



# Control variates for stochastic gradient MCMC

Jack Baker<sup>1</sup> · Paul Fearnhead<sup>1</sup> · Emily B. Fox<sup>2</sup> · Christopher Nemeth<sup>1</sup>

Received: 16 June 2017 / Accepted: 31 July 2018 / Published online: 4 August 2018  
© The Author(s) 2018

## Abstract

It is well known that Markov chain Monte Carlo (MCMC) methods scale poorly with dataset size. A popular class of methods for solving this issue is stochastic gradient MCMC (SGMCMC). These methods use a noisy estimate of the gradient of the log-posterior, which reduces the per iteration computational cost of the algorithm. Despite this, there are a number of results suggesting that stochastic gradient Langevin dynamics (SGLD), probably the most popular of these methods, still has computational cost proportional to the dataset size. We suggest an alternative log-posterior gradient estimate for stochastic gradient MCMC which uses control variates to reduce the variance. We analyse SGLD using this gradient estimate, and show that, under log-concavity assumptions on the target distribution, the computational cost required for a given level of accuracy is independent of the dataset size. Next, we show that a different control-variate technique, known as zero variance control variates, can be applied to SGMCMC algorithms for free. This postprocessing step improves the inference of the algorithm by reducing the variance of the MCMC output. Zero variance control variates rely on the gradient of the log-posterior; we explore how the variance reduction is affected by replacing this with the noisy gradient estimate calculated by SGMCMC.

**Keywords** Stochastic gradient MCMC · Langevin dynamics · Scalable MCMC · Control variates · Computational cost · Big data

## 1 Introduction

Markov chain Monte Carlo (MCMC), one of the most popular methods for Bayesian inference, scales poorly with dataset size. This is because standard methods require the whole dataset to be evaluated at each iteration. Stochastic gradient MCMC (SGMCMC) are a class of MCMC algorithms that aim to account for this issue. The algorithms have recently gained popularity in the machine learning literature. These methods use efficient MCMC proposals based on discretised dynamics that use gradients of the log-posterior. They reduce the computational cost by replacing the gradients with an unbiased estimate which uses only a subset of the data, referred to as a minibatch. They also bypass the acceptance step by making small discretisation steps (Welling and

Teh 2011; Chen et al. 2014; Ding et al. 2014; Dubey et al. 2016).

These new algorithms have been successfully applied to a range of state of the art machine learning problems (e.g. Patterson and Teh 2013; Wenzhe Li 2016). There is now software available to implement these methods (Tran et al. 2016; Baker et al. 2017). In particular, Baker et al. (2017) implements the control-variate methodology we discuss in this article.

This paper investigates stochastic gradient Langevin dynamics (SGLD), a popular SGMCMC algorithm that discretises the Langevin diffusion. There are a number of results suggesting that whilst SGLD has a lower per iteration computational cost compared with MCMC, its overall computational cost is proportional to the dataset size (Welling and Teh 2011; Nagapetyan et al. 2017). This motivates improving the computational cost of SGLD, which can be done by using control variates (Ripley 2009). Control variates can be applied to reduce the Monte Carlo variance of the gradient estimate in stochastic gradient MCMC algorithms. We refer to SGLD using this new control-variate gradient estimate as SGLD-CV.

We analyse the algorithm using the Wasserstein distance between the distribution defined by SGLD-CV and the true

---

✉ Jack Baker  
j.baker1@lancaster.ac.uk

<sup>1</sup> STOR-i Centre for Doctoral Training, Department of Mathematics and Statistics, Lancaster University, Lancaster, UK

<sup>2</sup> Department of Statistics, University of Washington, Seattle, WA, USA

posterior distribution, by adapting recent results by Dalalyan and Karagulyan (2017). This is quite a strong assumption, but has become common for the analysis of these methods (see e.g. Durmus and Moulines 2016; Dalalyan and Karagulyan 2017; Nagapetyan et al. 2017) since it ensures that errors do not accumulate from one iteration to the next. We provide an empirical comparison on a wide variety of models that do not necessarily satisfy these conditions in order to fully explore the results. If the strongly log-concave condition does not hold, then provided the posterior contracts with the number of observations, there should still be some benefit from the variance reduction but we leave a full analysis in the nonconvex case for future work.

We get bounds on the Wasserstein distance between the target distribution and the distribution we sample from at a given step of SGLD-CV. These bounds are in terms of the tuning constants chosen when implementing SGLD-CV. Under assumptions on how the posterior changes as we get more data, we are able to show that, after an initialisation step, the cost of running SGLD-CV to a required level of accuracy does not grow with the number of data points. The initialisation step involves finding a centring value  $\hat{\theta}$  using optimisation and evaluating the gradient of the log-posterior at this value. Both these are  $O(N)$  calculations, where  $N$  is the dataset size, but we show in Sect. 3.4 these each require just a single pass through the dataset. We also suggest starting the algorithm from the centring value, essentially replacing the burn-in step of SGLD with the optimisation step. The experiments show this optimisation step is often more efficient than the burn-in step of SGLD. Our results in Sect. 3.3 quantifies the impact on performance of obtaining a poor centring value.

The use of control variates has also been shown to be important for other Monte Carlo algorithms for simulating from a posterior with a cost that is sublinear in the number of data points (Bardenet et al. 2017; Bierkens et al. 2016; Pollock et al. 2016; Nagapetyan et al. 2017). For previous work that suggests using control variates within SGLD, see Dubey et al. (2016) and Chen et al. (2017). These latter papers, whilst showing benefits of using control variates, do not show that the resulting algorithm can have sublinear cost in the number of data points. A recent paper, Nagapetyan et al. (2017), does investigate how SGLD-CV performs in the data limit under similar log-concavity assumptions on the posterior distribution. They have results that are qualitatively similar to ours, including the sublinear computational cost of SGLD-CV. Though they measure accuracy of the algorithm through the mean squared error of Monte Carlo averages rather than through the Wasserstein distance.

Not only can control variates be used to speed up stochastic gradient MCMC by enabling smaller minibatches to be used; we show also that they can be used to improve the inferences made from the MCMC output. In particular, we can use post-

processing control variates (Mira et al. 2013; Papamarkou et al. 2014; Friel et al. 2016) to produce MCMC samples with a reduced variance. The postprocessing methods rely on the MCMC output as well as gradient information. Since stochastic gradient MCMC methods already compute estimates of the gradient, we explore replacing the true gradient in the postprocessing step with these free estimates. We also show theoretically how this affects the variance reduction factor, and empirically demonstrate the variance reduction that can be achieved by using these postprocessing methods.

## 2 Stochastic gradient MCMC

Throughout this paper, we aim to make inference on a vector of parameters  $\theta \in \mathbb{R}^d$ , with data  $\mathbf{x} = \{x_i\}_{i=1}^N$ . We denote the probability density of  $x_i$  as  $p(x_i|\theta)$  and assign a prior density  $p(\theta)$ . The resulting posterior density is then  $p(\theta|\mathbf{x}) \propto p(\theta) \prod_{i=1}^N p(x_i|\theta)$ , which defines the posterior distribution  $\pi$ . For brevity we write  $f_i(\theta) = -\log p(x_i|\theta)$  for  $i = 1, \dots, N$ ,  $f_0(\theta) = -\log p(\theta)$  and  $f(\theta) = -\log p(\theta|\mathbf{x})$ .

Many MCMC algorithms are based upon discrete-time approximations to continuous-time dynamics, such as the Langevin diffusion, that are known to have the posterior as their invariant distribution. The approximate discrete-time dynamics are then used as a proposal distribution within a Metropolis–Hastings algorithm. The accept-reject step within such an algorithm corrects for any errors in the discrete-time dynamics. Examples of such an approach include the Metropolis-adjusted Langevin algorithm (MALA; see e.g. Roberts and Tweedie 1996) and Hamiltonian Monte Carlo (HMC; see e.g. Neal 2010).

### 2.1 Stochastic gradient Langevin dynamics

SGLD, first introduced by Welling and Teh (2011), is a mini-batch version of the unadjusted Langevin algorithm (Roberts and Tweedie 1996). At each iteration, it creates an approximation of the true gradient of the log-posterior by using a small sample of data.

The SGLD algorithm is based upon the discretisation of a stochastic differential equation known as the Langevin diffusion. A Langevin diffusion for a parameter vector  $\theta$  with posterior  $p(\theta|\mathbf{x}) \propto \exp(-f(\theta))$  is given by

$$\theta_t = \theta_0 - \int_0^t \nabla f(\theta_s) ds + \sqrt{2d} B_t, \quad (1)$$

where  $B_t$  is a  $d$ -dimensional Wiener process. The stationary distribution of this diffusion is  $\pi$ . This means that it will target the posterior exactly, but in practice we need to discretise the dynamics to simulate from it, which introduces error. A bottleneck for this simulation is that calculating  $\nabla f(\theta)$

is an  $O(N)$  operation. To get around this, Welling and Teh (2011) replace the log-posterior gradient with the following unbiased estimate

$$\nabla \hat{f}(\theta) := \nabla f_0(\theta) + \frac{N}{n} \sum_{i \in S_k} \nabla f_i(\theta) \tag{2}$$

for some subsample  $S_k$  of  $\{1, \dots, N\}$ , with  $|S_k| = n$  representing the minibatch size. A single update of SGLD is then

$$\theta_{k+1} = \theta_k - \frac{h_k}{2} \nabla \hat{f}(\theta_k) + \zeta_k, \tag{3}$$

where  $\zeta_k \sim N(0, h_k)$ .

MALA uses a Metropolis–Hastings accept-reject step to correct for the discretisation of the Langevin process. Welling and Teh (2011) bypass this acceptance step, as it requires calculating  $p(\theta|\mathbf{x})$  using the full dataset, and instead use an adaptive rather than fixed stepsize, where  $h_k \rightarrow 0$  as  $k \rightarrow \infty$ . The motivation is that the noise in the gradient estimate disappears faster than the process noise, so eventually, the algorithm will sample the posterior approximately. In practice, we found the algorithm does not mix well when the stepsize is decreased to zero, so in general a fixed small stepsize  $h$  is used in practice, as suggested by Vollmer and Zygalakis (2016).

### 3 Control variates for SGLD efficiency

The SGLD algorithm has a reduced per iteration computational cost compared to traditional MCMC algorithms. However, there have been a number of results suggesting that the overall computational cost of SGLD is still  $O(N)$  (Welling and Teh 2011; Nagapetyan et al. 2017). What we mean by this is that in order for the algorithm to reach a desired distance from the true posterior, the minibatch size  $n$ , and the total number of iterations  $K$ , need to be set so that the computational cost  $Kn$  is  $O(N)$ . The main reason for this result is that in order to control the variance in the gradient estimate,  $\nabla \hat{f}(\theta)$ , we need  $n$  to increase linearly with  $N$ . Therefore, we would assume that reducing this variance would lead to an improved computational cost of the algorithm. A natural choice is to reduce this variance through control variates (Ripley 2009).

Control variates applied to SGLD have also been investigated by Dubey et al. (2016) and Chen et al. (2017), who show that the convergence bound of SGLD is reduced when they are used. Theoretical results, similar and independent to ours, show how control variates can improve the computational cost of SGLD (Nagapetyan et al. 2017).

