Stochastic Gradient Barker Dynamics

Jan 2022

Roadmap

1. Introduction
2. The Barker Proposal
3. vanilla-SGBD + bias $\hat{p}$
4. corrected-SGBD: $\tilde{p}$, bias correction and breaking point. + extreme-SGBD
5. Simulations: Toy case, Bayesian Log reg with scale heterogeneity, HD Bayesian Logistic regression, BPMF
6. appendix: Proof Prop on $\hat{p}$, toy example, toy log reg, additional log reg (sepsis + arr), ICA, PPMF

1 Introduction

In modern applications, large scale datasets are arising more and more frequently imposing the need for inference methods that scale easily with the number of data points. Bayesian inference is an appealing alternative in statistics and machine learning as it provides a natural way to quantify uncertainty and prevent overfitting, by characterizing the full posterior distribution. In many settings, however, sampling from the posterior distribution is a difficult task as standard methods, such as the Random Walk Metropolis [**rmw**], provide highly correlated samples and are inefficient in high dimensions. State-of-the-art Markov Chain Monte Carlo algorithms (MCMC) [21, 8, 16] tackle this problem by exploiting the gradient of the logarithm of the posterior distribution, which allows a fast exploration of the state space. However, computing the gradient requires a full pass over the entire dataset at each iteration determining a computational cost that grows linearly in the size of the dataset. This poses a challenge for Bayesian inference as the computational cost of gradient-based MCMC becomes impractical with massive datasets. Over the last years, there have been various attempts to develop sampling techniques able to scale to large dataset. One class of these methods relies on stochastic gradients, estimates of the true gradient obtained using a mini-batch subsampled from the original dataset. Such methods, known as stochastic gradient MCMC (SGMCMC), essentially consist in replacing the gradient with a computationally cheaper estimate in gradient-based schemes [26, 7, 18, 5, 6]. This idea has been originally developed in optimization [20], and applied for the first time in the context of Bayesian sampling by [26]. SGMCMC have obtained considerable popularity among practitioners as they combine a fast exploration of the state space with the scalability to dataset of massive dimensions. Most of these methods have been shown to converge to the true posterior distribution when the step-size is appropriately annealed to zero [26, 23]. However, decreasing the step-size may severely affect mixing and the step-size is often held fixed by practitioners. When the step-size is not decreased to zero, the invariant distribution may differ from the desired one [3], and the required computational budget to achieve any given level of accuracy does not differ from the non-stochastic case [15]. Since, under such circumstance, the choice of the step-size affects the stationary distribution of the algorithm, developing algorithms that are robust to this choice appears to be of paramount importance. In this paper, we present the stochastic gradient Barker dynamics algorithm (SGBD), by applying the stochastic gradient framework to the Barker dynamics algorithm [13]. The Barker dynamics algorithm is a novel gradient-based MCMC which has been to shown to outperform state of the art alternatives in terms of *robustness to hyperparameter tuning*. Our goal is to develop its scalable variant, which hopefully inherits its robustness. The main difference from existing gradient-based schemes is that the gradient is not used to generate a linear drift in the dynamics but controlls the skewness of the proposal. Hence, the magnitude of the increment of the parameter can be suitably controlled avoiding potential issues from exploding gradients in certain region of the state space. We compare SGBD to the stochastic gradient Langevin dynamics algorithm (SGLD) [23] in a variety of machine learning tasks and find that it displays a greater robustness to the choice of the hyperparameter with irregular posterior distributions and often a better predictive performance than SGLD on unseen data in a variety of machine learning tasks[1].

[1] The supporting R code to replicate the experiments can be found at ADD REPO

2 Analytical Setting

Let us consider a target distribution of the form

$$\pi(\theta) \propto \exp\left(g(\theta)\right). \tag{1}$$

In a classical Bayesian setting with conditional independent data,

$$\pi(\theta) \propto p(\theta) \prod_{i=1}^{N} p(y_i|x_i, \theta) \tag{2}$$

where $p(\theta)$ is the prior distribution of the parameter, and $p(y_i|x_i, \theta)$ is the likelihood component of the i^{th} data point. State-of-the-art MCMC algorithms [21, 8, 16] employ the information encoded in the gradient of the logarithm of the posterior distribution, $g(\theta)$, to propose new values in a informed way. The gradient of $g(\theta)$ can be written as the sum of N data points components

$$\nabla g(\theta) = \sum_{i=1}^{N} \nabla g_i(\theta) \tag{3}$$

where $\nabla g_i(\theta) = \frac{1}{N} \nabla \log p(\theta) + \nabla \log p(y_i|x_i, \theta)$. Computing $\nabla g(\theta)$ requires a loop over the entire dataset determining a cost $O(N)$ per iteration and represents the main computational bottleneck for applications with large datasets. This results in an important challenge for Bayesian inference in "big data" settings.

3 The Barker Proposal Algorithm

The *Barker Proposal Algorithm* [13] is a novel gradient-based MCMC which combines the robustness to *hyperparameter tuning* of simple schemes, such as the Random-Walk Metropolis Algorithm (RWM), with favourable high-dimensional scaling properties of gradient-based ones. In particular, the algorithm displays a decay of the order $d^{-1/3}$, as the dimension of the state space, d, increases [24], which is the same of the Metropolis-adjusted Langevin algorithm (MALA) [21], while it outperforms other gradient based alternatives in terms of robustness to hyperparameter tuning. In [13] the authors study behavior of the algorithms as the tuning parameters are chosen to be increasingly unsuitable for the problem at hand, showing that standard gradient based algorithms suffer from an exponential spectral gap decay in the degree of mismatch between the scales of the proposal and target distributions, whereas the Barker Proposal displays a linear decay as RWM.

The main idea of the Barker Proposal is to construct a continuous time Markov Jump Process, which approximately preserves the target distribution π, where the approximation is introduced to make the expression of the jump intensity tractable. The Jump Kernel of the process is then used as proposal distribution in a MCMC scheme.

Algorithm 1: The Barker Proposal Algorithm in $\mathbb{R}$

```
Input: $\theta^{(0)}, \sigma$
for $t$ =1,..., $T$ do
    Draw $z \sim \mu_\sigma(\cdot)$;
    Define $p(\nabla g(\theta^{(k-1)}), z) = \frac{1}{1+\exp(-z\nabla g(\theta^{(k-1)}))}$;
    Set $b = 1$ with probability $p(\nabla g(\theta^{(k-1)}), z)$ otherwise $b = -1$;
    Propose $\theta^{(k)} \leftarrow \theta^{(k-1)} + z \times b$ ;
end
```

Algorithm 1 proposes the increments from a base distribution. Subsequently, the sign of increment can be flipped with a probability depending on the gradient at the current location. Intuitively, the algorithm moves more likely into the direction of the gradient. If $z\nabla g(\theta) > 0$ ($z\nabla g(\theta) < 0$, respectively), then $p(\nabla g(\theta), z) > \frac{1}{2}$ ($p(\nabla g(\theta), z) < \frac{1}{2}$) and $p(\nabla g(\theta), z) \uparrow 1$ ($p(\nabla g(\theta), z) \downarrow 0$) as $z\nabla g(\theta) \uparrow +\infty$ ($z\nabla g(\theta) \downarrow -\infty$). It can be shown that the resulting proposal distribution is a skew-symmetric distribution [13]. A Metropolis-Hastings accept-reject (MH) step is then applied to remove the bias of the algorithm. Algorithm 1 is designed for $1-$dimensional targets, but the $d-$dimensional version can be derived by applying the $1-$dimensional algorithm independently to each component. Alternatively, one could define $q\left(\nabla g(\theta), z\right) = \frac{1}{1+\exp\left(-z^t\nabla g(\theta)\right))}$ and allow only two possible moves $(z, -z)$, respectively with probability $q\left(\nabla g(\theta), z\right)$ and $1-q\left(\nabla g(\theta), z\right)$. However, the authors suggest to rely on the first strategy as, for each sampled proposed increment z, it considers 2^d possible moves. One possible choice for the increment distribution is Normal distribution centered in σ with a standard deviation of 0.1σ; it can be shown that it is equivalent to a mixtures of two Normal distributions with equal weights and standard deviations (0.1σ), respectively centered in σ and $-\sigma$. Other solutions are equally viable, but a bimodal distribution is the suggested choice for the noise distribution as it gives rise to more efficient algorithm compared to a zero-mean Normal distribution [24]. Algorithm 1 has a comparable computational cost of MALA, requiring a $O(N)$ number of operations at each iteration. Therefore, the computation of the gradient represents the major computational bottleneck with large datasets, as in other gradient-based schemes.

4 The Stochastic Gradient Barker Dynamics Algorithm

In this section, we derive the stochastic-gradient version of the Barker Proposal algorithm (SGBD). In the following analysis, we work under the implicit assumption that the error generated by the finite time discretization of the diffusion is negligible with respect to the noise induced by the estimate of the gradient. At each iteration, we replace the true gradient with a minibatch estimate:

$$\hat{\nabla} g(\theta) = \frac{N}{n} \sum_{i \in \mathcal{S}_n} \nabla g_i(\theta) \tag{4}$$

where $\mathcal{S}_n$ is a minibatch of size n (with $n << N$) subsampled with or without replacement from the original dataset. One remark about our implementation is that to maintain the computational savings induced by (4), we also omit the accept-reject step. Although there are attempts to approximate the MH step using mini-batches [12, 1], this is the most common approach in the stochastic gradient MCMC literature [26, 5, 6].

4.1 Vanilla-SGBD

The vanilla implementation of SGBD (v-SGBD) consists in substituting the gradient with the estimate in equation (4) inside the Barker Proposal Algorithm. For simplicity, we only report the $1-$dimensional algorithm, since, as for Algorithm 1, the algorithm extends trivially to higher dimensions. At each iteration, we define $\hat{p}(\theta, z) = p\left(\hat{\nabla} g(\theta), z\right)$ and update θ by adding the increment z with probability $\hat{p}$, or $-z$ otherwise.

Algorithm 2: Vanilla Stochastic Gradient Barker Dynamics Algorithm

```
Input: $\theta^{(0)}, \sigma$
for $t$ =1,..., $T$ do
    Draw $\mathcal{S}_n \subset \{1, \dots, N\}$;
    Estimate $\hat{\nabla} g\left(\theta^{(k-1)}\right)$ using (4);
    Draw $z \sim N(\sigma, 0.1\sigma)$ ;
    Define $\hat{p}\left(\theta^{(k-1)}, z\right) = \frac{1}{1+\exp\left(-z\hat{\nabla} g\left(\theta^{(k-1)}\right)\right)}$;
    Set $b = 1$ with probability $\hat{p}\left(\theta^{(k-1)}, z\right)$ otherwise $b = -1$;
    Update $\theta^{(k)} \leftarrow \theta^{(k-1)} + z \times b$ ;
end
```

Understanding how the gradient noise affects the dynamics is quite challenging since p is a non-linear function of the gradient. Thus, even if an unbiased estimate of the gradient is used, we typically have $\mathbb{E}\left[p\left(\hat{\nabla} g(\theta), z\right)\right] \neq p(\nabla g(\theta), z)$, where the expectation on the left hand side is taken with respect the subsampling mechanism. This result is quite unsatisfactory, since it shows that on average the noise does not balance out. However, we show in the next section that, under suitable assumptions, we can determine the direction of the bias an approximation of such expectation, as well as, a strategy to address the bias. In particular, we identify the conditions under which it is possible to eliminate the bias. For the following analysis, we consider the $1-$dimensional case, as all considerations extend to the $d-$dimensional scenario by considering each dimension independently.

First, we identify the direction of the bias.

Proposition 1. *If the gradient noise has a symmetric distribution, then*

$$\left|p\left(\nabla g(\theta), z\right) - 0.5\right| > \left|\mathbb{E}\left[p\left(\hat{\nabla} g(\theta), z\right)\right] - 0.5\right|. \tag{5}$$

The proof of Proposition 1 is provided in the appendix. One remark is that the proposition identifies the direction of the bias. In particular, the expectation of $\hat{p}$ is shrunk towards 0.5. This phenomenon is also identified in our simulations. and intuitively, it inflates the variance of the stationary distribution as the algorithm moves less frequently towards a local mode of the distribution. This is analogous to what happens with other SGMCMC algorithms: for instance, when the step-size is held fixed, the stochastic gradient noise increseas the error of SGLD-based estimates when no correction is taken into account [25].

4.2 Corrected-SGBD

Equation (4) provides an unbiased estimate of the true gradient, but, we additionally require some regularity conditions on the distribution of the stochastic gradient noise:

Condition 1. *The stochastic gradient noise follows a unimodal and symmetric about 0 distribution: hence,*

$$\eta(\theta) \sim f_\theta(\cdot) \tag{6}$$

where $f_\theta(x)$ is non-decreasing for $x < 0$ and non-increasing elsewhere and $f_\theta(x) = f_\theta(-x) \quad \forall x \in \mathbb{R}$.

