Stochastic Gradient Methods for Markov Chain Monte Carlo

Avik Pal[1*], **Jayant Ranwka**[2*], **and Rahul Sharma**[1*]
[1]Computer Science and Engineering, [2]Mechanical Engineering
Indian Institute of Technology Kanpur
{avikpal, jayant, rsharma}@iitk.ac.in

Abstract

Data-Driven Bayesian Statistical Modeling relies on computing posterior distributions. However, posterior distributions are often intractable, and we have to use approximations. Approximations like Laplace Approximation, Variational Inference, MCMC, etc., have proven effective in generating a tractable approximation of these posterior distributions. In this paper, we focus on the theoretical and intuitive foundations of Stochastic Gradient MCMC Methods. Additionally, we discuss the merits of using cyclic learning rates (cSGLD) over monotonically decreasing learning rates (SGLD) and provide extensive experimental evidence validating the same.

1 Introduction

Modeling uncertainty in complex statistical machine learning models has become an area of growing interest. Learning the posterior distribution of certain unknowns $p(\theta \mid X)$ given a dataset X implicitly provides us with a model ensemble, allowing more robust predictions, and has led to exciting advances in sequential decision making (Reinforcement Learning [1, 2], Bayesian Optimization [3], etc.). Learning the true posterior can be expensive and often intractable for most interesting problems. Hence, we often construct approximations of the true posterior with a simpler tractable distribution using Variational Bayes [4, 5], Laplace Approximation [6], and Expectation Propagation [7].

Alternatively, we could approximate the posterior by an empirical distribution using i.i.d. samples generated from the true posterior. We could accomplish this by using Rejection sampling (RS). However, RS suffers from a high rejection rate and therefore useful only for low dimensional problems. For high dimensional applications, we have to use Markov Chain Monte Carlo (MCMC) methods to approximate the target posterior arbitrarily well via a Markov chain. However, unlike RS, the MCMC samples are correlated. Formally, we can describe this approximate distribution as:

$$p_L(A) = \sum_{l=1}^{L} w_l \delta_{z^{(l)}}(A) \quad \sum_{l=1}^{L} w_l = 1 \quad \delta_z(A) = \begin{cases} 0 & \text{if } z \notin A \\ 1 & \text{otherwise,} \end{cases}$$

where the samples $\left\{z^{(l)}\right\}_{l=1}^{L}$ are generated from some sampling routine (RS or MCMC). This paper explores a particular MCMC algorithm – Stochastic Gradient Langevin Dynamics (SGLD) – which incorporates gradient information in MCMC updates to generate samples from the posterior. We motivate the intuition of the algorithm from a molecular dynamics and stochastic optimization perspective and provide necessary mathematical foundations for the same. However, the simplicity of the method leads to problems in sampling from multi-modal distributions, and consequently, we discuss some latest advances which circumvent these issues. Additionally, we showcase these methods using a complex Bayesian Neural Network on a suite of real-world datasets.

Preprint. Under review.

2 Background

2.1 Bayesian Inference

In Bayesian Inference, we have a prior distribution $p(\theta)$ and a set of i.i.d. training examples $\{x_1, \ldots, x_N\}$. Given the likelihood of the model $p(x_i \mid \theta)$, our task is to learn the true posterior $p(\theta \mid x_1, \ldots, x_N)$.

$$p(X \mid \theta) = \prod_{n=1}^{N} p(x_n \mid \theta) \qquad p(\theta \mid X) = \frac{p(X \mid \theta)p(\theta)}{\int p(X \mid \theta)p(\theta)d\theta}$$

Estimating the posterior allows us to perform more robust predictions by computing Posterior Predictive Distributions $p(x_* \mid X) = \int p(x_* \mid \theta)p(\theta \mid X)d\theta$, rather than relying on point estimates like MLE. However, for most problems, the denominator $\int p(X \mid \theta)p(\theta)d\theta$ is intractable, and hence we can evaluate the posterior only up to a proportionality constant. Additionally, for machine learning applications, $\theta \in \mathbb{R}^d$ is extremely high dimensional, and sampling techniques like Rejection Sampling are not efficient. These properties make MCMC a strong candidate for generating samples from this posterior.

2.2 Stochastic Gradient Optimization

Gradient Based Optimization is used to find the local optima $\theta^* \in \mathbb{R}^d$ of a differentiable loss function $J(\theta)$. For a Bayesian MAP Estimation problem, $J(\theta) = -\log p(\theta, X) = -\log p(X \mid \theta) - \log p(\theta)$, since the normalization constant doesn't affect our optima.

To locate the local minima, we use gradient descent and move opposite to the direction of the gradient $\nabla_\theta J(\theta)$. Since machine learning models have millions (and even billions) of parameters, using the gradients obtained via backpropagation provides us with an efficient search strategy to navigate the parameter space. We train these statistical models with lots of data points, and computing the gradient using all these points per parameter update becomes highly inefficient. A simple strategy to circumvent this issue would be to perform the update per data sample (Stochastic Gradient Descent) or using a mini-batch of data samples (Mini-Batch Gradient Descent).

$$\theta_{t+1} = \theta_t - \eta_{t+1}\left(\frac{N}{B}\sum_{i=1}^{B} \nabla_{\theta_t} J\left(\theta_t\right)\right)$$

$$\theta_{t+1} = \theta_t + \eta_{t+1}\left(\frac{N}{B}\sum_{i=1}^{B} \nabla_{\theta_t}\left(\log p(x_i \mid \theta_t) + \log p(\theta_t)\right)\right)$$

Choosing a learning rate η_t becomes tedious for high-dimensional problems. Additionally, we might want different learning rates for each dimension, making the process even more complicated. Some popular methods like ADAM [8], AdaGrad [9], AdaDelta [10], etc. overcome this issue by adaptively modifying the learning rate depending on whether those dimensions change slowly or rapidly. [11] has recently proposed some exciting strategies for adaptive SGLD methods, but for the rest of this paper, we shall focus primarily on non-adaptive SGLD methods.

