DG-LMC: A Turn-key and Scalable Synchronous Distributed MCMC Algorithm via Langevin Monte Carlo within Gibbs

Vincent Plassier [1 2 *] Maxime Vono [2 *] Alain Durmus [3 *] Eric Moulines [1]

Abstract

Performing reliable Bayesian inference on a big data scale is becoming a keystone in the modern era of machine learning. A workhorse class of methods to achieve this task are Markov chain Monte Carlo (MCMC) algorithms and their design to handle distributed datasets has been the subject of many works. However, existing methods are not completely either reliable or computationally efficient. In this paper, we propose to fill this gap in the case where the dataset is partitioned and stored on computing nodes within a cluster under a master/slaves architecture. We derive a user-friendly centralised distributed MCMC algorithm with provable scaling in high-dimensional settings. We illustrate the relevance of the proposed methodology on both synthetic and real data experiments.

1. Introduction

In the current machine learning era, data acquisition has seen significant progress due to rapid technological advances which now allow for more accurate, cheaper and faster data storage and collection. This data quest is motivated by modern machine learning techniques and algorithms which are now well-proven and have become common tools for data analysis. In most cases, the empirical success of these methods are based on a very large sample size (Bardenet et al., 2017; Bottou et al., 2018). This need for data is also theoretically justified by data probabilistic modelling which asserts that under appropriate conditions, the more data can be processed, the more accurate the inference can be performed. However, in recent years, several challenges have emerged regarding the use and access to data in mainstream machine learning methods. Indeed, first the amount of data is now so large that it has outpaced the increase in computation power of computing resources (Verbraeken et al., 2020). Second, in many modern applications, data storage and/or use are not on a single machine but shared across several units (Raicu et al., 2006; Bernstein & Newcomer, 2009). Third, life privacy is becoming a prime concern for many users of machine learning applications who are therefore asking for methods preserving data anonymity (Shokri & Shmatikov, 2015; Abadi et al., 2016). Distributed machine learning aims at tackling these issues. One of its popular paradigms, referred to as data-parallel approach, is to consider that the training data are divided across multiple machines. Each of these units constitutes a worker node of a computing network and can perform a *local* inference based on the data it has access. Regarding the choice of the network, several options and frameworks have been considered. We focus here on the master/slaves architecture where the worker nodes communicate with each other through a device called the *master* node.

Under this framework, we are interested in carrying Bayesian inference about a parameter $\boldsymbol{\theta} \in \mathbb{R}^d$ based on observed data $\{\mathbf{y}_k\}_{k=1}^n \in \mathsf{Y}^n$ (Robert, 2001). The dataset is assumed to be partitioned into S *shards* and stored on S machines among a collection of b worker nodes. The subset of observations associated to worker $i \in [b]$ is denoted by D_i, where $[b] = \{1, \dots, b\}$. Potentially, $\mathrm{D}_i = \{\emptyset\}$ if $i \in [S+1 : b]$ for $b > S$, where we use the notation $[S+1 : b] = \{S+1, \dots, b\}$. The posterior distribution of interest is assumed to admit a density with respect to (w.r.t.) the d-dimensional Lebesgue measure which factorises across workers, *i.e.,*

$$\pi(\boldsymbol{\theta} \mid \mathbf{y}_{1:n}) = Z_\pi^{-1} \prod_{i=1}^{b} \mathrm{e}^{-U_i(\mathbf{A}_i \boldsymbol{\theta})} , \tag{1}$$

where $Z_\pi = \int_{\mathbb{R}^d} \prod_{i=1}^b \mathrm{e}^{-U_i(\mathbf{A}_i\boldsymbol{\theta})} \mathrm{d}\boldsymbol{\theta}$ is a normalisation constant and $\mathbf{A}_i \in \mathbb{R}^{d_i \times d}$ are matrices that might act on the parameter of interest. For $i \in [b]$, the potential function $U_i : \mathbb{R}^{d_i} \to \mathbb{R}$ is assumed to depend only on the subset of observations D_i. Note that for $i \in [S+1 : b]$, $b > S$, U_i does not depend on the data but only on the prior. For the sake of brevity, the dependency of π w.r.t. the observations

[*] Equal contribution [1] CMAP, Ecole Polytechnique, Institut Polytechnique de Paris, Palaiseau, France [2] Lagrange Mathematics and Computing Research Center, Paris, 75007, France [3] Universit Paris-Saclay, Ecole Normale Supérieure Paris-Saclay, Cachan, France. Correspondence to: Vincent Plassier <vincent.plassier@huawei.com>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

$\{\mathrm{D}_i\}_{i=1}^b$ is notationally omitted.

To sample from π given by (1) in a distributed fashion, a large number of approximate methods have been proposed in the past ten years (Johnson et al., 2013; Neiswanger et al., 2014; Ahn et al., 2014; Rabinovich et al., 2015; Scott et al., 2016; Hasenclever et al., 2017; Nemeth & Sherlock, 2018; Chowdhury & Jermaine, 2018; Bui et al., 2018; Rendell et al., 2020; Vehtari et al., 2020). Despite multiple research lines, to the best of authors' knowledge, none of these proposals has been proven to be satisfactory. Indeed, the latter are not completely either computationally efficient in high-dimensional settings, reliable or theoretically grounded (Jordan et al., 2019).