In Sect. 3.1, we show how control variates can be used to reduce the variance in the gradient estimate in SGLD, leading to the algorithm SGLD-CV. Then, in Sect. 3.3, we analyse

the Wasserstein distance between the distribution defined by SGLD-CV and the true posterior. There are a number of quantities that affect the performance of SGLD-CV, including the stepsize  $h$ , the number of iterations  $K$  and the minibatch size  $n$ . We provide sufficient conditions on  $h$ ,  $K$  and  $n$  in order to bound the Wasserstein distance. We show under certain assumptions, the computational cost, measured as  $Kn$ , required to bound the Wasserstein distance is independent of  $N$ .

### 3.1 Control variates for SGMCMC

Let  $\hat{\theta}$  be a fixed value of the parameter, chosen to be close to the mode of the posterior  $p(\theta|\mathbf{x})$ . The log-posterior gradient can then be rewritten as

$$\nabla f(\theta) = \nabla f(\hat{\theta}) + [\nabla f(\theta) - \nabla f(\hat{\theta})],$$

where the first term on the right-hand side is a constant and the bracketed term on the right-hand side can be unbiasedly estimated by

$$\begin{aligned} & [\nabla \hat{f}(\theta) - \nabla \hat{f}(\hat{\theta})] \\ &= \nabla f_0(\theta) - \nabla f_0(\hat{\theta}) + \frac{1}{n} \sum_{i \in S} \frac{1}{p_i} [\nabla f_i(\theta) - \nabla f_i(\hat{\theta})] \end{aligned}$$

where  $p_1, \dots, p_N$  are user-chosen, strictly positive probabilities,  $S$  is a random sample from  $\{1, \dots, N\}$  such that  $|S| = n$  and the expected number of times  $i$  is sampled is  $np_i$ . The standard implementation of control variates would set  $p_i = 1/N$  for all  $i$ . Yet we show below that there can be advantages in having these probabilities vary with  $i$ ; for example to give higher probabilities to sampling data points for which  $\nabla f_i(\theta) - \nabla f_i(\hat{\theta})$  has higher variability.

If the gradient of the likelihood for a single observation is smooth in  $\theta$ , then we will have

$$\nabla f_i(\theta) \approx \nabla f_i(\hat{\theta}) \text{ if } \theta \approx \hat{\theta}.$$

Hence, for  $\theta \approx \hat{\theta}$ , we would expect the unbiased estimator

$$\nabla \tilde{f}(\theta) = \nabla f(\hat{\theta}) + [\nabla \hat{f}(\theta) - \nabla \hat{f}(\hat{\theta})], \tag{4}$$

to have a lower variance than the simpler unbiased estimator (2). This is because when  $\theta$  is close to  $\hat{\theta}$  we would expect the terms  $\nabla \hat{f}(\theta)$  and  $\nabla \hat{f}(\hat{\theta})$  to be correlated. This reduction in variance is shown formally in Lemma 1, stated in Sect. 3.2.

The gradient estimate (4) can be substituted into any stochastic gradient MCMC algorithm in place of  $\nabla \hat{f}(\theta)$ . We refer to SGLD using this alternative gradient estimate as SGLD-CV. The full procedure is outlined in Algorithm 1.

**Algorithm 1** SGLD-CV

**Require:**  $\hat{\theta}, \nabla f(\hat{\theta}), h$ .  
 Set  $\theta_0 \leftarrow \hat{\theta}$ .  
**for**  $t \in 1, \dots, T$  **do**  
     Update  $\nabla \tilde{f}(\theta_k)$  using (4)  
     Draw  $\zeta_t \sim N(0, \epsilon I)$   
      $\theta_{k+1} \leftarrow \theta_k - \frac{h}{2} \nabla \tilde{f}(\theta_k) + \zeta_k$   
**end for**

Implementing this in practice means finding a suitable  $\hat{\theta}$ , which we refer to as the *centring value*. We show below, under a strongly log-concave assumption, that the Wasserstein distance between the distribution defined by SGLD-CV and the posterior distribution can be bounded by some arbitrary constant; and that this can be done such that the computational cost  $Kn$  is  $O(1)$ . For this to be the case though, we require both  $\hat{\theta}$  and the starting point of SGLD-CV,  $\theta_0$ , to be  $O(N^{-\frac{1}{2}})$  distance from the posterior mean. This requires some preprocessing steps: an optimisation step to find  $\hat{\theta}$ , and calculating  $f(\hat{\theta})$ . These steps are both  $O(N)$ , but we suggest starting the algorithm from  $\hat{\theta}$ , meaning the optimisation step essentially replaces the burn-in of the chain. We find in the experiments that these initial steps are often faster than the burn-in of SGLD.

In practice, we find  $\hat{\theta}$  using *stochastic optimisation* (Robbins and Monro 1951), and then calculate the full log-posterior gradient at this point  $\nabla f(\hat{\theta})$ . We then start the algorithm from  $\hat{\theta}$ . In our implementations we use a simple stochastic optimisation method, known as *stochastic gradient descent* (SGD, see e.g. Bottou 2010). The method works similarly to the standard optimisation method gradient descent, but at each iteration replaces the true gradient of the function with an unbiased estimate. A single update of the algorithm is as follows:

$$\theta_{k+1} = \theta_k - h_k \nabla \hat{f}(\theta), \tag{5}$$

where  $\nabla \hat{f}(\theta)$  is as defined in (2) and  $h_k > 0$  is a small tuning constant referred to as the stepsize. Provided the stepsizes  $h_k$  satisfy the following conditions  $\sum_k h_k^2 < \infty$  and  $\sum_k h_k = \infty$  then this algorithm will converge to a local maximum.

We show in Sect. 3.4, under our assumptions of log-concavity of the posterior, that finding  $\hat{\theta}$  using SGD has a computational cost that is linear in  $N$ , and we can achieve the required accuracy with just a single pass through the data. As we then start SGLD-CV with this value for  $\theta$ , we can view finding the centring value as a replacement for the burn-in phase of the algorithm, and we find, in practice, that the time to find a good  $\hat{\theta}$  is often quicker than the time it takes for SGLD to burn-in. One downside of this procedure is that the SGD algorithm, as well as the SGLD-CV algorithm itself needs to be tuned, which adds to the tuning burden.

In comparison with SGLD-CV, the SAGA algorithm by Dubey et al. (2016) also uses control variates to reduce the variance in the gradient estimate of SGLD. They show that this reduces the MSE of SGLD. The main difference is that their algorithm uses a previous state in the chain as the control variate, rather than an estimate of the mode. This means that SAGA does not require the additional optimisation step, so tuning should be easier. However we show in the experiments of Sect. 5, that the algorithm gets more easily stuck in local stationary points, especially during burn-in. For more complex examples, the algorithm was prohibitively slow to burn-in because of this tendency to get trapped. Dubey et al. (2016) also do not show that SAGA has favourable computational cost results.

**3.2 Variance reduction**

The improvements of using the control-variate gradient estimate (4) over the standard (2) become apparent when we calculate the variances of each. For our analysis, we make the assumption that the posterior is strongly log-concave, formally defined in Assumption 1. This has become a common assumption when analysing gradient based samplers that do not have an acceptance step (Durmus and Moulines 2016; Dalalyan and Karagulyan 2017). In all the following analysis, we use  $\|\cdot\|$  to denote the Euclidean norm.

**Assumption 1** Strongly log-concave posterior: there exists positive constants  $m$  and  $M$ , such that the following conditions hold for the negative log-posterior

$$f(\theta) - f(\theta') - \nabla f(\theta')^\top (\theta - \theta') \geq \frac{m}{2} \|\theta - \theta'\|^2 \tag{6}$$

$$\|\nabla f(\theta) - \nabla f(\theta')\| \leq M \|\theta - \theta'\|. \tag{7}$$

for all  $\theta, \theta' \in \mathbb{R}^d$ .

We further need a Lipschitz condition for each of the likelihood terms in order to bound the variance of our control-variate estimator of the gradient.

**Assumption 2** Lipschitz: there exists constants  $L_0, \dots, L_N$  such that

$$\|\nabla f_i(\theta) - \nabla f_i(\theta')\| \leq L_i \|\theta - \theta'\|, \quad \text{for } i = 0, \dots, N.$$

Using Assumption 2, we are able to derive a bound on the variance of the gradient estimate of SGLD-CV. This bound is formally stated in Lemma 1.

**Lemma 1** Under Assumption 2. Let  $\theta_k$  be the state of SGLD-CV at the  $k$ th iteration, with stepsize  $h$  and centring value  $\hat{\theta}$ . Assume we estimate the gradient using the control-variate estimator with  $p_i = L_i / \sum_{j=1}^N L_j$  for  $i = 1, \dots, N$ . Define

$\xi_k := \nabla \tilde{f}(\theta_k) - \nabla f(\theta_k)$ , so that  $\xi_k$  measures the noise in the gradient estimate  $\nabla \tilde{f}$  and has mean 0. Then, for all  $\theta_k, \hat{\theta} \in \mathbb{R}^d$ , and all  $k = 1, \dots, K$ , we have

$$\mathbb{E} \|\xi_k\|^2 \leq \frac{(\sum_{i=1}^N L_i)^2}{n} \mathbb{E} \|\theta_k - \hat{\theta}\|^2. \tag{8}$$

Here, the expectation is over both the noise in the minibatch choice, as well as the distribution of  $\theta_k$ . All proofs are relegated to the ‘‘Appendix’’. If Assumption 1 also holds, then we can choose an  $M = \sum_{i=0}^N L_i$ , and the bound (8) implies

$$\mathbb{E} \|\xi_k\|^2 \leq \frac{M^2}{n} \mathbb{E} \|\theta_k - \hat{\theta}\|^2.$$

We will use this form of the bound for the rest of the analysis. In many situations, each data point will have a similar amount of information about the parameters. We would thus have a global bound on the Lipschitz constants, as in Assumption 3, and it is natural to choose  $p_i = 1/N$ . We use  $p_i = 1/N$  in all our implementations in the experiments of Sect. 5.

In order to consider how SGLD-CV scales with  $N$ , we need to make assumptions on the properties of the posterior and how these change with  $N$ . To make discussions concrete, we will focus on the following, strong, assumption that each likelihood-term in the posterior is strongly log-concave. As we discuss later, our results apply under weaker conditions.

**Assumption 3** Assume there exists positive constants  $L$  and  $l$  such that  $f_i$  satisfies the following conditions

$$f_i(\theta) - f_i(\theta') - \nabla f_i(\theta')^\top (\theta - \theta') \geq \frac{l}{2} \|\theta - \theta'\|^2$$

$$\|\nabla f_i(\theta) - \nabla f_i(\theta')\| \leq L \|\theta - \theta'\|.$$

for all  $i \in 0, \dots, N$  and  $\theta, \theta' \in \mathbb{R}^d$ .

Under this assumption the log-concavity constants,  $m$  and  $M$ , of the posterior both increase linearly with  $N$ , as shown by the following Lemma.

**Lemma 2** Suppose Assumption 3 holds. Then, the log-posterior,  $f$ , satisfies the following

$$f(\theta) - f(\theta') - \nabla f(\theta')^\top (\theta - \theta') \geq \frac{l(N+1)}{2} \|\theta - \theta'\|^2$$

$$\|\nabla f(\theta) - \nabla f(\theta')\| \leq L(N+1) \|\theta - \theta'\|.$$

Thus, the posterior is strongly log-concave with parameters  $M = (N+1)L$  and  $m = (N+1)l$ .

We can see that the bound on the gradient estimate variance in (8) depends on the distance between  $\theta_k$  and  $\hat{\theta}$ . Appealing to the Bernstein-von Mises theorem (Le Cam

2012), under standard asymptotics we would expect the distance  $\mathbb{E} \|\theta_k - \hat{\theta}\|^2$  to be  $O(1/N)$ , if  $\hat{\theta}$  is within  $O(N^{-1/2})$  of the posterior mean, once the MCMC algorithm has burnt in. As  $M$  is  $O(N)$ , this suggests that  $\mathbb{E} \|\xi_k\|^2$  will be  $O(N)$ .