2.3 Metropolis Adjusted Langevin Algorithm (MALA)

In the previous section, we discussed methods to find a local optima. However, in many cases like Bayesian Inference we want to explore the complete posterior density which motivates the use of sampling routines like MCMC. In particular, MCMC constructs a Markov chain such that the stationary distribution of Markov chain is exactly equal to the target posterior. Some popular MCMC routines are Metropolis Hastings (MH) [12], Metropolis Adjusted Langevin Dynamics (MALA) [13], Hamiltonian Monte Carlo (HMC) [14], and Riemann extensions of MH [15].

MALA builds upon Langevin Stochastic Differential Equation (SDE) with the equilibrium distribution $\pi(x) \propto e^{\log p(X,\theta)}$:

$$d\theta_t = -\nabla L(\theta_t)dt + \sqrt{2}dB_t$$

where $L(\theta_t) = -\log p(X, \theta_t)$, $\quad (B_t)_{t \geq 0}$ is a d-dimensional Brownian Motion
ΔB_t are i.i.d. Gaussian Random Variables

The diffusion paths for the discretized SDE are given by Euler-Maruyama (EM) Scheme [16, 17]:

$$\theta_{t+1} = \theta_t - \eta_{t+1} \nabla L(\theta_t) + \sqrt{2\eta_{t+1}} \epsilon_t \quad \epsilon_t \sim \mathcal{N}(0, I_d)$$

We can rewrite this formulation as unnormalized SGD updates followed by sampling from a Gaussian Distribution centered at the current parameter estimate.

$$\theta_* = \theta_t + \eta_{t+1} \nabla_{\theta_t} \left(\log p(X \mid \theta_t) + \log p(\theta_t) \right) \quad \theta_{t+1} \sim \mathcal{N}\left(\theta_*, 2\eta_{t+1} I_d\right)$$

Finally, in Metropolis Adjusted Langevin Algorithm (MALA), we generate the Markov Chain by performing the Metropolis Hastings (MH) Accept-Reject step. On keeping the η_t constant, MALA generates a homogeneous Markov Chain $(\theta_t)_{t \geq 1}$; While if η_t is non-increasing and non-constant, MALA generates a non-homogeneous Markov Chain $(\theta_t)_{t \geq 1}$.

2.4 Heuristics to Compare different Samplers

In this section, we shall discuss some metrics we will use to compare SGLD and cSGLD samples. Consider a target density $p(x)$ and we want to estimate $\mathbb{E}[g(x)] = \int_{\mathcal{X}} g(x)p(x)dx$. We generate MCMC samples $\left\{X^{(s)}\right\}_{s=1}^{S}$ from $p(X)$, and estimate $\mathbb{E}[g(x)] = \hat{\mu}_g = \frac{1}{S}\sum_{s=1}^{S} g(x^{(s)})$. We can use Asymptotic Variance (Spectral Variance Estimator, Batch Means Estimator) [18], Sample Variance and Effective Sample Size[19, 20] to reason about which sampler gives a better estimate for $\hat{\mu}_g$.

- **Sample Variance** ($\hat{\lambda}^2$) is the variance of the samples generated assuming no correlation across them.

  $$\hat{\lambda}^2 = \frac{1}{S-1} \sum_{s=1}^{S} \left(g\left(x^{(s)}\right) - \hat{\mu}_g \right)^2$$

  The sample variance is a heuristic to estimate if the markov chain explored the support space. A higher $\hat{\lambda}^2$ essentially means that the chain explored low probability areas. Hence, we want a *higher* $\hat{\lambda}^2$.

- **Asymptotic Variance Estimators** is the infinite sum of lag covariances. Since we have n samples, which is typically large, computing this exactly is highly expensive. Hence, we rely on approximations of this value. Since this essentially measures the correlations across samples and we desire uncorrelated samples we want this estimate to be *lower* for better samples. The natural estimator for lag k covariance is given by:

  $$\hat{R}(k) = \frac{1}{S} \sum_{s=1}^{S-k} \left(g\left(x^{(s)}\right) - \hat{\mu}_g \right) \left(g\left(x^{(s+k)}\right) - \hat{\mu}_g \right)$$

  - **Spectral Variance Estimator** ($\hat{\sigma}_{sv}^2$) is a reliable but slightly computationally expensive estimate of the asymptotic variance. It truncates the lag to a truncation point b and uses different weights for each lag.

    $$\hat{\sigma}_{sv}^2 = \sum_{k=-(b-1)}^{b-1} w\left(\frac{k}{b-1}\right) \hat{R}(k)$$

    In this paper, we exclusively use the Bartlett Lag Window $w(x) = (1 - |x|)\mathbb{I}[|x| \leq 1]$ and set $b = \left\lfloor S^{\frac{n}{2}} \right\rfloor$