This work is an attempt to fill this gap. To this purpose, we follow the data augmentation approach introduced in Vono et al. (2020b) and referred to as asymptotically exact data augmentation (AXDA). Given a tolerance parameter $\boldsymbol{\rho} \in \mathbb{R}_+^b$, the main idea behind this methodology is to consider a joint distribution $\Pi_{\boldsymbol{\rho}}$ on the extended state space $\mathbb{R}^d \times \prod_{i=1}^b \mathbb{R}^{d_i}$ such that $\Pi_{\boldsymbol{\rho}}$ has a density w.r.t. the Lebesgue measure of the form $(\boldsymbol{\theta}, \mathbf{z}_{1:b}) \mapsto \prod_{i=1}^b \Pi_{\boldsymbol{\rho}}^i(\boldsymbol{\theta}, \mathbf{z}_i)$. $\Pi_{\boldsymbol{\rho}}$ is carefully designed so that its marginal w.r.t. $\boldsymbol{\theta}$, denoted by $\pi_{\boldsymbol{\rho}}$, is a proxy of (1) for which quantitative approximation bounds can be derived and are controlled by $\boldsymbol{\rho}$. In addition, for any $i \in [b]$, $\Pi_{\boldsymbol{\rho}}^i(\boldsymbol{\theta}, \mathbf{z}_i)$ only depends on the data D_i, and therefore plays a role similar to the local posterior $\pi^i(\boldsymbol{\theta}) \propto \mathrm{e}^{-U_i(\mathbf{A}_i\boldsymbol{\theta})}$ in popular embarrassingly parallel approaches (Neiswanger et al., 2014; Scott et al., 2016). However, compared to this class of methods, AXDA does not seek for each worker to sample from $\Pi_{\boldsymbol{\rho}}^i$. Following a data augmentation strategy based on Gibbs sampling, AXDA instead requires each worker to sample from the conditional distribution $\Pi_{\boldsymbol{\rho}}(\mathbf{z}_i|\boldsymbol{\theta})$ and to communicate its sample to the master. $\Pi_{\boldsymbol{\rho}}$ is generally chosen such that sampling from $\Pi_{\boldsymbol{\rho}}(\boldsymbol{\theta}|\mathbf{z}_{1:b})$ is easy and does not require to access to the data. However, two main challenges remain: one has to sample efficiently from the conditional distribution $\Pi_{\boldsymbol{\rho}}(\mathbf{z}_i|\boldsymbol{\theta})$ for $i \in [b]$ and avoid too frequent communication rounds on the master. Existing AXDA-based approaches unfortunately do not fulfill these important requirements (Vono et al., 2019b; Rendell et al., 2020). In this work, we leverage these issues by considering the use of the Langevin Monte Carlo (LMC) algorithm to approximately sample from $\Pi_{\boldsymbol{\rho}}(\mathbf{z}_i|\boldsymbol{\theta})$ (Rossky et al., 1978; Roberts & Tweedie, 1996).

Our contributions are summarised in what follows. (1) We introduce in Section 2 a new methodology called Distributed Gibbs using Langevin Monte Carlo (DG-LMC). (2) Importantly, we provide in Section 3 a detailed quantitative analysis of the induced bias and show explicit convergence results. This stands for our main contribution and to the best of authors' knowledge, this theoretical study is one of the most complete among existing works which focused on distributed Bayesian machine learning with a master/slaves architecture. In particular, we discuss the complexity of our algorithm, the choice of hyperparameters, and provide practitioners with simple prescriptions to tune them. Further, we provide a thorough comparison of our method with existing approaches in Section 4. (3) Finally, in Section 5, we show the benefits of the proposed sampler over popular and recent distributed MCMC algorithms on several numerical experiments. Given the limited page count, all the proofs are postponed to the supplementary material.

Notations and conventions. The Euclidean norm on $\mathbb{R}^d$ is denoted by $\|\cdot\|$. For $n \geq 1$, we refer to $\{1, \ldots, n\}$ with the notation $[n]$ and for $i_1, i_2 \in \mathbb{N}$, $i_1 \leq i_2$, $\{i_1, \ldots, i_2\}$ with the notation $[i_1 : i_2]$. For $0 \leq i < j$ and $(\mathbf{u}_k; k \in \{i, \cdots, j\})$, we use the notation $\mathbf{u}_{i:j}$ to refer to the vector $[\mathbf{u}_i^\top, \cdots, \mathbf{u}_j^\top]^\top$. We denote by $\mathrm{N}(\mathbf{m}, \boldsymbol{\Sigma})$ the Gaussian distribution with mean vector $\mathbf{m}$ and covariance matrix $\boldsymbol{\Sigma}$. For a given matrix $\mathbf{M} \in \mathbb{R}^{d\times d}$, we denote its smallest eigenvalue by $\lambda_{\min}(\mathbf{M})$. We denote by $\mathcal{B}(\mathbb{R}^d)$ the Borel σ-field of $\mathbb{R}^d$. We define the Wasserstein distance of order 2 for any probability measures μ, ν on $\mathbb{R}^d$ with finite 2-moment by $W_2(\mu, \nu) = (\inf_{\zeta \in \mathcal{T}(\mu,\nu)} \int_{\mathbb{R}^d\times\mathbb{R}^d} \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|^2 \mathrm{d}\zeta(\boldsymbol{\theta}, \boldsymbol{\theta}'))^{1/2}$, where $\mathcal{T}(\mu, \nu)$ is the set of transference plans of μ and ν.

2. Distributed Gibbs using Langevin Monte Carlo (DG-LMC)

In this section, we present the proposed methodology which is based on the AXDA statistical framework and the popular LMC algorithm.

