^{(\tau)} + \Delta\tau \mathbf{M}^{-1}\widetilde{\boldsymbol{\rho}}^{(\tau+\Delta\tau/2)} \tag{10}$$

$$\widetilde{\boldsymbol{\rho}}^{(\tau+\Delta\tau)} = \widetilde{\boldsymbol{\rho}}^{(\tau+\Delta\tau/2)} + \frac{\Delta\tau}{2}\nabla \log p\left(\widetilde{\boldsymbol{\theta}}^{(\tau+\Delta\tau))}\right) \tag{11}$$

As the total energy is only approximately conserved with the Leapfrog integrator, a bias is introduced in the corresponding joint density $p(\boldsymbol{\theta}, \boldsymbol{\rho})$ which can be corrected by an MH accept-reject step as shown in algorithm 3.

Algorithm 3 Hamilton Monte Carlo (HMC) Algorithm

Initialization: Choose an initial state $\boldsymbol{\theta}^{(\mathbf{0})}$, the discretization step $\Delta\tau$, the number of integration steps L and total number of iterations T
for $t \leftarrow 1$ to T **do**
$\mathbf{u} \sim \boldsymbol{\mathcal{U}}(0, \mathbf{M})$

Generate particle trajectories from the Hamiltonian Dynamics Equations 8,
using L steps of Leapfrog method starting from
$\widetilde{\boldsymbol{\theta}}^{(0)} = \boldsymbol{\theta}^{(t-1)}$ and $\widetilde{\boldsymbol{\rho}}^{(0)} \sim \mathcal{N}(0, \mathbf{I})$, setting
$\boldsymbol{\theta}^* = \widetilde{\boldsymbol{\theta}}^{(L\Delta\tau)}$ and $\boldsymbol{\rho}^* = -\widetilde{\boldsymbol{\rho}}^{(L\Delta\tau)}$

Calculate acceptance ratio:
$\alpha_t = \alpha(\boldsymbol{\theta}^*, \boldsymbol{\rho}^*; \boldsymbol{\theta}^{(t-1)}, \boldsymbol{\rho}^{()t-1}) = \min\left\{1, \exp(-H(\boldsymbol{\theta}^*, \boldsymbol{\rho}^*) + H(\boldsymbol{\theta}^{(t-1)}, \boldsymbol{\rho}^{(t-1)}))\right\}$

if $u < \alpha_t$ **then**
Set $\boldsymbol{\theta}^t \leftarrow \boldsymbol{\theta}^*$
Record $\boldsymbol{\theta}^t$
end if
end for

HMC allows for an improved exploration of the state space than Random Walk Metropolis-Hastings 1 and MALA 2, especially in higher dimensions as the proposal mechanism generates a trajectory of new states, leading to quicker exploration of distant local modes. The optimal scaling of the step size $\Delta\tau$ has been analyzed in [3] and they prove that it requires $\mathcal{O}(D_\theta^{1/4})$ steps to traverse the state space and the asymptotically optimal acceptance rate for HMC is 0.651, highest of all algorithms discussed.

Table 1: Acceptance Rates and Convergence Rates for the Random Walk MH, MALA and HMC. We can clearly see that Dynamics-Based MCMCs are much better than Random Walk MH. And HMC is the best out of the three.

	RWMH	MALA	HMC
Convergence Rate	$\mathcal{O}(D_\theta)$	$\mathcal{O}(D_\theta^{1/3})$	$\mathcal{O}(D_\theta^{1/4})$
Acceptance Rate	0.234	0.5474	0.651

By combining the eqns (9) & (10) we get the $\boldsymbol{\theta}$'s single step update of the form

$$\widetilde{\boldsymbol{\theta}}^{(\tau+\Delta\tau)} = \widetilde{\boldsymbol{\theta}}^{(\tau)} + \Delta\tau \mathbf{M}^{-1}\widetilde{\boldsymbol{\rho}}^{(\tau)} + \frac{\Delta\tau}{2}\mathbf{M}^{-1}\nabla \log p\left(\widetilde{\boldsymbol{\theta}}^{(\tau)}\right) \tag{12}$$

We notice that this is similar to Euler discretized Preconditioned Langevin Diffusion Proposal eqn(4). Therefore we can also notice that the mass matrix is similar to the preconditioning matrix discussed above (2.1). It determines the scaling of the momentum, and therefore affects the magnitude and direction of the momentum updates, ultimately having a significant impact on the performance of HMC. A poorly chosen mass matrix can result in inefficient exploration of the state space, leading to slow convergence and poor mixing of the Markov chain. On the other hand, a well-chosen mass matrix can lead to more efficient sampling by adapting to the geometry of the target distribution.