  - **Batch Means Variance Estimator** ($\hat{\sigma}_{bm}^2$) is a cheap estimate which splits the samples into a batches of batch size b. Then we simply compute the mean of each batch and estimate the sample variance assuming that these means are our samples.

    $$\bar{Y}_k = \frac{1}{b} \sum_{s=1}^{b} g\left(x^{((k-1)b+s)}\right) \quad \hat{\sigma}_{bm}^2 = \frac{b}{a-1} \sum_{k=1}^{a} \left(\bar{Y}_k - \hat{\mu}_g\right)^2$$

- **Effective Sample Size** (ESS) is the number of iid samples that would produce the same standard error as the correlated MCMC samples. We would want this number to be *high*, since it would mean that we have generated "more" uncorrelated samples. We estimate this using the sample variance and asymptotic variance. In our experiments we report ESS wrt both $\hat{\sigma}^2_{sv}$ and $\hat{\sigma}^2_{bm}$. If the asymptotic variance estimate is $\hat{\sigma}^2_g$ and the sample variance is $\hat{\lambda}^2$, ESS is given by:

  $$ESS = S\frac{\hat{\lambda}^2}{\hat{\sigma}^2_g}$$

  For high dimensional problems we rely on the multiESS metric proposed in [20].

3 Method

3.1 Stochastic Gradient Langevin Dynamics

Stochastic Gradient Langevin Dynamics (SGLD) is an optimization algorithm popularly used to generate samples from posteriors in Bayesian Learning applications. [2] proposed SGLD by combining stochastic (mini-batch) gradient descent (SGD) with LD to obtain an algorithm to generate a non-stationary Markov Chain and provided extensive experiments to establish that this Markov Chain can generate samples from the posterior.

$$\Delta\theta_t = \eta_t \left(\nabla \log p(\theta_t) + \frac{N}{n} \sum_{i=1}^{n} \nabla \log p((x_i)_t|\theta_t) \right) + \epsilon_t$$

$$\sum_{t=1}^{\infty} \eta_t = \infty \qquad \sum_{t=1}^{\infty} \eta_t^2 < \infty \quad \epsilon_t \sim \mathcal{N}(0, 2\eta_t I_d)$$

This structure to the updates pose an issue where the curvature of different dimensions of the parameters are vastly different. The authors propose to use a symmetric preconditioning matrix M to alleviate these issues. The update rule is thus augmented to become:

$$\Delta\theta_t = \eta_t M \left(\nabla \log p(\theta_t) + \frac{N}{n} \sum_{i=1}^{n} \nabla \log p((x_i)_t|\theta_t) \right) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, 2\eta_t M)$$

However, this requires us to explicitly define a preconditioning matrix. Recently [11] proposed a class of adaptive SGLD updates for Bayesian Machine Learning Applications.

$$\Delta\theta_t = \eta_t \left(\nabla \log p(\theta_t) + \frac{N}{n} \sum_{i=1}^{n} \nabla \log p((x_i)_t|\theta_t) + aA_t \right) + \sqrt{2\eta_t \tau}\epsilon_t$$

$$\text{where } \epsilon_t \sim \mathcal{N}(0, I_d), \quad a \text{ is the bias factor}, \quad A_t \text{ is the adaptive bias term}$$

These adaptive SGLD methods are widely applicable, but are beyond the scope of this paper. Unlike traditional MCMC methods which require an explicit Accept-Reject step, SGLD doesn't require that due to the monotonically decreasing learning rate. SGLD Algorithm can be broadly divided into two phases:

1. **Exploration Phase**: In this phase, the parameters converge towards the local maxima of the distribution. Initially, when the learning rate is relatively large, the SGD updates dominate the added noise. This is the burn-in phase and the samples generated in this phase are not retained.
2. **Sampling Phase**: Once the parameters have converged to a local maxima, the noise term dominates since the gradients tend to zero. The samples generated are from high probability regions around the maxima and it has been shown empirically that the acceptance probability tends to one as the learning rate decreases. Hence, there is no need to explicitly perform the Accept-Reject Step.

This strategy poses a clear problem, of the samples being generated around a local maxima. Hence, SGLD will never be able to generate samples from a multi-modal distribution. This problem is alleviated to a certain extent using Cyclic SGLD.

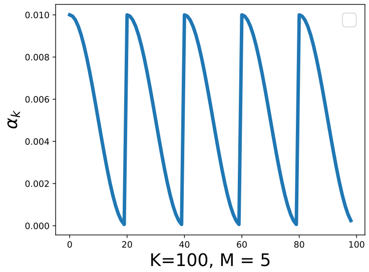

Figure 1: **Cosine learning rate schedule**

3.2 Cyclic SGLD

In the previous section, we introduced the method of SGLD. However, as pointed out earlier, SGLD often gets stuck in a local maximum. In contrast cyclic SGLD [21] employs a cyclic learning rate to explore multiple modes thereby solving the issue. Specifically, a cosine step size scheduler is used to set the learning rate. Let the iteration number be k, the learning rate α_k is defined as follows:

$$\alpha_k = \frac{\alpha_0}{2}\left[1 + \cos\left(\frac{\pi \text{mod}(k-1, \lceil \frac{K}{M} \rceil)}{\lceil \frac{K}{M} \rceil}\right)\right],$$

where α_0 is the initial learning rate, M decides the number of cycles, and K is the total number of iterations. For visualization, please refer to figure (1). C-SGLD also operates in two phases: *exploration* phase and a *sampling* phase as discussed in the previous section. In particular, when α_k is less than a threshold we start collecting samples. Note, however, in contrast to SGLD sample collection is done multiple times due to the cyclic nature of SGLD.