To see the potential benefit of using control variates to estimate the gradient in situations where  $N$  is large, we now compare this  $O(N)$  result for SGLD-CV, with a result on the variance of the simple estimator,  $\nabla \hat{f}(\theta)$ . If we randomly pick some data point index  $I$  and fix some point  $\theta = \vartheta$ , then define  $V_j(\vartheta)$  to be the empirical variance of  $\partial_j f_I(\vartheta)$  given the data  $\mathbf{x}$ ; and set  $V(\vartheta) = \sum_{j=1}^d V_j(\vartheta)$ . Then, defining  $\hat{\xi}(\vartheta) = \nabla \hat{f}(\theta) - \nabla f(\theta)$ , if we assume we are sampling the minibatch without replacement then

$$\mathbb{E} \|\hat{\xi}(\vartheta)\|^2 = \frac{N(N-n)}{n} V(\vartheta).$$

Now, suppose that for large  $N$ , the true parameter value converges to  $\theta_0$ . Then, we would expect that, for  $\vartheta$  close to  $\theta_0$ ,  $\mathbb{E} \|\hat{\xi}(\vartheta)\|^2 \approx \frac{N^2}{n} V(\theta_0)$ , so that the estimator will be  $O(N^2)$ . More precisely, if we assume that as  $N \rightarrow \infty$ , the posterior converges to a point mass at some value  $\theta_0$ , then we can choose  $\epsilon > 0$  and assume  $V(\vartheta) \geq \sigma^2 > 0$  for all  $\vartheta$  in an epsilon ball around  $\theta_0$ . Then, there is a constant  $c > 0$  such that

$$\frac{n}{N^2} \mathbb{E} \|\hat{\xi}(\theta)\|^2 \rightarrow c, \quad \text{as } N \rightarrow \infty. \tag{9}$$

This suggests using the estimate  $\nabla \tilde{f}$ , rather than  $\nabla \hat{f}$ , could give an  $O(N)$  reduction in variance, and this plays a key part in the computational cost improvements we show in the next section.

### 3.3 Computational cost of SGLD-CV

In this section, we investigate how applying control variates to the gradient estimate of SGLD reduces the computational cost of the algorithm.

In order to show this, we investigate the Wasserstein–Monge–Kantorovich (Wasserstein) distance  $W_2$  between the distribution defined by the SGLD-CV algorithm at each iteration and the true posterior as  $N$  is changed. For two measures  $\mu$  and  $\nu$  defined on the probability space  $(\mathbb{R}^d, B(\mathbb{R}^d))$ , and for a real number  $q > 0$ , the distance  $W_q$  is defined by

$$W_q(\mu, \nu) = \left[ \inf_{\gamma \in \Gamma(\mu, \nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|\theta - \theta'\|^q d\gamma(\theta, \theta') \right]^{\frac{1}{q}},$$

where the infimum is with respect to all joint distributions  $\Gamma$  having  $\mu$  and  $\nu$  as marginals. The Wasserstein distance is a natural distance measure to work with for Monte Carlo



algorithms, as discussed in Durmus and Moulines (2016) and Dalalyan and Karagulyan (2017).

One issue when working with the Wasserstein distance is that it is not invariant to transformations. For example, scaling all entries of  $\theta$  by a constant will scale the Wasserstein distance by the same constant. A linear transformation of the parameters will result in a posterior that is still strongly log-concave, but with different constants  $m$  and  $M$ . To account for this, we suggest measuring error by the quantity  $\sqrt{m}W_2$ , which is invariant to scaling  $\theta$  by a constant. Theorem 1 of Durmus and Moulines (2016) bounds the standard deviation of any component of  $\theta$  by a constant times  $1/\sqrt{m}$ , so we can view the quantity  $\sqrt{m}W_2$  as measuring the error on a scale that is relative to the variability of  $\theta$  under the posterior distribution.

There are a number of quantities that will affect the performance of SGLD and SGLD-CV. These include the stepsize  $h$ , the minibatch size  $n$  and the total number of iterations  $K$ . In the analysis to follow, we find conditions on  $h$ ,  $n$  and  $K$  that ensure the Wasserstein distance between the distribution defined by SGLD-CV and the true posterior distribution  $\pi$  are less than some  $\epsilon > 0$ . We use these conditions to calculate the computational cost, measured as  $Kn$ , required for this algorithm to reach the satisfactory error  $\epsilon$ .

The first step is to find an upper bound on the Wasserstein distance between SGLD-CV and the posterior distribution in terms of  $h$ ,  $n$ ,  $K$  and the constants  $m$  and  $M$  declared in Assumption 1.

**Proposition 1** *Under Assumptions 1 and 2, let  $\theta_K$  be the state of SGLD-CV at the  $K$ th iteration of the algorithm with stepsize  $h$ , initial value  $\theta_0$ , centring value  $\hat{\theta}$ . Let the distribution of  $\theta_K$  be  $\nu_K$ . Denote the expectation of  $\theta$  under the posterior distribution  $\pi$  by  $\bar{\theta}$ . If  $h < \frac{2m}{2M^2+m^2}$ , then for all integers  $K \geq 0$ ,*

$$W_2(\nu_K, \pi) \leq (1 - A)^K W_2(\nu_0, \pi) + \frac{C}{A} + \frac{B^2}{C + \sqrt{AB}},$$

where

$$A = 1 - \sqrt{\frac{2h^2M}{n} + (1 - mh)^2},$$

$$B = \sqrt{\frac{2h^2M^2}{n} \left[ \mathbb{E} \left\| \hat{\theta} - \bar{\theta} \right\|^2 + \frac{d}{m} \right]},$$

$$C = \alpha M (h^3 d)^{\frac{1}{2}},$$

$\alpha = 7\sqrt{2}/6$  and  $d$  is the dimension of  $\theta_k$ .

The proof of this proposition is closely related to the proof of Proposition 2 of Dalalyan and Karagulyan (2017). The extra complication comes from our bound on the variance of

the estimator of the gradient; which depends on the current state of the SGLD-CV algorithm, rather than being bounded by a global constant.

We can now use Proposition 1 to find conditions on  $K$ ,  $h$  and  $n$  in terms of the constants  $M$  and  $m$  such that the Wasserstein distance is bounded by some positive constant  $\epsilon_0/\sqrt{m}$  at its final iteration  $K$ .

**Theorem 1** *Under Assumptions 1 and 2, let  $\theta_K$  be the state of SGLD-CV at the  $K$ th iteration of the algorithm with stepsize  $h$ , initial value  $\theta_0$ , centring value  $\hat{\theta}$ . Let the distribution of  $\theta_K$  be  $\nu_K$ . Denote the expectation of  $\theta$  under the posterior distribution  $\pi$  by  $\bar{\theta}$ . Define  $R := M/m$ . Then, for any  $\epsilon_0 > 0$ , if the following conditions hold:*

$$h \leq \frac{1}{m} \max \left\{ \frac{n}{2R^2 + n}, \frac{\epsilon_0^2}{64R^2\alpha^2 d} \right\}, \tag{10}$$

$$Kh \geq \frac{1}{m} \log \left[ \frac{4m}{\epsilon_0^2} \left( \mathbb{E} \left\| \theta_0 - \bar{\theta} \right\|^2 + d/m \right) \right], \tag{11}$$

$$n \geq \frac{64R^2\beta}{\epsilon_0^2} m \left[ \mathbb{E} \left\| \hat{\theta} - \bar{\theta} \right\|^2 + \frac{d}{m} \right], \tag{12}$$

where

$$\beta = \max \left\{ \frac{1}{2R^2 + 1}, \frac{\epsilon_0^2}{64R^2\alpha^2 d} \right\},$$

$\alpha = 7\sqrt{2}/6$ , and  $d$  is the dimension of  $\theta_k$ , then  $W_2(\nu_K, \pi) \leq \epsilon_0/\sqrt{m}$ .

As a corollary of this result, we have the following, which gives a bound on the computational cost of SGLD, as measured by  $Kn$ , to achieve a required bound on the Wasserstein distance.

**Corollary 1** *Assume that Assumptions 1 and 3 and the conditions of Theorem 1 hold. Fix  $\epsilon_0$  and define*

$$C_1 = \min \left\{ 2R^2 + 1, \frac{64R^2\alpha^2 d}{\epsilon_0^2} \right\},$$

and  $C_2 := 64R^2\beta/\epsilon_0^2$ . We can implement an SGLD-CV algorithm with  $W_2(\nu_K, \pi) < \epsilon_0/\sqrt{m}$  such that

$$Kn \leq \left[ C_1 \log \left[ m \mathbb{E} \left\| \theta_0 - \bar{\theta} \right\|^2 + d \right] + C_1 \log \frac{4}{\epsilon_0^2} + 1 \right] \left[ C_2 m \mathbb{E} \left\| \hat{\theta} - \bar{\theta} \right\|^2 + C_2 d + 1 \right].$$

The constants,  $C_1$  and  $C_2$ , in the bound on  $Kn$ , depend on  $\epsilon_0$  and  $R = M/m$ . It is simple to show that both constants

are increasing in  $R$ . Under Assumption 3, we have that  $R$  is a constant as  $N$  increases. Corollary 1 suggests that provided  $\|\theta_0 - \bar{\theta}\| < c/\sqrt{m}$  and  $\|\hat{\theta} - \bar{\theta}\| < c/\sqrt{m}$ , for some constant  $c$ ; then the computational cost of SGLD-CV will be bounded by a constant. Since we suggest starting SGLD-CV at  $\hat{\theta}$ , then technically we just need this to hold for  $\|\hat{\theta} - \bar{\theta}\|$ . Under Assumption 3, we have that  $m$  increases linearly with  $N$ , so this corresponds to needing  $\|\hat{\theta} - \bar{\theta}\| < c_1/\sqrt{N}$  as  $N$  increases. Additionally, by Theorem 1 of Durmus and Moulines (2016), we have that the variance of the posterior scales like  $1/m = 1/N$  as  $N$  increases, so we can interpret the  $1/\sqrt{N}$  factor as being a measure of the spread of the posterior as  $N$  increases. The form of the corollary makes it clear that a similar argument would apply under weaker assumptions than Assumption 3. We only need that the ratio of the log-concavity constants,  $M/m$ , of the posterior remains bounded as  $N$  increases.

This corollary also gives insight into the computational penalty you pay for a poor choice of  $\theta_0$  or  $\hat{\theta}$ . The bound on the computational cost will increase logarithmically with  $\|\theta_0 - \bar{\theta}\|$  and linearly with  $\|\hat{\theta} - \bar{\theta}\|$ .

Similar bounds for the computational cost of SGLD have been derived (e.g. Nagapetyan et al. 2017; Chatterji et al. 2018). For example, Chatterji et al. (2018) state the computational cost of SGLD in order to sample from a distribution with  $W_2$  distance that is within  $\epsilon$  distance of the target is  $O(1/\epsilon^2)$ . For our required level of accuracy,  $\epsilon = \epsilon_0/\sqrt{m}$ , this corresponds to a bound on the computational cost that is  $O(N)$ , as compared to  $O(1)$  for SGLD-CV. However, a formal proof that SGLD is  $O(N)$  requires showing that any implementation of SGLD with  $Kn$  that is sublinear in  $N$  cannot reach arbitrary accuracy  $\epsilon$ . This requires a lower bound on the  $W_2$  distance, rather than an upper bound. Whilst the result of Chatterji et al. (2018) is just an upper bound on the  $W_2$  distance, there are empirical results that suggest a linear computational cost for SGLD, including those in Nagapetyan et al. (2017), as well as our own experiments in Sect. 5.

