Adaptively Weighted Stochastic Gradient MCMC for Global Optimization in Deep Learning

Faming Liang

Bayesian Learning and SGLD

Bayesian statistics is important to machine learning, which captures uncertainty and leads to safe AI.

For big data problems, traditional MCMC algorithms can become extremely slow as they typically require a large number of iterations and a complete scan of the full dataset for each iteration. To tackle this difficulty, a variety of scalable algorithms have been developed under the umbrella of subsampling:

- Stochastic gradient MCMC
- split-and-merge
- mini-batch Metropolis-Hastings algorithms
- nonreversible Markov process

Stochastic gradient Langevin dynamics (SGLD)

- Let $\boldsymbol{X}_N = (X_1, X_2, \ldots, X_N)$ denote a set of N independent and identically distributed (iid) observations.
- Let $L(\boldsymbol{\theta}|\boldsymbol{X}_N) = \sum_{i=1}^{N} \log f(x_i|\boldsymbol{\theta}) + \log \pi(\boldsymbol{\theta})$ denote the log-posterior density of a statistical model parameterized by $\boldsymbol{\theta}$, where $f(x_i|\boldsymbol{\theta})$ is the density function of X_i, and $\pi(\boldsymbol{\theta})$ is the prior density of $\boldsymbol{\theta}$.

The SGLD algorithm iterates via the equation

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \frac{\epsilon_{t+1}}{2} \partial_{\boldsymbol{\theta}} \widehat{L}(\boldsymbol{\theta}_t|\boldsymbol{X}_N) + \sqrt{\epsilon_{t+1}\tau}\eta_{t+1}, \quad \eta_{t+1} \sim N(0, I_d), \tag{1}$$

where d is the dimension of $\boldsymbol{\theta}$, I_d is a $d \times d$-identity matrix, ϵ_{t+1} is the step size (also known as the learning rate), τ is the temperature, and $\hat{L}(\boldsymbol{\theta}|\boldsymbol{X}_N)$ denotes an estimate of $L(\boldsymbol{\theta}|\boldsymbol{X}_N)$.

Stochastic Gradient MCMC (SGMCMC)

- Stochastic gradient Langevin dynamics (SGLD)
- stochastic gradient Hamiltonian Monte Carlo
- stochastic gradient thermostats
- stochastic gradient Fisher scoring
- preconditioned SGLD

Refer to Ma et al. (2015) for a complete framework of these algorithms.

Attractive Features of SGLD

- Their transitional kernel is defined by a stochastic differential equation with the moving direction guided by the gradient of the log-target density function.
- Only an inaccurate gradient of the log-target density function, which can be evaluated on a mini-batch of data, is required for each transition.

These two distinctive features make SGMCMC algorithms highly scalable, exhibiting similar costs as many stochastic optimization algorithms.

Difficulties with SGLD

- For convex energy functions, SGLD converges to the stationary distribution in polynomial time.
- For non-convex energy functions, however, an exponentially long mixing time is unavoidable in general. According to Bonis (2016), it takes the Langevin diffusion at least $\exp(\Omega(h/\tau))$ steps to escape a depth-h basin of attraction.

How to overcome the local-trap problem is crucial to the success of Bayesian learning.

Strategies used to Improve SGMCMC

- Auxiliary variable: using auxiliary variables to increase dynamics of the simulation. For example, SGHMC introduces a moment term and SGT introduces further another auxiliary variable.
- Geometry information of the manifold: using the geometry information of the target distribution to adapt the dynamics of the simulation. For example, SGRLD, SGFS and pSGLD use the fisher information metric or its estimator.

When the learning rate ϵ_t becomes very small after a sufficient burn-in period, the trajectory of the samples can be local. Therefore, it is very difficult for the existing SGMCMC algorithms to traverse between different local energy minima even with above two strategies.

Adaptively Weighted SGLD

AWSGLD employs a new strategy to resolve the local trap problem, which is to adaptively update the gradient used in (1) to facilitate escaping from local traps.