4 Theoretical Analysis

4.1 Weak Convergence

Weak Convergence estimates the expectation of a suitable test function $\phi(\theta)$ by taking the average over the generated samples $(\theta_k)_{k=1}^K$. We can then measure the accuracy of SGLD algorithm by using the mean square error of the estimator. [22] considered this in the case of decreasing step sizes. MSE can be partitioned into square bias term and variance term. Following [21], we will prove weak convergence of cSGLD.

Define the posterior average as: $\bar{\phi} := \int_{\mathcal{X}} \phi(\theta)\rho(\theta)\, d\theta$ and the sample average as: $\hat{\phi} := \frac{1}{K}\sum_{k=1}^{K}\phi(\theta_k)$. In the analysis, we define a functional ψ that solves the following Poisson Equation:

$$\mathcal{L}\psi(\theta_k) = \phi(\theta_k) - \bar{\phi}, \text{ or equivalently, } \frac{1}{K}\sum_{k=1}^{K}\mathcal{L}\psi(\theta_k) = \hat{\phi} - \bar{\phi}.$$

The solution functional $\psi(\theta_k)$ characterizes the difference between $\phi(\theta_k)$ and $\bar{\phi}$ for every θ_k, thus would typically possess a unique solution, which is at least as smooth as ϕ under the elliptic or hypoelliptic settings [23].

Assumption 1. ψ and its up to 3rd-order derivatives, $\mathcal{D}^k\psi$, are bounded by a function $\mathcal{V}$, i.e., $\|\mathcal{D}^k\psi\| \leq H_k\mathcal{V}^{p_k}$ for $k = (0, 1, 2, 3)$, $H_k, p_k > 0$. Furthermore, the expectation of $\mathcal{V}$ on $\{\theta_k\}$ is bounded: $\sup_l \mathbb{E}\mathcal{V}^p(\theta_k) < \infty$, and $\mathcal{V}$ is smooth such that $\sup_{s\in(0,1)}\mathcal{V}^p(s\theta + (1-s)\theta') \leq C(\mathcal{V}^p(\theta) + \mathcal{V}^p(\theta')), \forall\theta, \theta', p \leq \max\{2p_k\}$ for some $C > 0$.

Lemma 1. Let $S_K := \sum_{k=1}^{K} h_k$. Under Assumptions 1 , for a smooth test function ϕ, the bias and MSE of a decreasing-step-size SG-MCMC with a Nth-order integrator at time S_K are bounded as:

$$\text{BIAS} : |\mathbb{E}\hat{\phi} - \bar{\phi}| = O\left(\frac{1}{S_K} + \frac{\sum_{k=1}^{K} h_k^{N+1}}{S_K}\right)$$

$$\text{MSE} : \mathbb{E}\left(\hat{\phi} - \bar{\phi}\right)^2 \leq C\left(\sum_k \frac{h_k^2}{S_K^2}\mathbb{E}\|\Delta V_k\|^2 + \frac{1}{S_K} + \frac{\left(\sum_{k=1}^{K} h_k^{N+1}\right)^2}{S_K^2}\right)$$

ΔV_k measures the difference between stochastic gradient and true gradient of the potential energy.

$$\Delta V_k = \nabla_\theta \tilde{U}_k - \nabla_\theta U$$

Proof. The proof for this lemma is beyond the scope of this paper. Refer to [24] for the detailed proof. ■

Theorem 1. Under Assumption 1, for a smooth test function ϕ, the bias and MSE of cSGLD are bounded as:

$$\text{BIAS} : |\mathbb{E}\hat{\phi} - \bar{\phi}| = O\left(\frac{1}{\alpha_0 K} + \alpha_0\right), \ \text{MSE} : \mathbb{E}\left(\hat{\phi} - \bar{\phi}\right)^2 = O\left(\frac{1}{\alpha_0 K} + \alpha_0^2\right).$$

Proof. This result is a special case of Lemma 1. Also, cSGLD adopts a first order integrator, thus $N = 1$.

$$\begin{aligned} S_K &= \sum_{k=1}^{K} \alpha_k = \frac{\alpha_0}{2}\sum_{k=1}^{K}\left[\cos\left(\frac{\pi \mod (k-1, \lceil K/M \rceil)}{\lceil K/M \rceil}\right) + 1\right] \\ &= \frac{\alpha_0}{2}(M + K) \quad \text{using telescopic sum [25]} \\ &= O(\alpha_0 K) \quad \text{as } M \leq K. \end{aligned}$$

Now, we will prove the following result for just 1 cycle and assume that $K \mid M$ with $\frac{K}{M} = T$ and then generalise it for M cycles by multiplying with M.