AXDA relies on the decomposition of the target distribution π given in (1) to introduce an extended distribution which enjoys favorable properties for distributed computations. This distribution is defined on the state space $\mathbb{R}^d \times \mathsf{Z}$, $\mathsf{Z} = \prod_{i=1}^b \mathbb{R}^{d_i}$, and admits a density w.r.t. the Lebesgue measure given, for any $\boldsymbol{\theta} \in \mathbb{R}^d$, $\mathbf{z}_{1:b} \in \mathsf{Z}$, by

$$\Pi_{\boldsymbol{\rho}}(\boldsymbol{\theta}, \mathbf{z}_{1:b}) \propto \prod_{i=1}^b \tilde{\Pi}_{\boldsymbol{\rho}}^i(\boldsymbol{\theta}, \mathbf{z}_i) \ , \qquad (2)$$

where $\tilde{\Pi}_{\boldsymbol{\rho}}^i(\boldsymbol{\theta}, \mathbf{z}_i) = \exp(-U_i(\mathbf{z}_i) - \|\mathbf{z}_i - \mathbf{A}_i\boldsymbol{\theta}\|^2/2\rho_i)$ and $\boldsymbol{\rho} = \{\rho_i\}_{i=1}^b \in \mathbb{R}_+^b$ is a sequence of positive tolerance parameters. Note that $\tilde{\Pi}_{\boldsymbol{\rho}}^i$ is not necessarily a probability density function. Actually, for $\Pi_{\boldsymbol{\rho}}$ to define a proper probability density, *i.e.* $\int_{\mathbb{R}^d\times\mathsf{Z}} \prod_{i=1}^b \tilde{\Pi}_{\boldsymbol{\rho}}^i(\boldsymbol{\theta}, \mathbf{z}_i)\mathrm{d}\boldsymbol{\theta}\mathrm{d}\mathbf{z}_{1:b} < \infty$, some conditions are required.

H 1. *There exists $b' \in [b-1]$ such that the following conditions hold: $\min_{i\in[b']} \inf_{\mathbf{z}_i \in \mathbb{R}^{d_i}} U_i(\mathbf{z}_i) > -\infty$, and $\max_{i\in[b'+1:b]} \int_{\mathbb{R}^{d_i}} \mathrm{e}^{-U_i(\mathbf{z}_i)}\mathrm{d}\mathbf{z}_i < \infty$. In addition, $\sum_{j=b'+1}^b \mathbf{A}_j^\top \mathbf{A}_j$ is invertible.*

The next result shows that these mild assumptions are sufficient to guarantee that the extended model (2) is well-defined.

Proposition 1. *Assume **H**1. Then, for any $\boldsymbol{\rho} \in \mathbb{R}_+^b$, $\Pi_{\boldsymbol{\rho}}$ in (2) is a proper density.*

The data augmentation scheme (2) is approximate in the sense that the $\boldsymbol{\theta}$-marginal defined by

$$\pi_{\boldsymbol{\rho}}(\boldsymbol{\theta}) = \int_{\mathsf{Z}} \Pi_{\boldsymbol{\rho}}(\boldsymbol{\theta}, \mathbf{z}_{1:b})\mathrm{d}\mathbf{z}_{1:b} \;, \tag{3}$$

coincides with (1) only in the limiting case $\max_{i\in[b]} \rho_i \downarrow 0$ (Scheffé, 1947). For a fixed $\boldsymbol{\rho}$, quantitative results on the induced bias in total variation distance can be found in Vono et al. (2019b). The main benefit of working with (2) is that conditionally upon $\boldsymbol{\theta}$, auxiliary variables $\{\mathbf{z}_i\}_{i=1}^b$ are independent. Therefore, they can be sampled in parallel within a Gibbs sampler. For $i \in [b]$, the conditional density of $\mathbf{z}_i$ given $\boldsymbol{\theta}$ writes

$$\Pi_{\boldsymbol{\rho}}(\mathbf{z}_i \mid \boldsymbol{\theta}) \propto \exp\left(- U_i(\mathbf{z}_i) - \tfrac{\|\mathbf{z}_i - \mathbf{A}_i\boldsymbol{\theta}\|^2}{2\rho_i}\right) . \tag{4}$$

On the other hand, the conditional distribution of $\boldsymbol{\theta}$ given $\mathbf{z}_{1:b}$ is a Gaussian distribution

$$\Pi_{\boldsymbol{\rho}}(\boldsymbol{\theta} \mid \mathbf{z}_{1:b}) = \mathrm{N}(\boldsymbol{\mu}(\mathbf{z}_{1:b}), \mathbf{Q}^{-1}) \;, \tag{5}$$

with precision matrix $\mathbf{Q} = \sum_{i=1}^b \mathbf{A}_i^\top \mathbf{A}_i/\rho_i$ and mean vector $\boldsymbol{\mu}(\mathbf{z}_{1:b}) = \mathbf{Q}^{-1} \sum_{i=1}^b \mathbf{A}_i^\top \mathbf{z}_i/\rho_i$. Under **H**1, note that $\mathbf{Q}$ is invertible and therefore this conditional Gaussian distribution is well-defined. Since sampling from high-dimensional Gaussian distributions can be performed efficiently (Vono et al., 2020a), this Gibbs sampling scheme is interesting as long as sampling from (4) is cheap. Vono et al. (2019b) proposed the use of a rejection sampling step requiring to set $\rho_i = \mathcal{O}(1/d_i)$. When $d_i \gg 1$, this condition unfortunately leads to prohibitive computational costs and hence prevents its practical use for general Bayesian inference problems. Instead of sampling exactly from (4), Rendell et al. (2020) rather proposed to use Metropolis-Hastings algorithms. However, it is not clear whether this choice indeed leads to efficient sampling schemes. To tackle these issues, we propose to build upon LMC to end up with a distributed MCMC algorithm which is both simple to implement, efficient and amenable to a theoretical study. LMC stands for a popular way to approximately generate samples from a given distribution based on the Euler-Maruyama discretisation scheme of the overdamped Langevin stochastic differential equation (Roberts & Tweedie, 1996). At iteration t of the considered Gibbs sampling scheme and given a current parameter $\boldsymbol{\theta}^{(t)}$, LMC applied to (4) considers, for $i \in [b]$, the recursion