- AWSGLD will converge to a distribution (denoted by $\tilde{\pi}(\boldsymbol{x}|\boldsymbol{D})$) different from the target distribution $\pi(\boldsymbol{x}|\boldsymbol{D})$, while all the existing variants will converge to $\pi(\boldsymbol{x}|\boldsymbol{D})$.
- Compared to $\pi(\boldsymbol{x}|\boldsymbol{D})$, $\tilde{\pi}(\boldsymbol{x}|\boldsymbol{D})$ biases sampling toward low energy regions and thus facilitates optimization of the energy function.
- AWSGLD can be incorporated into the existing variants of SGLD. For example, with the use of auxiliary variables, we can have new algorithms such as adaptively weighted SGHMC or adaptively weight SGT; with the use of the fisher information metric, we can have a new algorithm adaptively weighted SGRLD.

AWSGLD: Idea

- Suppose that we are interested in sampling from a posterior density function

$$\pi(\boldsymbol{x}|\boldsymbol{D}) \propto \exp(-U(\boldsymbol{x}|\boldsymbol{D})/\tau), \tag{2}$$

where $\boldsymbol{D}$ denotes the observed data, τ denotes the temperature, and $U(\boldsymbol{x}|\boldsymbol{D})$ is the energy function. Let $g(u)$ denote the marginal density function of energy, which can be defined through $g(u) = dG(u)/du$ and

$$G(u) = \int_{\{\boldsymbol{x}:U(\boldsymbol{x}|\boldsymbol{D})\leq u\}} \pi(\boldsymbol{x}|\boldsymbol{D})d\boldsymbol{x}. \tag{3}$$

AWSGLD: Idea

- We propose to simulate from a weighted density function

$$\tilde{\pi}(\boldsymbol{x}|\boldsymbol{D}) \propto \frac{\pi(\boldsymbol{x}|\boldsymbol{D})}{\theta(U(\boldsymbol{x}))^{\zeta}}, \tag{4}$$

where ζ is a positive constant, and $\theta(u)$ is a strictly increasing function of u. Sampling from (4) leads to the following marginal density function of energy

$$P(u) \propto \frac{g(u)}{\theta^{\zeta}(u)},$$

which biases sampling toward low energy subregions.

AWSGLD: Setup

- A straightforward calculation shows

$$\nabla_{\boldsymbol{x}} \log \tilde{\pi}(\boldsymbol{x}|\boldsymbol{D}) = -\left[1 + \zeta \frac{\partial \theta(u)/\partial u}{\theta(u)}\right] \nabla_{\boldsymbol{x}} U(\boldsymbol{x}|\boldsymbol{D}). \quad (5)$$

- partitioning the sample space according to the energy function into m disjoint subregions: $E_1 = \{\boldsymbol{x} : U(\boldsymbol{x}|\boldsymbol{D}) \leq u_1\}$, $E_2 = \{\boldsymbol{x} : u_1 < U(\boldsymbol{x}|\boldsymbol{D}) \leq u_2\}$, ..., $E_{m-1} = \{\boldsymbol{x} : u_{m-2} < U(\boldsymbol{x}|\boldsymbol{D}) \leq u_{m-1}\}$, and $E_m = \{\boldsymbol{x} : U(\boldsymbol{x}|\boldsymbol{D}) > u_{m-1}\}$, where $-\infty < u_1 < u_2 < \cdots < u_m < \infty$ are specified by the user.
- Let $\{\epsilon_t : t = 1, 2, \ldots\}$ and $\{\gamma_t : t = 1, 2, \ldots\}$ denote two positive, non-increasing sequence satisfying certain conditions as given in stochastic approximation.