$$\begin{aligned} \sum_{t=1}^{T} \alpha_t^2 &= \frac{\alpha_0^2}{4}\sum_{t=1}^{T}\left[\cos\left(\frac{t-1}{T}\pi\right) + 1\right]^2 \\ &= \frac{\alpha_0^2}{4}\sum_{t=1}^{T}\left[\cos^2\left(\frac{t-1}{T}\pi\right) + 1 + 2\cos\left(\frac{t-1}{T}\pi\right)\right] \\ &= \frac{\alpha_0^2}{4}\left(\sum_{t=1}^{T}\left[\frac{1 + \cos\left(2\frac{t-1}{T}\pi\right)}{2} + 1\right] + 2\right) \quad \text{as } \cos(\pi - x) = -\cos(x) \\ &= \frac{\alpha_0^2}{4}\left(\frac{3T}{2} + \frac{1}{2}\sum_{t=1}^{T}\cos\left(2\frac{t-1}{T}\pi\right) + 2\right) \\ &= \begin{cases} \frac{\alpha_0^2}{4}\left(\frac{3T}{2} + 2\right) & \text{if } T \neq 1 \\ \frac{\alpha_0^2}{4}\left(2T + 2\right) & \text{if } T = 1 \end{cases} \end{aligned}$$

For the last step, we used telescopic sum [25]. For M cycles,

$$\begin{aligned} \sum_{k=1}^{K} \alpha_k^2 &= \begin{cases} \frac{\alpha_0^2}{4}\left(\frac{3T}{2} + 2\right)M & \text{if } K/M \neq 1 \\ \frac{\alpha_0^2}{4}\left(2T + 2\right)M & \text{if } K/M = 1 \end{cases} \\ &= \begin{cases} \frac{\alpha_0^2}{4}\left(\frac{3K}{2} + 2M\right) & \text{if } K/M \neq 1 \\ \frac{\alpha_0^2}{4}\left(2K + 2M\right) & \text{if } K/M = 1 \end{cases} \end{aligned}$$

Thus,

$$\begin{aligned} |\mathbb{E}\hat{\phi} - \bar{\phi}| &= O\left(\frac{1}{\alpha_0 K} + \frac{\alpha_0^2 (K+M)}{\alpha_0 K}\right) \\ &= O\left(\frac{1}{\alpha_0 K} + \alpha_0 + \frac{M}{K}\alpha_0\right) \\ &= O\left(\frac{1}{\alpha_0 K} + \alpha_0\right) \quad \text{as } M \le K \end{aligned}$$

For the MSE, the first term has higher order than others. So, we will ignore it in the order notation.

$$\begin{aligned} \mathbb{E}\left(\hat{\phi} - \bar{\phi}\right)^2 &= O\left(\frac{1}{\alpha_0 K} + \frac{\alpha_0^4 (K+M)^2}{\alpha_0^2 K^2}\right) \\ &= O\left(\frac{1}{\alpha_0 K} + \alpha_0^2 \left(1 + \frac{M}{K}\right)^2\right) \\ &= O\left(\frac{1}{\alpha_0 K} + \alpha_0^2\right) \quad \text{as } M \le K \end{aligned}$$

■

4.2 Convergence under the Wasserstein distance

Convergence under the Wasserstein distance considers the distribution μ_K that SGLD samples from at iteration K and measures its Wasserstein distance from the target distribution. 2-Wasserstein distance being a metric has its own advantages over other such functions like KL divergence.

$$W_2^2(\mu, \nu) := \inf_{\gamma} \left\{ \int_{\Omega \times \Omega} \|\theta - \theta'\|_2^2 \, d\gamma(\theta, \theta') : \gamma \in \Gamma(\mu, \nu) \right\}$$

where $\Gamma(\mu, \nu)$ is the set of joint distributions over (θ, θ') such that the two marginals equal μ and ν, respectively.

Denote the distribution of θ_t in the SDE as ν_t. The stationary distribution ν_∞ matches our target distribution [26]. We will derive a convergence bound on $W_2(\mu_K, \nu_\infty)$ for cSGLD.

Assumption 2. Following [27], the following standard assumptions have been adopted.

- $\exists$ some constants $A \ge 0$ and $B \ge 0$, such that $U(0) \le A$ and $\nabla U(0) \le B$.
- The function U is L_U-smooth: $\|\nabla U(w) - \nabla U(v)\| \le L_U \|w - v\|$.
- The function U is $(m_u, b)-$ dissipative, which means for some $m_U > 0$ and $b > 0$ $\langle w, \nabla U(w) \rangle \ge m_U \|w\|^2 - b$.
- $\exists$ some constant $\delta \in [0, 1)$, such that $\mathbb{E}\left[\|\nabla \tilde{U}_k(w) - \nabla U(w)\|^2\right] \le 2\sigma\left(M_U^2 \|w\|^2 + B^2\right)$.
- We can choose μ_0 which satisfies the requirement: $\kappa_0 := \log \int e^{\|w\|^2} \mu_0(w)\, dw < \infty$.

Theorem 2. Under Assumption 2, $\exists$ constants (C_0, C_1, C_2, C_3) independent of the stepsizes such that the convergence rate of the proposed cSGLD is bounded $\forall$ K satisfying $(K \mod M = 0)$, as $W_2(\mu_K, \nu_\infty) \le C_3 \exp\left(-\frac{K\alpha_0}{2C_4}\right) + \left(6 + \frac{C_2 K \alpha_0}{2}\right)^{\frac{1}{2}} \left[\left(C_1 \frac{3\alpha_0^2 K}{8} + \sigma C_0 \frac{K\alpha_2}{2}\right)^{\frac{1}{2}} + \left(C_1 \frac{3\alpha_0^2 K}{16} + \sigma C_0 \frac{K\alpha_2}{4}\right)^{\frac{1}{4}}\right]$.