3 MCMC and Manifolds

To fully utilize the effectiveness of both MALA and HMC approaches, it is beneficial to use transitions that consider the local structure of the target distribution when proposing transitions to different probability regions. This can enhance the mixing of the Markov chain and improve its performance. Additionally, rather than using a constant covariance matrix $\mathbf{M}$, it would be advantageous to adopt a covariance matrix that is tailored to the position being evaluated. Girolami and Calderhead [11] addressed these enclaves and proposed Riemann Manifold MCMC methods. We will first discuss how we can use manifolds with MCMC and then give a brief preliminary on Riemannian Manifolds. Thereafter we will elaborate on Riemann Manifold MCMCs in Section 4

3.1 Manifolds and Markov Chains

The methods discussed above make several assumptions about the state space $\boldsymbol{\Theta}$ in which our Markov chains evolve. They all assume that it is a Euclidean Space $\boldsymbol{\Theta} = \mathbb{R}^D$ with the induced distance metric:

$$d(x, y) = \sqrt{\sum_i (x_i - y_i)^2} = \sqrt{\langle x - y, x - y \rangle} \tag{13}$$

For MCMC methods that explore this parameter space *locally* it may be advantageous to have a local metric structure on $\boldsymbol{\Theta}$ as discussed above. In simpler terms, one can imagine that distances in $\boldsymbol{\Theta}$ are defined in a way that if the current position $\boldsymbol{\theta}$ in the Markov chain is far away from a region of $\boldsymbol{\Theta}$ that is expected to have a high probability under the target distribution, then the distance to that typical set can be shortened. Once the chain reaches this region, the space can be expanded or modified so that it can be explored more effectively. This approach allows the Markov chain to move more efficiently through the space and explore the target distribution more thoroughly.

3.2 Manifolds - a brief overview

A manifold is a mathematical object that can be *locally* approximated by Euclidean space. In other words, it is a topological space that is locally homeomorphic to Euclidean space. This means that each point on the manifold has a neighborhood that looks like a piece of Euclidean space, but the global structure of the manifold may be more complicated. Formally, a manifold $\mathcal{M}$ of dimension

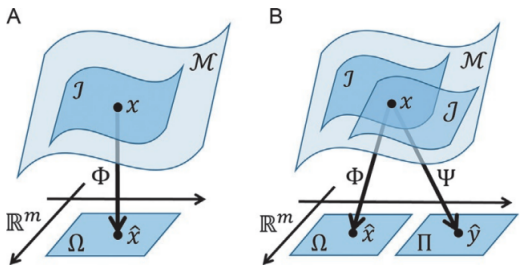


Figure 1: Illustration of the concepts of manifold and coordinate system. (A) Concept of manifold and local coordinate system. (B) Multiple charts defining a Manifold.[17]

m is a Hausdorff topological space that is second-countable and locally Euclidean of dimension m. This means that every point on $\mathcal{M}$ has a neighborhood $\mathcal{J}$ that is homeomorphic to an open subset $\boldsymbol{\Omega} \subseteq \mathbb{R}^m$. Meaning that there is a continuous bijection $\boldsymbol{\Phi} : \mathcal{J} \to \boldsymbol{\Omega}$ whose inverse is also continuous.

$\mathbf{\Phi}$ is called a chart of the manifold such that $\hat{x} = \mathbf{\Phi}(x) \in \mathbb{R}^m$ where $(\mathcal{J}, \mathbf{\Phi})$ is the local coordinate system, and $\mathbf{\Omega}$ the coordinate space (See Figure (1.A). A function $f : \mathcal{M} \to \mathbb{R}$ on the manifold can also be concretized as a m-dimensional multivariate function $f \circ \mathbf{\Phi}^{-1} : \mathbb{R}^m \to \mathbb{R}$ which holds the same continuity as f.