Moreover, if the minibatch size is sufficiently large, say of the order of hundreds, by Central Limit Theorem (CLT), the stochastic gradient noise is approximately Gaussian. Assuming the stochastic gradient noise to be normally distributed is a very common requirement in the stochastic gradient MCMC literature [7, 5, 6, 14] and helps the theoretical analysis. Such assumption is made precise in the following condition.

Condition 2. *The stochastic gradient noise is normally distributed: hence,*

$$\hat{\nabla} g(\theta) \sim \nabla g(\theta) + \eta(\theta) \quad \eta(\theta) \sim \mathcal{N}(0, \tau(\theta)) \tag{7}$$

where $\tau(\theta)$ is the standard deviation of the stochastic gradient noise at location θ.

The symmetry and unimodality of the noise distribution of Condition 1 is met by Condition 2, but for many result we require only Condition 1, which accommodates more general scenarios with heavy-tailed distributions.

Proposition 2. *Under Condition 2, we can derive the following bound:*

$$\left| \mathbb{E}\left[p\left(\hat{\nabla} g(\theta), z\right)\right] - p\left(\frac{1.702}{\sqrt{1.702^2 + z^2\tau^2(\theta)}} \nabla g(\theta), z \right) \right| < 0.019. \tag{8}$$

The proof of Proposition 2 is provided in the appendix. Proposition 2 provides, under Condition 2, an approximation of the expected value of $p(\hat{\nabla} g(\theta), z)$ whose absolute error can be suitably bounded. Moreover, note that the approximation corresponds to the probability that we would obtain had the true gradient shrunk by a multiplicative factor of $\frac{1.702}{\sqrt{1.702^2+z^2\tau^2(\theta)}}$. Thus, the average effect of the noise is equivalent to reducing the magnitude of the gradient by such factor in a multiplicative fashion. The result in (8) quantifies approximately the magnitude of the bias of $\hat{p}$ and suggests a strategy for eliminating it. Let us define a new class of estimators of the probability of not flipping the sign of the increment, $\tilde{p}_\alpha$,

$$\tilde{p}_\alpha\left(\hat{\nabla}g(\theta), z\right) = \frac{1}{1+\exp\left(-z\alpha\hat{\nabla}g(\theta)\right)}. \tag{9}$$

This class of estimators includes $\hat{p}$, as $\hat{p} = \tilde{p}_\alpha$, when α is set to 1. For ease of notation, we omit the explicit dependence of $\tilde{p}_\alpha$ from z and $\hat{\nabla}g(\theta)$.

We first show that under suitable assumptions α has a monotonic effect on the expected value of $\tilde{p}_\alpha$. Next, we propose an estimator of p that is approximately unbiased under suitable conditions. Finally, in the next section, we identify such conditions and suggest a strategy to adopt when they are not met.

Proposition 3. *Under Condition 1, for every $\alpha > 0$, if $z\nabla g(\theta) > 0$, $\frac{\partial}{\partial\alpha}\mathbb{E}\left[\tilde{p}_\alpha\right] > 0$, otherwise $\frac{\partial}{\partial\alpha}\mathbb{E}\left[\tilde{p}_\alpha\right] < 0$.*

The proof of the proposition is reported in the appendix. Given the result in Proposition 3, the goal is to detect the conditions under which it is possible to eliminate the bias identified in (8). In particular, in the next section, we identify, under Condition 2, the "breaking-point" of SGBD, τ^*, that is the maximum amount of gradient estimator standard deviation that SGBD can tolerate. Namely, under Condition 2, if $\tau(\theta) < \tau^*$, there exists an unbiased estimator of p in the class $\tilde{p}_\alpha$.

In general, the value τ^*, solving (12), is not known in closed form, however, resorting to the approximation of the logistic distribution function via the standard normal cumulative distribution function used to derive the bound Proposition 1, we recover an analytical estimate: $\tau^* \approx \frac{1.702}{|z|}$ (for more details see section A.2.1 in the appendix) .

If the variance of $\hat{\nabla}g(\theta)$ is not excessively large, we can effectively define an estimator of p, that is approximately unbiased under Condition 2. If $\tau(\theta) < \frac{1.702}{|z|}$ (or, equivalently, $|z| < \frac{1.702}{\tau(\theta)}$), we define the *corrected*-estimator of p as $\tilde{p}\left(\hat{\nabla}g(\theta), z\right) = \tilde{p}_{\hat{\alpha}}(\hat{\nabla}g(\theta), z)$, where $\hat{\alpha} = \frac{1.702}{\sqrt{1.702^2 - \tau^2(\theta)z^2}}$. Then, the following proposition holds.

Proposition 4. *Under Condition 2, if $\tau(\theta) < \frac{1.702}{|z|}$, then*

$$\left|\mathbb{E}_{S_n}\tilde{p}\left(\hat{\nabla}g(\theta), z\right) - p(\nabla g(\theta), z)\right| < 0.019 \tag{10}$$

Hence, under suitable conditions, $\tilde{p}$ is an approximately unbiased estimator of p. To prove Proposition 4, we derive a bound on the absolute error of the estimate of the expected value of $\tilde{p}_\alpha$ following the steps we did for Proposition 1 and replacing z with $z\alpha$. Choosing $\alpha = \frac{1.702}{\sqrt{1.702^2 - \tau^2(\theta)z^2}}$ leads to the desired result[2]. We perform a simulation to empirically illustrate these findings: we take the last sample from the vanilla-SGBD chain used to produce the Figure 5, we repeatedly subsample a mini-batch, store the gradient for a randomly selected coordinate, estimate its standard deviation, and compute $\hat{p}$ and $\tilde{p}$. Figure 1 reports the Monte Carlo average of $\hat{p}$ and $\tilde{p}$ as the value of the proposed increment. z, varies. When proposed increments are very small, the expected value of $\hat{p}$ is very close to the true value of p (i.e. the value obtained using the full gradient), and the correction performed by $\tilde{p}$ is limited. However, for greater values of z, the shrinkage effect is more evident and $\tilde{p}$ successfully reduces the bias. Moreover, this figures illustrates also the presence of the breaking-point. As z increases, the breaking-point decreases and a bias on the expectation of $\tilde{p}$ emerges.

To obtain the result in Proposition 4, we assumed the value of the variance of the gradient noise to be known. In practical applications, the theoretical value must be replaced by an estimate. In particular, we adopt an online estimate across iterations, as shown in Algorithm 3.

We refer to the resulting algorithm by substituting the naive estimate of p, $\hat{p}$, with $\tilde{p}$ as *corrected*-SGBD (c-SGBD).

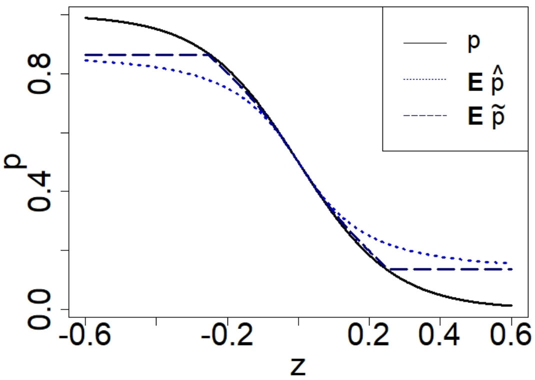


Fig. 1. Shrinkage effect and bias correction: p (black line) and Monte Carlo averages of $\hat{p}$ (dotted blue line), and $\tilde{p}$ (dashed dark blue line) vs the proposed increment z.

[2] See the section A.2.4 for the complete proof.

Algorithm 3: Corrected Stochastic Gradient Barker Dynamics Algorithm

```
Input: $\theta^{(0)}, \sigma, \beta$
for $t$ =1,..., $T$ do
    Draw $\mathcal{S}_n \subset \{1, \dots, N\}$;
    Estimate $\hat{\nabla} g(\theta^{(k-1)})$ using (4);
    Estimate $\hat{\tau}_k = \sqrt{\frac{1}{n-1} \sum_{i \in \mathcal{S}_n} (\nabla g_i(\theta^{(k-1)}) - \frac{1}{n} \sum_{i \in \mathcal{S}_n} \nabla g_i(\theta))^2}$;
+   Update $\hat{\tau}^{(k)} \leftarrow (1-\beta)\hat{\tau}^{(k-1)} + \beta \hat{\tau}_k$;
    Draw $z \sim N(\sigma, 0.1\sigma)$;
    Set $\alpha = \frac{1.702}{\sqrt{1.702^2 - \hat{\tau}^{(k)2} z^2}}$;
    Define $\tilde{p}\left(\hat{\nabla} g(\theta^{(k-1)}), z\right) = \frac{1}{1+\exp(-z\alpha \hat{\nabla} g(\theta^{(k-1)})^{(n)})}$;
    Set $b = 1$ with probability $\tilde{p}\left(\hat{\nabla} g(\theta^{(k-1)}), z\right)$ otherwise $b = -1$;
    Update $\theta^{(k)} \leftarrow \theta^{(k-1)} + z \times b$ ;
end
```

Deriving the unbiased estimate of p requires $|z| < \frac{1.702}{\tau(\theta)}$. This gives also a strategy for tuning adaptively the hyperparameter of the algorithm: when the increment z is sampled from a bimodal distribution as in Algorithm 3, around 99% of the sampled increments will be bounded, in absolute value, by 1.233σ. Hence, at each iteration, we can exploit the online estimate of the gradient estimator standard deviation $\hat{\tau}^{(k)}$ and set $\sigma^{(k)} = \frac{1.702}{\hat{\tau}^{(k)} 1.233}$. Reducing the stepsize when the stochastic gradient noise is large is somehow analogous to other strategies present in the SGMCMC literature, such as the use of “thermostats” as auxiliary variables to introduce a friction in the dynamics [6].

If this condition is not met, the solution is to use the *extreme*-estimator of p as, by Corollary 2, this strategy achieves the maximum reduction possible of the bias of $\tilde{p}$.

In our simulations, we also study the performance of the algorithm when this strategy is always applied, independently from the value of the standard deviation of the gradient noise. The resulting algorithm is referred to as *extreme*-SGBD (e-SGBD).

Algorithm 4: Extreme Stochastic Gradient Barker Dynamics Algorithm

```
Input: $\theta^{(0)}, \sigma$
for $t$ =1,..., $T$ do
    Draw $\mathcal{S}_n \subset \{1, \dots, N\}$;
    Estimate $\hat{\nabla} g(\theta^{(k-1)})$ using (4);
    Draw $z \sim N(\sigma, 0.1\sigma)$;
    Set $b = 1$ if $z\hat{\nabla} g(\theta^{(k-1)})^{(n)} > 0$ otherwise $b = -1$;
    Update $\theta^{(k)} \leftarrow \theta^{(k-1)} + z \times b$ ;
end
```

e-SGBD acts as stochastic optimization method, moving at each iteration in the direction of the gradient. Indeed, it is very similar to applying the stochastic optimization method AdaGrad [9] independently to each coordinate, where the sampled increment z acts as the step-size.

4.3 Breaking-point of SGBD

In this section, we show that, under Condition 2, there exists a maximum value of τ that SGBD can tolerate. We refer to such value as "breaking-point" of SGBD and indicate it with τ^*.

Let us define the class of symmetric estimator of p, namely the class of estimators $\hat{\hat{p}}$ satisfying the following condition:

Definition 1. *An estimator $\hat{\hat{p}}$ of p is said symmetric if it respects the following condition:*

$$\hat{\hat{p}}(\hat{\nabla} g(\theta), z) + \hat{\hat{p}}(-\hat{\nabla} g(\theta), z) = 1 \quad \forall z \in \mathbb{R}, \quad \forall \hat{\nabla} g(\theta) \in \mathbb{R}. \tag{11}$$

Notice that the class of symmetric estimators of p contains but it is not limited to the class of estimators $\tilde{p}_\alpha$. In particular, the condition in equation (11) can be equivalently restated as $\hat{\hat{p}}(\hat{\nabla} g(\theta), z) - 0.5 = 0.5 - \hat{\hat{p}}(-\hat{\nabla} g(\theta), z)$ and corresponds to requiring that the estimator as symmetric behaviour about 0.5 when the sign of the gradient is flipped. Moreover, we claim that this is the only class of estimators of p that is of practical interest, otherwise the estimator would be unjustifiably biased towards positive or negative values. Next, we show that resorting to the extreme-estimator $\bar{p}$ amounts to apply the maximum correction possible among all the symmetric estimators. We make this concept explicit in the following proposition:

Proposition 5. *Under Condition 1, if* $z\nabla g(\theta) > 0$*,*

$$\mathbb{E}\left[\hat{\hat{p}}\left(\hat{\nabla} g(\theta), z\right)\right] < \mathbb{E}\left[\bar{p}\left(\hat{\nabla} g(\theta), z\right)\right],$$

otherwise,

$$\mathbb{E}\left[\bar{p}\left(\hat{\nabla} g(\theta), z\right)\right] < \mathbb{E}\left[\hat{\hat{p}}\left(\hat{\nabla} g(\theta), z\right)\right]$$

$\forall z \in \mathbb{R}, \forall$ *symmetric estimator* $\hat{\hat{p}}$*.*

The proof of the proposition is reported in the appendix. This proposition guarantees that, under Condition 1, resorting to the extreme-estimator $\bar{p}$ is the correct strategy to reduce the bias when the noise standard deviation is greater than τ^*.

