Stochastic Gradient MCMC

Connie Trojan

March 2026

Introduction

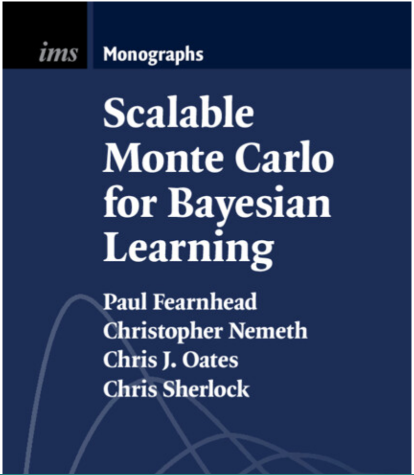
Inference in Bayesian deep learning is challenging due to:

1. the complex and high-dimensional posterior distribution
2. the large dataset size required

Sampling methods like MCMC can perform exact inference without making simplifying approximations to the posterior distribution.

However, since each iteration has linear cost in dataset size, they can quickly become computationally infeasible.

Stochastic gradient MCMC methods aim to reduce this runtime by using a subsample of the data at each iteration.



ims Monographs

Scalable Monte Carlo for Bayesian Learning

Paul Fearnhead
Christopher Nemeth
Chris J. Oates
Chris Sherlock

We will loosely follow the introduction to SGMCMC given in ← this book.

Unadjusted Langevin Algorithm (ULA)

The overdamped Langevin diffusion has π as its stationary distribution:

$$d\theta_t = \frac{1}{2}\nabla \log \pi(\theta_t)\mathrm{d}t + \mathrm{d}W_t$$

We can simulate it approximately with an Euler-Maruyama discretisation:

$$\theta_{t+1} = \theta_t + \frac{\delta}{2}\nabla \log \pi(\theta_t) + \sqrt{\delta}Z, \quad Z \sim N(0,1)$$

For $\delta > 0$, this samples from an approximation to π: choosing δ to be small reduces this bias at the cost of increasing the number of steps required to control variance.

Metropolis-adjusted Langevin Algorithm (MALA)

In MALA, an accept/reject step is used to correct this discretisation error: a sample is proposed from the discretised SDE and accepted with probability

$$a(\theta^*, \theta_t) = \min\left\{1, \frac{\pi(\theta^*)q(\theta_t|\theta^*)}{\pi(\theta_t)q(\theta^*|\theta_t)}\right\}$$

This allows us to sample exactly from π, so we can use larger step sizes without introducing error.
Rules of thumb based on acceptance rate work well to choose a δ that promotes fast convergence to π.

Stochastic Gradient Langevin Dynamics (SGLD)

Standard MCMC is computationally infeasible for large datasets due to the $O(N)$ cost of evaluating

1. the gradient $\nabla \log \pi$ in the proposal
2. the acceptance probability in the Metropolis adjustment

Let $\pi_i(\theta) := p(\theta)^{1/N} \ell(x_i|\theta)$. Since (1) is simply the sum of the $\nabla \log \pi_i$, we can construct an unbiased estimator using a random subsample S_t:

$$\hat{\nabla} \log \pi(\theta_t) = \frac{N}{m} \sum_{i \in S_t} \nabla \log \pi_i(\theta_t)$$

This estimate can be plugged into the ULA update to obtain the **SGLD** algorithm [Welling and Teh, 2011]:

$$\theta_{t+1} = \theta_t + \frac{\delta}{2}\hat{\nabla} \log \pi(\theta_t) + \sqrt{\delta} Z$$

- This is equivalent to performing SGD on $-\log \pi$, with an injection of Gaussian noise at each iteration
- Due to the additional noise in this gradient estimate, SGLD targets a different distribution to ULA

We can analogously formulate subsampling-based versions of other MCMC samplers, e.g. SG-HMC.