### 3.4 Setup costs

There are a number of known results on the convergence of SGD under the strongly log-concave conditions of Assumption 1. These will allow us to quantify the setup cost of finding the point  $\hat{\theta}$  in this setting. More complex cases are explored empirically in the experiments in Sect. 5. Lemma 3 due to Nemirovski et al. (2009) quantifies the convergence of the final point of SGD.

**Lemma 3** (Nemirovski et al. 2009) *Under Assumption 1, let  $\hat{\theta}$  denote the final state of SGD with stepsizes  $h_k = 1/(mk)$  after  $K$  iterations. Suppose  $\mathbb{E} \|\nabla \hat{f}(\theta)\|^2 \leq D^2$  and denote*

*the true mode of  $f$  by  $\theta^*$ . Then it holds that*

$$\mathbb{E} \|\hat{\theta} - \theta^*\|^2 \leq \frac{4D^2}{m^2 K}.$$

By using a similar argument to (9), we would expect that  $D^2$  is  $O(N^2/n)$ . This means that under Assumption 3, we will need to process the full dataset once before the SGD algorithm has converged to a mode  $\hat{\theta}$  within  $O(N^{-\frac{1}{2}})$  of the posterior mean. It follows that, for these cases there are two one off  $O(N)$  setup costs, one to find an acceptable mode  $\hat{\theta}$  and one to find the full log-posterior gradient at this point  $\nabla f(\hat{\theta})$ .

## 4 Postprocessing control variates

Control variates can also be used to improve the inferences made from MCMC by reducing the variance of the output directly. The general aim of MCMC is to estimate expectations of functions,  $g(\theta)$ , with respect to the posterior  $\pi$ . Given an MCMC sample  $\theta^{(1)}, \dots, \theta^{(M)}$ , from the posterior  $\pi$ , we can estimate  $\mathbb{E}[g(\theta)]$  unbiasedly as

$$\mathbb{E}[g(\theta)] \approx \frac{1}{M} \sum_{i=1}^M g(\theta^{(i)}).$$

Suppose there exists a function  $h(\theta)$ , which has expectation 0 under the posterior. We can then introduce an alternative function,

$$\tilde{g}(\theta) = g(\theta) + h(\theta),$$

where  $\mathbb{E}[\tilde{g}(\theta)] = \mathbb{E}[g(\theta)]$ . If  $h(\cdot)$  is chosen so that it is negatively correlated with  $g(\theta)$ , then the variance of  $\tilde{g}(\theta)$  will be reduced considerably.

Mira et al. (2013) introduce a way of choosing  $h(\theta)$  almost automatically by using the gradient of the log-posterior. Choosing  $h(\cdot)$  in this manner is referred to as a zero variance (ZV) control variate. Friel et al. (2016) showed that, under mild conditions, we can replace the log-posterior gradient with an unbiased estimate and still have a valid control variate. SGMCMC methods produce unbiased estimates of the log-posterior gradient, and so it follows that these gradient estimates can be applied as ZV control variates. For the rest of this section, we focus our attention on SGLD, but these ideas are easily extendable to other stochastic gradient MCMC algorithms. We refer to SGLD with these postprocessing control variates as SGLD-ZV.

Given the setup outlined above, Mira et al. (2013) propose the following form for  $h(\theta)$ ,

$$h(\theta) = \Delta Q(\theta) + \nabla Q(\theta) \cdot \mathbf{z},$$

**Algorithm 2** SGLD-ZV

**Require:**  $\{\theta_k, \nabla \hat{f}(\theta_k)\}_{k=1}^K$  ▷ SGLD output  
 Set  $\mathbf{z}_k \leftarrow \frac{1}{2} \nabla \hat{f}(\theta_k)$   
 Estimate  $V_{\mathbf{z}} \leftarrow \text{Var}(\mathbf{z}), C_{g,\mathbf{z}} \leftarrow \text{Cov}(g(\theta), \mathbf{z})$   
 $\hat{\mathbf{a}}_j \leftarrow [V_{\mathbf{z}}]^{-1} C_{g,\mathbf{z}}$   
**for**  $k = 1 \dots K$  **do**  
 $\hat{g}(\theta_k) \leftarrow g(\theta_k) + \hat{\mathbf{a}}^T \mathbf{z}_k$   
**end for**

here  $Q(\theta)$  is a polynomial of  $\theta$  to be chosen and  $\mathbf{z} = f(\theta)/2$ .  $\Delta$  refers to the Laplace operator  $\frac{\partial^2}{\partial \theta_1^2} + \dots + \frac{\partial^2}{\partial \theta_d^2}$ . In order to get the best variance reduction, we simply have to optimise the coefficients of the polynomial  $Q(\cdot)$ . In practice, first or second degree polynomials  $Q(\theta)$  often provide good variance reduction (Mira et al. 2013). For the rest of this section, we focus on first degree polynomials, so  $Q(\theta) = \mathbf{a}^T \theta$ , but the ideas are easily extendable to higher orders (Papamarkou et al. 2014).

The SGLD algorithm only calculates an unbiased estimate of  $\nabla f(\theta)$ , so we propose replacing  $h(\theta)$  with the unbiased estimate

$$\hat{h}(\theta) = \Delta Q(\theta) + \nabla Q(\theta) \cdot \hat{\mathbf{z}}, \tag{13}$$

where  $\hat{\mathbf{z}} = \nabla \hat{f}(\theta)/2$ . By identical reasoning to Friel et al. (2016),  $\hat{h}(\theta)$  is a valid control variate. Note that  $\hat{\mathbf{z}}$  can use any unbiased estimate, and as we will show later, the better the gradient estimate, the better this control variate performs. The expectation of  $\hat{h}$  is zero because  $\mathbb{E}[h(\theta)] = 0$ , and  $\hat{\mathbf{z}}$  is an unbiased estimate of  $\mathbf{z}$  that is independent of  $\theta$ , depending only on the minibatch choice. See ‘‘Appendix’’ for the full calculation.

As  $Q(\theta)$  is a linear polynomial  $\mathbf{a}^T \theta$ , so our SGLD-ZV estimate will take the following form

$$\hat{g}(\theta) = g(\theta) + \mathbf{a}^T \hat{\mathbf{z}}. \tag{14}$$

Similar to standard control variates (Ripley 2009), we need to find optimal coefficients  $\hat{\mathbf{a}}$  in order to minimise the variance of  $\tilde{g}(\cdot)$ , defined in (14). In our case, the optimal coefficients take the following form (Friel et al. 2016)

$$\hat{\mathbf{a}} = \text{Var}^{-1}(\hat{\mathbf{z}}) \text{Cov}(\hat{\mathbf{z}}, g(\theta)).$$

This means that SGLD already calculates all the necessary terms for these control variates to be applied for free. So the postprocessing step can simply be applied once when the SGLD algorithm has finished, provided the full output plus gradient estimates are stored. The full details are given in Algorithm 2. For higher order polynomials, the calculations are much the same, but more coefficients need to be estimated (Papamarkou et al. 2014).

The efficiency of ZV control variates in reducing the variance of our MCMC sample is directly affected by using an estimate of the gradient rather than the truth. For the remainder of this section, we investigate how the choice of the gradient estimate, and the minibatch size  $n$ , affects the variance reduction.

**Assumption 4**  $\text{Var}[\phi(\theta)] < \infty$  and  $\text{Var}[\hat{\psi}(\theta)] < \infty$ .  $\mathbb{E}_{\theta|\mathbf{x}} \|\nabla f_i(\theta)\|^2$  is bounded by some constant  $\sigma$  for all  $i = 0, \dots, N, \theta \in \mathbb{R}^d$ .

**Theorem 2** Under Assumption 4, define the optimal variance reduction for ZV control variates using the full gradient estimate to be  $R$ , and the optimal variance reduction using SGLD gradient estimates to be  $\hat{R}$ . Then, we have that

$$\hat{R} \geq \frac{R}{1 + [\sigma(N + 1)]^{-1} \mathbb{E}_{\theta|\mathbf{x}} [\mathbb{E}_S \|\xi_S(\theta)\|^2]}, \tag{15}$$

where  $\xi_S(\theta)$  is the noise in the log-posterior gradient estimate.

The proof of this result is given in the ‘‘Appendix’’. An important consequence of Theorem 2 is that if we use the standard SGLD gradient estimate, then the denominator of (15) is  $O(n/N)$ , so our variance reduction diminishes as  $N$  gets large. However, if we use the SGLD-CV estimate instead, then under standard asymptotics, the denominator of (15) is  $O(n)$ , so the variance reduction does not diminish with increasing dataset size. The same holds for SAGA, and other control-variate algorithms that reduce the gradient error to  $O(N)$ . It follows that for best results, we recommend using the ZV postprocessing step after running the SGLD-CV algorithm, especially for large  $N$ . The ZV postprocessing step can be immediately applied in exactly the same way to other stochastic gradient MCMC algorithms, such as SGHMC and SGNHT (Chen et al. 2014; Ding et al. 2014).

It is worth noting that there are some storage constraints for SGLD-ZV. This algorithm requires storing the full MCMC chain, as well as the gradient estimates at each iteration. So the storage cost is twice the storage cost of a standard SGMCMC run. However, in some high-dimensional cases, the required SGMCMC test statistic is estimated on the fly using the most recent output of the chain and thus reducing the storage costs. We suggest that if the dimensionality is not too high, then the additional storage cost of recording the gradients to apply the ZV postprocessing step can offer significant variance reduction for free. However, for very high-dimensional parameters, the cost associated with storing the gradients may preclude the use of the ZV step.



## 5 Experiments

In the experiments to follow, we compare SGLD-CV to SGLD in a number of different scenarios. To empirically demonstrate the scalability advantages, we infer each model for three different dataset sizes. The preprocessing steps for SGLD-CV are included in the plots. We include the method SAGA (Dubey et al. 2016) in our results, an alternative variance reduction method for SGLD.

We also provide box-plots of the log predictive density of iterations of SGLD-CV before and after it has been post-processed using ZV control variates, to demonstrate the variance reduction and improved predictive performance of the method.

A fixed stepsize scheme is used for all methods, and these are tuned using a grid search (for SGLD-CV, both SGD and SGLD steps are tuned using a grid search). Full details of the tuning constants and alternative results using a decreasing stepsize scheme are given in “Appendix”. The minibatch size is fixed across all the dataset sizes. All results are timed when the algorithms are run on four cores of a 2.3 GHz Intel Xeon CPU.