The manifold can be further characterized by a set of charts $(\mathbf{\Phi}, \mathbf{\Psi})$ (See Figure 1.B). However, we only consider smooth/differentiable manifolds for which a global coordinate chart exists, meaning that the same mapping $\mathbf{\Phi}$ exists for all points on the manifold and it is differentiable and invertible and whose inverse $\mathbf{\Phi}^{-1}$ is also differentiable.

Our Dynamics-based MCMC techniques require taking the gradient of the log posterior $f(\theta) = \log p(\boldsymbol{\theta})$, we need to ensure that our likelihood function should be differentiable on the smooth manifold. The differentiability of a function f on a smooth manifold can be consistently determined by the differentiability of $f \circ \mathbf{\Phi}^{-1}$ and since f and $\mathbf{\Phi}^{-1}$ are both differentiable, f is differentiable on the manifold. As stated in eqn(13), for a Euclidean manifold the distance metric is induced by a

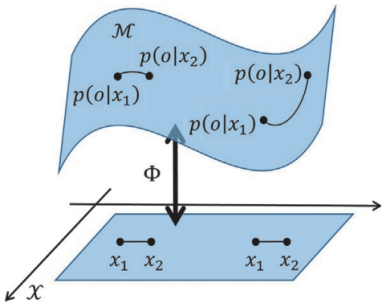


Figure 2: [17]

Euclidean inner product. Similarly, we need a metric to calculate the distance between two points on a non-Euclidean manifold. To aid intuition take a look at Figure (2) and think about the distance between x_1 and x_2 as the curve γ between $p(o|x_1)$ and $p(o|x_2)$. For simplicity let, $\gamma : [0, 1] \to \mathbb{R}^m$, where $\gamma(0) = x_1$ and $\gamma(1) = x_2$. Then the length of the curve is defined as

$$L(\gamma) = \int_0^1 \sqrt{\langle \gamma'(t), \gamma'(t) \rangle} dt \tag{14}$$

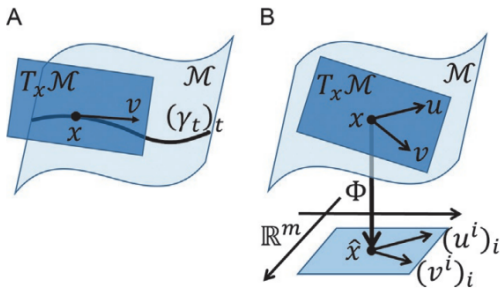


Figure 3: Illustration of tangent vector, tangent space on a manifold M. (A) Tangent vector on a curve. (B) Tangent vector and tangent space on a manifold.[17]

Now lets generalize this and define a curve on a manifold M as $\gamma_{\mathcal{M}} : [0, 1] \to \mathcal{M}$. At each point $\gamma_{\mathcal{M}}(t) = p \in \mathcal{M}$ the derivative of the curve $\gamma'_{\mathcal{M}}(t)$ lies in a m-dimensional vector space which touches $\mathcal{M}$ at p. These are known as tangent spaces $T_P\mathcal{M}$ which can be thought of as a vector space that approximates the manifold at the point p. It consists of all possible tangent vectors to the manifold at p (See Figure (3)). In a vector space, metric properties can always be induced through an inner product. Therefore we can use the tangent space to define the length of a curve by defining a

metric $g_p : T_P\mathcal{M} \to \mathbb{R}$ and generalizing eqn(14) to

$$L(\gamma_{\mathcal{M}}) = \int_0^1 \sqrt{g_p\langle\gamma'_{\mathcal{M}}(t), \gamma'_{\mathcal{M}}(t)\rangle}dt \tag{15}$$

This metric allows us to measure distances between points on the manifold, even though the manifold may not have a global inner product. We can see that Euclidean Space is a manifold where the metric is an Identity Matrix.