Variance reduction

The additional bias introduced by the gradient approximation can be reduced by using estimators with lower variance:

- **Control variates** [Baker et al., 2019]: fix a $\hat{\theta}$, compute the full $\nabla \log \pi(\hat{\theta})$, and estimate the perturbation $\nabla \log \pi(\theta_t) - \nabla \log \pi(\hat{\theta})$ instead

- **Preferential subsampling** [Putcha et al., 2023]: take a weighted sample from X, giving higher weight to x_i with large values of $\|\nabla \log \pi_i(\hat{\theta})\|$

Control variate SGLD

Trivially,

$$\nabla \log \pi(\theta_t) = \nabla \log \pi(\hat{\theta}) + \underbrace{\left[\nabla \log \pi(\theta_t) - \nabla \log \pi(\hat{\theta})\right]}_{(*)} .$$

We can estimate (*) by subsampling to obtain an unbiased estimate of $\nabla \log \pi(\theta_t)$:

$$\nabla \log \pi(\hat{\theta}) + \frac{N}{m} \sum_{x_i \in S_t} \left[\nabla \log \pi_i(\theta_t) - \nabla \log \pi_i(\hat{\theta})\right]$$

This will have lower variance than the simple MC estimator when θ_t is **close to** $\hat{\theta}$, e.g. if $\hat{\theta}$ is at the mode of π.

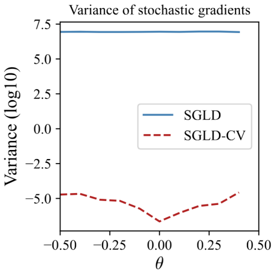


Figure: Taken from Fearnhead et al. [2025]

Step size schedules

For asymptotic consistency with π, SGMCMC requires a step size schedule satisfying the Robbins-Monro criteria:

1. $\sum_{t=1}^{\infty} \delta_t = \infty$
2. $\sum_{t=1}^{\infty} \delta_t^2 < \infty$, i.e. we need $\delta_t \to 0$

In practice a fixed step size is typically used, e.g. the asymptotic variance in Bernstein von-Mises theorem suggests this should scale as $1/N$.

δ must typically chosen to be small in order to control the bias from discretisation and stochastic approximation.

Cyclical schedules

A small step size is problematic for mixing in multimodal posterior distributions, e.g. BNN posteriors (!)

Zhang et al. [2020] propose using a **cyclical** step size schedule, which periodically resets the step size to escape local minima

They alternate between using SGD with a large step size to find a new mode, and running SGMCMC with a decreasing step size to sample locally

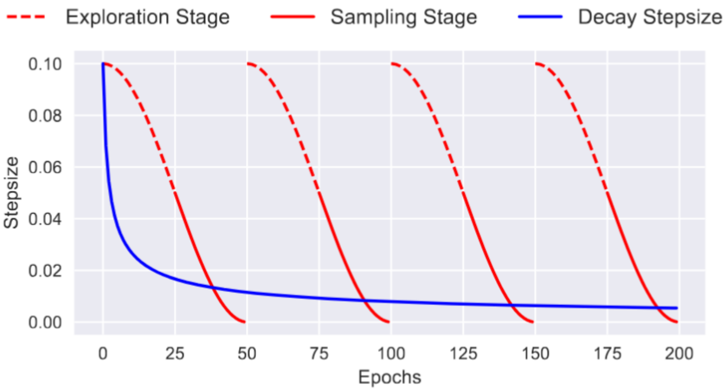
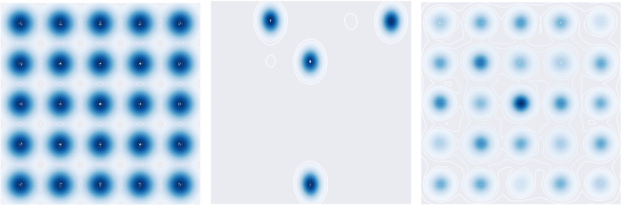


Figure: From Zhang et al. [2020]



(a) Target (b) SGLD (c) cSGLD

Figure: From Zhang et al. [2020]

Does SGMCMC "work"?