AWSGLD: Algorithm

1. (Data subsampling) Draw a subsample of size n, denoted by $\boldsymbol{D}_t$, from the dataset $\boldsymbol{D}$ which contains N iid samples.
2. (SGLD sampling) Draw $\boldsymbol{x}_{t+1}$ using the SGLD algorithm based on the current sample $\boldsymbol{x}_t$, i.e.,

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \epsilon_{t+1}\frac{N}{n}\left[1 + \frac{\zeta}{\Delta u}\left(1 - \frac{\hat{\theta}_{t,J(\boldsymbol{x}_t)-1}}{\hat{\theta}_{t,J(\boldsymbol{x}_t)}}\right)\right] \times \nabla_{\boldsymbol{x}} U(\boldsymbol{x}_t|\boldsymbol{D}_t) + \sqrt{2\epsilon_{t+1}}\eta_{t+1}, \tag{6}$$

   where $\eta_{t+1} \sim N(0, I_d)$, d denotes the dimension of $\boldsymbol{x}$, and ϵ_{t+1} is the learning rate.
3. (Adaptive parameter updating) Update the estimate of $\theta(u_i)$'s for $i = 1, 2, \ldots, m$ by setting

$$\hat{\theta}_{t+1,i} = \begin{cases} (1-\gamma_{t+1})\hat{\theta}_{t,i}, & i < J(\boldsymbol{x}_{t+1}), \\ (1-\gamma_{t+1})\hat{\theta}_{t,i} + \gamma_{t+1}\hat{\theta}_{t,J(\boldsymbol{x}_{t+1})}, & i \geq J(\boldsymbol{x}_{t+1}). \end{cases}$$

AWSGLD: remarks

- The estimate of $\theta(u)$ changes from iteration to iteration, AWSGLD is an adaptive SGMCMC algorithm.
- AWSGLD possesses a *self-adjusting mechanism* for escaping local traps: If it falls into a local energy minimum, the ratio $\hat{\theta}_{t,J(\boldsymbol{x}_t)-1}/\hat{\theta}_{t,J(\boldsymbol{x}_t)}$ will gradually decrease in the followed iterations as implied by step 3 of the algorithm, rendering a larger gradient multiplier to facilitate escaping from the local trap.
- A large value of ζ enhances the ability of AWSGLD to escape from local traps, especially in the early period of the simulation. In practice, the value of ζ can be gradually increased to a constant via an adaptive mechanism.

AWSGLD vs SGLD

Compared to SGLD, AWSGLD allows a higher temperature to be used in simulations due to the inclusion of the gradient multiplier. This is very important for global optimization: AWSGLD can flatten the energy landscape using a high temperature and thus reduce the risk of getting local traps, while making the sampling more focused on low energy regions via an importance sampling mechanism. As a result, this shortens its hitting time to the low energy set.

Adaptive SGLD: General algorithm

1. (SGLD sampling) Simulate $\boldsymbol{x}_{t+1}$ using SGLD based on the current parameter estimate $\boldsymbol{\theta}_t$, i.e.

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \epsilon_{t+1} \frac{\partial}{\partial \boldsymbol{x}} \tilde{L}(\boldsymbol{x}_t, \boldsymbol{\theta}_t) + \sqrt{2\epsilon_{t+1}}\eta_{t+1},$$

   where $\eta_{t+1} \sim \mathcal{N}(0, \boldsymbol{I})$.

2. (Parameter Update) Update the parameter $\boldsymbol{\theta}_t$ according to the recursion

$$\begin{aligned} \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t + \gamma_{t+1} \left(\xi(\boldsymbol{\theta}_t, \boldsymbol{x}_{t+1}) - \boldsymbol{\theta}_t \right) \\ &= (1 - \gamma_{t+1})\boldsymbol{\theta}_t + \gamma_{t+1}\xi(\boldsymbol{\theta}_t, \boldsymbol{x}_{t+1}), \end{aligned} \tag{7}$$

   where $\xi(\cdot, \cdot)$ is a mapping to derive to the optimal parameter value $\boldsymbol{\theta}$ based on the samples $\boldsymbol{x}_t$'s.