If $z\nabla g(\theta) > 0$, we know that $0.5 < \mathbb{E}\hat{p}\left(\hat{\nabla} g(\theta), z\right) < p\left(\nabla g(\theta), z\right)^3$, and, if $z\nabla g(\theta) < 0$, the reverse inequalities hold. By Corollary 2, the maximum correction that we can apply is resort to $\bar{p}$. Under Condition 2, the expectation of $\bar{p}$ is given by: $E_{S_n}\bar{p}\left(\hat{\nabla} g(\theta), z\right) = \mathbb{P}(z\hat{\nabla} g(\theta) > 0) = \begin{cases} \Phi\left(\frac{|\nabla g(\theta)|}{\tau(\theta)}\right) & \text{if } z\nabla g(\theta) > 0 \\ \Phi\left(-\frac{|\nabla g(\theta)|}{\tau(\theta)}\right) & \text{otherwise} \end{cases}$. These consideration allow us to determine the breaking point of SGBD, which is implicitly determined by the following equation:

$$\begin{cases} \Phi\left(\frac{|\nabla g(\theta)|}{\tau^*}\right) = \frac{1}{1+\exp\left(-z\nabla g(\theta)\right)} & \text{if } z\nabla g(\theta) > 0 \\ \Phi\left(-\frac{|\nabla g(\theta)|}{\tau^*}\right) = \frac{1}{1+\exp\left(-z\nabla g(\theta)\right)} & \text{otherwise} \end{cases}. \tag{12}$$

where right hand side (RHS) corresponds to the true probability p, while the left hand side (LHS) represents the expected value of $\bar{p}$. Notice that the value of the breaking point , $\tau^*\left(\nabla g(\theta), z\right) = \tau^*$ is a function of both the value of the true gradient at location θ and of the sampled increment z and, differently from other gradient-based schemes, does not depend directly on the hyperparameter σ. Therefore, τ^* is the value of the stochastic gradient noise standard deviation which equates the expected value of the extreme-estimator to p. It also represents the breaking-point of SGBD, since, as the LHS is strictly decreasing (increasing) in $\tau(\theta)$ if $z\nabla g(\theta) > 0$ ($z\nabla g(\theta) < 0$), for values of the standard deviation greater then τ^*, there is no unbiased estimator of p in the considered class $\tilde{p}_\alpha$. This impossibility result follows as simple corollary of Corollary 2.

[3] Proposition 1 identifies the 2^{nd} inequality. As for the 1^{st} one, a simple proof is provided in the appendix.

Corollary 1. *Under Condition 2, if $\tau(\theta) > \tau^*$, where τ^* is the solution to equation (12), there does not exist an unbiased estimator satisfying Condition 1.*

However, Corollary 2 also ensures that resorting to $\bar{p}$ is the optimal strategy when $\tau(\theta) > \tau^*$:

Corollary 2. *Under Condition 1, if the standard deviation noise is greater than the breaking point τ^*, where τ^* is the solution to equation (12), , then $\bar{p}$ achieves the maximum bias reduction in the class of symmetric estimators. i.e.*

$$\left|p(\nabla g(\theta), z) - \mathbb{E}\left[\bar{p}\left(\hat{\nabla} g(\theta), z\right)\right]\right| < \left|p(\nabla g(\theta), z) - \mathbb{E}\left[\hat{\hat{p}}\left(\hat{\nabla} g(\theta), z\right)\right]\right| \quad \forall \alpha > 0, \forall z \in \mathbb{R}, \tag{13}$$

$\forall \hat{\hat{p}}$ satisfying Condition 1.

To prove the corollary, it is enough to apply Proposition 5, when $\tau(\theta) > \tau^*$.

Proposition 6. *Under Condition 1, if $\tau(\theta) \leq \tau^*$, where τ^* is the solution to equation (12), there exists an unbiased estimator of p, in the class $\tilde{p}_\alpha$.*

To prove Proposition 6, consider that, if $\tau(\theta) < \tau^*$ and $z\nabla g(\theta) > 0$, $\mathbb{E}\left[p(\hat{\nabla} g(\theta), z)\right] < p(\nabla g(\theta), z) < \mathbb{E}\left[p(\hat{\nabla} g(\theta), z)\right]$, while the reserve inequalities hold when $z\nabla g(\theta) < 0$. Therefore, by continuity of $\tilde{p}_\alpha$ and by Proposition 3, if $\tau(\theta) < \tau^*$, there exists a value of α, such that $\tilde{p}_\alpha$ is an unbiased estimator of p. When $\tau(\theta) = \tau^*$, the extreme-estimator $\bar{p}$ is unbiased by how τ^* is defined in equation 12.

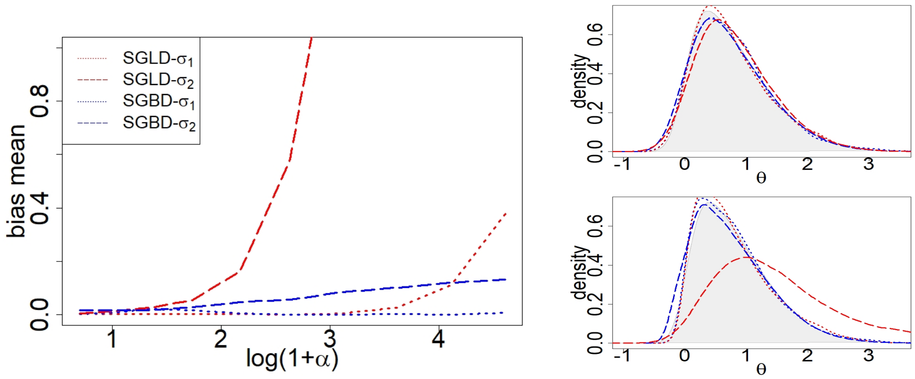
In general, the value τ^*, solving (12), is not known in closed form, however, resorting to the approximation of the logistic distribution function via the standard normal cumulative distribution function used to derive the bound Proposition 1, see section A.2.1 in the appendix for more details, we recover an analytical estimate: $\tau^* \approx \frac{1.702}{|z|}$.

5 Simulations

To study the performance of SGBD we apply the algorithm to a variety of machine learning applications and compare it to the stochastic gradient Langevin dynamics algorithm (SGLD) [26]. As regards SGLD, we consider the vanilla variant, which corresponds to the stochastic gradient version of ULA [17], a corrected variant by adjusting the standard deviation of the artificial noise to keep into consideration the stochastic gradient noise (c-SGLD) (see Algorithm 7 in section A.1 in the appendix), and an extreme variant where the maximum correction is applied and no artificial noise is added. The ladder corresponds to the stochastic gradient descent algorithm (SGD) [19]. Notice that we also tested the modified variant of SGLD proposed in [25], obtaining comparable results to c-SGLD. We report here the main results. Additional material and simulations can be found in the appendix.

5.0.1 Toy Example

We present a simplified scenario where at each iteration we add isotropic Gaussian noise to the true gradient of the target distribution to study how skewness affects the behavior of the algorithm. In particular, we use a skew-normal target distribution and study how the relative bias of the mean varies as the shape parameter α, and, correspondingly, the degree of skewness increase. Analyzing the robustness to skewness is interesting as skewed distributions arise often in machine learning tasks. Moreover, skewness generates heterogeneity in the magnitude of the gradient in different regions of the state space, making tuning gradient-based methods more challenging. In this scenario, we expect SGLD to suffer from the heterogeneity of magnitute of the gradient, while the greater robustness of the Barker scheme to tuning should allow to better handle skewed target distributions.



(a) **(b)**

Fig. 2. Toy Example: Skew-Normal target with isotropic Gaussian Noise. Shape parameter (in log-scale) vs relative bias of mean (left) and invariant distribution of the algorithm for different levels of α: 5 (top) and 20 (bottom) (right). Red refers to Langevin-based schemes and blue to Barker-based schemes. Dotted (dashed resp.) lines are produced using a step size equal to 10% (50% resp.) of the target standard deviation. The grey shaded area in the right plots represents the target distribution density.

We set the noise standard deviation τ equal to the value of the target standard deviation (a different value is also considered in the appendix) and consider two configurations of step-size σ (10% and 50% of the target standard deviation).

Figure 2a reports how the first-order bias increases as the level of skewness grows. The first remark is that algorithms based on the Langevin dynamics are less robust to skweness in the target distribution. As α increases, the target becomes increasingly unsuitable for the Langevin-based schemes which is based on a linear drift proportional to the gradient. For both choices of hyperpameter, there exists a level of skewness above which the performance of Langevin-based algorithms degrades quickly. In particular, there is a region of the state space where the gradient becomes increasingly large as α increases. When the algorithm gravitates in such region the linear drift term becomes extremely large and subsequent samples lie in the right tail of the distribution, determining a large bias in the Monte-Carlo estimates. Consequently, as α grows, the first order bias of the algorithm explodes. For the Barker-based schemes, instead, the size increment at each iteration is $|z|$, where $z \sim \mu_\sigma(\cdot)$. This allows Barker to avoid the difficulties related to exploding gradients in certain region of the space and to have a stable behaviour for all the level of skewness considered. The second remark is that the two algorithms show a different robustness to hyperpameter tuning. The Langevin-based scheme with a large step-size starts to perform poorly at moderate level of skewness while the algorithms based on the Barker dynamics displays only a small increase in the the first order bias when the step-size is increased. This difference can also be appreciated from the stationary distributions of the samplers in Figure 2b. With a slightly skewed target distribution both algorithms perform reasonably well, but, when the skewness is increased, Langevin-based schemes fail completely to sample from the correct distribution is the step size is too big (red dashed line in the right bottom plot).

5.1 Scale Heterogeneity and High-Dimensional Binary Regression

We study the performance of SGBD on two Bayesian Binary Regression tasks. The input is of the form $\mathbf{Z} = \{x_i, y_i\}$, where $x_i \in \mathbf{R}^d$ and $y_i \in \{0, 1\}$. We use the Bayesian logistic regression model which is formulated as follow:

$$\begin{aligned} y_i|\mathbf{x}_i, \theta \sim \textit{Bernoulli}\left(\frac{1}{1+\exp\left(-\theta^t \mathbf{x}_i\right)}\right) \quad & i = 1, \dots, N \\ \theta \sim \mathcal{N}_d(0, \lambda^2 I_d). & \end{aligned} \tag{14}$$

θ is assigned a zero-mean Gaussian prior distribution with diagonal covariance $\Sigma_d = \tau^2 \mathbf{I}_d$ and the likelihood of the model is given by:

$$p(\mathbf{y}|\mathbf{X}, \theta) = \prod_{i=1}^{N} \left(\frac{1}{1 + \exp\left(-\theta^t \mathbf{x}_i\right)} \right)^{y_i} \left(1 - \frac{1}{1 + \exp\left(-\theta^t \mathbf{x}_i\right)} \right)^{1-y_i}. \tag{15}$$

In our experiments, we set τ to 1. To obtain posterior quantities of interest, we run the STAN [4] implementation of the No-U-Turn sampler [11], ensuring convergence for all parameters.

5.1.1 Scale Heterogenity

In the first experiment, we focus on the robustness of SGBD to heterogeneity in the scale of the coordinates of the target distribution. In particular, we apply the samplers to a Bayesian Logistic Regression problem where variables have not been standardized, resulting in a remarkable degree of heterogeneity in the scales of univariate posterior distributions of the parameters. In particular, we use the Sepsis dataset[4], which contains 110204 instances and 4 covariates. We apply a $80\% - 20\%$ train-test split, use a mini-batch of $1\%N$ and run the algorithms for 200000 iterations. In this scenario, where N/d is high and there are not extreme outliers in the dataset, marginal posterior distributions are well approximated by Gaussian distributions. However, the absence of standardization leads to a remarkable heterogeneity of scales between the covariates. In particular, the scale of the first coordinate is several orders of magnitude smaller than the one of the other three coordinates. The algorithms are initialized in the posterior mean to focus only on mixing and sampling accuracy and remove convergence issues from the analysis.

This scenario is particularly challenging for tuning a unique step-size. Indeed, tuning the step-size to match the first coordinate results in a step-size too small for the other coordinates. Instead, with a larger step-size the accuracy of the marginal samples for the first coordinate decreases.