Particularly, if we further assume $\alpha_0 = O\left(K^{-\beta}\right)$ $\forall \beta > 1$, $W_2(\mu_K, \nu_\infty) \le C_3 + \left(6 + \frac{C_2}{K^{\beta-1}}\right)^{\frac{1}{2}} \left[\left(\frac{2C_1}{K^{2\beta-1}} + \frac{2C_0}{K^{\beta-1}}\right)^{\frac{1}{2}} + \left(\frac{C_1}{K^{2\beta-1}} + \frac{C_0}{K^{\beta-1}}\right)^{\frac{1}{4}}\right]$.

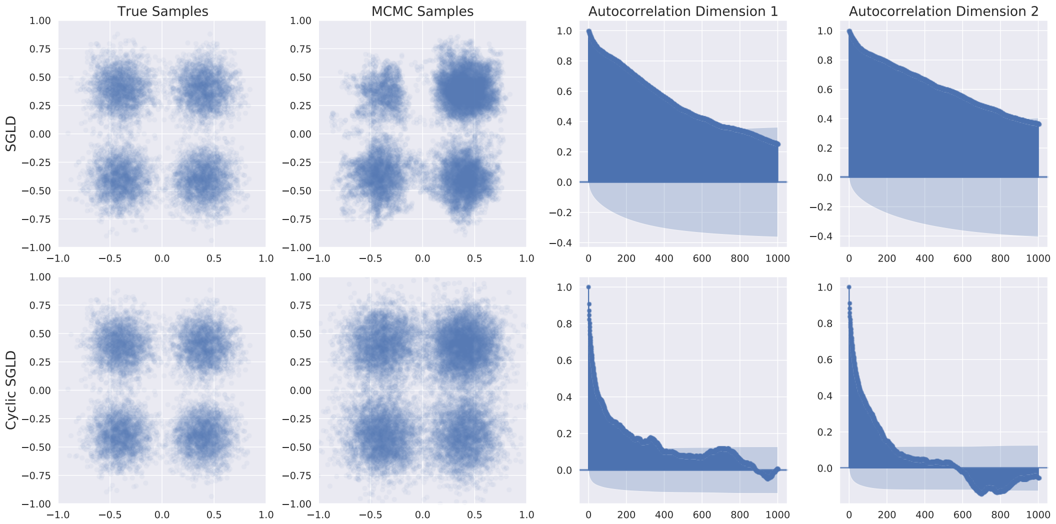

Figure 2: **Sampling from Mixture of Gaussians** Both SGLD and cSGLD are able to discover all the modes of the distribution, but SGLD samples are skewed towards a few modes while cSGLD ones are well distributed. Additionally SGLD samples have an extremely high autocorrelation lag.

Proof. This proof is beyond the scope of this paper, please refer to [21]. ■

5 Experiments

5.1 Mixture of Gaussians

In this toy experiment, we generate samples from a Mixture of Gaussians with all parameters known:

$$p(X) = \frac{1}{4} \sum_{\substack{i \in \{-1,1\} \\ j \in \{-1,1\}}} \mathcal{N}\left(X \mid \begin{bmatrix} 0.4i \\ 0.4j \end{bmatrix}, \begin{bmatrix} 0.02 & 0.0 \\ 0.0 & 0.02 \end{bmatrix}\right)$$

Though sampling from this distribution would be easier if we sample from a Multinomial Distribution followed by the respective Gaussian Distribution, this experiment serves as an interesting baseline to understand the differences between SGLD and cSGLD. For each of the runs we start from the origin $(0, 0)$ and generate a total $20k$ samples. We discard the first $1k$ samples generated in the burn-in phase. Typically, in cSGLD we discard a group of alternating samples due to the multiple exploration phases, but for this simple problem with close modes we can retain them. we use the learning rate scheduling from [21, 28]:

$$(\eta_t)_{SGLD} = \begin{cases} \eta_0 \times (t+1)^{-0.55} & \text{if } t < 200 \\ \eta_0 \times 200^{-0.55} & otherwise \end{cases}$$

$$(\eta_t)_{cSGLD} = \eta_0 \times \left(\gamma_t \times sign\left(\frac{d\gamma_t}{dt}\right)\right) + 0.0005$$

$$\gamma_t = \cos\left(\frac{\pi \omega t}{360}\right) \qquad \eta_0 = 0.01 \quad \omega = 0.3$$

In Figure 2, we can qualitatively observe that cSGLD samples are more evenly distributed around all the modes of the true distribution. The cyclic learning rate increases the probability of sample from escaping a particular mode and explore other regions of higher probability. In case of SGLD, even though we obtain samples around all the modes, these samples are highly skewed since the only way to escape a mode is to sample a point in a very low density region from the gaussian noise. Table 1 quantifies these observations using heuristics defined in Section 2.4. These results clearly suggest that the samples generated by cSGLD are of higher quality.

	x-component					y-component				
Sampler	$\hat{\lambda}^2$	$\hat{\sigma}^2_{sv}$	ESS $S\frac{\hat{\lambda}^2}{\hat{\sigma}^2_{sv}}$	$\hat{\sigma}^2_{bm}$	ESS $S\frac{\hat{\lambda}^2}{\hat{\sigma}^2_{bm}}$	$\hat{\lambda}^2$	$\hat{\sigma}^2_{sv}$	ESS $S\frac{\hat{\lambda}^2}{\hat{\sigma}^2_{sv}}$	$\hat{\sigma}^2_{bm}$	ESS $S\frac{\hat{\lambda}^2}{\hat{\sigma}^2_{bm}}$
SGLD	0.1507	19.119	157.70	206.23	14.620	0.1684	21.624	155.75	467.98	7.1970
cSGLD	**0.1805**	**12.660**	**285.25**	**18.772**	**192.37**	**0.1853**	**13.840**	**267.81**	**14.489**	**255.80**