### 5.1 Logistic regression

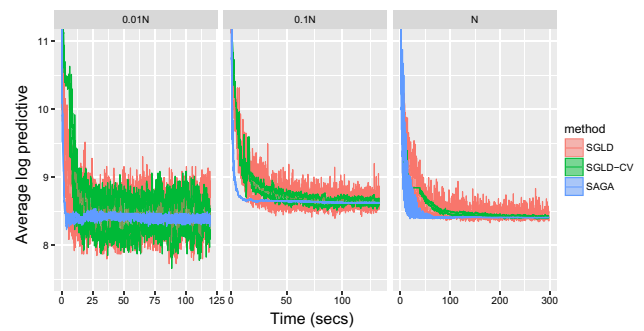
We examine our approaches on a Bayesian logistic regression problem. The probability of the  $i$ th output  $y_i \in \{-1, +1\}$  is given by

$$p(y_i|x_i, \beta) = \frac{1}{1 + \exp(-y_i \beta^T x_i)}.$$

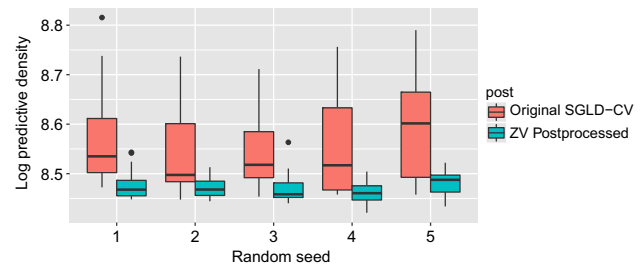
We use a Laplace prior for  $\beta$  with scale 1.

We used the cover type dataset (Blackard and Dean 1999), which has 581012 observations, which we split into a training and test set. First, we run SGLD, SGLD-CV and SAGA on the dataset, all with minibatch size 500. The method SAGA was discussed at the end of Sect. 3.1. To empirically support the scalability results of Theorem 1, we fit the model 3 times. In each fit, the dataset size is varied, from about 1% of the full dataset to the full dataset size  $N$ . The performance is measured by calculating the log predictive density on a held out test set every 10 iterations. Some of our examples are high-dimensional, so our performance measure aims to reduce dimensionality whilst still capturing important quantities such as the variance of the chain. We include the burn-in of SGLD and SAGA, to contrast with the optimisation step and full gradient calculation required for SGLD-CV, which is included in the total computational time.

The results are plotted against time in Fig. 1. The results illustrate the efficiency gains of SGLD-CV over SGLD as the dataset size increases, as expected from Theorem 1. SAGA outperforms SGLD-CV in this example because SAGA con-



**Fig. 1** Log predictive density over a test set every 10 iterations of SGLD, SGLD-CV and SAGA fit to a logistic regression model as the data size  $N$  is varied



**Fig. 2** Plots of the log predictive density of an SGLD-CV chain when ZV postprocessing is applied versus when it is not, over 5 random runs. Logistic regression model on the cover type dataset (Blackard and Dean 1999)

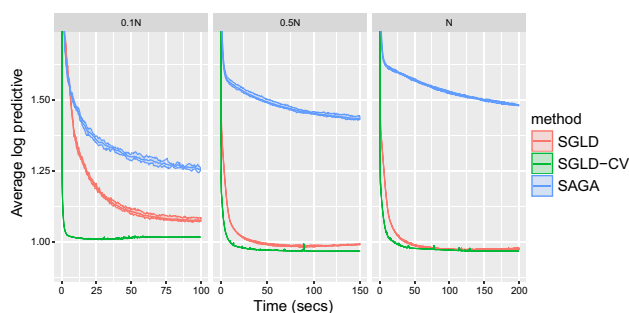
verges quickly in this simple setting. In the more complicated examples to follow, we show that SAGA can get stuck in local stationary points.

We also compare the log predictive density over a test set for SGLD-CV with and without ZV postprocessing, averaged over 5 runs at different seeds. We apply the method to SGLD-CV rather than SGLD due to the favourable scaling results as discussed after Theorem 2. Results are given in Fig. 2. The plot shows box-plots of the log predictive density of the SGLD sample before and after postprocessing using ZV control variates. The plots show good variance reduction of the chain.

### 5.2 Probabilistic matrix factorisation

A common recommendation system task is to predict a user’s rating of a set of items, given previous ratings and the ratings of other users. The end goal is to recommend new items that the user will rate highly. Probabilistic matrix factorisation (PMF) is a popular method to train these models (Mnih and Salakhutdinov 2008). As the matrix of ratings is sparse, overfitting is a common issue in these systems, and Bayesian approaches are a way to account for this (Chen et al. 2014; Ahn et al. 2015).

In this experiment, we apply SGLD, SGLD-CV and SAGA to a Bayesian PMF (BPMF) problem, using the for-

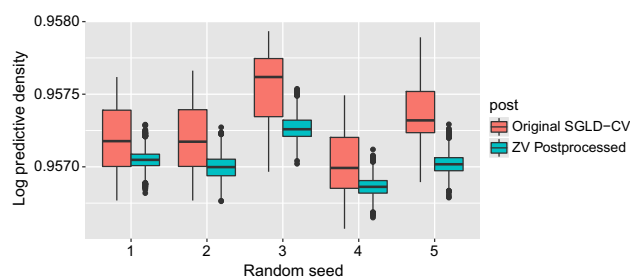


**Fig. 3** Log predictive density over a test set of SGLD, SGLD-CV and SAGA fit to a Bayesian probabilistic matrix factorisation model as the number of users is varied, averaged over 5 runs. We used the MovieLens ml-100k dataset

mulation of Chen et al. (2014). The data for BPMF are a matrix of ratings  $R$ , of size  $L \times M$ , where  $L$  is the number of users and  $M$  is the number of items. Each entry contains a rating of a particular item by that user, with a lot of missing entries. The aim is to predict the values of the missing entries. This is done by factorising  $R$  into two matrices  $U$  and  $V$ , so that  $R \approx U^T V$ , where  $U$  and  $V$  are size  $D \times L$  and  $D \times M$ , respectively. Here,  $D$  is some user-specified parameter, which we choose to be 20. We use the MovieLens dataset ml-100k<sup>1</sup>, which contains 100,000 ratings from almost 1000 users and 1,700 movies. We use batch sizes of 5000, with a larger minibatch size chosen due to the high-dimensional parameter space. As before, we compare performance by calculating the predictive distribution on a held out dataset every 10 iterations. Full details of chosen hyperparameters are detailed in “Appendix”.

We investigate the scaling results of SGLD-CV and SAGA versus SGLD by varying the dataset size. We do this by limiting the number of users in the dataset, ranging from 100 users to the full 943. The results are given in Fig. 3. In this example, SAGA converges slowly in comparison even with SGLD. In fact, the algorithm converges slowly in all our more complex experiments. The problem is particularly bad for large  $N$ . This is likely a result of the starting point for SAGA being far from the posterior mode. Empirically, we found that the gradient direction and magnitude can update very slowly in these cases. This is not an issue for simpler examples such as logistic regression, but for more complex examples we believe it could be a sign that the algorithm is getting stuck in, or moving slowly through, local modes where the gradient is comparatively flatter. The problem appears to be made worse for large  $N$  when it takes longer to update  $g_\alpha$ . This is an example where the optimisation step of SGLD-CV is an advantage, as the algorithm is immediately started close to the posterior mode and so the efficiency gains are quickly noted. This issue with SAGA could be related to the starting

<sup>1</sup> <https://grouplens.org/datasets/movielens/100k/>.



**Fig. 4** Plots of the log predictive density of an SGLD-CV chain when ZV postprocessing is applied versus when it is not, over five random runs. SGLD-CV algorithm applied to a Bayesian probabilistic matrix factorisation problem using the MovieLens ml-100k dataset

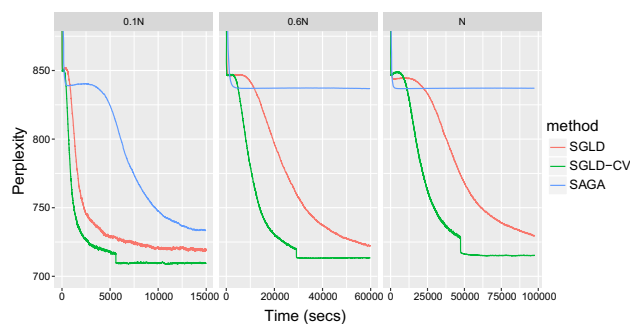
point condition for SGLD-CV as detailed in Corollary 1. Due to the form of the Wasserstein bound, it is likely that SAGA would have a similar starting point condition.

Once again we compare the log predictive density over a test set for SGLD-CV with and without ZV postprocessing when applied to the Bayesian PMF problem, averaged over five runs at different seeds. Results are given in Fig. 4. The plot shows box-plots of the log predictive density of the SGLD sample before and after postprocessing using ZV control variates. The plots show excellent variance reduction of the chain.

### 5.3 Latent Dirichlet allocation

Latent Dirichlet allocation (LDA) is an example of a topic model used to describe collections of documents by sets of discovered topics (Blei et al. 2003). The input consists of a matrix of word frequencies in each document, which is very sparse, motivating the use of a Bayesian approach to avoid over-fitting.

Due to storage constraints, it was not feasible to apply SGLD-ZV to this problem, so we focus on SGLD-CV. We scraped approximately 80,000 documents from Wikipedia, and used the 1000 most common words to form our



**Fig. 5** Perplexity of SGLD and SGLD-CV fit to an LDA model as the data size  $N$  is varied, averaged over 5 runs. The dataset consists of scraped Wikipedia articles

document-word matrix input. We used a similar formulation to Patterson and Teh (2013), though we did not use a Riemannian sampler, which should test the methods due to the gradient instability. We also instead use the expanded-natural parameterisation, since according to empirical results by Patterson and Teh (2013) that was the best performing parameterisation for the non-Riemannian case. We set the number of topics  $K$  to be 20. Full details of hyperparameters are given in “Appendix”.

Once again in our comparison of SGLD, SGLD-CV and SAGA, we vary the dataset size, this time by changing the number of documents used in fitting the model, from 10,000 to the full 81,538. We use batch sizes of 50 documents. We measure the performance of LDA using the *perplexity* on held out words from each document, a standard performance measure for this model, for more detail see Patterson and Teh (2013). The results are given in Fig. 5. Here, the scalability improvements of using SGLD-CV over SGLD are clear as the dataset size increases. This time the batch size is small compared to the dataset size, which probably makes the scalability improvements more obvious. The sudden drop in perplexity for the SGLD-CV plot occurs at the switch from the stochastic optimisation step to SGLD-CV. This is likely a result of the algorithm making efficient use of the Gibbs step to simulate the latent topics.

An interesting aspect of this problem is that it appears to have a pronounced local mode where each of the methods become trapped (this can be seen by the blip in the plot at a perplexity of around 850). SGLD-CV is the first to escape followed by SGLD, but SAGA takes a long time to escape. This is probably due to a similar aspect as the one discussed in the previous experiment (Sect. 5.2). Similar to the previous experiment, we find that whilst SAGA seems trapped, its gradient estimate changes very little, which could be a sign that the algorithm is moving very slowly through an area with a relatively flat gradient, such as a local mode. A simple solution would be to start SAGA closer to the mode using a stochastic optimisation scheme.

## 6 Discussion

We have used control variates for stochastic gradient MCMC to reduce the variance in the gradient estimate. We have shown that in the strongly log-concave setting, and under standard asymptotics, this proposed SGLD-CV algorithm reduces the computational cost of stochastic gradient Langevin dynamics to  $O(1)$ . Our theoretical results give results on the computational cost under nonstandard asymptotics also, and show there should be some benefit provided distance between the centring value  $\hat{\theta}$  and the posterior mean inversely depends on  $N$ . The algorithm relies on a setup cost that estimates the posterior mode which replaces the

burn-in of SGLD. We have explored the cost of this step both theoretically and empirically. We have empirically supported these scalability results on a variety of interesting and challenging problems from the statistics and machine learning literature using real-world datasets. The simulation study also revealed that SGLD-CV was less susceptible to getting stuck in local stationary points than an alternative method that performs variance reduction using control variates, SAGA (Dubey et al. 2016). An interesting future extension would be to reduce the start-up cost of SGLD-CV, along with introducing automatic stepsize tuning.