We notice that SGLD is much more sensitive to heterogeneity of scale. Tuning the step-size to match the scale of the first coordinate leads to very poor mixing in the other coordinates. Instead with a slightly larger step-size, it greatly inflates the variance of the first component while still suffering from a poor mixing in the other coordinates (Top plots in Figure 3). Moreover, further increasing the step-size provides samples of the first coordinate with an even bigger variance and introduces a bias in the other coordinates (bottom plots in

[4] https://archive.ics.uci.edu/ml/datasets/Sepsis+survival+minimal+clinical+records

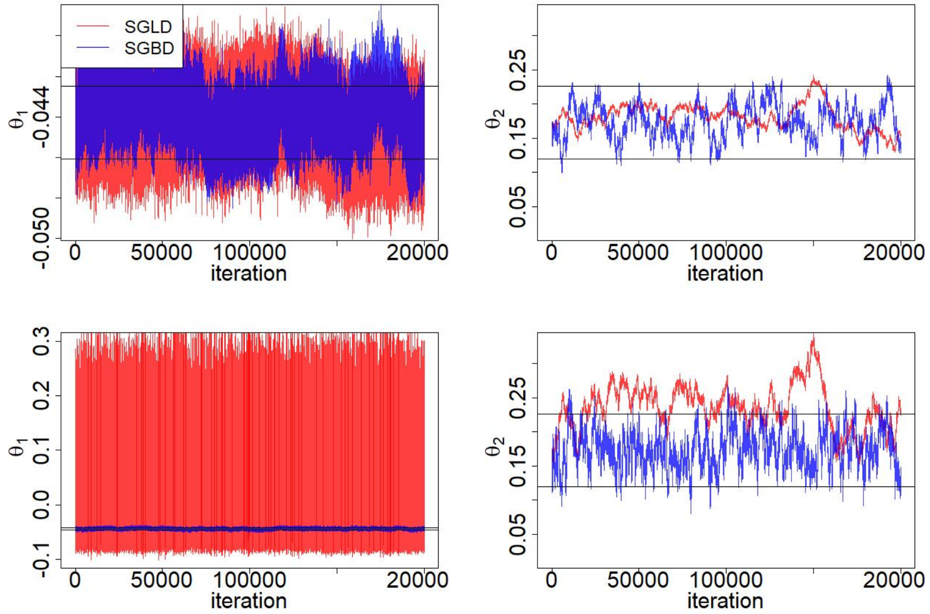


Fig. 3. Logistic Regression with Scale Heterogeneity on the Sepsis dataset. Traceplots of two coordinates with a different scale (the coordinate with the smallest scale is on the left) with two step-size configuration: small σ (top), large σ (bottom). Red refers to SGLD and blue to the SGBD. Black horizontal lines represent the interval around the posterior mean with a two standard deviations width

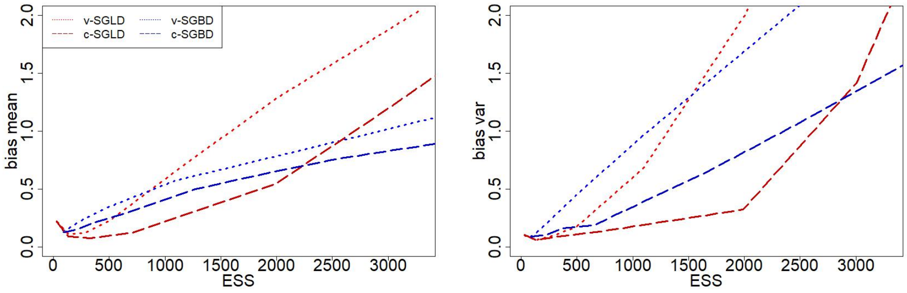
Figure 3). Instead, SGBD is remarkably more robust to scale heterogeneity. It only inflates the variance of the first coordinates and has a small bias for the mean, and shows a good mixing for the other coordinates, providing good approximations of their marginals (for the marginal density estimates see plots in the appendix). In addition, it shows an appealing robustness to hyperpameter tuning. Increasing the step-size only leads to a limited increase in the variance of the marginal stationary distribution in the first component.

5.1.2 High-Dimensional

Then, we study the performance of SGBD on ill-conditioned high dimensional logistic regression task using model (14). We apply the model to the Arrhythmia dataset from the UCI repository [5]. The dataset contains 452 instances and 279 covariates, from which we retain the first 100.

[5] https://archive.ics.uci.edu/ml/datasets/arrhythmia

We are interested in how hyperparameter tuning affects the sampling accuracy of the algorithm. In particular, we study the trade-off between mixing and sampling accuracy, since increasing the step size of the algorithms produces better mixing but less accurate chains, as no MH step is used. Mixing is measured with the median effective sample size (ESS) across the parameters and sampling accuracy with the mean standardized 1^{st} and 2^{nd} order bias[6]. SGBD appears to be more robust to hyperparameter tuning. In particular, it enjoys a more favourable mixing-accuracy trade off when the step size is chosen increasingly large. While SGLD shows a greater accuracy for low values of mixing (i.e. when the step-size is small), as the step-size is increased, SGBD displays a greater robustness and a better sampling accuracy.



(a) ESS vs bias of mean

(b) ESS vs bias of variance

Fig. 4. Logistic Regression Mixing-Accuracy trade off on the Arrhythmia dataset. Median ESS vs mean bias of mean (left) and mean bias of variance (right). Red refers to SGLD and blue to the SGBD. Lighter dotted lines refer to vanilla implementations of the algorithm, darker dashed lines to their corrected variants.

We also evaluate the log-loss on the test est for each configuration of hyperparameter, considering also the extreme variants of the samplers. The log-loss is defined as

$$l(\theta, T) = -\frac{1}{|T|} \sum_{i \in T} y_i \log\left(\hat{p}(\mathbf{x}_i, \theta)\right) + (1 - y_i) \log\left(1 - \hat{p}(\mathbf{x}_i, \theta)\right) \tag{16}$$

where T is the test set and $\hat{p}(\mathbf{x}_i, \theta)$ is the MCMC estimate of the probability of $Y_i = 1$ given the predictors and the parameter θ.

[6] We compute the bias of the mean and variance of each coordinate rescaled by its posterior standard deviation.

Figure 5 reports the log-loss on the test set of a single MCMC chain. Hyper-parameters are chosen such that the median ESS of the samples is equal to 1000. In general, SGBD outperforms SGLD in all variants, and the corrected variant of SGBD obtains the highest accuracy. Similar results are obtained using different configurations of step-sizes (see Figure 15 in the appendix).

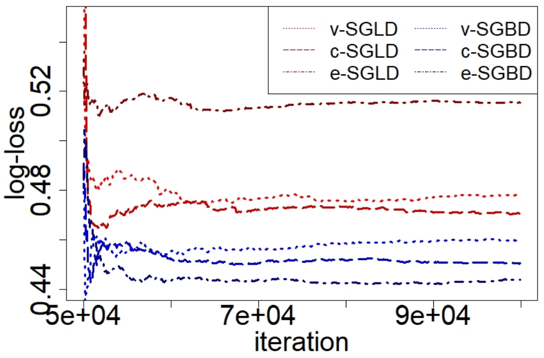


Fig. 5. Logistic Regression Mixing-Accuracy Predictive Accuracy on the Arrhythmia dataset. Log-loss of MCMC estimates. Red refers to SGLD and blue to the SGBD. For both algorithms, the vanilla (lighter dotted lines), corrected (medium scale dashed lines) and extreme (darker dotted-dashed lines) versions are displayed.

5.2 Independent Component Analysis

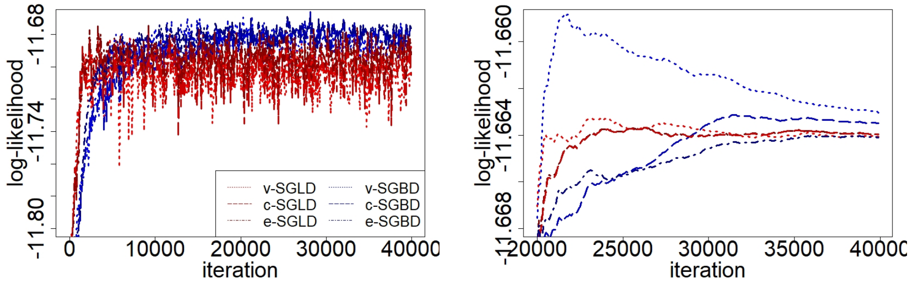
We evaluate the performance of SGBD on an independent component analysis. The input data is of the form $\mathbf{X} = \{x_i\}_{i=1}^N$ where each $x_i \in \mathbb{R}^d$. The model is defined as follows:

$$p(x_i|W) \propto |\det W| \prod_{k=1}^{d} p(y_{ik}) \quad W_{ij} \sim N(0, \lambda^{-1}) \tag{17}$$

where $y_{ik} = w_k^t x_i$ and $p(y) = \frac{1}{4\cosh^2(0.5y)}$. We apply the model to the MEG dataset[7], which contains 17730 data points of dimensionality 100. To perform our experiments, we extract the first 10 channels and set $\lambda = 1$. We perform a $80\% - 20\%$ train-test split and compare log-likelihood on the unseen test set. We run the samplers for 40000 iterations choosing a batch-size of 100.

We test the performance on the unseen test set, computing the log-likelihood produced by each sample, as well as, by the MCMC estimates. Both algorithms require very small hyperparmeters to work. Under such settings, SGBD and SGLD have a very similar behavior. In addition, we do not highlight particular differences between vanilla and corrected versions and both algorithm converge

[7] http://research.ics.aalto.fi/ica/eegmeg/MEG_data.html



(a) Samples log-likelihood

(b) MCMC estimates log-likelihood

Fig. 6. ICA: log likelihood on the MEG dataset. Log likelihood produced by each sample (left) and by the MCMC estimate (right). Red refers to SGLD and blue to the SGBD. For both algorithms, the vanilla (lighter dotted lines), corrected (medium scale dashed lines) and extreme (darker dotted-dashed lines) versions are displayed.

considerably fast to stationarity. The corrected-SGBD estimates are able to produce higher levels of log-likelihood on unseen data.

5.3 Bayesian Probabilistic Matrix Factorization

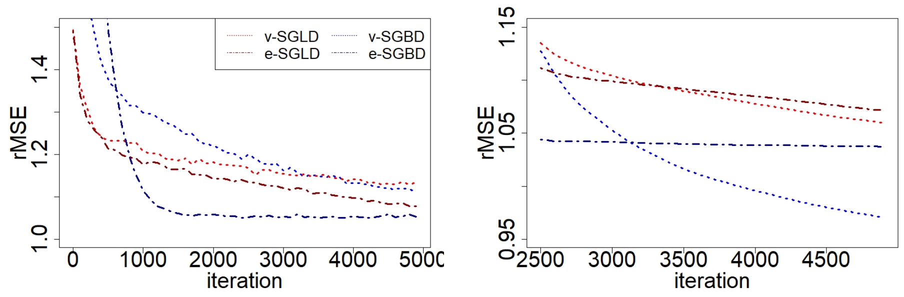
Finally, we study the performance of SGBD on a content recommendation task tackled with probabilistic matrix factorization techniques. In particular, let us consider to have U users rating M items. The data comes in the form of a matrix $\mathbf{R} \in \mathbb{R}^{U \times M}$, where the entry R_{ij} corresponds to rating given by user i to the item j. Clearly, the matrix $\mathbf{R}$ contains many zeros and can be factorized into two lower dimensional matrices. We use the MovieLens dataset[8] containing 100000 ratings (taking values in $\{1, 2, 3, 4, 5\}$) of 1000 users on 1700 movies to which we apply a 80%−20% train-test split. We present here the results using the classical Bayesian matrix factorization model (BPMF) [22]; in the appendix, we report the results of an alternative probabilistic matrix factorization technique. BPMF uses a Normal likelihood and a Normal prior on the mean of the columns of the two lower dimensional matrices. In particular, the ratings matrix is factorized into the matrices $\mathbf{U} \in \mathbb{R}^{U \times d}$ and $\mathbf{V} \in \mathbb{R}^{M \times d}$ and the BPMF model is formulated

[8] https://grouplens.org/datasets/movielens/100k

as follows:

$$\begin{aligned} R_{ij}|\mathbf{U},\mathbf{V},\alpha,I_{ij}=1 &\sim \mathcal{N}(\mathbf{U}_i^t\mathbf{V}_j,\alpha^{-1}) \quad i=1,\dots,U \quad j=1,\dots,M \\ \mathbf{U}_i|\mu_{\mathbf{U}},\Lambda_{\mathbf{U}} &\sim \mathcal{N}_d(\mu_{\mathbf{U}},\Lambda_{\mathbf{U}}^{-1}) \quad i=1,\dots,U \\ \mathbf{V}_j|\mu_{\mathbf{V}},\Lambda_{\mathbf{V}} &\sim \mathcal{N}_d(\mu_{\mathbf{V}},\Lambda_{\mathbf{V}}^{-1}) \quad j=1,\dots,M \\ \mu_{\mathbf{U}}|\mu_0,\Lambda_{\mathbf{U}} &\sim \mathcal{N}_d(\mu_0,\Lambda_{\mathbf{U}}^{-1}) \\ \mu_{\mathbf{V}}|\mu_0,\Lambda_{\mathbf{V}} &\sim \mathcal{N}_d(\mu_0,\Lambda_{\mathbf{V}}^{-1}) \\ \Lambda_{\mathbf{U}}|a_0,b_0 &\sim \Gamma(a_0,b_0) \\ \Lambda_{\mathbf{V}}|a_0,b_0 &\sim \Gamma(a_0,b_0) \end{aligned} \tag{18}$$

where $I_{ij}=1$ if the i^{th} user has rated the j^{th} item. For our experiments, we set $d=20$, $\alpha=3$, $\mu_0=0$, $a_0=1$ and $b_0=5$ and used a mini batch of size 800, which corresponds to $1\%N$.