$$\mathbf{z}_i^{(t+1)} = \left(1 - \tfrac{\gamma_i}{\rho_i}\right)\mathbf{z}_i^{(t)} + \tfrac{\gamma_i}{\rho_i}\mathbf{A}_i\boldsymbol{\theta}^{(t)} - \gamma_i \nabla U_i\left(\mathbf{z}_i^{(t)}\right) + \sqrt{2\gamma_i}\boldsymbol{\xi}_i^{(t)}$$

where $\gamma_i > 0$ is a fixed step-size and $(\boldsymbol{\xi}_i^{(k)})_{k\in\mathbb{N}, i\in[b]}$ a sequence of independent and identically distributed (i.i.d.) d-dimensional standard Gaussian random variables. Only using a single step of LMC on each worker might incur important communication costs. To mitigate the latter while increasing the proportion of time spent on exploring the state-space, we instead allow each worker to perform $N_i \geq 1$ LMC steps (Dieuleveut & Patel, 2019; Rendell et al., 2020). Letting N_i varies across workers prevents Algorithm 1 to suffer from a significant block-by-the-slowest delay in cases where the response times of the workers are unbalanced (Ahn et al., 2014). The proposed algorithm, coined Distributed Gibbs using Langevin Monte Carlo (DG-LMC), is depicted in Algorithm 1 and illustrated in Figure 1.

Algorithm 1 Distributed Gibbs using LMC (DG-LMC)

```
Input: burn-in $T_{\mathrm{bi}}$; for $i \in [b]$, tolerance parameters $\rho_i > 0$, step-sizes $\gamma_i \in (0, \rho_i/(1+\rho_i M_i)]$, local LMC steps $N_i \geq 1$.
Initialise $\boldsymbol{\theta}^{(0)}$ and $\mathbf{z}_{1:b}^{(0)}$.
for $t = 0$ to $T-1$ do
  // Sampling from $\Pi_{\boldsymbol{\rho}}(\mathbf{z}_{1:b}|\boldsymbol{\theta})$
  for $i = 1$ to $b$ // In parallel on the $b$ workers do
    $\mathbf{u}_i^{(0)} = \mathbf{z}_i^{(t)}$
    for $k = 0$ to $N_i - 1$ // $N_i$ local LMC steps do
      $\boldsymbol{\xi}_i^{(k,t)} \sim \mathrm{N}(\mathbf{0}_{d_i}, \mathbf{I}_{d_i})$
      $\mathbf{g}_i = \left(1 - \frac{\gamma_i}{\rho_i}\right)\mathbf{u}_i^{(k)} + \frac{\gamma_i}{\rho_i}\mathbf{A}_i\boldsymbol{\theta}^{(t)} - \gamma_i\nabla U_i\left(\mathbf{u}_i^{(k)}\right)$
      $\mathbf{u}_i^{(k+1)} = \mathbf{g}_i + \sqrt{2\gamma_i}\boldsymbol{\xi}_i^{(k,t)}$ // See (4)
    end for
    $\mathbf{z}_i^{(t+1)} = \mathbf{u}_i^{(N_i)}$
  end for
  // Sampling from $\Pi_{\boldsymbol{\rho}}(\boldsymbol{\theta}|\mathbf{z}_{1:b})$
  $\boldsymbol{\theta}^{(t+1)} \sim \mathrm{N}\left(\boldsymbol{\mu}\left(\mathbf{z}_{1:b}^{(t+1)}\right), \mathbf{Q}^{-1}\right)$ // See (5)
end for
Output: samples $\{\boldsymbol{\theta}^{(t)}\}_{t=T_{\mathrm{bi}}-1}^{T}$.
```

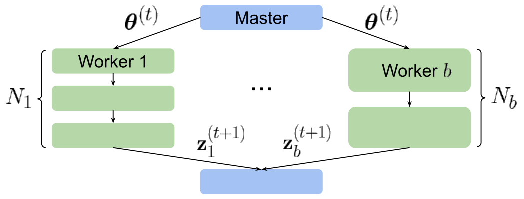

Figure 1. Illustration of one global iteration of Algorithm 1. For each worker, the width of the green box represents the amount of time required to perform one LMC step.

3. Detailed analysis of DG-LMC

In this section, we derive quantitative bias and convergence results for DG-LMC and show that its mixing time only scales quadratically w.r.t. the dimension d. We also discuss the choice of hyperparameters and provide guidelines to tune them.

3.1. Non-Asymptotic Analysis

The scope of our analysis will focus on smooth and strongly log-concave target posterior distributions π. While these assumptions may be restrictive in practice, they allow for a detailed theoretical study of the proposed algorithm.

H2. *(i) For any $i \in [b]$, U_i is twice continuously differentiable and* $\sup_{\mathbf{z}_i \in \mathbb{R}^{d_i}} \|\nabla^2 U_i(\mathbf{z}_i)\| \leq M_i$.
(ii) For any $i \in [b]$, U_i is m_i-strongly convex: there exists $m_i > 0$ such that $m_i \mathbf{I}_{d_i} \preceq \nabla^2 U_i$.

Under these assumptions, it is shown in Lemma S16 in the supplementary material that $-\log \pi$ is strongly convex with constant

$$m_U = \lambda_{\min}(\textstyle\sum_{i=1}^b m_i \mathbf{A}_i^\top \mathbf{A}_i) \ . \qquad (6)$$

Behind the use of LMC, the main motivation is to end up with a simple hybrid Gibbs sampler amenable to a non-asymptotic theoretical analysis based on previous works (Durmus & Moulines, 2019; Dalalyan & Karagulyan, 2019). In the following, this study is carried out using the Wasserstein distance of order 2.