Adaptive SGLD: Convergence

Theorem 1 [L^2 convergence rate] For the ASGLD algorithm, if $\epsilon \in (0, \mathrm{Re}(\frac{m-\sqrt{m^2-2M(M+1)}}{2M(M+1)}))$, then there exists a constant λ such that $E\left[\|\boldsymbol{\theta}_t - \boldsymbol{\theta}_*\|^2\right] \leq \lambda\gamma_k$.

Adaptive SGLD: Convergence

Theorem 2 [Ergodicity and WLLN] For decaying $\epsilon \in (0, \mathrm{Re}(\frac{m-\sqrt{m^2-2M(M+1)}}{2M(M+1)}))$, ASGLD is ergodic. In addition, for any bounded measurable function $\rho(\cdot)$, $\frac{1}{T}\sum_{t=1}^{T}\rho(\boldsymbol{X}_t) \to \pi_{\boldsymbol{\theta}_*}(\rho)$ in probability as $T \to \infty$, where $\pi_{\boldsymbol{\theta}_*}(\rho) = \int_{\mathcal{X}} \rho(\boldsymbol{x})\pi_{\boldsymbol{\theta}_*}(d\boldsymbol{x})$.

AWSGLD: Convergence

AWSGLD is a special case of ASGLD, for which $\boldsymbol{\theta}_*$ can be determined by solving the equation

$$\boldsymbol{\theta}_*(u) = \int \frac{e^{-U(x)}}{\boldsymbol{\theta}_*^{\zeta}(u)} \boldsymbol{\theta}_*(u) 1_{\{u \geq U(x)\}} dx = \frac{G(u)}{\boldsymbol{\theta}_*^{\zeta-1}(u)}. \tag{8}$$

That is,

$$\boldsymbol{\theta}_*(u) = (G(u))^{1/\zeta}.$$

Furthermore, for any ζ, we have

$$\zeta \frac{\partial \boldsymbol{\theta}(u)/\partial u}{\boldsymbol{\theta}(u)} = \frac{g(u)}{G(u)}. \tag{9}$$

The choice of ζ will not affect much the performance of the algorithm. However, a large value of ζ does enhance the convergence of the algorithm toward a lower energy region in the early period of the simulation.

AWSGLD: Hitting time

Theorem 3 [Upper bound of hitting time] For AWSGLD, the expected number of iterations required to hit the optimal set $K_0^\rho := \{x : d(x, K_0) \leq \rho\}$ is $\frac{m\tau_{\text{aw-sgld}}}{1-2\delta}$ for any $\rho \lesssim \Delta u$ and $\delta < 0.5$, where $K_0 = \{\boldsymbol{x} : U(\boldsymbol{x}) \leq \min_{\boldsymbol{x}} U(\boldsymbol{x}) + \varepsilon_0\}$ for any $\varepsilon_0 > 0$, $d(\cdot, \cdot)$ denotes a distance metric, and $\tau_{\text{aw-sgld}}$ is a constant factor polynomially dependent on the dimension d.