(a) Sample rMSE

(b) MCMC rMSE

Fig. 7. Bayesian Probabilistic Matrix Factorization: Predictive Accuracy on the MovieLens dataset. Sample (left) and MCMC (right) estimates rMSE. Red refers to SGLD and blue to the SGBD. Lighter and dotted lines refer to vanilla implementations of the algorithm, darker and dashed-dotted lines to their extreme variants.

We evaluate the root mean squared error (rMSE) of the vanilla and extreme variants of the samplers. The rMSE is computed after clipping the ratings predicted values at 1 and 5, where the predicted rating for the j^{th} item given by the i^{th} user is $\hat{R}_{ij}=\mathbf{U}_i^t\mathbf{V}_j$. Similar results are obtained without clipping the predictions. We consider both each sample predictions, as well as, MCMC estimates of the predicted ratings, which are given by the the ergodic averages of the predicted ratings, after the burn-in. In general, SGBD outperforms SGLD in all variants. The exteme variant of stochastic Barker converges faster and single samples have a better predictive accuracy. However vanilla-SGBD MCMC estimates have the best overall predictive accuracy.

6 Conclusion

The "Stochastic Gradient Barker Dynamics" algorithm is a novel technique for approximate posterior sampling. We have presented its vanilla version as well as corrected variants to deal with the effect of the stochastic gradient noise. SGBD represents a valid alternative to SGLD with a minimal algorithmic change.

This method outperforms SGLD in terms of robustness to tuning and to irregularities in the posterior distribution, and shows an appealing predictive accuracy on unseen data on a variety of machine learning tasks.

Interesting extensions include adding momentum and characterizing how the invariant distribution is affected by the choice of the step-size. In addition, it would be worthwile to further explore the connection with optimization schemes, which we leave to future work.

A Appendix

A.1 Miscellaneous

We report the pseudo-code for the algorithm used in the simulations reported in section 5.

Algorithm 5: Stochastic Gradient Langevin Dynamics (SGLD)

```
Input: $\theta^{(0)}, \{h_1, \ldots, h_K\}$
for $t$ =1,..., $T$ do
    Draw $\mathcal{S}_n \subset \{1, \ldots, N\}$;
    Estimate $\hat{\nabla} U(\theta^{(t-1)})^{(n)}$ using (4);
    Estimate $\hat{\tau}_k = \sqrt{\frac{1}{n-1} \sum_{i \in \mathcal{S}_n} (\nabla g_i(\theta^{(k-1)}) - \frac{1}{n} \sum_{i \in \mathcal{S}_n} \nabla g_i(\theta))^2}$;
    Draw $\epsilon_k \sim N(0, h_k I)$;
    Update $\theta^{(t)} \leftarrow \theta^{(t-1)} - \frac{h_k}{2} \hat{\nabla} U(\theta^{(t-1)})^{(n)} + \epsilon_k$;
end
```

Algorithm 6: Modified Stochastic Gradient Langevin Dynamics (m-SGLD)

```
Input: $\theta^{(0)}, \{h_1, \ldots, h_K\}$
for $t$ =1,..., $T$ do
    Draw $\mathcal{S}_n \subset \{1, \ldots, N\}$;
    Estimate $\hat{\nabla} U(\theta^{(t-1)})^{(n)}$ using (4);
    Estimate $\mathit{cov}\left(\hat{\nabla} U(\theta^{(t-1)})^{(n)}\right)$;
    Draw $\epsilon_k \sim N(0, \sqrt{h_k} \left(I - \frac{h_k}{2} c\hat{o}v \left(\hat{\nabla} U(\theta^{(t-1)})^{(n)}\right)\right)$;
    Update $\theta^{(t)} \leftarrow \theta^{(t-1)} - \frac{h_k}{2} \hat{\nabla} U(\theta^{(t-1)})^{(n)} + \epsilon_k$;
end
```

Algorithm 7: Corrected Stochastic Gradient Langevin Dynamics (c-SGLD)

```
Input: $\theta^{(0)}, \{h_1, \dots, h_K\}$
for $t$ =1,..., $T$ do
    Draw $\mathcal{S}_n \subset \{1, \dots, N\}$;
    Estimate $\hat{\nabla} U(\theta^{(t-1)})^{(n)}$ using (4);
    Estimate $\hat{\tau}_k = \sqrt{\frac{1}{n-1} \sum_{i \in \mathcal{S}_n} (\nabla g_i(\theta^{(k-1)}) - \frac{1}{n} \sum_{i \in \mathcal{S}_n} \nabla g_i(\theta))^2}$;
    Update $\hat{\tau}^{(k)} \leftarrow (1-\beta)\hat{\tau}^{(k-1)} + \beta \hat{\tau}_k$;
    Draw $\epsilon_k \sim N\left(0, \sqrt{h_k^2 - \frac{\hat{\tau}^{(k)2}}{4} h_k^4}\right)$;
    Update $\theta^{(t)} \leftarrow \theta^{(t-1)} - \frac{h_k}{2} \hat{\nabla} U(\theta^{(t-1)})^{(n)} + \epsilon_k$;
end
```

A.2 Proofs

In this section, we report the computations relative to the expected value of the probability of the sign flip. In the following analysis, we consider the unidimensional case. The arguments extend to the multidimensional case coordinate-wise. For ease of notation, we drop the explicit dependence of all the quantities on the location θ. Let us make the dependence of the probability of the sign flip on the gradient noise explicit:

$$p\left(\nabla g + \eta, z\right) = (1 + \exp(-z(\nabla g + \eta)))^{-1}. \tag{19}$$

We are interested in

$$E_\eta p\left(\nabla g + \eta, z\right) = \int_{-\infty}^{+\infty} p\left(\nabla g + \eta, z\right) dP_\eta(\eta) \tag{20}$$

where η is the stochastic gradient noise, z and ∇g are considered fixed, and the expectation is taken with respect to the noise distribution P_η. Since the only source of the stochastic gradient noise at any given location θ is the subsampling mechanism, this is equivalent to taking the expectation with respect to that mechanism.

A.2.1 Proof of Proposition 1

We first show that, if the distribution of the gradient noise is symmetric, the expected value of $\hat{p}$ is shrunk towards 0.5 (i.e $|p - 0.5| > |\mathbb{E}\hat{p} - 0.5|$).

Let us assume that $z\nabla g > 0$. We first show that under such assumptions $\mathbb{E}\hat{p} < p$. The arguments extend trivially to other cases. If P_η is symmetric, the equation (21) can be written as

$$\mathbb{E}_\eta p\left(\nabla g + \eta, z\right) = \int_0^{+\infty} \left(p\left(\nabla g + \eta, z\right) + p\left(\nabla g - \eta, z\right)\right) dP_\eta(\eta) \tag{21}$$

Note that

$$\frac{1}{2}\left(p\left(\nabla g + \eta, z\right) + p\left(\nabla g - \eta, z\right)\right) = \frac{1}{2}\left(\frac{1}{1 + ae^{-z\eta}} + \frac{1}{1 + ae^{z\eta}}\right) < \frac{1}{1+a} = \frac{1}{1+e^{-z\nabla g}} = p(\nabla g, z) \tag{22}$$

where $a = \exp\left(-z\nabla g\right) \in (0,1)$ The term on the left hand side is equal to

$$\frac{1}{2}\frac{2 + a(e^{(-z\eta)} + e^{(z\eta)})}{1 + a(e^{(-z\eta)} + e^{(z\eta)}) + a^2} \tag{23}$$

Let us define N and D respectively as the numerator and the denominator in (23). Next, we show that and $N(1+a) < D$. $N(1+a) = 2 + a(e^{-z\eta} + e^{z\eta}) + 2a + a^2(e^{-z\eta} + e^{z\eta})$ and $D - N(1+a) = (a - a^2)(e^{-z\eta} + e^{z\eta} - 2) > 0$ since $a \in (0,1)$. The inequality (22) follows. Then, equation (21) can be upper bounded by

$$2\int_0^{+\infty} p(\nabla g(\theta), z) dP_\eta(\eta) = p(\nabla g(\theta), z) \tag{24}$$

where the last equality follows from $p(\nabla g(\theta), z)$ not depending on η and $P_\eta(\eta)$ being symmetric about 0. Hence, we showed taht, when the distribution of the noise is symmetric about 0 and $z\nabla g > 0$, the probability of the sign flip is shrunk towards 0.5

$$\mathbb{E}\left[\hat{p}\left(\hat{\nabla} g(\theta), z\right)\right] < p(\nabla g(\theta), z). \tag{25}$$

Moreover, it can be easily shown that $\mathbb{E}\hat{p} > 0.5$. In particular, $p(\nabla g - \eta, z) + \hat{p}(\nabla g - \eta, z) > 0.5$, therefore, a lower bound to (21) is given by

$$\int_0^{+\infty} 0.5 dP(\eta) = 0.5$$

. Hence we proved that, if $zg > 0$, $0.5 < \mathbb{E}\hat{p} < p$. Analogously, if $z\nabla g < 0$, it is possible to show the reverse inequality (i.e. $p < \mathbb{E}\hat{p} < 0.5$) leading to the bound in Proposition 1. □

A.2.2 Proof of Proposition 2

Next, we provide an approximation of (21) under Condition 2 with a corresponding bound on the absolute error. If we assume that the gradient noise is normally distributed (as in (2)), (21) reduces to

$$\mathbb{E}_\eta \tilde{p}\left(\nabla g + \eta, z\right) = \int_{-\infty}^{\infty} \frac{1}{1 + \exp\left(-z(\nabla g + \eta)\right)} \frac{1}{\sqrt{2\pi}\tau} \exp\left(-\frac{1}{2}\frac{\eta^2}{\tau^2}\right) d\eta \tag{26}$$

where τ is the standard deviation of the gradient noise. The integral in equation (26) is intractable, but we can resort to an approximation. The cumulative distribution function of a random variable following a logistic distribution, $F(\cdot)$, can be approximated with the standard Normal CDF, $\Phi(\cdot)$, in the following manner

$$F(x) \approx \Phi\left(\frac{x}{1.702}\right) \quad \forall x \in \mathbb{R}. \tag{27}$$

(27) is the approximation of the logistic CDF via a standard Normal CDF that minimizes the maximum absolute variation. In particular, $\left|F(x) - \Phi\left(\frac{x}{1.702}\right)\right| < 0.0095 \quad \forall x \in \mathbb{R}$ [2]. Let us approximate (26), with

$$\mathbb{E}\left[\hat{p}\left(\hat{\nabla} g, z\right)\right] \approx \mathbb{E}_\eta \Phi\left(\frac{z(\nabla g + \eta)}{1.702}\right) \tag{28}$$

where the expectation on the right hand side is taken with respect to the distribution of the gradient noise. The right hand side of (28) exits in closed form:

$$\mathbb{E}_\eta \Phi\left(\frac{z(\nabla g + \eta)}{1.702}\right) = \Phi\left(\frac{z\nabla g}{\sqrt{1.702^2 + z^2\tau^2}}\right) \tag{29}$$

Let us derive the bound in Proposition 1:

$$\begin{gathered}\left|\mathbb{E}\left[\hat{p}\left(\hat{\nabla} g(\theta), z\right)\right] - \Phi\left(\frac{z\nabla g}{\sqrt{1.702^2 + z^2\tau^2}}\right)\right| = \\ \left|\int_{-\infty}^{\infty} \left(\hat{p}\left(\hat{\nabla} g, z\right) - \Phi\left(\frac{z(\nabla g + \eta)}{1.702}\right)\right) \frac{1}{\sqrt{2\pi}\tau} \exp\left(-\frac{1}{2}\frac{\eta^2}{\tau^2}\right) d\eta\right|\end{gathered} \tag{30}$$

(30) can be upper bounded by

$$\begin{gathered}\int_{-\infty}^{\infty} \left|\hat{p}\left(\hat{\nabla} g, z\right) - \Phi\left(\frac{z(\nabla g + \eta)}{1.702}\right)\right| \frac{1}{\sqrt{2\pi}\tau} \exp\left(-\frac{1}{2}\frac{\eta^2}{\tau^2}\right) d\eta < \\ \int_{-\infty}^{\infty} \max\left|\hat{p}\left(\hat{\nabla} g, z\right) - \Phi\left(\frac{z(\nabla g + \eta)}{1.702}\right)\right| \frac{1}{\sqrt{2\pi}\tau} \exp\left(-\frac{1}{2}\frac{\eta^2}{\tau^2}\right) d\eta = 0.0095\end{gathered} \tag{31}$$