For the purpose of using manifolds in MCMC sampling we are concerned with Riemannian Manifolds [1]. Riemannian manifolds are real, smooth manifolds with a Riemannian metric eqn(16). The Riemannian metric (tensor) is a family of inner products

$$g_p : T_P\mathcal{M} \times T_P\mathcal{M} \to \mathbb{R}, p \in \mathcal{M} \tag{16}$$

such that $p \to g_p(X(p), Y(p))$ for any two tangent vectors $X(p), Y(p)$ is a smooth function of p. Note that this is a family of metric tensors, that is, we have a different tensor for every point on the manifold. The implication of this is that even though each adjacent tangent space can be different (the manifold curves therefore the tangent space changes), the inner product varies smoothly between adjacent points. Intuitively, Riemannian manifolds have all the nice "smoothness" properties we would want for Dynamics-Based MCMC sampling [13].

4 MCMCs on Riemannian Manifolds

The relationship between Riemann geometry and statistics had been employed in the development of, primarily asymptotic, statistical theory [21, 2]. Girolami and Calderhead [11] exploited geometric concepts of distance, curvature, and manifolds to develop Riemann manifold Metropolis adjusted Langevin algorithm (RMMALA) and Riemann manifold Hamilton Monte Carlo (RMHMC).

Rao [24] defined the distance between two parameterized density functions $p(y; \boldsymbol{\theta})$ and $p(y; \boldsymbol{\theta} + \delta\boldsymbol{\theta})$ to be $\delta\boldsymbol{\theta}^T\mathbf{G}(\boldsymbol{\theta})\delta\boldsymbol{\theta}$ where $\mathbf{G}(\boldsymbol{\theta})$ was shown to be equal to

$$\mathbf{G}(\boldsymbol{\theta}) = -\mathbb{E}_{y|\theta}\left[\frac{\partial^2}{\partial\theta^2}\log p(y|\theta)\right] = \text{cov}\left[\frac{\partial}{\partial\theta}\log p(y|\theta)\right] \tag{17}$$

the expected Fisher information matrix. Rao also noted that $\mathbf{G}(\boldsymbol{\theta})$ is a positive definite matrix and a position-specific metric of a Riemannian manifold. Therefore the space of parameterized probability density functions is endowed with a natural Riemann geometry. Given this geometry, Rao went further and showed that expressions for the curvature of the manifold and distances on the manifold between two densities could be derived [24].

Given the Bayesian Perspective, $p(y|\boldsymbol{\theta}) \propto p(y, \boldsymbol{\theta})$, which is given by the product of the likelihood and prior. Taking this into account the authors employed this joint probability of data and parameters when defining the metric tensor

$$\mathbf{G}(\boldsymbol{\theta}) = -\mathbb{E}_{y|\theta}\left[\frac{\partial^2}{\partial\theta^2}\log p(y, \theta)\right] \tag{18}$$

which is the expected Fisher information matrix plus the negative Hessian of the log-prior.

The parameter space of a statistical model is indeed a Riemann manifold, and the natural geometric structure of the posterior density model $p(\boldsymbol{\theta})$ is defined by the Riemann manifold and associated metric tensor. This means that the curvature and distance properties of the parameter space are influenced by the metric tensor $\mathbf{G}(\boldsymbol{\theta})$. Adopting a good metric tensor can lead to more effective transitions between different points in the parameter space and improve the algorithm's overall performance.

4.1 Riemannian Manifold Langevin Dynamics

Now, we can modify the MALA in such a way that the SDE evolves along the Riemannian manifold with $\mathbf{G}(\boldsymbol{\theta})$ defined as eqn(18). The modified SDE looks like

$$d\boldsymbol{\theta}(\tau) = \tilde{\mathbf{f}}(\boldsymbol{\theta}(\tau))d\tau + d\tilde{\mathbf{b}}(\tau) \tag{19}$$

$$\tilde{\mathbf{f}}(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{G}^{-1}(\boldsymbol{\theta})\nabla \log p(\boldsymbol{\theta}) \tag{20}$$

$$d\tilde{b}_i = |\mathbf{G}(\boldsymbol{\theta})|^{-1/2}\sum_{j=1}^{D}\frac{\partial}{\partial\theta_j}[\mathbf{G}^{-1}(\boldsymbol{\theta})]_{ij}|\mathbf{G}(\boldsymbol{\theta})|^{-1/2}dt + [\mathbf{G}^{-1/2}(\boldsymbol{\theta})d\mathbf{b}]_i \tag{21}$$

This gives us the general SDE Langevin equation and when the metric tensor is an identity matrix we get the SDE on Euclidean Space. The first term of eqn(21) corresponds to changes in the local curvature of the manifold and reduces to 0 when curvature is constant everywhere. The second right-hand term provides a position-specific axis alignment of the Brownian motion based on the local metric by the transformation of the independent Brownian motion.