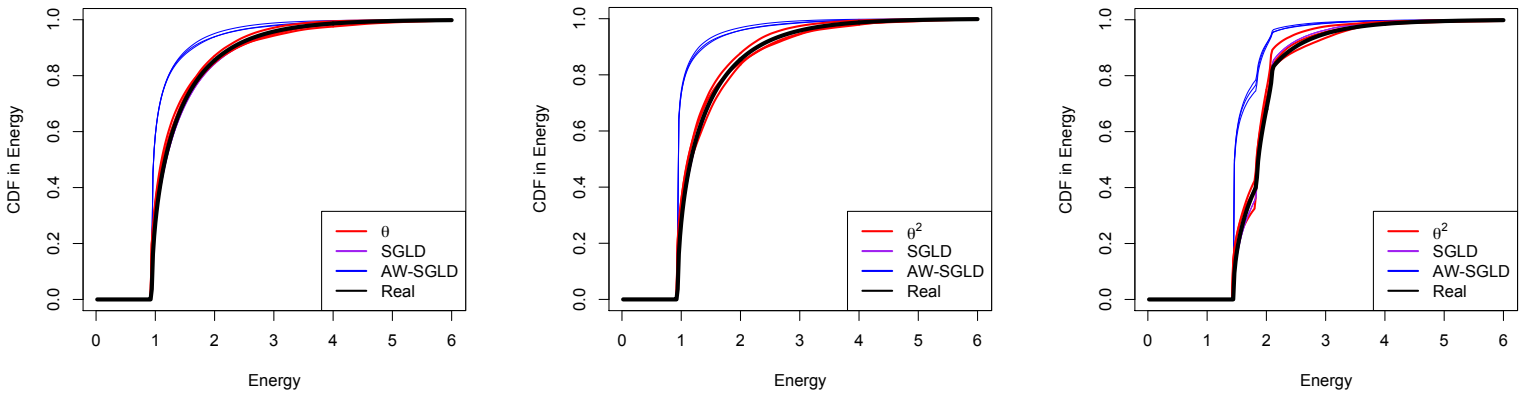
Table: Expected hitting time of SGLD, SA-SGLD and AWSGLD.

Method	Expected hitting time
SGLD	$\mathcal{O}(\tau_{\text{sgld}})e^{\mathcal{O}(\Xi)}/(1-\delta)$
SA-SGLD	$\mathcal{O}(\tau_{\text{sa-sgld}})\tilde{m}/(1-6\delta)^{\tilde{m}}$
AWSGLD	$\mathcal{O}(\tau_{\text{aw-sgld}})m/(1-2\delta)$

Estimation of CDF in Energy

- Standard Gaussian distribution, i.e., $\pi(\boldsymbol{x}|\boldsymbol{D}) = N(0,1)$.
- Mixture Gaussian distribution $\frac{2}{5}N(-2,1) + \frac{3}{5}N(1,1)$, which represents a multimodal distribution.

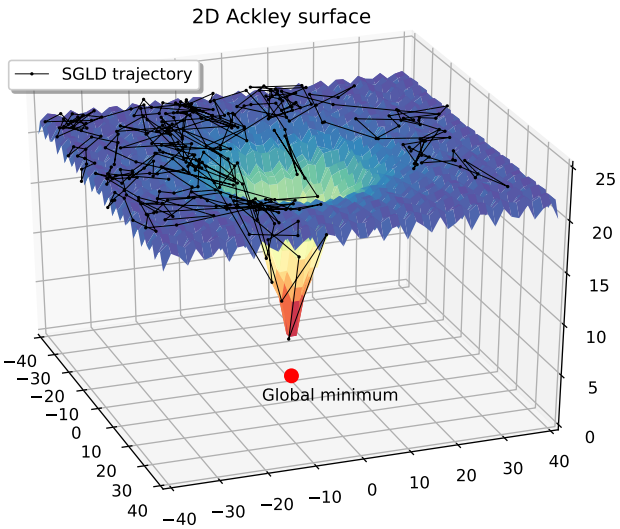
Estimation of CDF in Energy



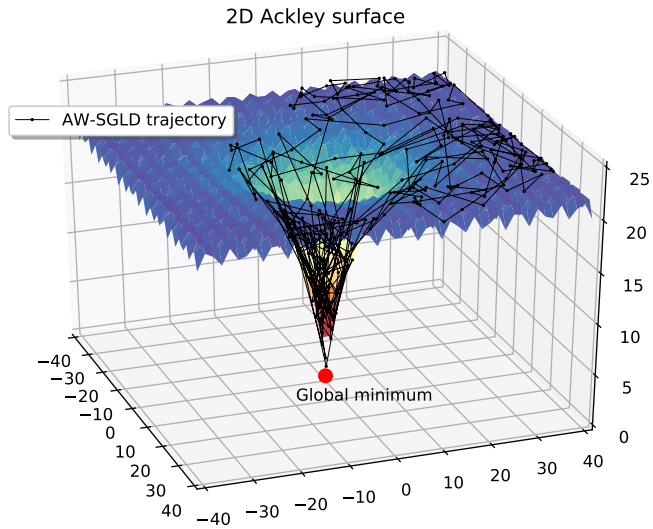
(a) Standard Gaussian $\zeta = 1$. (b) Standard Gaussian $\zeta = 2$. (c) Gaussian Mixture $\zeta = 2$.