While the per-iteration cost of SGLD is constant in dataset size, the step-size is typically set to be much smaller than traditional MCMC, so **more steps are required**.

For vanilla SGLD, the overall computational cost of approximating the posterior to a given degree of accuracy is **still linear** in N [Nagapetyan et al., 2017].
→ you can get samples very quickly but distributional accuracy isn't free.

However, for log-concave posteriors, SGLD-CV does have a sampling cost that is **constant** in dataset size*.
*Though the initial cost of finding $\nabla \log \pi(\hat{\theta})$ is still $O(N)$

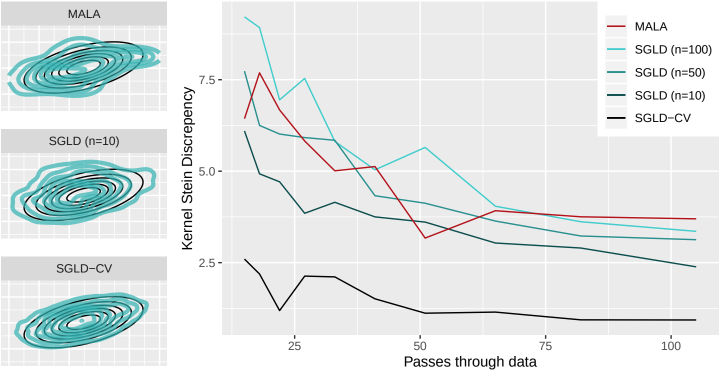


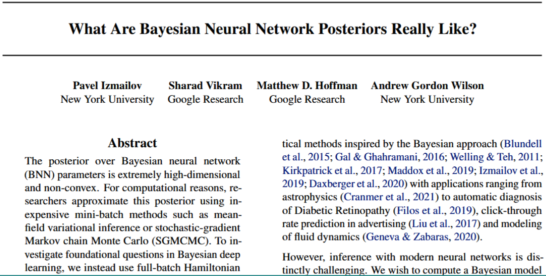
Figure: A graph I was very proud of as an intern

(For a fixed compute budget, MALA and SGLD have similar KSD but SGLD-CV is better.)

BNN posteriors

Performance evaluation on complex posteriors like BNNs typically looks at predictive performance/log-likelihoods for simplicity.

SGMCMC/Bayesian approaches will often outperform optimisation-based approaches like SGD on these metrics.



What Are Bayesian Neural Network Posteriors Really Like?

Pavel Izmailov New York University
Sharad Vikram Google Research
Matthew D. Hoffman Google Research
Andrew Gordon Wilson New York University

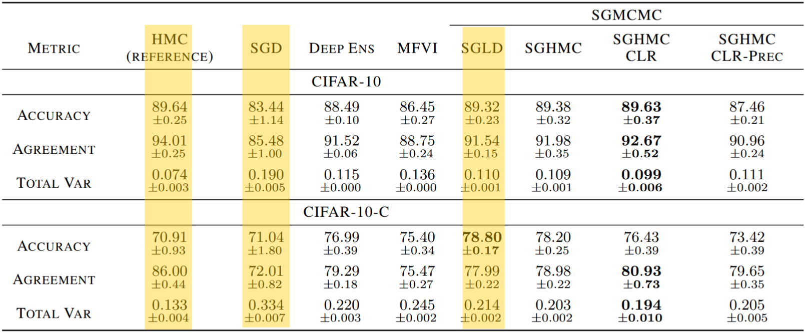
Abstract

The posterior over Bayesian neural network (BNN) parameters is extremely high-dimensional and non-convex. For computational reasons, researchers approximate this posterior using inexpensive mini-batch methods such as mean-field variational inference or stochastic-gradient Markov chain Monte Carlo (SGMCMC). To investigate foundational questions in Bayesian deep learning, we instead use full-batch Hamiltonian

tical methods inspired by the Bayesian approach (Blundell et al., 2015; Gal & Ghahramani, 2016; Welling & Teh, 2011; Kirkpatrick et al., 2017; Maddox et al., 2019; Izmailov et al., 2019; Daxberger et al., 2020) with applications ranging from astrophysics (Cranmer et al., 2021) to automatic diagnosis of Diabetic Retinopathy (Filos et al., 2019), click-through rate prediction in advertising (Liu et al., 2017) and modeling of fluid dynamics (Geneva & Zabaras, 2020).