We get the final update rule by employing a first-order Euler integrator to the SDE and the update rule, analogous to eqns(3 & 4) is

$$\begin{aligned}\boldsymbol{\theta}_i^{(\tau_{n+1})} \approx \boldsymbol{\theta}_i^{(\tau_n)} + \frac{\Delta\tau}{2}\left[\mathbf{G}^{-1}(\boldsymbol{\theta}^{(\tau_n)})\nabla\log p(\boldsymbol{\theta}^{(\tau_n)})\right]_i - \\ \Delta\tau\sum_{j=1}^{D}\left[\mathbf{G}^{-1}(\boldsymbol{\theta}^{(\tau_n)})\frac{\partial\mathbf{G}(\boldsymbol{\theta}^{(\tau_n)})}{\partial\boldsymbol{\theta}_j}\mathbf{G}^{-1}(\boldsymbol{\theta}^{(\tau_n)})\right]_{ij} + \\ \frac{\Delta\tau}{2}\sum_{j=1}^{D}\left[\mathbf{G}^{-1}(\boldsymbol{\theta}^{(\tau_n)})\right]_{ij}\text{tr}\left[\mathbf{G}^{-1}(\boldsymbol{\theta}^{(\tau_n)})\frac{\partial\mathbf{G}(\boldsymbol{\theta}^{(\tau_n)})}{\partial\boldsymbol{\theta}_j}\right] + \sqrt{\Delta\tau\mathbf{G}^{-1}(\boldsymbol{\theta}^{(\tau_n)})}\mathbf{z}_n\end{aligned} \tag{22}$$

forming the proposal distribution $q(\boldsymbol{\theta}^*|\boldsymbol{\theta})$ as a Gaussian with the first four terms of eqn(22) as the mean and square of the last term as the variance. We replace the update rule and proposal distribution in Algorithm 2 to get the Riemann Manifold Metropolis adjusted Langevin algorithm (RMMALA)

4.2 Riemannian Manifold Hamilton Monte Carlo

Now lets take a look at how HMC is adopted for Riemannian Manifolds. We replace the global mass matrix M with the position-specific metric tensor $\mathbf{G}(\boldsymbol{\theta})$ in eqns (6) to get the Hamiltonian on a Riemann Manifold

$$H(\boldsymbol{\theta},\boldsymbol{\rho}) = -\log p(\boldsymbol{\theta}) + \frac{1}{2}\log|2\pi\mathbf{G}(\boldsymbol{\theta})| + \frac{1}{2}\boldsymbol{\rho}^T\mathbf{G}^{-1}(\boldsymbol{\theta})\boldsymbol{\rho}, \tag{23}$$

and the Hamiltonian Dynamics are given by adapting eqn(8)

$$\frac{d\boldsymbol{\theta}}{d\tau} = \nabla_{\boldsymbol{\rho}}H = \mathbf{G}^{-1}(\boldsymbol{\theta})\boldsymbol{\rho} \tag{24}$$

$$\frac{d\boldsymbol{\rho}}{d\tau} = -\nabla_{\boldsymbol{\theta}}H = \nabla\log p(\boldsymbol{\theta}) + \mathbf{h}(\boldsymbol{\theta}) \tag{25}$$

where the additional term $\mathbf{h}(\boldsymbol{\theta})$ is due to the metric tensor being position specific and hence dependent on $\boldsymbol{\theta}$. It is given by

$$h_i(\boldsymbol{\theta}) = -\frac{1}{2}\text{tr}\left\{\mathbf{G}^{-1}(\boldsymbol{\theta})\frac{\partial\mathbf{G}(\boldsymbol{\theta})}{\partial\theta_i}\right\} + \frac{1}{2}\boldsymbol{\rho}^T\mathbf{G}^{-1}(\boldsymbol{\theta})\frac{\partial\mathbf{G}(\boldsymbol{\theta})}{\partial\theta_i}\mathbf{G}^{-1}(\boldsymbol{\theta})\boldsymbol{\rho}. \tag{26}$$