Figure: Results of AWSGLD where the lines are annotated as follows: black: true function $G(u)$; red: AWSGLD estimate of $G(u)$; blue: empirical CDF of AWSGLD samples; and purple: empirical CDF of SGLD samples.

2D Ackley Function



(a) SGLD trajectory.



(b) AWSGLD trajectory.

Figure: Trajectories of AWSGLD and SGLD on 2D Ackley surface.

Multimodal Sampling

The second experiment is to show that AWSGLD has the capability to traverse over a rugged energy landscape with many local energy minima. The target density function is given by $\pi(\boldsymbol{x}) \propto \exp(-U(\boldsymbol{x})/\tau)$, where $\tau = 20$ and $U(\boldsymbol{x})$ is given by

$$\begin{aligned} U(\boldsymbol{x}) = &-\{x_1 \sin(20x_2) + x_2 \sin(20x_1)\}^2 \cosh(\sin(10x_1)x_1) \\ &- \{x_1 \cos(10x_2) - x_2 \sin(10x_1)\}^2 \cosh(\cos(20x_2)x_2), \end{aligned}$$

which has been used in multiple papers as a multimodal example.

Multimodal Sampling

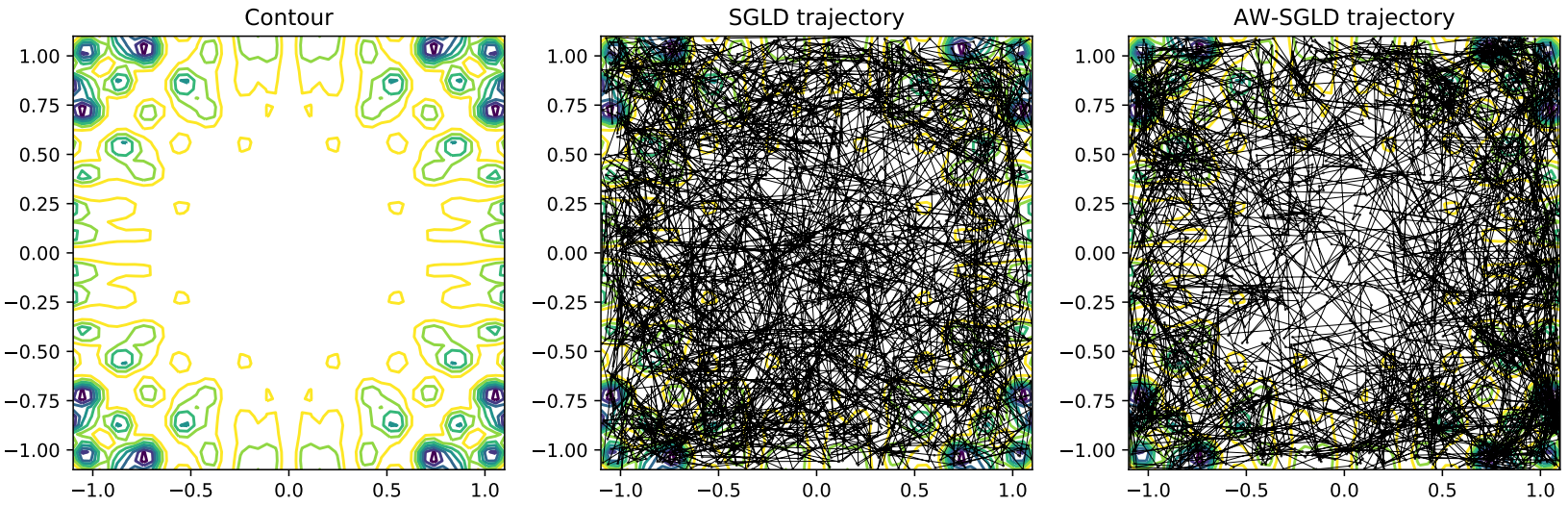


Figure: Sample paths of SGLD and AWSGLD on a rugged energy landscape.