Hence, $\left|\mathbb{E}\left[\hat{p}\left(\hat{\nabla}g, z\right)\right] - \Phi\left(\frac{z\nabla g}{\sqrt{1.702^2+z^2\tau^2)}}\right)\right| < 0.0095$. Moreover, considering the bound on the approximation (27) and applying the triangle inequality lead to (8). □

A.2.3 Proof of Proposition 3

In this section, we prove that for an estimator defined as in (9), its expected value is increasing in α assuming that $z\nabla g > 0$ (and it is decreasing otherwise) and that the stochastic gradient noise had a unimodal and symmetric about zero density. Therefore, for the class of estimators, $\tilde{p}$, of p, the extreme estimator achieves the maximum bias reduction possible when the noise is sufficiently large. Let us consider the derivative with respect to α of $\mathbb{E}\left[\tilde{p}\left(\hat{\nabla}g, z\right)\right]$:

$$\frac{\partial}{\partial\alpha}\mathbb{E}\left[\tilde{p}\left(\hat{\nabla}g, z\right)\right] = \frac{\partial}{\partial\alpha}\int_{-\infty}^{\infty}\frac{1}{1+\exp\left(-z\alpha(\nabla g+\eta)\right)}f_0(|\eta|)d\eta \tag{32}$$

where $f_0(\cdot)$ is the density of η and, since $f_0(\cdot)$ is symmetric about 0, $f_0(x) = f_0(|x|)$ for all $x \in \mathbb{R}$ In particular, as the partial derivative of the integrand with respect to α is continuous we can apply the Leibniz rule and interchange the integral with the derivate. If we further perform a change of variable ($x := \nabla g + \eta$), (32) is equal to

$$\frac{\partial}{\partial\alpha}\mathbb{E}\left[\tilde{p}\left(\hat{\nabla}g, z\right)\right] = \int_{-\infty}^{\infty}\frac{\exp\left(-z\alpha x\right)}{\left(1+\exp\left(-z\alpha x\right)\right)^2}zxf_0(|x-\nabla g|)dx \tag{33}$$

which corresponds to the integral of an odd function (positive on the interval $(0, +\infty)$), with respect to a measure with a unimodal and symmetric density about $\nabla g > 0$. When $z < 0$ and $\nabla g < 0$, (32) corresponds to an integral of an odd function (negative on the interval $(0, +\infty)$), with respect to a measure with a unimodal and symmetric density about $\nabla g < 0$. Therefore, in both cases, we can conclude that

$$\frac{\partial}{\partial\alpha}\mathbb{E}\left[\tilde{p}\left(\hat{\nabla}g, z\right)\right] > 0. \tag{34}$$

Similarly, it possible to show that if $z\nabla g < 0$, (32) is negative. □

As a final remark, we note that the unimodality and symmetry assumptions are met by Condition 2 but hold for a much more general class of distributions.

A.2.4 Proof of Proposition 4

To prove Proposition 4, it is enough to follow the steps of the proof of Proposition 1, replacing z wit $z\alpha$. In particular, we obtain

$$\left|\mathbb{E}\left[\tilde{p}_\alpha\left(\hat{\nabla} g, z\right)\right] - p\left(\frac{1.702\alpha}{\sqrt{1.702^2 + z^2\tau^2\alpha^2}}\nabla g, z\right)\right| < 0.019. \tag{35}$$

If we choose $\alpha = \frac{1.702}{\sqrt{1.702^2 - \tau^2 z^2}}$ the second term in the absolute value of (35) simplifies to $p(\nabla g, z)$ leading to the desired result. □

A.2.5 Proof of Proposition 5

In this section, we show that, if the distribution of the gradient estimator is unimodal and symmetric about its mean, the estimator of p, $\bar{p}$, defined in section 4.3, is the *extreme*-estimator among all estimators $\hat{\hat{p}}$ satisfying the following condition $\forall z \in \mathbb{R} \quad \forall \hat{\nabla} g \in \mathbb{R} \quad \text{such that} \quad z\hat{\nabla} g > 0 \quad \hat{\hat{p}}(\hat{\nabla} g, z) - \frac{1}{2} = \frac{1}{2} - \hat{\hat{p}}(-\hat{\nabla} g, z)$, namely $\mathbb{E}\bar{p}(\hat{\nabla} g, z) > \mathbb{E}\hat{\hat{p}}(\hat{\nabla} g, z)$ if $\hat{\nabla} g z > 0$ and $\mathbb{E}\left[\bar{p}(\hat{\nabla} g, z)\right] < \mathbb{E}\left[\hat{\hat{p}}(\hat{\nabla} g, z)\right.$ otherwise for all such $\hat{\hat{p}}$. We prove the case when $z > 0$ and $\hat{\nabla} g > 0$, since, as in the previous case, the proof can be easily extended to all other cases. Let us consider the expectation of such estimator:

$$\begin{gathered}\mathbb{E}\left[\hat{\hat{p}}(\hat{\nabla} g, z)\right] = \int_{-\infty}^{+\infty} \hat{\hat{p}}(\hat{\nabla} g, z) f_{\nabla g}(|\hat{\nabla} g - \nabla g|) d\hat{\nabla} g = \\ \int_{-\infty}^{0} \hat{\hat{p}}(\hat{\nabla} g, z) f_{\nabla g}(|\hat{\nabla} g - \nabla g|) d\hat{\nabla} g + \int_{0}^{+\infty} \hat{\hat{p}}(\hat{\nabla} g, z) f_{\nabla g}(|\hat{\nabla} g - \nabla g|) d\hat{\nabla} g\end{gathered} \tag{36}$$

Notice that we can rewrite the expectation of $\bar{p}$ as follows:

$$\begin{aligned}\mathbb{E}\left[\bar{p}(\hat{\nabla} g, z)\right] &= \int_{-\infty}^{+\infty} \bar{p}(\hat{\nabla} g, z) f_{\nabla g}(|\hat{\nabla} g - \nabla g|) d\hat{\nabla} g = \int_{0}^{+\infty} 1 f_{\nabla g}(|\hat{\nabla} g - \nabla g|) d\hat{\nabla} g \\ &= \int_{0}^{+\infty} \hat{\hat{p}}(-\hat{\nabla} g, z) f_{\nabla g}(|\hat{\nabla} g - \nabla g|) d\hat{\nabla} g + \int_{0}^{+\infty} \hat{\hat{p}}(\hat{\nabla} g, z) f_{\nabla g}(|\hat{\nabla} g - \nabla g|) d\hat{\nabla} g\end{aligned} \tag{37}$$

where the last equality follows from the fact that $\hat{\hat{p}}(-\hat{\nabla} g, z) + \hat{\hat{p}}(\hat{\nabla} g, z) = 1$ for all $\hat{\nabla} g, z$. In addition, by a change of variable, we can rewrite the first summand in (36) as:

$$\int_{0}^{\infty} \hat{\hat{p}}(-\hat{\nabla} g, z) f_{\nabla g}(|-\hat{\nabla} g - \nabla g|) d\hat{\nabla} g \tag{38}$$

which is smaller than the first summand in (37) since, $\forall \hat{\nabla} g \quad f_{\nabla g}(|-\hat{\nabla} g - \nabla g|) < f_{\nabla g}(|\hat{\nabla} g - \nabla g|)$, for any unimodal distribution symmetric about ∇g, $f_{\nabla g}(\cdot)$. □

A.3 Approximation of the Breaking-Point τ^*

The breaking-point is implicitly determined equation 12.

$$\begin{cases} \Phi\left(\frac{|\nabla g(\theta)|}{\tau^*}\right) = \frac{1}{1+\exp\left(-z\nabla g(\theta)\right)} & \text{if } z\nabla g(\theta) > 0 \\ \Phi\left(-\frac{|\nabla g(\theta)|}{\tau^*}\right) = \frac{1}{1+\exp\left(-z\nabla g(\theta)\right)} & \text{otherwise} \end{cases}. \tag{39}$$

We focus on the case $z > 0, \nabla g > 0$. Then the breaking point is given by $\Phi\left(\frac{|\nabla g|}{\tau^*}\right) = \frac{1}{1+\exp\left(-z\nabla g\right)}$. Next, we can approximate the RHS of the equation resorting to the approximation mentioned in the proof of Proposition 2 to obtain the following:

$$\Phi\left(\frac{\nabla g}{\tau^*}\right) \approx \Phi\left(\frac{z\nabla g(\theta)}{1.702}\right) \tag{40}$$

or, equivalently,

$$\tau^* \approx \frac{1.702}{z} \tag{41}$$

It easy to extend this computation to all other cases of z and ∇g, leading to following the approximation for the general case

$$\tau^* \approx \frac{1.702}{|z|}. \tag{42}$$

Moreover, it possible to get an interval range of the breaking point instead of a point approximation. Recall that

$$\Phi\left(\frac{\nabla g}{\tau^*}\right) \in \left[\Phi\left(\frac{z\nabla g(\theta)}{1.702}\right) - 0.0095, \Phi\left(\frac{z\nabla g(\theta)}{1.702}\right) + 0.0095\right] \tag{43}$$

which is equivalent to

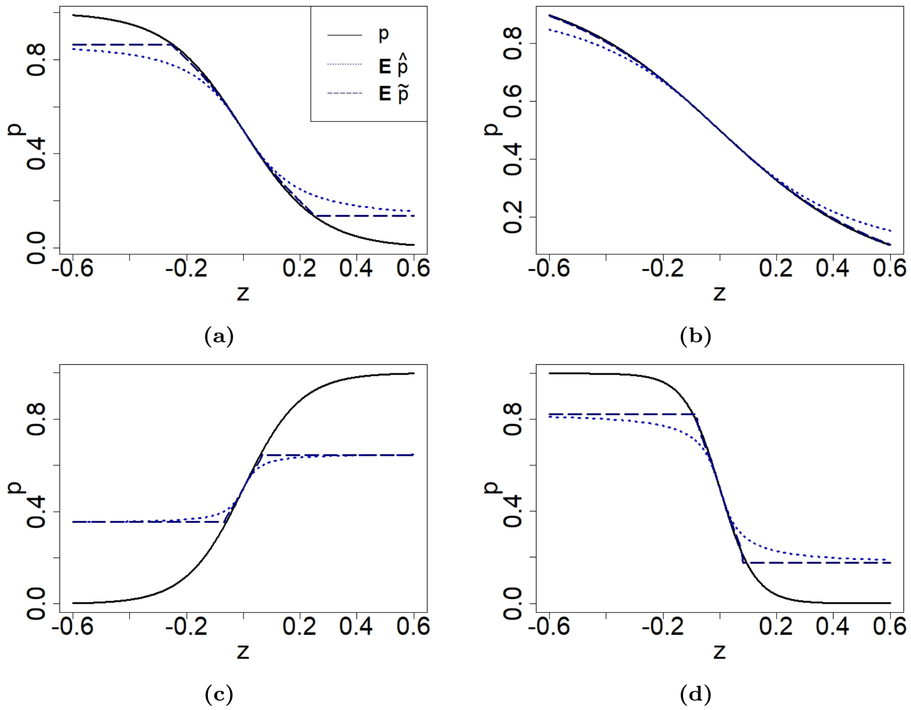
$$\frac{\nabla g}{\tau^*} \in \left[\frac{z\nabla g(\theta)}{1.702} - \Phi^{-1}\left(0.0095\right), \frac{z\nabla g(\theta)}{1.702} + \Phi^{-1}\left(0.0095\right)\right] \tag{44}$$

and, from (44) we derive

$$\tau^* \in \left[\frac{1}{\frac{|z|}{1.702} + \frac{\Phi^{-1}(0.0095)}{|\nabla g|}}, \frac{1}{\frac{|z|}{1.702} - \frac{\Phi^{-1}(0.0095)}{|\nabla g|}}\right] \tag{45}$$

A.3.1 Empirical Simulation

We perform a simulation to empirically validate the findings in the previous section: we take the last sample from the vanilla-SGBD chain used to produce the Figure 5, we repeatedly subsample a mini-batch, store the gradient for each coordinate, estimate its standard deviation, and compute $\hat{p}$ and $\tilde{p}$.



(a)

(b)

(c)

(d)

Fig. 8. Shrinkage effect and bias correction: p (black line) and Monte Carlo averages of $\hat{p}$ (dashed blue line), and $\tilde{p}$ (dotted dark blue line) vs the proposed increment z.