We showed that stochastic gradient MCMC methods calculate all the information needed to apply zero variance postprocessing control variates. This improves the inference of the output by reducing its variance. We explored how the variance reduction is affected by the minibatch size and the gradient estimate, and show using SGLD-CV or SAGA rather than SGLD can achieve a better variance reduction. We demonstrated this variance reduction empirically. A limitation of these postprocessing control variates is they require the whole chain, which can lead to high storage costs if the dimensionality of the sample space is high. Future work could explore ways to reduce the storage costs of stochastic gradient MCMC.

**Acknowledgements** The first author gratefully acknowledges the support of the EPSRC funded EP/L015692/1 STOR-i Centre for Doctoral Training. This work was supported by EPSRC grants EP/K014463/1, EP/R018561/1, EP/S00159X/1 and EP/R01860X/1 and ONR Grant N00014-15-1-2380 and NSF CAREER Award IIS-1350133.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## A Computational cost proofs

### A.1 Proof of Proposition 1

**Proof** Let  $\pi$  be the invariant distribution of the underlying dynamics, so that it has density  $e^{-f(\theta)} = p(\theta|\mathbf{x})$ , and define  $W_2(\nu_k, \pi)$  to be the Wasserstein distance between  $\nu_k$  and  $\pi$ . Define  $\xi_k$  to be the SGLD-CV gradient noise term. Then, we can write a single step of SGLD-CV as

$$\theta_{k+1} = \theta_k + h\nabla f(\theta_k) + h\xi_k + \sqrt{2h}\zeta_k,$$

We have that  $\theta_k \sim \nu_k$ , and follow similarly to the proof of Dalalyan and Karagulyan (2017, Proposition 2). First define  $Y_0$  to be a draw from the invariant distribution  $\pi$ , such that the joint distribution of  $Y_0$  and  $\theta_k$  minimises  $\mathbb{E} \|Y_0 - \theta_k\|^2$ .

Here  $\|\cdot\|$  denotes the Euclidean distance for  $\mathbb{R}^d$ . It follows that  $\mathbb{E} \|Y_0 - \theta_k\|^2 = W_2^2(\nu_k, \pi)$ .

Let  $B_t$  be a  $d$ -dimensional Wiener process, independent of  $\theta_k, Y_0$  and  $\xi_k$  but which we couple to the injected noise  $\zeta_k$  so that  $B_h = \sqrt{h}\zeta_k$ . Now let  $Y_t, t > 0$ , follow the diffusion

$$Y_t = Y_0 + \int_0^t \nabla f(Y_s) ds + \sqrt{2}B_t. \tag{16}$$

Let  $\Delta_k = Y_0 - \theta_k$  and  $\Delta_{k+1} = Y_h - \theta_{k+1}$ . Since we started the process  $Y_t$  from  $Y_0 \sim \pi$ , then it follows that  $Y_t \sim \pi$  for all  $t > 0$ . Also since  $W_2^2(\nu_{k+1}, \pi)$  minimises the expected squared distance between two random variables with marginals  $\nu_{k+1}$  and  $\pi$  then it follows that  $W_2^2(\nu_{k+1}, \pi) \leq \mathbb{E} \|\Delta_{k+1}\|^2$ .

Let us define

$$U = \nabla f(\theta_k + \Delta_k) - \nabla f(\theta_k), \tag{17}$$

$$V = \int_0^h [\nabla f(Y_t) - \nabla f(Y_0)] dt. \tag{18}$$

Then, by the unbiasedness of the gradient estimation,  $\xi_k$  has mean 0 regardless of the value of  $\theta_k$ . Thus

$$\begin{aligned} \mathbb{E} \|\Delta_{k+1}\|^2 &= \mathbb{E} \|\Delta_k + hU + V\|^2 + \mathbb{E} \|\xi_k\|^2 \\ &\leq [\mathbb{E} \|\Delta_k - hU\| + \mathbb{E} \|V\|]^2 + h^2 \mathbb{E} \|\xi_k\|^2. \end{aligned}$$

We can then apply Lemmas 2 and 4 in Dalalyan and Karagulyan (2017), stated in Lemmas 4 and 5, as well as applying the gradient noise bound in Lemma 1, to obtain a bound on  $W_2^2(\nu_{k+1}, \pi)$  given  $W_2^2(\nu_k, \pi)$ .

**Lemma 4** *With  $U$  as defined in (17), if  $h < 2m/(2M^2 + m^2)$ , then  $\|\Delta_k - hU\| \leq (1 - mh) \|\Delta_k\|$ .*

The original lemma by Dalalyan and Karagulyan (2017) assumed  $h < 2/(m + M)$ , but this holds when  $h < 2m/(2M^2 + m^2)$  as  $m \leq M$ .

**Lemma 5** *Under Assumption 1. Let  $V$  be as defined in (18), then*

$$\mathbb{E} \|V\| \leq \frac{1}{2}(h^4 M^3 d)^{\frac{1}{2}} + \frac{2}{3}(2h^3 d)^{\frac{1}{2}} M.$$

Finally, we can apply Lemma 1, as stated in the main body, to get

$$\begin{aligned} \mathbb{E} \|\xi_k\|^2 &\leq \frac{M^2}{n} \mathbb{E} \|\theta_k - \hat{\theta}\|^2 \\ &\leq \frac{2M^2}{n} \mathbb{E} \|\theta_k - Y_0\|^2 + \frac{2M^2}{n^2} \mathbb{E} \|Y_0 - \hat{\theta}\|^2 \\ &\leq \frac{2M^2}{n} W_2^2(\nu_k, \pi) + \frac{2M^2}{n} \mathbb{E} \|Y_0 - \hat{\theta}\|^2 \end{aligned}$$

Using Theorem 1 of Durmus and Moulines (2016)

$$\mathbb{E} \|Y_0 - \hat{\theta}\|^2 \leq \mathbb{E} \|\hat{\theta} - \bar{\theta}\|^2 + \frac{d}{m}. \tag{19}$$

It follows that

$$\mathbb{E} \|\xi_k\|^2 \leq \frac{2M^2}{n} W_2^2(\nu_k, \pi) + \frac{2M^2}{n} \left[ \mathbb{E} \|\hat{\theta} - \bar{\theta}\|^2 + \frac{d}{m} \right]. \tag{20}$$

Now using that  $W_2^2(\nu_{k+1}, \pi) \leq \mathbb{E} \|\Delta_{k+1}\|^2$  we get the following

$$\begin{aligned} W_2^2(\nu_{k+1}, \pi) &\leq \left[ (1 - mh) W_2(\nu_k, \pi) + \alpha M (h^3 d)^{\frac{1}{2}} \right]^2 \\ &\quad + \frac{2h^2 M^2}{n} W_2^2(\nu_k, \pi) + \frac{2h^2 M^2}{n} \left[ \mathbb{E} \|\hat{\theta} - \bar{\theta}\|^2 + \frac{d}{m} \right], \end{aligned}$$

where  $\alpha = 7\sqrt{2}/6$ . Gathering like terms we can further bound  $W_2^2(\nu_{k+1}, \pi)$  to get the following recursive formula

$$W_2^2(\nu_{k+1}, \pi) \leq [(1 - A)W_2(\nu_k, \pi) + C]^2 + B^2$$

where

$$\begin{aligned} A &= 1 - \sqrt{\frac{2h^2 M^2}{n} + (1 - mh)^2} \\ B &= \sqrt{\frac{2h^2 M^2}{n} \left[ \mathbb{E} \|\hat{\theta} - \bar{\theta}\|^2 + \frac{d}{m} \right]} \\ C &= \alpha M (h^3 d)^{\frac{1}{2}}. \end{aligned}$$

We can now apply Lemma 1 of Dalalyan and Karagulyan (2017), as stated below to solve this recurrence relation.

**Lemma 6** *Let  $A, B$  and  $C$  be nonnegative numbers such that  $A \in (0, 1)$ . Assume that the sequence of nonnegative numbers  $x_k, k = 0, 1, \dots$ , satisfies the recursive inequality*

$$x_k^2 \leq [(1 - A)x_k + C]^2 + B^2$$

for every integer  $k > 0$ . Then, for all integers  $k \geq 0$

$$x_k \leq (1 - A)^k x_0 + \frac{C}{A} + \frac{B^2}{C + \sqrt{AB}}$$

To complete the proof all that remains is to check  $A \in (0, 1)$  so that Lemma 6 can be applied. Clearly  $A < 1$ , since  $n \geq 1$  we have

$$A \geq 1 - \sqrt{2h^2 M^2 - (1 - mh)^2},$$

and the RHS is positive when  $h \in (0, 2m/(2M^2 + m^2))$ .



### A.2 Proof of Theorem 1

**Proof** Starting from Proposition 1, we have that

$$W_2(\nu_K, \pi) \leq (1-A)^K W_2(\nu_0, \pi) + \frac{C}{A} + \frac{B^2}{C + \sqrt{AB}}. \tag{21}$$

where

$$A = 1 - \sqrt{\frac{2h^2M^2}{n} + (1-mh)^2},$$

$$B = \sqrt{\frac{2h^2M^2}{n} \left[ \mathbb{E} \|\hat{\theta} - \bar{\theta}\|^2 + \frac{d}{m} \right]},$$

$$C = \alpha M (h^3d)^{\frac{1}{2}},$$

Suppose we stop the algorithm at iteration  $K$ . Using (21), the following are sufficient conditions that ensure  $W_2^2(\nu_K, \pi) < \epsilon_0/\sqrt{m}$ ,

$$(1-A)^K W_2(\nu_0, \pi) \leq \frac{\epsilon_0}{2\sqrt{m}}, \tag{22}$$

$$\frac{C}{A} \leq \frac{\epsilon_0}{4\sqrt{m}}, \tag{23}$$

$$\frac{B^2}{C + \sqrt{AB}} \leq \frac{\epsilon_0}{4\sqrt{m}}. \tag{24}$$

The starting point  $\theta_0$  is deterministic, so from Theorem 1 of Durmus and Moulines (2016)

$$W_2^2(\nu_0, \pi) \leq \mathbb{E} \|\theta_0 - \bar{\theta}\|^2 + \frac{d}{m}. \tag{25}$$

If we rewrite

$$h = \frac{\gamma}{m} \left[ \frac{2n}{2R^2 + n} \right], \tag{26}$$

where  $\gamma \in (0, 1)$  is some constant and  $R := M/m$  as defined in the theorem statement, then it follows that we can write

$$A = 1 - \sqrt{1 - 2mh(1-\gamma)}. \tag{27}$$

Since we have the condition

$$h \leq \frac{1}{m} \left[ \frac{n}{2R^2 + n} \right],$$

then  $\gamma \leq \frac{1}{2}$ .

Now suppose, using (27), we set

$$Kh \geq \frac{1}{m} \log \left[ \frac{4m}{\epsilon_0^2} \left( \mathbb{E} \|\theta_0 - \bar{\theta}\|^2 + d/m \right) \right] \tag{28}$$

Then using the result for the deterministic starting point  $\theta_0$  (25), we find that (28) implies that

$$\begin{aligned} \frac{\epsilon_0}{2\sqrt{m}} &\geq \exp[-mhK/2] \sqrt{\mathbb{E} \|\theta_0 - \bar{\theta}\|^2 + \frac{d}{m}} \\ &\geq [1 - mh]^{\frac{K}{2}} W_2(\nu_0, \pi) \\ &\geq (1-A)^K W_2(\nu_0, \pi), \end{aligned}$$

Using (27) and that our conditions imply  $\gamma < 1/2$ . Hence, (22) holds.