3.1.1. Convergence Results

DG-LMC introduced in Algorithm 1 defines a homogeneous Markov chain $(V_t)_{t\in\mathbb{N}} = (\theta_t, Z_t)_{t\in\mathbb{N}}$ with realisations $(\boldsymbol{\theta}^{(t)}, \mathbf{z}_{1:b}^{(t)})_{t\in\mathbb{N}}$. We denote by $P_{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{N}}$ the Markov kernel associated with $(V_t)_{t\in\mathbb{N}}$. Since no Metropolis-Hastings step is used in combination with LMC, the proposed algorithm does not fall into the class of Metropolis-within-Gibbs samplers (Roberts & Rosenthal, 2006). Therefore, a first step is to show that $P_{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{N}}$ admits an unique invariant distribution and is geometrically ergodic. We proceed via an appropriate synchronous coupling which reduces the convergence analysis of $(V_t)_{t\in\mathbb{N}}$ to that of the marginal process $(Z_t)_{t\in\mathbb{N}}$. While the proof of the convergence of $(Z_t)_{t\in\mathbb{N}}$ shares some similarities with LMC (Durmus & Moulines, 2019), the analysis of $(Z_t)_{t\in\mathbb{N}}$ is much more involved and especially in the case $\max_{i\in[b]} N_i > 1$. We believe that the proof techniques we developed to show the next result can be useful to the study of other MCMC approaches based on LMC.

Proposition 2. *Assume **H** 1-**H** 2 and let $c > 0$ and $\boldsymbol{\gamma} = \{\gamma_i\}_{i=1}^b$ $\boldsymbol{N} = \{N_i\}_{i=1}^b$ satisfying $\max_{i\in[b]} \gamma_i \leq \bar{\gamma}$, $\min_{i\in[b]}\{N_i\gamma_i\}/\max_{i\in[b]}\{N_i\gamma_i\} \geq c$ and $\max_{i\in[b]}\{N_i\gamma_i\} \leq C_1$ where $\bar{\gamma}, C_1$ are explicit constants only depending on $(m_i, M_i, \rho_i)_{i\in[b]}$[1][2]. Then, there exists a probability measure $\Pi_{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{N}}$ such that $\Pi_{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{N}}$ is invariant for $P_{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{N}}$. Moreover there exists $C_2 > 0$ such that for any integer $t \geq 0$ and $\mathbf{v} = (\boldsymbol{\theta}, \mathbf{z}) \in \mathbb{R}^d \times \mathsf{Z}$, we have*

$$W_2(\delta_{\mathbf{v}} P^t_{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{N}}, \Pi_{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{N}}) \leq C_2 \cdot (1 - \min_{i\in[b]}\{N_i\gamma_i m_i\}/2)^t \times W_2(\delta_{\mathbf{v}}, \Pi_{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{N}}) \ .$$

Explicit expressions for C_1 and C_2 are given in Proposition S13 in the supplementary material. Finally, if $\boldsymbol{N} = N\mathbf{1}_b$ for $N \geq 1$, then $\Pi_{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{N}} = \Pi_{\boldsymbol{\rho},\boldsymbol{\gamma},\mathbf{1}_b}$.

We now discuss Proposition 2. If we set, for any $i \in [b]$, $N_i = 1$, the convergence rate in Proposition 2 becomes equal to $1 - \min_{i\in[b]}\{\gamma_i m_i\}/2$. In this specific case, we show in Proposition S5 in the supplementary material that DG-LMC actually admits the tighter convergence rate $1 - \min_{i\in[b]}\{\gamma_i m_i\}$ which simply corresponds to the rate at which the slowest LMC conditional kernel converges. On the other hand, when $\max_{i\in[b]} N_i > 1$, the convergence of $P_{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{N}}$ towards $\Pi_{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{N}}$ only holds if $\max_{i\in[b]}\{N_i\gamma_i\}$ is sufficiently small. This condition is necessary to ensure a contraction in W_2 and can be understood intuitively as follows in the case where $\boldsymbol{N} = N\mathbf{1}_b$ and $\boldsymbol{\gamma} = \gamma\mathbf{1}_b$. Given two vectors (θ_k, θ'_k) and an appropriate coupling (Z_{k+1}, Z'_{k+1}), we can show that $Z_{k+1} - Z'_{k+1}$ involves two competing terms: one keeping $Z_{k+1} - Z'_{k+1}$ close to $Z_k - Z'_k$ and another one driving $Z_{k+1} - Z'_{k+1}$ away from $\theta_k - \theta'_k$ (and therefore of $Z_k - Z'_k$) as N increases. This implies that N stands for a trade-off and the product $N\gamma$ cannot be arbitrarily chosen. Finally, it is worth mentioning that the tolerance parameters $\{\rho_i\}_{i\in[b]}$ implicitly drive the convergence rate of DG-LMC. In the case $N_i = 1$, a sufficient condition on the step-sizes to ensure a contraction is $\gamma_i \leq 2/(M_i + m_i + 1/\rho_i)$. We can denote that the smaller ρ_i, the smaller γ_i and the slower the convergence.