We use the Leapfrog integrator as mentioned above and carry out the Algorithm(3) with this new Hamilton Dynamics and get the Riemannian Manifold Hamilton Monte Carlo (RMHMC)

5 Illustrative Results

To get a better intuitive understanding of the superior performance of Riemann Manifold MCMC methods, let's take a look at some comparative examples from [11, 4]

A synthetic dataset of observations drawn from a Normal Distribution $\mathcal{N}(\mu = 0, \sigma = 10)$, defined in hyperbolic space. Figure(4) shows the Markov Chain formed by MALA (left) and RMMALA (right) while sampling on the 2-dimensional space. We observe that MALA proposals take inefficient steps

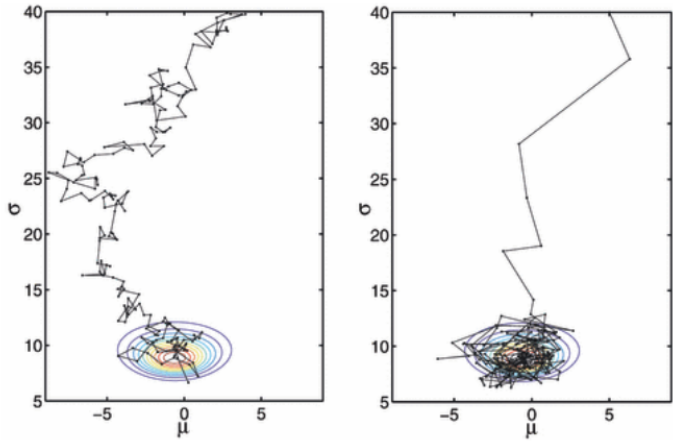


Figure 4: The contours represent sample estimates of $p(\mu, \sigma|X)$. Both MALA and RMMALA were simulated from starting point of $\mu_0 = 5$ and $\sigma_0 = 40$ with a step size $\Delta\tau = 0.75$ for 200 steps [11].

of almost equal length throughout. In contrast, RMMALA proposals are defined by the metric of hyperbolic space, and hence the distances covered at each step reflect the nature of distances on the manifold, resulting in much more efficient traversal of the space.

A stochastic volatility model [18, 14] is defined with latent volatilities taking the form of an auto-regressive AR(1) process. The evolution of Markov Chain on these models using HMC (left) and RMHMC (right) is shown in Figure(5). Notice how the use of the metric on a Riemannian manifold

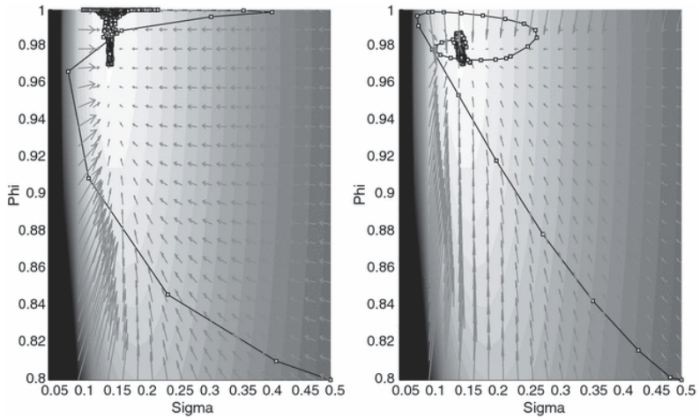


Figure 5: Contours plotted from the stochastic volatility model. HMC (left) is using a unit mass matrix and both HMC and RMHMC Markov chains starting at the same initial point. [11]

allows the RMHMC to converge much more quickly to the target density. Whereas the HMC does get near the target quickly but gets stuck in a local maximum at the top of the high-density region (Figure 6).