Multimodal Sampling

Table: Average iteration numbers required by SGLD and AWSGLD to locate a global energy minimum for 10 benchmark functions, where AW-10 is short for AWSGLD with 10 partitions and each test was repeated 40 times

No	Function \ Iters $\times 10^3$	Dim	SGLD	AW-10	AW-100	AW-1000
1	Rastrigin (RT_{20})	20	8.7±0.8	16.8±1.3	9.4±1	7.1±0.6
2	Griewank (G_{20})	20	467.8±13	**4.3±1**	5.3±0.2	13.7±0.7
3	Sum Squares (SS_{20})	20	∞	81.4±6	78.3±6	96.9±3
4	Rosenbrock (R_{20})	20	61.0±10	**19.9±5**	21.6±6	26.3±6
5	Zakharov (Z_{20})	20	66.8±20	10.0±5	**5.8±4**	10.1±5
6	Powell (PW_{24})	24	98.4±1	34.8±4	**20.0±3**	51.7±6
7	Dixon&Price (DP_{25})	25	32.6±3	23.1±3	**20.0±3**	23.0±3
8	Levy (L_{30})	30	23.3±2	**0.1±0.004**	0.4±0.01	1.0±0.04
9	Sphere (SR_{30})	30	0.5±0.02	**0.1±0.001**	**0.1±0.001**	0.2±0.001
10	Ackley (AK_{30})	30	11.7±2	16.1±6	**3.4±0.4**	7.4±1

Landset Data

AWSGLD was applied to train a deep neural network (DNN) for the dataset *Landset* which is available at the UCI Machine Learning Repository.

- The dataset consists of 4435 training instances and 2000 testing instances
- Each instance consists of 36 attributes representing features of an image of the earth surface taken from a satellite.
- The training instances consist of 6 classes.
- We trained a DNN with structure 36-30-30-6 and the activation function *tanh*.