Starting from the results of Proposition 2, we can analyse the convergence properties of DG-LMC. We specify our result to the case where we take for the specific initial distribution

$$\mu^\star_{\boldsymbol{\rho}} = \delta_{\mathbf{z}^\star} \otimes \Pi_{\boldsymbol{\rho}}(\cdot|\mathbf{z}^\star) \ , \qquad (7)$$

where $\mathbf{z}^\star = ([\mathbf{A}_1\boldsymbol{\theta}^\star]^\top, \cdots, [\mathbf{A}_b\boldsymbol{\theta}^\star]^\top)^\top$, $\boldsymbol{\theta}^\star = \arg\min\{-\log\pi\}$ and $\Pi_{\boldsymbol{\rho}}(\cdot|\mathbf{z}^\star)$ is defined in (5). Note that sampling from $\mu^\star_{\boldsymbol{\rho}}$ is straightforward and simply consists in setting $\mathbf{z}^{(0)} = \mathbf{z}^\star$ and drawing $\boldsymbol{\theta}^{(0)}$ from $\Pi_{\boldsymbol{\rho}}(\cdot \mid \mathbf{z}^\star)$. For $t \geq 1$, we consider the marginal law of θ_t initialised at $\mathbf{v}^\star$ with distribution $\mu^\star_{\boldsymbol{\rho}}$ and denote it $\Gamma^t_{\mathbf{v}^\star}$. As mentioned previously, the proposed approach relies on two approximations which both come with some bias we need to control. This naturally brings us to consider the following inequality based on the triangular inequality and the definition of the Wasserstein distance:

$$W_2(\Gamma^t_{\mathbf{v}^\star}, \pi) \leq W_2(\mu^\star_{\boldsymbol{\rho}} P^t_{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{N}}, \Pi_{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{N}}) + W_2(\Pi_{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{N}}, \Pi_{\boldsymbol{\rho}}) + W_2(\pi_{\boldsymbol{\rho}}, \pi) \ , \qquad (8)$$

[1] When $\boldsymbol{N} = \mathbf{1}_b$, $C_1 = \bar{\gamma} = 1/\max_{i\in[b]}\{M_i + \rho_i^{-1}\}$.

[2] When $\max_{i\in[b]} N_i > 1$, C_1 is of order $\min_{i\in[b]} \rho_i^2$ when $\max_{i\in[b]} \rho_i \to 0$, see Lemma S12 in the supplement.

where $\Pi_{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{N}}$, $\Pi_{\boldsymbol{\rho}}$ and $\pi_{\boldsymbol{\rho}}$ are defined in Proposition 2, (2) and (3), respectively. In Proposition S14 in the supplementary material, we provide an upper bound on the first term on the right hand side based on Proposition 2. In the next section, we focus on controlling the last two terms on the right hand side.

3.1.2. Quantitative Bounds on the Bias

The error term $W_2(\pi_{\boldsymbol{\rho}}, \pi)$ in (8) is related to the underlying AXDA framework which induces an approximate posterior representation $\pi_{\boldsymbol{\rho}}$. It can be controlled by the sequence of positive tolerance parameters $\{\rho_i\}_{i=1}^b$. By denoting $\bar{\rho} = \max_{i\in[b]} \rho_i$, Proposition 3 shows that this error can be quantitatively assessed and is of order $\mathcal{O}(\bar{\rho})$ for sufficiently small values of this parameter.

Proposition 3. *Assume **H**1, **H**2. Let* $\mathbf{A} = [\mathbf{A}_1^\top, \dots, \mathbf{A}_b^\top]^\top$ *and denote* $\sigma_U^2 = \|\mathbf{A}^\top\mathbf{A}\| \max_{i\in[b]}\{M_i^2\}/m_U$, *where* m_U *is defined in* (6). *Then, for any* $\bar{\rho} \le \sigma_U^2/12$,

$$W_2(\pi_{\boldsymbol{\rho}}, \pi) \le \sqrt{2/m_U}\max(A_{\boldsymbol{\rho}}, B_{\boldsymbol{\rho}}) \ ,$$

where $A_{\boldsymbol{\rho}} = d\mathcal{O}(\bar{\rho})$ *and* $B_{\boldsymbol{\rho}} = d^{1/2}\mathcal{O}(\bar{\rho})$ *for* $\bar{\rho} \downarrow 0$. *Explicit expressions for* $A_{\boldsymbol{\rho}}, B_{\boldsymbol{\rho}}$ *are given in Section S3 in the supplementary material.*

In the case where π is Gaussian, the approximate distribution $\pi_{\boldsymbol{\rho}}$ admits an explicit expression and is Gaussian as well (e.g. when $b = 1$, the mean is the same and the covariance matrix is inflated by a factor $\rho\mathbf{I}_d$), see for instance Rendell et al. (2020, Section S2) and Vono et al. (2020b, Section 5.1). Hence, an explicit expression for $W_2(\pi_{\boldsymbol{\rho}}, \pi)$ can be derived. Based on this result, we can check that the upper bound provided by Proposition 3 matches the same asymptotics as $\rho \to 0$ and $d \to \infty$.

The second source of approximation error is induced by the use of LMC within Algorithm 1 to target the conditional distribution $\Pi_{\boldsymbol{\rho}}(\mathbf{z}_{1:b} \mid \boldsymbol{\theta})$ in (4). The stationary distribution of $P_{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{N}}$ whose existence is ensured in Proposition 2 differs from $\Pi_{\boldsymbol{\rho}}$. The associated bias is assessed quantitatively in Proposition 4.