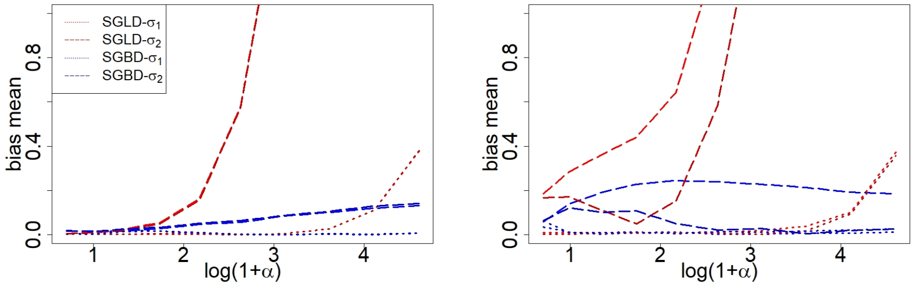
Figure 8 reports the Monte Carlo average of $\hat{p}$ and $\tilde{p}$ as the value of the proposed increment. z, varies. When proposed increments are very small, the expected value of $\hat{p}$ is very close to the true value of p (i.e. the value obtained using the full gradient), and the correction performed by $\tilde{p}$ is limited. However, in some cases (see Figure 8b, 8c and 8d), when the proposed increment is larger in absolute value than the breaking point, the bias cannot be completely eliminated.

A.4 Additional Simulations

We report the additional simulations results and plots.

A.4.1 Toy Example: Skew-Normal target with isotropic Gaussian Noise

We report additional results from the toy example with isotropic noise. In particular, we report the results obtained using also corrected variants of the algorithms.



(a) small τ

(b) large τ

Fig. 9. Toy Example: Skew-Normal target with isotropic Gaussian Noise. Shape parameter (in log-scale) vs relative bias of mean with noise standard deviation equal to 1 (left) and 10 (right) times the target standard deviation. Red refers to Langevin and blue to Barker. Lighter lines refer to vanilla implementations of the algorithm, darker lines to their corrected variants. Dotted (dashed resp.) lines are produced using a step size equal to 10% (50% resp.) of the target standard deviation.

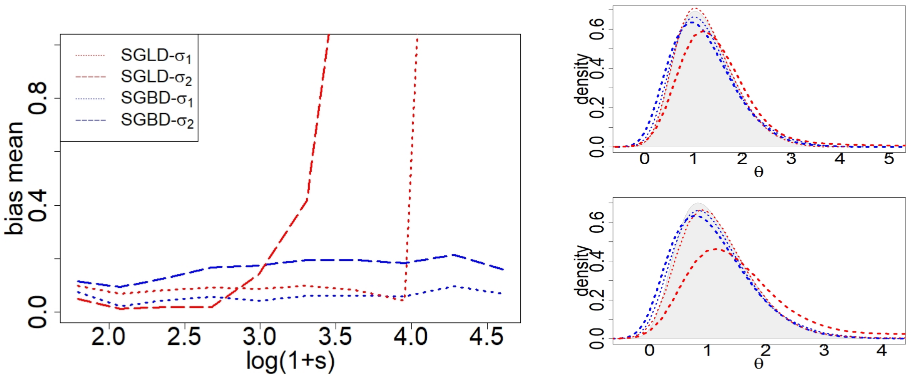
With a gradient noise standard deviation equal to the one of the target (Figure 9a), there is little difference between vanilla and corrected variants of the algorithms. When the stardard deviation if large (Figure 9b), corrected variants achieve a lower bias then their vanilla counterparts with a large step-size. Under all configurations, the sampler based on the Barker dynamics outperforms the one based on the Langevin dynamics in terms of robustness to skewness as well as robustness to tuning.

A.4.2 Toy Example: Unidimensional Logistic Regression with heavy-tailed and imbalanced dataset

We apply the Bayesian Logistic Regression model (14) to an imbalanced dataset and covariates with extreme values. We generate the data as follows:

$$\begin{aligned} y_i &= 1 \quad i = 1, \ldots, N \\ X_i &\sim \textit{Half-Cauchy}(s) \quad i = 1, \ldots, N \end{aligned} \tag{46}$$

for increasing values of the scale parameter s and $N = 100$. As s is increased, some of the $X_i's$ assume more extreme value, and, consequently, the degree of skewness of resulting posterior distribution of the parameter increases. Again, we consider two configurations of step-size σ (10% and 20% of the target standard deviation). This simulation is similar to the toy example presented above as increasing the scale parameter has a comparable effect on the target distribution as increasing α in the previous example. The difference consists in using real data. Consequently, the stochastic gradient noise ceases to be isotropic and can potentially have a non-Gaussian distribution. Algorithms are run using a minibatch-size of 1.



(a) **(b)**

Fig. 10. Toy Example: Logistic Regression with heavy-tailed covariates. Scale parameter (in log-scale) vs relative bias of mean and invariant distribution of the algorithm for two different levels of s. Red refers to SGLD and blue to SGBD. Dotted (dashed resp.) lines are produced using a step size equal to 10% (20% resp.) of the target standard deviation. The grey shaded area in the right plots represents the target distribution density.

SGBD is more robust to skweness in the target distribution, and the greater robustness becomes more evident as the step-size increases (Figure (10)).

As $s \uparrow \infty$, the posterior distribution becomes increasingly skewed with more extreme values of the gradient in some region of the state space. While SGLD suffers from this situation also with a small step-size, SGBD is remarkably stable. In particular, SGLD fails to sample accurately from the right distribution, also when the step size is small, but the target distribution is moderately or highly skewed (right plots in Figure 10). In fact, when SGLD samples from a region with a very large gradient, subsequent samples lie in the right tail of the distribution, determining a large bias in the Monte-Carlo estimates of the mean. By contrast, SGBD does not suffer from the increasing skewness of the posterior showing a low first-order bias in all cases. With a larger step-size (bottom plot in figure 10b), SGLD fails completely to sample from the right distribution. SGBD instead displays a reduction in the sampling accuracy with a larger step-size but the bias is limited and does not increases with the skewness of the posterior distribution. Overall, the results are very similar to the ones found in the simulation with isotropic noise.

A.4.3 Binary Regression with Scale Heterogeneity

We report additional plots of the simulation presented in section 5.1.1.

We also report the predictive performance on the held-out test set. In general, smaller step-sizes produce better estimates. SGBD with a small step-size obtains the best predictive accuracy and the performance SGBD appears to be more robust to tuning.

We performed a second experiment selecting step-sizes respectively equal to 10% and 20% of the marginal posterior standard deviations. The aim of this experiment is to study the performance of the algorithm with a correct tuning of the step-size across the coordinates. However, we note that this is not replicable in practice as it requires to have access to posterior quantities which are in general unknown.

In this scenario, clearly both algorithm perform better than in the previous one and sample accurately with small step-sizes (dotted lines in 14). With a larger step-size (dashed lines in 14), SGLD remarkably inflate the variance of the marginal distribution, while SGBD's invariant distributions match more closely target densities. If the step-sizes are further increased, SGLD completely fails to sample accurately, instead SGBD only slightly inflates the variances while correctly centering the marginal distributions. Therefore, SGBD outperforms SGLD also when the step-sizes are tuned to keep into account the scale heterogeneity of the target.

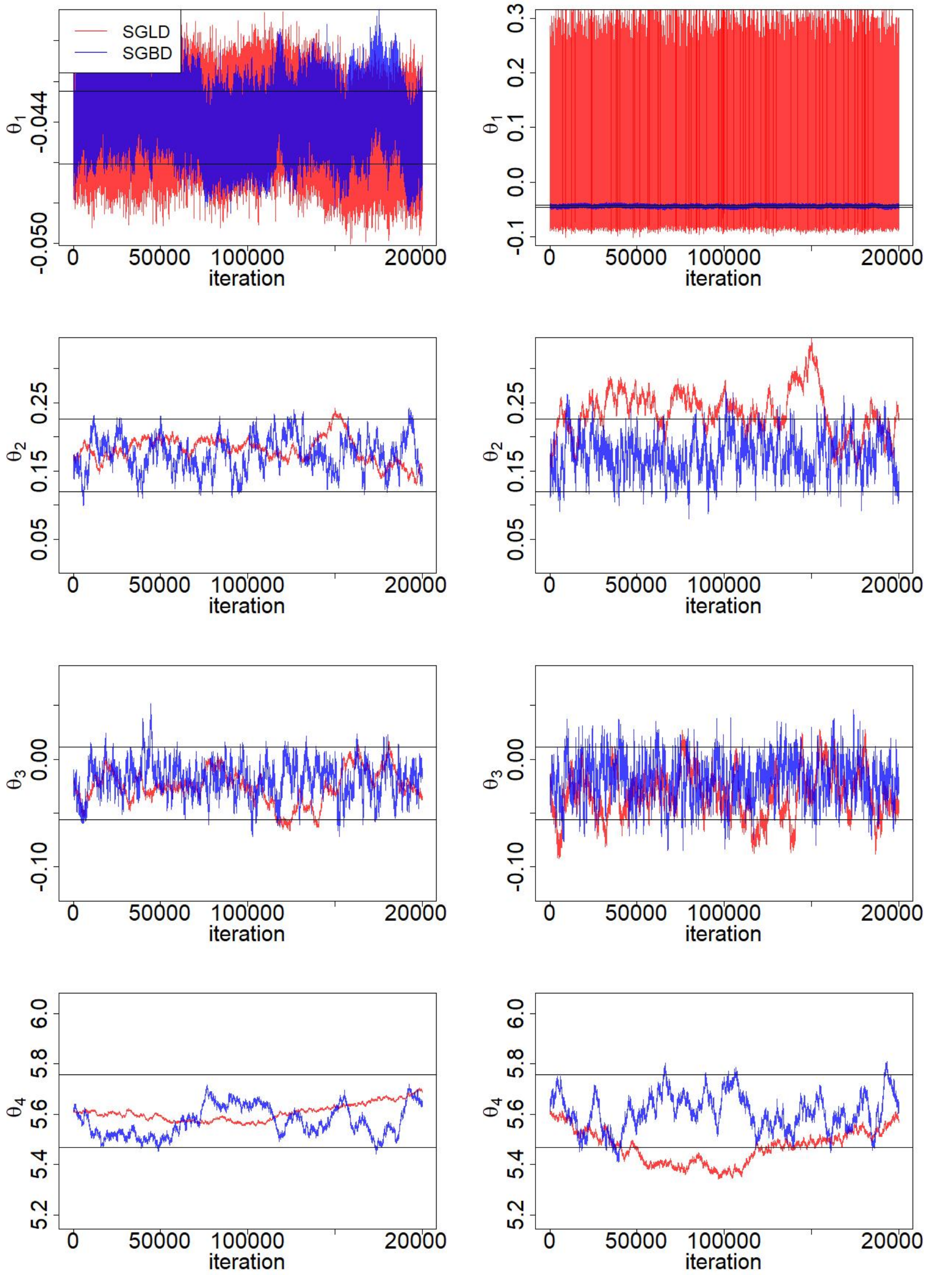


Fig. 11. Logistic Regression with Scale Heterogeneity on the Sepsis dataset. Traceplots of all the coordinates with two step-size configurations: small σ (left), large σ (right). Red refers to SGLD and blue to the SGBD. Black horizontal lines represent the interval around the posterior mean with a two standard deviations width

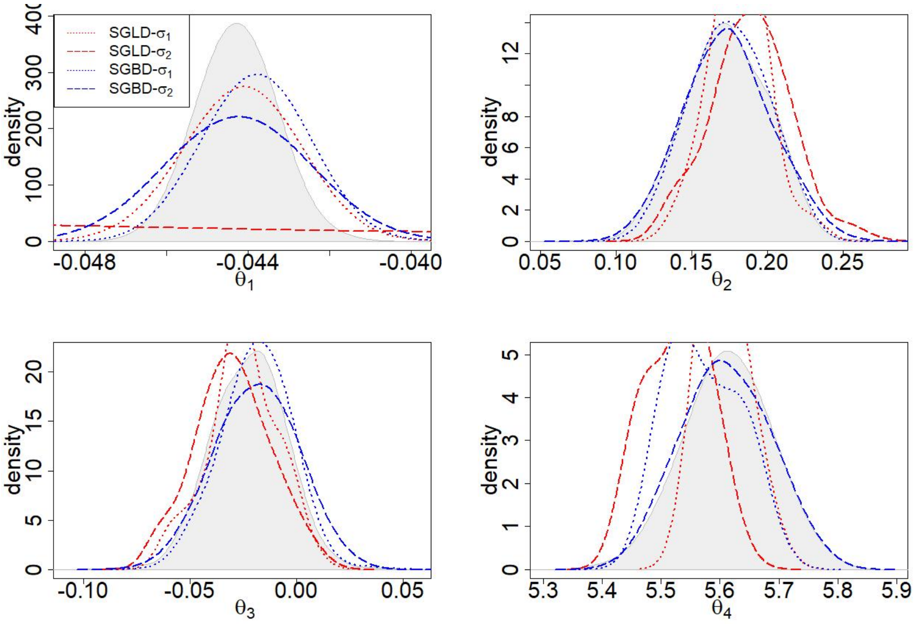


Fig. 12. Logistic Regression: Univariate distributions using the Sepsis dataset. Blue lines represent the density estimates of vanilla-SGBD samples. Grey areas represent the estimate of the marginal posterior densities obtained with STAN. Dotted (dashed resp.) lines are produced using a smaller (larger resp.) step size

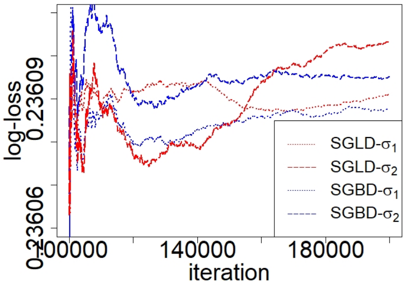


Fig. 13. Logistic Regression Mixing-Accuracy Predictive Accuracy on the Seps dataset. Log-loss of MCMC estimates. Red refers to SGLD and blue to the SGBD. We present the performance algorithm with two configurations of step-size: smaller σ (dotted lines), larger σ (dashed lines).