Table 1: **Sample Statistics for the Mixture of Gaussians Experiment**

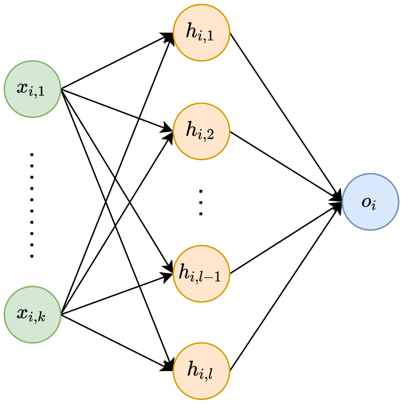

Figure 3: **Neural Network Architecture**

5.2 Bayesian Neural Net

We compare SGLD and C-SGLD for Bayesian neural net regression on datasets from the UCI repository. we use a single-layer neural net with 10 hidden units and ReLU activation function (Figure 3). Our goal is to infer the posterior $p(\theta, \sigma^2 \mid x)$, where θ represents the weights of the neural net and σ^2 is the variance of the likelihood term. Specifically, we can express the posterior density as follows:

$$p(\theta, \sigma^2 | X, y) \propto \mathcal{N}(\theta \mid \mathbf{0}, \mathbf{I}) \prod_{i=1}^{N} \mathcal{N}(y_i \mid \text{NN}(x_i, \theta), \sigma^2)$$

Each data-set was randomly split into 80 $\%$ training set and 20 $\%$ test set. We evaluated the negative log-likelihood (NLL) and RMSE on the test data to compare the performance. Each experiment was repeated 20 times to get an effective measure on the performance.

For estimating the gradient we used a batch size of 32 with initial learning rate 10^{-4} for all the data-sets and gradually reduced it via polynomial decay for SGLD. Once the learning rate was below 10^{-5} we started collecting samples for both SGLD as well as C-SGLD. The total number of epochs for all the data-sets were 2000. Further, for C-SGLD we chose $M = 5$ (number of cycles). Table 2 summarizes the results. For a single training set we also evaluated multivariate ESS [20] for 2000 samples from both SGLD as C-SGLD. As evident from the results we are getting superior performance on the test data along with superior sample quality.

6 Discussions

In our experiments, it was evident that SGLD samples tend to collapse to a particular mode unless explicit care is taken to avoid the same. Apart from Cyclic Learning Rates (cSGLD), methods like Hybrid/Hamiltonian Monte Carlo (HMC) [29] and Stochastic Gradient HMC [30] alleviate the problem of exploration. HMC is motivated by Hamiltonian Dynamics:

$$\frac{\partial \theta}{\partial t} = \frac{\partial H}{\partial r} = \frac{\partial K}{\partial r} \qquad \frac{\partial r}{\partial t} = -\frac{\partial H}{\partial \theta} = -\frac{\partial K}{\partial \theta}$$

$$H(\theta, r) = U(\theta) + \frac{1}{2} r^T M^{-1} r = U(\theta) + K(r)$$

The discretized form of this dynamics controls the transition of the parameters $\theta \mapsto \theta^*$. The momentum term $\frac{\partial r}{\partial t}$ ensures exploration of the entire support space rather than getting stuck near

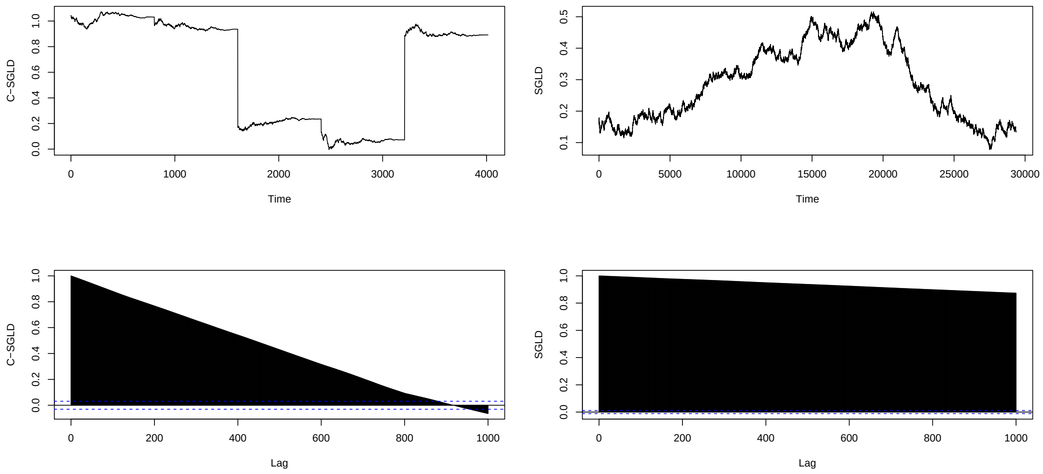

Figure 4: Comparison of trace plot and acf for 1 dimension of **Boston** data-set **left**: C-SGLD, observe that there are jumps between different modes, **right**: SGLD, observe that the chain is stuck at a single place