Proposition 4. *Assume **H**1-**H**2. For any* $i \in [b]$, *define* $\tilde{M}_i = M_i + 1/\rho_i$ *and let* $\boldsymbol{\gamma} \in (\mathbb{R}_+^*)^b$, $\boldsymbol{N} \in (\mathbb{N}^*)^b$ *such that for any* $i \in [b]$,

$$\gamma_i \le \frac{m_i}{40\tilde{M}_i^2}\min_{i\in[b]}(m_i/\tilde{M}_i)^2/\max_{i\in[b]}(m_i/\tilde{M}_i)^2 \ , \tag{9}$$

$$N_i = \left\lfloor m_i \min_{i\in[b]}\{m_i/\tilde{M}_i\}^2/(20\gamma_i\tilde{M}_i^2\max_{i\in[b]}\{m_i/\tilde{M}_i\}^2)\right\rfloor . \tag{10}$$

Then, we have

$$W_2^2\left(\Pi_{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{N}}, \Pi_{\boldsymbol{\rho}}\right) \le C_3\sum_{i=1}^b d_i\gamma_i\tilde{M}_i^2 \ ,$$

where $C_3 > 0$ *only depends of* $(m_i, M_i, \mathbf{A}_i, \rho_i)_{i=1}^b$ *and is explicitly given in Proposition S29 in the supplementary material.*

With the notation $\bar{\gamma} = \max_{i\in[b]}\gamma_i$, Proposition 4 implies that $W_2(\Pi_{\boldsymbol{\rho}}, \Pi_{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{N}}) \le \mathcal{O}(\bar{\gamma}^{1/2})(\sum_{i=1}^b d_i)^{1/2}$ for $\bar{\gamma} \downarrow 0$. Note that this result is in line with Durmus & Moulines (2019, Corollary 7) and can be improved under further regularity assumptions on U, as shown below.

H3. *U is three times continuously differentiable and there exists $L_i > 0$ such that for all* $\mathbf{z}_i, \mathbf{z}_i' \in \mathbb{R}^{d_i}$, $\|\nabla^2 U_i(\mathbf{z}_i) - \nabla^2 U_i(\mathbf{z}_i')\| \le L_i\|\mathbf{z}_i - \mathbf{z}_i'\|$.

Proposition 5. *Assume **H**1-**H**2-**H**3. For any* $i \in [b]$, *define* $\tilde{M}_i = M_i + 1/\rho_i$ *and let* $\boldsymbol{\gamma} \in (\mathbb{R}_+^*)^b$, $\boldsymbol{N} \in (\mathbb{N}^*)^b$ *such that for any* $i \in [b]$, (9) *and* (10) *hold. Then, we have*

$$W_2^2\left(\Pi_{\boldsymbol{\rho},\boldsymbol{\gamma},\boldsymbol{N}}, \Pi_{\boldsymbol{\rho}}\right) \le C_4\sum_{i\in[b]} d_i\gamma_i(1/\tilde{M}_i^2 + \gamma_i\tilde{M}_i^2) \ ,$$

where $C_4 > 0$ *only depends on* $(m_i, M_i, L_i, \mathbf{A}_i, \rho_i)_{i=1}^b$ *and is explicitly given in Proposition S33 in the supplementary material.*

3.1.3. Mixing Time with Explicit Dependencies

Based on explicit non-asymptotic bounds shown in Propositions 2, 3 and 4 and the decomposition (8), we are now able to analyse the scaling of Algorithm 1 in high dimension. Given a prescribed precision $\varepsilon > 0$ and an initial condition $\mathbf{v}^\star$ with distribution $\mu_{\boldsymbol{\rho}}^\star$ given in (7), we define the ε-mixing time associated to $\Gamma_{\mathbf{v}^\star}$ by

$$t_{\text{mix}}(\varepsilon; \mathbf{v}^\star) = \min\left\{t \in \mathbb{N} : W_2\left(\Gamma_{\mathbf{v}^\star}^t, \pi\right) \le \varepsilon\right\} .$$

This quantity stands for the minimum number of DG-LMC iterations such that the $\boldsymbol{\theta}$-marginal distribution is at most at an ε W_2-distance from the initial target π. Under the condition that $b\max_{i\in[b]} d_i = \mathcal{O}(d)$ and by assuming for simplicity that for any $i \in [b]$, $m_i = m, M_i = M, L_i = L, \rho_i = \rho, \gamma_i = \gamma$ and $N_i = N$, Table 1 gathers the dependencies w.r.t. d and ε of the parameters involved in Algorithm 1 and of $t_{\text{mix}}(\varepsilon; \mathbf{v}^\star)$ to get a W_2-error of at most ε. Note that the mixing time of Algorithm 1 scales at most quadratically (up to polylogarithmic factors) in the dimension. When **H** 3 holds, we can see that the number of local iterations becomes independent of d and ε which leads to a total number of gradient evaluations with better dependencies w.r.t. to these quantities. Up to the authors' knowledge, these explicit results are the first among the centralised distributed MCMC literature and in particular give the dependency w.r.t. d and ε of the number of local LMC iterations on each worker. Overall, the proposed approach appears as a scalable and reliable alternative for high-dimensional and distributed Bayesian inference.

Table 1. For the specific initialisation $\mathbf{v}^\star$ with distribution $\mu_{\boldsymbol{\rho}}^\star$ given in (7), dependencies w.r.t. d and ε of the parameters involved in Algorithm 1 and of $t_{\mathrm{mix}}(\varepsilon; \mathbf{v}^\star)$ to get a W_2-error of at most ε.