Figure(6) takes a closer look at Figure(5) and we can see that RMHMC sampling effectively normalizes the gradients in each direction, whereas HMC, with a unit mass matrix, exhibits stronger gradients along the horizontal direction compared with the vertical direction and therefore takes longer to converge to the target density.

Figure(7), displays the HMC and RMHMC on a warped bivariate Gaussian distribution. We see that RMHMC is able to track the contours of the density and reach the furthest tails of the ridge, adapting

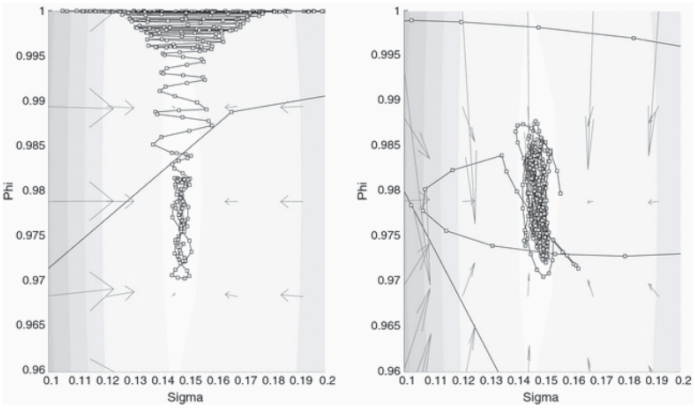
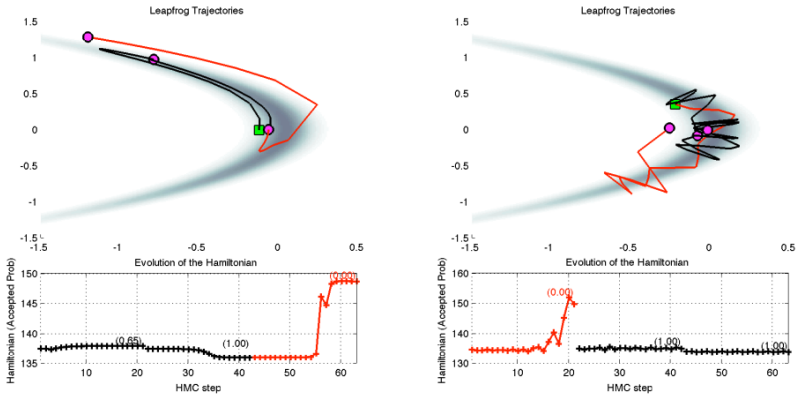


Figure 6: Close-up of the Markov Chain paths shown in Figure (5)



(a) RMHMC (b) HMC

Figure 7: 3 typical consecutive trajectories of length 20 steps each, with a step size of 0.1, for the HMC (right) and RMHMC (left) on a warped bivariate Gaussian. [4]

to the local geometry, whereas the HMC trajectories oscillate back and forth across the center of the ridge.

For detailed experimental results of Riemann Manifold MCMC refer to [11]

6 Conclusion

Dynamics-Based MCMC and the adapted Riemann Manifold MCMCs are a few examples of the several methods that have been developed for Bayesian Posterior Sampling and MCMC methods remain an active area of research to date. There have been various alternative approaches to the methods described in the previous sections and novel methods developed like Stochastic Gradient Hamilton Monte Carlo [7], Stochastic Gradient Riemann Hamilton Monte Carlo [19] Constrained Monte Carlo [5], Geodesic Monte Carlo [6], and Nose-Hoover Thermostats [9]. An interesting line of research has focussed on parallelizing the MCMC process to adapt for large-scale datasets [8, 23].

References

[1] Shun-ichi Amari and Hiroshi Nagaoka. *Methods of information geometry*, volume 191. American Mathematical Soc., 2000.

[2] Ole E Barndorff-Nielsen, David Roxbee Cox, and Nancy Reid. The role of differential geometry in statistical theory. *International Statistical Review/Revue Internationale de Statistique*, pages 83–96, 1986.

[3] Alexandros Beskos, Natesh Pillai, Gareth Roberts, Jesus-Maria Sanz-Serna, and Andrew Stuart. Optimal tuning of the hybrid monte carlo algorithm. 2013.