Using that for some real number  $y \in [0, 1]$ ,  $\sqrt{1-y} \leq 1 - y/2$ , we can bound  $A$  by

$$A \geq 1 - \sqrt{1 - 2mh(1-\gamma)} \geq mh(1-\gamma) := A_0. \tag{29}$$

As  $\gamma \leq 1/2$ , for (23) to hold, it is sufficient that

$$\frac{\epsilon_0}{4\sqrt{m}} \geq \frac{C}{A_0},$$

where  $C/A_0 \geq 2\alpha M \sqrt{hd}/m$ . This leads to the following sufficient condition on  $h$ ,

$$h \leq \frac{1}{m} \left[ \frac{\epsilon_0^2}{64R^2\alpha^2d} \right] \tag{30}$$

Similarly, for (24), it is sufficient that

$$\frac{\epsilon_0}{4\sqrt{m}} \geq \frac{B}{\sqrt{A_0}}$$

Now

$$\frac{B}{\sqrt{A_0}} \geq \frac{2\sqrt{h}M \sqrt{\mathbb{E} \|\hat{\theta} - \bar{\theta}\|^2 + d/m}}{\sqrt{mn}}$$

Leading to the following sufficient condition on  $n$

$$n \geq \frac{64hM^2}{\epsilon_0^2} \left[ \mathbb{E} \|\hat{\theta} - \bar{\theta}\|^2 + \frac{d}{m} \right].$$

Now due to the conditions on  $h$ , define

$$\beta := \max \left\{ \frac{1}{2L^2 + 1}, \frac{\epsilon_0^2}{64L^2\alpha^2d} \right\}.$$

Then, (24) will hold when

$$n \geq \frac{64L^2\beta}{\epsilon_0^2} m \left[ \mathbb{E} \|\hat{\theta} - \bar{\theta}\|^2 + \frac{d}{m} \right] \tag{31}$$

□

### A.3 Proof of Lemma 1

**Proof** Our proof follows similarly to Dubey et al. (2016),

$$\begin{aligned} \mathbb{E} \|\xi_k\|^2 &= \mathbb{E} \left\| \nabla \tilde{f}(\theta_k) - \nabla f(\theta_k) \right\|^2 \\ &= \mathbb{E} \left\| \nabla f_0(\theta_k) - \nabla f_0(\hat{\theta}) \right. \\ &\quad \left. + \frac{1}{n} \sum_{i \in S_k} \frac{1}{p_i} \left[ \nabla f_i(\theta_k) - \nabla f_i(\hat{\theta}) \right] \right. \\ &\quad \left. - \left[ \nabla f(\theta_k) - \nabla f(\hat{\theta}) \right] \right\|^2 \\ &\leq \frac{1}{n^2} \mathbb{E} \sum_{i \in S_k} \left\| \left[ \nabla f(\theta_k) - \nabla f(\hat{\theta}) \right] - \left( \nabla f_0(\theta_k) \right. \right. \\ &\quad \left. \left. - \nabla f_0(\hat{\theta}) + \frac{1}{p_i} \left[ \nabla f_i(\theta_k) - \nabla f_i(\hat{\theta}) \right] \right) \right\|^2. \end{aligned}$$

Where the third line follows due to independence. For any random variable  $R$ , we have that  $\mathbb{E} \|R - \mathbb{E}R\|^2 \leq \mathbb{E} \|R\|^2$ . Using this, the Lipschitz results of Assumption 2 and our choice of  $p_i$ , gives the following, where  $\mathbb{E}_I$  refers to expectation with respect to the sampled datum index,  $I$ ,

$$\begin{aligned} \mathbb{E} \|\xi_k\|^2 &\leq \frac{1}{n} \mathbb{E}_I \left( \frac{1}{p_I} \left\| \nabla f_I(\theta_k) - \nabla f_I(\hat{\theta}) \right\|^2 \right) \\ &\leq \frac{1}{n} \sum_{i=1}^N \frac{\sum_{j=1}^N L_j}{L_i} \left( L_i \|\theta_k - \hat{\theta}\|^2 \right) \\ &= \frac{1}{n} \left\{ \sum_{i=1}^N \left( \sum_{j=1}^N L_j \right) L_i \right\} \|\theta_k - \hat{\theta}\|^2, \end{aligned}$$

from which the required bound follows trivially.

### A.4 Proof of Lemma 2

**Proof** *Lipschitz condition:* By the triangle inequality

$$\begin{aligned} \left\| \nabla f(\theta) - \nabla f(\theta') \right\| &\leq \sum_{i=0}^N \left\| \nabla f_i(\theta) - \sum_{i=0}^N f_i(\theta') \right\| \\ &\leq (N + 1)L \|\theta - \theta'\|. \end{aligned}$$

*Strong convexity:* We have that

$$\begin{aligned} f(\theta) - f(\theta') - \nabla f(\theta')^\top (\theta - \theta') &= \sum_{i=0}^N \left[ f_i(\theta) - f_i(\theta') - \nabla f_i(\theta')^\top (\theta - \theta') \right] \\ &\geq \frac{(N + 1)l}{2} \|\theta - \theta'\|_2^2. \end{aligned}$$

## B Postprocessing proofs

### B.1 Proof that $\mathbb{E}[\hat{h}(\theta)] = 0$

Proof due to Friel et al. (2016). Let  $S \subset \{1, \dots, N\}$  be the minibatch chosen to estimate  $\hat{z}$ , then using the law of total expectation

$$\begin{aligned} \mathbb{E}[\hat{h}(\theta)] &= \mathbb{E} \left[ \Delta Q(\theta) + \nabla Q(\theta) \cdot \mathbb{E}[\hat{z}|S] \right] \\ &= \mathbb{E} \left[ \Delta Q(\theta) + \nabla Q(\theta) \cdot \mathbf{z} \right] = \mathbb{E}[h(\theta)] = 0. \end{aligned}$$

### B.2 Proof of Theorem 2

**Proof** We start from the bound in Theorem 6.1 of Mira et al. (2013), stating for some control variate  $h$ , the optimal variance reduction  $R$  is given by

$$R = \frac{(\mathbb{E}_{\theta|\mathbf{x}} [g(\theta)h(\theta)])^2}{\mathbb{E}_{\theta|\mathbf{x}} [h(\theta)]^2},$$

so that in our case we have

$$\begin{aligned} \hat{R} &= \frac{(\mathbb{E}_{\theta|\mathbf{x}} [g(\theta)\hat{h}(\theta)])^2}{\mathbb{E}_{\theta|\mathbf{x}} [\hat{h}(\theta)]^2} \\ &= \frac{(\mathbb{E}_{\theta|\mathbf{x}} [g(\theta)h(\theta)])^2}{\mathbb{E}_{\theta|\mathbf{x}} [h(\theta)]^2 + \frac{1}{4} \mathbb{E}_{\theta|\mathbf{x}} [\mathbf{a} \cdot \xi_S(\theta)]^2} \\ &= \frac{R}{1 + \frac{\frac{1}{4} \mathbb{E}_{\theta|\mathbf{x}} [\mathbf{a} \cdot \xi_S(\theta)]^2}{\mathbb{E}_{\theta|\mathbf{x}} [h(\theta)]^2}}. \end{aligned}$$

□ Then we can apply Lemmas 7 and 8, defined in Sect. B.3, to get the desired result

$$\hat{R} \geq \frac{R}{1 + [\sigma(N + 1)]^{-1} \mathbb{E}_{\theta|\mathbf{x}} \mathbb{E}_S \|\xi_S(\theta)\|^2}. \tag{32}$$

□

### B.3 Lemmas

**Lemma 7** Define  $A = \sum_{i=1}^d a_i^2$ , and let  $\xi_S(\theta) = \nabla \log \widehat{p}(\theta|\mathbf{x}) - \nabla \log p(\theta|\mathbf{x})$  be the noise in the gradient estimate. Then

$$\mathbb{E}_{\theta|\mathbf{x}} [\mathbf{a} \cdot \xi_S(\theta)]^2 \leq A \mathbb{E}_{\theta|\mathbf{x}} \mathbb{E}_S \|\xi_S(\theta)\|^2.$$

**Proof** We can condition on the gradient noise, and then immediately apply the Cauchy-Schwarz inequality to get

$$\square \quad \mathbb{E}_{\theta|\mathbf{x}} [\mathbf{a} \cdot \xi_S(\theta)]^2 = \mathbb{E}_{\theta|\mathbf{x}} \mathbb{E}_S [\mathbf{a} \cdot \xi_S(\theta)]^2$$

**Table 1** Minibatch sizes for each of the experiments in 5 (they were fixed for SGLD, SGLD-CV and SAGA)

Model	Minibatch size
Logistic regression	500
Probabilistic matrix factorisation	5000
LDA	50

$$\leq \left( \sum_{i=1}^d a_i^2 \right) \mathbb{E}_{\theta|x} \mathbb{E}_S \|\xi_S(\theta)\|^2$$

□

**Lemma 8** Under Assumption 4, define  $A = \sum_{i=1}^d a_i^2$ . Then  $\mathbb{E}_{\theta|x} [h(\theta)]^2 \leq A\sigma(N + 1)/4$ .

**Proof** Applying the Cauchy-Schwarz inequality

$$\begin{aligned} \mathbb{E}_{\theta|x} [h(\theta)]^2 &\leq \frac{1}{4} \left( \sum_{i=1}^d a_i^2 \right) \mathbb{E}_{\theta|x} \|\nabla f(\theta)\|^2 \\ &\leq \frac{A(N + 1)}{4} \sigma \end{aligned}$$

□

## C Experiments

### C.1 Minibatch sizes

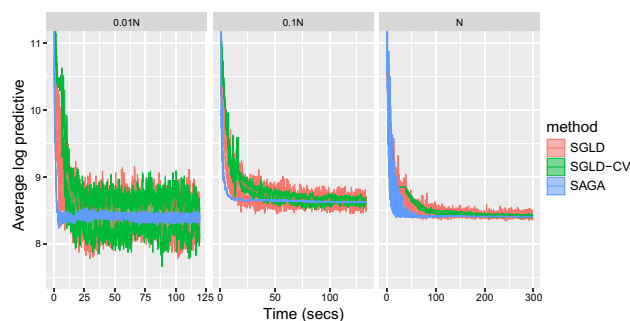
The minibatch sizes were kept fixed for each of the dataset sizes. They are given in the Table 1.

### C.2 Stepsize tuning and hyperparameters

When tuning the experiments, initially a wide grid search was used to obtain a stepsize to first order. Then, if convergence was insufficient, a more precise grid search was used. In the tables to follow, we detail the optimal stepsizes found in the different experiments. For SGLD-CV we list two sets of stepsizes: SGD and SGLD-CV. SGD corresponds to the stepsizes for the initial optimisation step, SGLD-CV corresponds to the stepsizes for the SGLD-CV algorithm itself.

#### C.2.1 Logistic regression

Alternative results using a decreasing stepsize scheme of the form  $h_k = h(1 + k/a)^{-b}$  are given in Fig. 6. We use the optimal value of  $b = .33$  (Teh et al. 2016). We again tune using a grid search and find optimal  $h$  values are the same for the fixed case, and  $a = 1000$  is the best for all dataset sizes. The tuned stepsizes are given in Table 2.