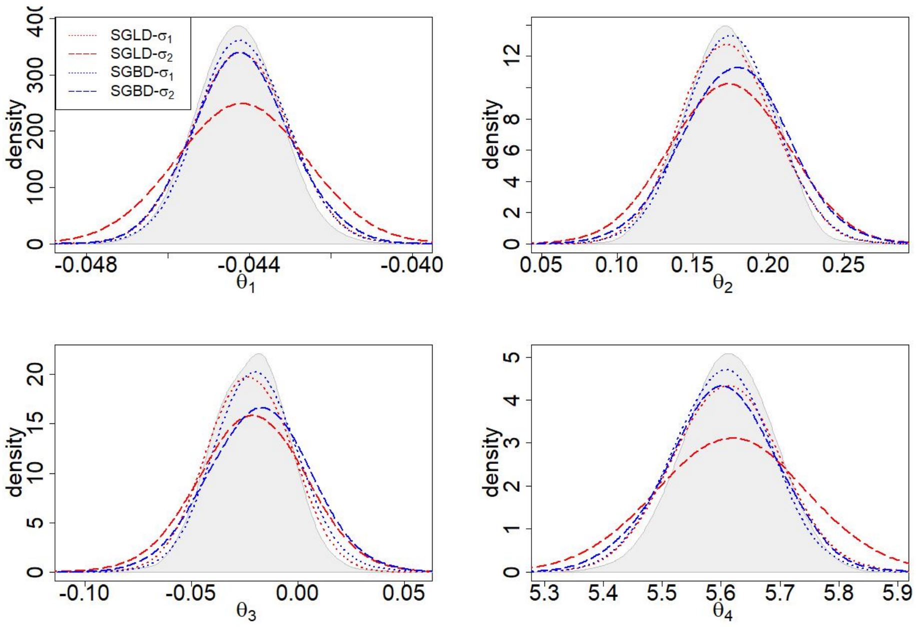


Fig. 14. Logistic Regression: Univariate distributions using the Sepsis dataset. Blue lines represent the density estimates of vanilla-SGBD samples. Grey areas represent the estimate of the marginal posterior densities obtained with STAN. Dotted (dashed resp.) lines are produced using a step size equal to 10% (20% resp.) of the target standard deviation

A.4.4 High Dimensional Binary Regression

We report the result regarding the log-loss with different step-sizes configurations on the Arrhythmia dataset.

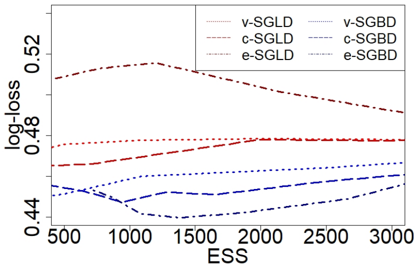


Fig. 15. Logistic Regression: Predictive Accuracy on Arrhythmia dataset. Median ESS vs Log-loss on unseen test set using all samples after the burn-in. Red lines refers to SGLD and blue lines to the SGBD. Lighter lines refer to vanilla implementations of the algorithm, medium scale lines to corrected variants, and darker lines to extreme ones.

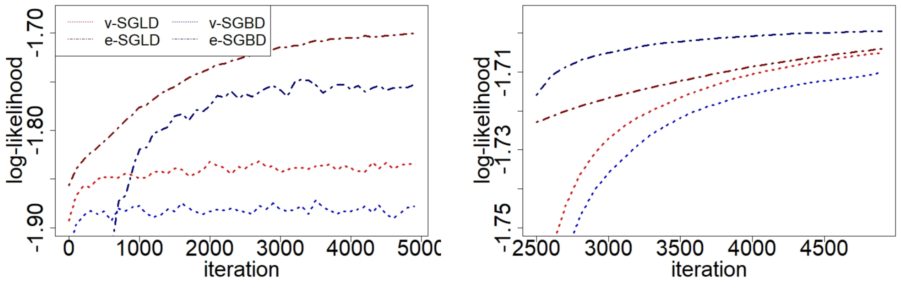
SGBD outperforms SGLD with all tuning configurations considered.

A.4.5 Poisson Probabilistic Matrix Factorization

We also study the performance of SGBD on the same content recommendation task of section 5.3 using a different model. More specifically, we apply to the same task the Poisson probabilistic matrix factorization model (PPMF) [10]. Ratings are modelled as Poisson random variables and entries of the lower dimensional matrices are assigned Gamma distributions. To avoid confusion from the BPMF model, we call the factorizing matrices $\theta \in \mathbb{R}^{U\times d}$ and $\beta \in \mathbb{R}^{M\times d}$. The PPMF model is specified as follows:

$$\begin{aligned}
R_{ij}|\theta,\beta,I_{ij}=1 \sim Poisson(\theta_i^t\beta_j) \quad i=1,\dots,N \quad j=1,\dots,M\\
\theta_{ik}|\varepsilon_i \sim \Gamma(a,\varepsilon_i) \quad k=1,\dots,d \quad i=1,\dots,U\\
\beta_{jk}|\nu_j \sim \Gamma(c,\nu_j) \quad k=1,\dots,d \quad j=1,\dots,M\\
\varepsilon_i \sim \Gamma\left(a',\frac{a'}{b'}\right) \quad i=1,\dots,U\\
\nu_j \sim \Gamma\left(c',\frac{c'}{d'}\right) \quad j=1,\dots,M.
\end{aligned} \tag{47}$$

We set d to 100, a', a, c', c to 0.3, and b', d' to 1. We evaluate the performance in terms of log-likelihood on the unseen test set.



(a) Sample log-likelihood

(b) MCMC log-likelihood

Fig. 16. Poisson Probabilistic Matrix Factorization: Predictive Accuracy on MovieLens dataset. Log-likelihood of raw (left) and clipped (right) predictions. Red refers to SGLD and blue to the SGBD. Lighter lines refer to vanilla implementations of the algorithm, darker lines to their extreme variants. Solid lines represent the performance computed at each sample, dashed ones are produced using MCMC estimates.

In this case, MCMC estimates of all algorithms have a similar with extreme-SGBD estimates being slightly better.

References

[1] Rémi Bardenet, Arnaud Doucet, and Chris Holmes. "Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 1. Bejing, China: PMLR, 22–24 Jun 2014, pp. 405–413.

[2] Shannon Bowling, Mohammad Khasawneh, Sittichai Kaewkuekool, and Byung Cho. "A logistic approximation to the cumulative normal distribution". In: *Journal of Industrial Engineering and Management* 2 (July 2009). DOI: `10.3926/jiem..v2n1.p114-127`.

[3] Nicolas Brosse, Alain Durmus, and Eric Moulines. "The promises and pitfalls of Stochastic Gradient Langevin Dynamics". In: *Advances in Neural Information Processing Systems* 31 (2018).

[4] Bob Carpenter, Andrew Gelman, Matthew Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. "Stan: A Probabilistic Programming Language". In: *Journal of Statistical Software* 76.1 (2017).

[5] Tianqi Chen, Emily Fox, and Carlos Guestrin. "Stochastic Gradient Hamiltonian Monte Carlo". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, June 2014, pp. 1683–1691.

[6] Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert Skeel, and Hartmut Neven. "Bayesian Sampling Using Stochastic Gradient Thermostats". In: vol. 4. Dec. 2014.

[7] Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert D Skeel, and Hartmut Neven. "Bayesian Sampling Using Stochastic Gradient Thermostats". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger. Vol. 27. Curran Associates, Inc., 2014.

[8] Simon Duane, A.D. Kennedy, Brian J. Pendleton, and Duncan Roweth. "Hybrid Monte Carlo". In: *Physics Letters B* 195.2 (1987), pp. 216–222. ISSN: 0370-2693. DOI: `https://doi.org/10.1016/0370-2693(87)91197-X`. URL: `https://www.sciencedirect.com/science/article/pii/037026938791197X`.

[9] John Duchi, Elad Hazan, and Yoram Singer. “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization”. In: *Journal of Machine Learning Research* 12.61 (2011), pp. 2121–2159. URL: http://jmlr.org/papers/v12/duchi11a.html.

[10] Prem Gopalan, Jake M. Hofman, and David M. Blei. *Scalable Recommendation with Poisson Factorization.* 2013. DOI: 10.48550/ARXIV.1311.1704. URL: https://arxiv.org/abs/1311.1704.

[11] Matthew D. Homan and Andrew Gelman. “The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo”. In: *J. Mach. Learn. Res.* 15.1 (2014), pp. 1593–1623.

[12] Anoop Korattikara, Yutian Chen, and Max Welling. “Austerity in MCMC Land: Cutting the Metropolis-Hastings Budget”. In: *Proceedings of the 31st International Conference on Machine Learning.* Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 1. Bejing, China: PMLR, June 2014, pp. 181–189.

[13] Samuel Livingstone and Giacomo Zanella. “The Barker proposal: Combining robustness and efficiency in gradient-based MCMC”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* (2022).

[14] Xiaoyu Lu, Valerio Perrone, Leonard Hasenclever, Yee Whye Teh, and Sebastian J. Vollmer. “Relativistic Monte Carlo”. In: *AISTATS.* 2017.

[15] Tigran Nagapetyan, Andrew B. Duncan, Leonard Hasenclever, Sebastian J. Vollmer, Lukasz Szpruch, and Konstantinos Zygalakis. *The True Cost of Stochastic Gradient Langevin Dynamics.* 2017. arXiv: 1706.02692.

[16] Radford Neal. “MCMC using Hamiltonian dynamics”. In: *Handbook of Markov Chain Monte Carlo* (2012).

[17] G. Parisi. “Correlation Functions and Computer Simulations”. In: *Nucl. Phys. B* 180 (1981), p. 378.

[18] Sam Patterson and Yee Whye Teh. “Stochastic Gradient Riemannian Langevin Dynamics on the Probability Simplex”. In: *Advances in Neural Information Processing Systems.* Ed. by C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger. Vol. 26. Curran Associates, Inc., 2013. URL: https://proceedings.neurips.cc/paper/2013/file/309928d4b100a5d75adff48a9bfc1ddb-Paper.pdf.

[19] Herbert Robbins and Sutton Monro. “A Stochastic Approximation Method”. In: *The Annals of Mathematical Statistics* 22.3 (1951), pp. 400–407.

[20] Herbert E. Robbins and David O. Siegmund. "A Convergence Theorem for Non Negative Almost Supermartingales and Some Applications". In: 1985.

[21] Gareth O. Roberts and Jeffrey S. Rosenthal. "Optimal scaling of discrete approximations to Langevin diffusions". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 60.1 (1995), pp. 255–268.

[22] Ruslan Salakhutdinov and Andriy Mnih. "Bayesian Probabilistic Matrix Factorization Using Markov Chain Monte Carlo". In: *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. Helsinki, Finland: Association for Computing Machinery, 2008, pp. 880–887. ISBN: 9781605582054. DOI: 10.1145/1390156.1390267. URL: https://doi.org/10.1145/1390156.1390267.

[23] Yee Whye Teh, Alexandre H. Thiery, and Sebastian J. Vollmer. "Consistency and Fluctuations for Stochastic Gradient Langevin Dynamics". In: *J. Mach. Learn. Res.* 17.1 (2016), pp. 193–225.

[24] Jure Vogrinc, Samuel Livingstone, and Giacomo Zanella. *Optimal design of the Barker proposal and other locally-balanced Metropolis-Hastings algorithms*. 2022. arXiv: 2201.01123 [stat.CO].

[25] Sebastian J. Vollmer, Konstantinos C. Zygalakis, and Yee Whye Teh. "Exploration of the (Non-)Asymptotic Bias and Variance of Stochastic Gradient Langevin Dynamics". In: *Journal of Machine Learning Research* 17.159 (2016), pp. 1–48. URL: http://jmlr.org/papers/v17/15-494.html.

[26] Max Welling and Yee Whye Teh. "Bayesian Learning via Stochastic Gradient Langevin Dynamics". In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*. ICML'11. Bellevue, Washington, USA: Omnipress, 2011, pp. 681–688. ISBN: 9781450306195.