However, inference with modern neural networks is distinctly challenging. We wish to compute a Bayesian model

← this paper tries to look at **distributional** similarity to HMC “ground truth”



					SGMCMC			
Metric	HMC (Reference)	SGD	Deep Ens	MFVI	SGLD	SGHMC	SGHMC CLR	SGHMC CLR-Prec
			CIFAR-10					
Accuracy	89.64 ±0.25	83.44 ±1.14	88.49 ±0.10	86.45 ±0.27	89.32 ±0.23	89.38 ±0.32	**89.63** **±0.37**	87.46 ±0.21
Agreement	94.01 ±0.25	85.48 ±1.00	91.52 ±0.06	88.75 ±0.24	91.54 ±0.15	91.98 ±0.35	**92.67** **±0.52**	90.96 ±0.24
Total Var	0.074 ±0.003	0.190 ±0.005	0.115 ±0.000	0.136 ±0.000	0.110 ±0.001	0.109 ±0.001	**0.099** **±0.006**	0.111 ±0.002
			CIFAR-10-C					
Accuracy	70.91 ±0.93	71.04 ±1.80	76.99 ±0.39	75.40 ±0.34	**78.80** **±0.17**	78.20 ±0.25	76.43 ±0.39	73.42 ±0.39
Agreement	86.00 ±0.44	72.01 ±0.82	79.29 ±0.18	75.47 ±0.27	77.99 ±0.22	78.98 ±0.22	**80.93** **±0.73**	79.65 ±0.35
Total Var	0.133 ±0.004	0.334 ±0.007	0.220 ±0.003	0.245 ±0.002	0.214 ±0.002	0.203 ±0.002	**0.194** **±0.010**	0.205 ±0.005

Figure: From Izmailov et al. [2021]

(SGMCMC methods have comparable accuracy to HMC, but the predictive distributions are different.)

Summary

- SGMCMC algorithms reduce the per-iteration cost of MCMC by using stochastic gradient estimators

- This makes them computationally feasible to run, however getting **distributional** accuracy is still hard due to the additional variance introduced

- Variance reduction techniques can alleviate this

- SGMCMC produces good results in Bayesian deep learning but doesn't quite match the output of MCMC

References I

J. Baker, P. Fearnhead, E. B. Fox, and C. Nemeth. Control variates for stochastic gradient MCMC. *Statistics and Computing*, 29(3):599–615, 2019. ISSN 0960-3174.

P. Fearnhead, C. Nemeth, C. J. Oates, and C. Sherlock. *Scalable Monte Carlo for Bayesian Learning*. Institute of Mathematical Statistics Monographs. Cambridge University Press, 2025.

P. Izmailov, S. Vikram, M. D. Hoffman, and A. G. G. Wilson. What are Bayesian neural network posteriors really like? In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4629–4640. PMLR, 18–24 Jul 2021.

T. Nagapetyan, A. B. Duncan, L. Hasenclever, S. J. Vollmer, L. Szpruch, and K. Zygalakis. The true cost of stochastic gradient Langevin dynamics. *arXiv preprint 1706.02692*, 2017.

References II

S. Putcha, C. Nemeth, and P. Fearnhead. Preferential subsampling for stochastic gradient Langevin dynamics. In F. Ruiz, J. Dy, and J.-W. van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 8837–8856. PMLR, 25–27 Apr 2023.

M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, ICML'11, page 681–688. Omnipress, 2011.

R. Zhang, C. Li, J. Zhang, C. Chen, and A. G. Wilson. Cyclical stochastic gradient MCMC for Bayesian deep learning. In *International Conference on Learning Representations*, 2020.