**Fig. 6** Log predictive density over a test set every 10 iterations of SGLD (with a decreasing stepsize scheme), SGLD-CV and SAGA fit to a logistic regression model as the data size  $N$  is varied

**Table 2** Tuned stepsizes for the Logistic regression experiment in Sect. 5.1

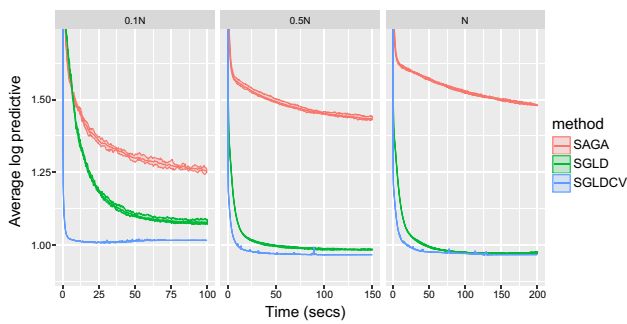
Method	0.01N	0.1N	N
SGLD	$5 \times 10^{-4}$	$5 \times 10^{-5}$	$5 \times 10^{-6}$
SGLD-CV	$5 \times 10^{-4}$	$5 \times 10^{-5}$	$5 \times 10^{-6}$
SGLD-CV (SGD)	$7 \times 10^{-5}$	$5 \times 10^{-5}$	$5 \times 10^{-6}$
SAGA	$1 \times 10^{-3}$	$1 \times 10^{-4}$	$1 \times 10^{-5}$

**Table 3** Tuned stepsizes for the Bayesian probabilistic matrix factorisation experiment in Sect. 5.2

Method	0.1N	0.5N	N
SGLD	$1 \times 10^{-3}$	$5 \times 10^{-3}$	$5 \times 10^{-3}$
SGLD-CV	$5 \times 10^{-7}$	$1 \times 10^{-6}$	$1 \times 10^{-6}$
SGLD-CV (SGD)	$1 \times 10^{-2}$	$1 \times 10^{-2}$	$1 \times 10^{-2}$
SAGA	$5 \times 10^{-3}$	$1 \times 10^{-2}$	$1 \times 10^{-2}$

#### C.2.2 Matrix factorisation

We use the formulation of BPMF as in Chen et al. (2014, Section H.2). We set  $\lambda_U, \lambda_V, \lambda_a, \lambda_b \sim \text{Gamma}(1, 300)$  and  $\tau = 3$ . This formulation has two matrix parameters  $U, V$  and two vector parameters  $a$  and  $b$  that are learnt by the chosen SGMCMC algorithm. We found these parameters tend to have quite different scales, so setting one global tuning parameter for all mixed poorly. However having four separate stepsizes would also prove difficult to tune. We opted instead to set one global tuning parameter  $h$ , detailed in Table 3, and then scale each of the stepsizes based on the relative size of  $\mathbb{E}[\|\nabla f(\theta)\|]$ . The scaling we opted for, for each parameter  $\theta$  was  $\mathbb{E} \left[ \left\| \nabla \hat{f}(\theta) \right\|^2 / d \right]$  for SGLD and SGD;  $\mathbb{E} \left[ \left\| \nabla \tilde{f}(\theta)^2 \right\| / d \right]$  for SGLD-CV; and for SAGA it was equivalent with the corresponding variance reduced gradient. Here, for a given matrix  $A, \|A\|$  corresponds to the Frobenius norm. The expectation was estimated using stochastic



**Fig. 7** Log predictive density over a test set of SGLD (with a decreasing stepsize scheme), SGLD-CV and SAGA fit to a Bayesian probabilistic matrix factorisation model as the number of users is varied, averaged over 5 runs. We used the Movielens ml-100k dataset

**Table 4** Tuned stepsizes for the latent Dirichlet allocation experiment in Sect. 5.3

Method	0.1N	0.6N	N
SGLD	$8 \times 10^{-4}$	$8 \times 10^{-5}$	$5 \times 10^{-5}$
SGLD-CV	$8 \times 10^{-4}$	$8 \times 10^{-5}$	$5 \times 10^{-5}$
SGLD-CV (SGD)	$7 \times 10^{-4}$	$1 \times 10^{-4}$	$1 \times 10^{-4}$
SAGA	$5 \times 10^{-4}$	$5 \times 10^{-5}$	$5 \times 10^{-5}$

optimisation, for example at each iteration  $k$  the scaling  $S_k$  for SGLD would be estimated by

$$S_k = \frac{1}{k} \left[ \frac{\|\nabla \hat{f}(\theta_k)\|^2}{d} - S_{k-1} \right].$$

Then, the local stepsize  $h_\theta$  for parameter  $\theta$  would be set to  $h_\theta = h/\sqrt{S_k}$ . The global stepsizes are detailed in the table.

Alternative results using a decreasing stepsize scheme of the form  $h_k = h(1 + k/a)^{-b}$  are given in Fig. 7. We use the optimal value of  $b = .33$  (Teh et al. 2016). We again tune using a grid search and find optimal  $h$  values are the same for the fixed case, and  $a = 1000$  is the best for all dataset sizes.

### C.2.3 Latent Dirichlet allocation

We used the LDA formulation of (Patterson and Teh 2013), integrating out  $\eta$ , but without the Riemannian information, and using the expanded-natural parameterisation. We use uninformative hyperparameters  $\alpha = \beta = 1$ . The tuned stepsizes are given in Table 4.

## References

Ahn, S., Korattikara, A., Liu, N., Rajan, S., Welling, M.: Large-scale distributed Bayesian matrix factorization using stochastic gradient MCMC. In: Proceedings of the 21th ACM SIGKDD interna-

- tional conference on knowledge discovery and data mining, pp. 9–18. ACM, New York (2015)
- Baker, J., Fearnhead, P., Fox, E.B., Nemeth, C.: sgmmc: An R package for stochastic gradient Markov Chain Monte Carlo. *J. Stat. Softw.* (2017). <https://arxiv.org/abs/1710.00578>
- Bardenet, R., Doucet, A., Holmes, C.: On Markov chain Monte Carlo methods for tall data. *J. Mach. Learn. Res.* **18**(47), 1–43 (2017)
- Bierkens, J., Fearnhead, P., Roberts, G.: The zig-zag process and super-efficient sampling for Bayesian analysis of big data. (2016). <https://arxiv.org/abs/1607.03188>
- Blackard, J.A., Dean, D.J.: Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Comput. Electron. Agric.* **24**(3), 131–151 (1999)
- Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)
- Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Proceedings of the 19th international conference on computational statistics, pp. 177–187. Springer, Berlin (2010)
- Chatterji, N.S., Flammarion, N., Ma, Y.-A., Bartlett, P.L., Jordan, M.I.: On the theory of variance reduction for stochastic gradient Monte Carlo. (2018). <https://arxiv.org/abs/1802.05431v1>
- Chen, C., Wang, W., Zhang, Y., Su, Q., Carin, L.: A convergence analysis for a class of practical variance-reduction stochastic gradient MCMC. (2017). <https://arxiv.org/abs/1709.01180>
- Chen, T., Fox, E., Guestrin, C.: Stochastic gradient Hamiltonian Monte Carlo. In: Proceedings of the 31st international conference on machine learning PMLR, pp. 1683–1691 (2014)
- Dalalyan, A.S., Karagulyan, A.G.: User-friendly guarantees for the Langevin Monte Carlo with inaccurate gradient. (2017). <https://arxiv.org/abs/1710.00095>
- Ding, N., Fang, Y., Babbush, R., Chen, C., Skeel, R.D., Neven, H.: Bayesian sampling using stochastic gradient thermostats. In: Advances in neural information processing systems 27, pp. 3203–3211. Curran Associates, Inc, Red Hook (2014)
- Dubey, K.A., Reddi, S.J., Williamson, S.A., Póczos, B., Smola, A.J., Xing, E.P.: Variance reduction in stochastic gradient Langevin dynamics. In: Advances in neural information processing systems, vol. 29, pp. 1154–1162. Curran Associates, Inc, Red Hook (2016)
- Durmus, A., Moulines, E.: High-dimensional bayesian inference via the unadjusted langevin algorithm. (2016). <https://hal.archives-ouvertes.fr/hal-01304430/>
- Friel, N., Mira, A., Oates, C.: Exploiting multi-core architectures for reduced-variance estimation with intractable likelihoods. *Bayesian Anal.* **11**(1), 215–245 (2016)
- Le Cam, L.: Asymptotic Methods in Statistical Decision Theory. Springer, Berlin (2012)
- Mira, A., Solgi, R., Imparato, D.: Zero variance Markov chain Monte Carlo for Bayesian estimators. *Stat. Comput.* **23**(5), 653–662 (2013)
- Mnih, A., Salakhutdinov, R.R.: Probabilistic matrix factorization. In: Advances in neural information processing systems, vol. 20, pp. 1257–1264. Curran Associates, Inc, Red Hook (2008)
- Nagapetyan, T., Duncan, A., Hasenclever, L., Vollmer, S.J., Szpruch, L., Zygalkis, K.: The true cost of stochastic gradient Langevin dynamics. (2017). <https://arxiv.org/abs/1706.02692>
- Neal, R.M.: MCMC using Hamiltonian dynamics. In: Brooks, S., Gelman, A., Jones, G.L., Meng, X.-L. (eds.) Handbook of Markov Chain Monte Carlo. Chapman & Hall, Boca Raton (2010)
- Nemirovski, A., Juditsky, A., Lan, G., Shapiro, A.: Robust stochastic approximation approach to stochastic programming. *SIAM J. Optim.* **19**(4), 1574–1609 (2009)
- Papamarkou, T., Mira, A., Girolami, M.: Zero variance differential geometric Markov chain Monte Carlo algorithms. *Bayesian Anal.* **9**(1), 97–128 (2014)



- Patterson, S., Teh, Y.W.: Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In: Advances in neural information processing systems, vol. 26, pp. 3102–3110. Curran Associates, Inc, Red Hook (2013)
- Pollock, M., Fearnhead, P., Johansen, A.M., Roberts, G.O.: The scalable Langevin exact algorithm: Bayesian inference for big data. (2016). <https://arxiv.org/abs/1609.03436>
- Ripley, B.D.: Stochastic Simulation. Wiley, Hoboken (2009)
- Robbins, H., Monro, S.: A stochastic approximation method. *Ann. Math. Stat.* **22**(3), 400–407 (1951)
- Roberts, G.O., Tweedie, R.L.: Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli* **2**(4), 341–363 (1996)
- Teh, Y.W., Thiéry, A.H., Vollmer, S.J.: Consistency and fluctuations for stochastic gradient Langevin dynamics. *J. Mach. Learn. Res.* **17**(7), 1–33 (2016)
- Tran, D., Kucukelbir, A., Dieng, A.B., Rudolph, M., Liang, D., Blei, D.M.: Edward: a library for probabilistic modeling, inference, and criticism. (2016). <https://arxiv.org/abs/1610.09787>
- Vollmer, S.J., Zygalkis, K.C.: (Non-) asymptotic properties of stochastic gradient Langevin dynamics. *J. Mach. Learn. Res.* **17**(159), 1–48 (2016)
- Welling, M., Teh, Y.W.: Bayesian learning via stochastic gradient Langevin dynamics. In: Proceedings of the 28th international conference on machine learning PMLR, pp. 681–688 (2011)
- Wenzhe Li, Sungjin Ahn, M.W.: Scalable MCMC for mixed membership stochastic blockmodels. In: Proceedings of the 19th international conference on artificial intelligence and statistics PMLR, pp. 723–731 (2016)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.