[4] Luke Bornn, Julien Cornebise, and Gareth W. Peters. Discussion of "riemann manifold langevin and hamiltonian monte carlo methods" by m. girolami and b. calderhead, 2010.

[5] Marcus Brubaker, Mathieu Salzmann, and Raquel Urtasun. A family of mcmc methods on implicitly defined manifolds. In *Artificial intelligence and statistics*, pages 161–172. PMLR, 2012.

[6] Simon Byrne and Mark Girolami. Geodesic monte carlo on embedded manifolds. *Scandinavian Journal of Statistics*, 40(4):825–845, 2013.

[7] Tianqi Chen, Emily B. Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo, 2014.

[8] Daniel Augusto de Souza, Diego Mesquita, Samuel Kaski, and Luigi Acerbi. Parallel mcmc without embarrassing failures, 2022.

[9] Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert D Skeel, and Hartmut Neven. Bayesian sampling using stochastic gradient thermostats. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL `https://proceedings.neurips.cc/paper_files/paper/2014/file/21fe5b8ba755eeaece7a450849876228-Paper.pdf`.

[10] Simon Duane, A.D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216–222, 1987. doi: 10.1016/0370-2693(87)91197-x.

[11] Mark Girolami and Ben Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 73(2):123–214, 2011. doi: 10.1111/j.1467-9868.2010.00765.x.

[12] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

[13] Brian Keng. Manifolds: A gentle introduction, Apr 2018. URL `https://bjlkeng.github.io/posts/manifolds/`.

[14] Sangjoon Kim, Neil Shephard, and Siddhartha Chib. Stochastic volatility: likelihood inference and comparison with arch models. *The review of economic studies*, 65(3):361–393, 1998.

[15] Peter E Kloeden, Eckhard Platen, Peter E Kloeden, and Eckhard Platen. *Stochastic differential equations*. Springer, 1992.

[16] Benedict Leimkuhler and Sebastian Reich. *Simulating hamiltonian dynamics*. Number 14. Cambridge university press, 2004.

[17] Chang Liu and Jun Zhu. Geometry in sampling methods: A review on manifold mcmc and particle-based variational inference methods. *Handbook of Statistics*, page 239–293, 2022. doi: 10.1016/bs.host.2022.07.004.

[18] Jun S Liu and Jun S Liu. *Monte Carlo strategies in scientific computing*, volume 75. Springer, 2001.

[19] Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient mcmc. *Advances in neural information processing systems*, 28, 2015.

[20] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6): 1087–1092, 1953.

[21] Michael K Murray and John W Rice. *Differential geometry and statistics*, volume 48. CRC Press, 1993.

[22] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11): 2, 2011.

[23] Willie Neiswanger, Chong Wang, and Eric Xing. Asymptotically exact, embarrassingly parallel mcmc, 2014.

[24] C Rao. Information and accuracy attainable in estimation of statistical parameters. bulletin of the calcutta mathematical society. 1945.

[25] Christian P Robert and George Casella. Monte carlo statistical methods. 2004. *Google Scholar Google Scholar Digital Library Digital Library*, 2004.

[26] G. O. Roberts. Langevin diffusions and metropolis-hastings algorithms. *Methodology and Computing in Applied Probability*, 4(4):337–357, 2002. doi: 10.1023/a:1023562417138.

[27] Gareth O. Roberts and Jeffrey S. Rosenthal. Optimal scaling of discrete approximations to langevin diffusions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 60(1):255–268, 1998. doi: 10.1111/1467-9868.00123.

[28] Gareth O. Roberts and Jeffrey S. Rosenthal. Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16(4):351 – 367, 2001. doi: 10.1214/ss/1015346320. URL `https://doi.org/10.1214/ss/1015346320`.

[29] Gareth O Roberts and Osnat Stramer. Langevin diffusions and metropolis-hastings algorithms. *Methodology and computing in applied probability*, 4:337–357, 2002.

[30] Chris Sherlock and Gareth Roberts. Optimal scaling of the random walk metropolis on elliptically symmetric unimodal targets. *Bernoulli*, 15(3), aug 2009. doi: 10.3150/08-bej176. URL `https://doi.org/10.3150%2F08-bej176`.