	SGLD			cSGLD		
Dataset	RMSE	NLL	mESS	RMSE	NLL	mESS
Boston	3.43 ± 0.18	**2.65 ± 0.18**	350.30	**3.22 ± 0.18**	2.69 ± 0.18	**1826.01**
Concrete	5.92 ± 0.09	3.19 ± 0.09	563.37	**5.74 ± 0.12**	**3.18 ± 0.12**	**711.17**
Energy	2.19 ± 0.09	2.18 ± 0.09	545.39	**0.72 ± 0.03**	**1.14 ± 0.03**	**656.66**
Power	**4.10 ± 0.03**	**2.82 ± 0.03**	188.82	4.19 ± 0.03	2.85 ± 0.04	**190.95**
Yacht	5.41 ± 0.27	3.25 ± 0.27	319.23	**1.21 ± 0.08**	**1.64 ± 0.08**	**401.66**

Table 2: **Performance of Bayesian Neural Network trained using SGLD and cSGLD** cSGLD performs better than SGLD in most of the experiments, and the cases where the performance is inferior the difference is very minor. Comparing the multiESS (mESS) values we can clearly see that cSGLD produces far superior samples compared to SGLD.

the MAP solution. Unlike SGLD, where each sample generated is accepted, in HMC we perform L leapfrop steps and at the end perform an MH Accept-Reject step. SGHMC allows us to skip this Accept-Reject Step.

7 Conclusion

In this paper, we have described a popular MCMC Algorithm, SGLD, which infuses gradient information in parameter updates to ensure that we reach areas of high probability densities faster. We followed this with a discussion on cSGLD, which allows the algorithm to handle low probability density areas and multi-modal distributions. We have reviewed some of the theoretical analyses for convergence of SGLD and cSGLD. Finally, we provide extensive experimental results on both toy and real-world datasets to show that cSGLD almost always performs better than SGLD.

References

[1] Nikos Vlassis, Mohammad Ghavamzadeh, Shie Mannor, and Pascal Poupart. Bayesian reinforcement learning. *Reinforcement learning*, pages 359–386, 2012.

[2] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

[3] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[4] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(5), 2013.

[5] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[6] Pierre Simon Laplace. Memoir on the probability of the causes of events. *Statistical science*, 1(3):364–378, 1986.

[7] Thomas P Minka. Expectation propagation for approximate bayesian inference. *arXiv preprint arXiv:1301.2294*, 2013.

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[9] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[10] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[11] Sehwan Kim, Qifan Song, and Faming Liang. Stochastic gradient langevin dynamics algorithms with adaptive drifts. *arXiv preprint arXiv:2009.09535*, 2020.

[12] Siddhartha Chib and Edward Greenberg. Understanding the metropolis-hastings algorithm. *The american statistician*, 49(4):327–335, 1995.

[13] Tatiana Xifara, Chris Sherlock, Samuel Livingstone, Simon Byrne, and Mark Girolami. Langevin diffusions and the metropolis-adjusted langevin algorithm. *Statistics & Probability Letters*, 91:14–19, 2014.

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

[15] 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.

[16] Yaozhong Hu. Semi-implicit euler-maruyama scheme for stiff stochastic equations. In *Stochastic Analysis and Related Topics V*, pages 183–202. Springer, 1996.

[17] H Lamba, Jonathan C Mattingly, and Andrew M Stuart. An adaptive euler–maruyama scheme for sdes: convergence and stability. *IMA journal of numerical analysis*, 27(3):479–506, 2007.

[18] James M Flegal, Galin L Jones, et al. Batch means and spectral variance estimators in markov chain monte carlo. *The Annals of Statistics*, 38(2):1034–1070, 2010.

[19] Augustine Kong, Jun S Liu, and Wing Hung Wong. Sequential imputations and bayesian missing data problems. *Journal of the American statistical association*, 89(425):278–288, 1994.

[20] Dootika Vats, James M. Flegal, and Galin L. Jones. Multivariate output analysis for markov chain monte carlo, 2017.

[21] Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient mcmc for bayesian deep learning. *arXiv preprint arXiv:1902.03932*, 2019.

[22] Yee Whye Teh, Alexandre Thiéry, and Sebastian Vollmer. Consistency and fluctuations for stochastic gradient langevin dynamics, 2015.

[23] Jonathan C. Mattingly, Andrew M. Stuart, and M. V. Tretyakov. Convergence of numerical time-averaging and stationary measures via poisson equations. *SIAM Journal on Numerical Analysis*, 48(2):552–577, Jan 2010.

[24] Changyou Chen, Nan Ding, and Lawrence Carin. On the convergence of stochastic gradient mcmc algorithms with high-order integrators. *NIPS*, 2015.

[25] S. Greitzer. Many cheerful facts. *Arbelos*, 5:14–17, 1986.

[26] Diffusion for global optimisation in rn. *SIAM J. Control Optim.*, pages 737–753, 1987.

[27] Maxim Raginsky, Alexander Rakhlin, and Matus Telgarsky. Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. *arXiv preprint arXiv:1702.03849*, 2017.

[28] Sungjin Ahn, Anoop Korattikara, and Max Welling. Bayesian posterior sampling via stochastic gradient fisher scoring. *arXiv preprint arXiv:1206.6380*, 2012.

[29] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

[30] Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691. PMLR, 2014.