Landset Data

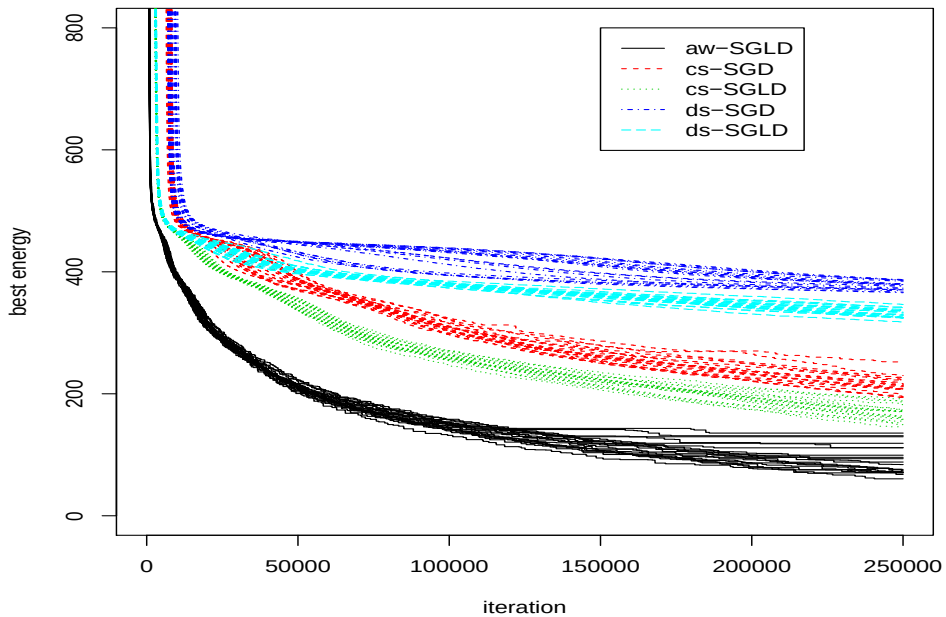


Figure: Sample paths of AWSGLD (aw-SGLD, black), constant stepsize SGD (cs-SGD, red), constant stepsize SGLD (cs-SGLD, green), decreasing stepsize SGD (ds-SGD, blue), decreasing stepsize SGLD (ds-SGLD, light blue) in training a DNN for Landset data.

Landset Data

Table: Training and prediction errors produced by AWSGLD, constant stepsize SGD (csSGD), constant stepsize SGLD (csSGLD), decreasing stepsize SGD (dsSGD), decreasing stepsize SGLD (dsSGLD) for the Landsat data.

		SGLD		SGD	
Method	AWSGLD	cs	ds	cs	ds
fit(%)	**1.87**	4.08	9.76	5.68	11.79
pred(%)	**7.65**	9.00	12.35	9.90	14.60

Computer Vision Data

- **CIFAR10** is a benchmark dataset with 10 classes. It contains 50,000 images for training and 10,000 RGB images for testing.
- **CIFAR100** dataset has 100 classes and each class consists of 500 training images.

Computer Vision Data

Table: Experiments on CIFAR10 & 100 datasets using Resnet20 models, where ϵ denotes the learning rate and τ the temperature. Each test is repeated for 10 times. The bold color means significant improvement.

Strategy	Dataset Criteria	CIFAR10 One Sample	CIFAR10 BMA	CIFAR100 One Sample	CIFAR100 BMA
Fixed ϵ Fixed τ	M-SGD	90.10 $\pm$ 0.06	92.79 $\pm$ 0.04	61.19 $\pm$ 0.08	68.03 $\pm$ 0.07
	SGHMC	89.73 $\pm$ 0.05	92.51 $\pm$ 0.03	61.72 $\pm$ 0.11	68.05 $\pm$ 0.08
	AWSGHMC	**91.20 $\pm$ 0.09**	**93.27 $\pm$ 0.03**	**65.20 $\pm$ 0.34**	**70.46 $\pm$ 0.08**
Decay ϵ_t Fixed τ	M-SGD	**93.14 $\pm$ 0.05**	93.50 $\pm$ 0.04	67.12 $\pm$ 0.26	70.14 $\pm$ 0.06
	SGHMC	90.79 $\pm$ 0.04	93.07 $\pm$ 0.07	64.65 $\pm$ 0.09	69.85 $\pm$ 0.09
	AWSGHMC	92.52 $\pm$ 0.04	93.66 $\pm$ 0.03	**68.74 $\pm$ 0.16**	**71.74 $\pm$ 0.08**
Decay ϵ_t Decay τ_t	M-SGD	93.16 $\pm$ 0.04	93.49 $\pm$ 0.05	67.60 $\pm$ 0.11	70.31 $\pm$ 0.11
	SGHMC	93.26 $\pm$ 0.04	93.58 $\pm$ 0.03	67.45 $\pm$ 0.14	70.22 $\pm$ 0.09
	AWSGHMC	**93.85 $\pm$ 0.04**	**94.10 $\pm$ 0.04**	**70.65 $\pm$ 0.09**	**72.04 $\pm$ 0.08**

Conclusion

- We have proposed the AWSGLD algorithm as a general Monte Carlo algorithm for Bayesian learning.
- We established the convergence theory and hitting time upper bound for AWSGLD, and tested its performance on multiple benchmark examples.
- The comparison with SGLD shows its superiority in stochastic optimization and Bayesian learning.
- The efficiency of AWSGLD can be further improved with the SGLD step replaced by its variants such as SGHMC, SGT or SGRLD.