Assumptions		ρ_ε	γ_ε	N_ε	$t_{\mathrm{mix}}(\varepsilon; \mathbf{v}^\star)$	Nb. of gradient evaluations
H1, **H**2	d	$\mathcal{O}(d^{-1})$	$\mathcal{O}(d^{-3})$	$\mathcal{O}(d)$	$\mathcal{O}(d^2 \log(d))$	$\mathcal{O}(d^3 \log(d))$
	ε	$\mathcal{O}(\varepsilon)$	$\mathcal{O}(\varepsilon^4)$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-2} \lvert \log(\varepsilon) \rvert)$	$\mathcal{O}(\varepsilon^{-4} \lvert \log(\varepsilon) \rvert)$
H1, **H**2, **H**3	d	$\mathcal{O}(d^{-1})$	$\mathcal{O}(d^{-2})$	$\mathcal{O}(1)$	$\mathcal{O}(d^2 \log(d))$	$\mathcal{O}(d^2 \log(d))$
	ε	$\mathcal{O}(\varepsilon)$	$\mathcal{O}(\varepsilon^2)$	$\mathcal{O}(1)$	$\mathcal{O}(\varepsilon^{-2} \lvert \log(\varepsilon) \rvert)$	$\mathcal{O}(\varepsilon^{-2} \lvert \log(\varepsilon) \rvert)$

3.2. DG-LMC in Practice: Guidelines for Practitioners

We now discuss practical guidelines for setting the values of hyperparameters involved in Algorithm 1. Based on Proposition 2, we theoretically show an optimal choice of order $N_i \gamma_i \asymp m_i \rho_i^2 / (\rho_i M_i + 1)^2$, see Lemma S26 in the supplementary material. Ideally, within the considered distributed setting, the optimal value for $(N_i, \gamma_i)_{i \in [b]}$ would boil down to optimise the value of $\max_{i \in [b]} \{N_i \gamma_i\}$ under the constraints derived in Proposition 2 combined with communication considerations. In particular, this would imply a comprehensive modelling of the communication costs including I/O bandwiths constraints. These optimisation tasks fall outside the scope of the present paper and therefore we let the search of optimal values for future works. Since our aim here is to provide practitioners with simple prescriptions, we rather focus on general rules involving tractable quantities.

3.2.1. Selection of $\boldsymbol{\gamma}$ and $\boldsymbol{\rho}$

From Durmus & Moulines (2017) and references therein, a simple sufficient condition on step-sizes $\boldsymbol{\gamma} = \{\gamma_i\}_{i=1}^b$ to guarantee the stability of LMC is $\gamma_i \leq \rho_i / (\rho_i M_i + 1)$ for $i \in [b]$. Both the values of γ_i and ρ_i are subject to a bias-variance trade-off. More precisely, large values yield a Markov chain with small estimation variance but high asymptotic bias. Conversely, small values produce a Markov chain with small asymptotic bias but which requires a large number of iterations to obtain a stable estimator. We propose to mitigate this trade-off by setting γ_i to a reasonably large value, that is for $i \in [b]$, $\gamma_i \in [0.1 \rho_i / (\rho_i M_i + 1), 0.5 \rho_i / (\rho_i M_i + 1)]$. Since γ_i saturates to $1/M_i$ when $\rho_i \to \infty$, there is no computational advantage to choose very large values for ρ_i. Based on several numerical studies, we found that setting ρ_i of the order of $1/M_i$ was a good compromise between computational efficiency and asymptotic bias.

3.2.2. $\boldsymbol{N}$: A Trade-Off between Asymptotic Bias and Communication Overhead

In a similar vein, the choice of $\boldsymbol{N} = \{N_i\}_{i=1}^b$ also stands for a trade-off but here between asymptotic accuracy and communication costs. Indeed, a large number of local LMC iterations reduces the communication overhead but at the expense of a larger asymptotic bias since the master parameter is not updated enough. Ahn et al. (2014) proposed to tune the number of local iterations N_i on a given worker based on the amount of time needed to perform one local iteration, denoted here by τ_i. Given an average number of local iterations N_{avg}, the authors set $N_i = q_i N_{\mathrm{avg}} b$ with $q_i = \tau_i^{-1} / \sum_{k=1}^b \tau_k^{-1}$ so that $b^{-1} \sum_{i=1}^b N_i = N_{\mathrm{avg}}$. As mentioned by the aforementioned authors, this choice allows to keep the block-by-the-slowest delay small by letting fast workers perform more iterations in the same wall-clock time. Although they showed how to tune N_i w.r.t. communication considerations, they let the choice of N_{avg} to the practitioner. Here, we propose a simple guideline to set N_{avg} such that N_i stands for a good compromise between the amount of time spent on exploring the state-space and communication overhead. As highlighted in the discussion after Proposition 2, as γ_i becomes smaller, more local LMC iterations are required to sufficiently explore the latent space before the global consensus round on the master. Assuming for any $i \in [b]$ that γ_i has been chosen following our guidelines in Section 3.2.1, this suggests to set $N_{\mathrm{avg}} = \lceil (1/b) \sum_{i \in [b]} \rho_i / (\gamma_i [\rho_i M_i + 1]) \rceil$.

4. Related work

As already mentioned in Section 1, hosts of contributions have focused on deriving distributed MCMC algorithms to sample from (1). This section briefly reviews the main existing research lines and draws a detailed comparison with the proposed methodology.

4.1. Existing distributed MCMC methods

Existing methodologies are mostly approximate and can be loosely speaking divided into three groups: *one-shot*,

Table 2. Synthetic overview of the main existing distributed MCMC methods under a master-slave architecture. The column *Exact* means that the Markov chain defined by the MCMC sampler admits (1) as invariant distribution. The column *Comm. overhead* reports the communication frequency. A value of 1 means that the sampler communicates after every iteration. T stands for the total number of iterations and $N < T$ is a tunable parameter to mitigate communication costs. The acronym D-SGLD stands for distributed stochastic gradient Langevin dynamics.

METHOD	TYPE	EXACT	COMM. OVERHEAD	BIAS BOUNDS	SCALING W.R.T. d
WANG & DUNSON (2013)	ONE-SHOT	$\times$	$1/T$	$\surd$	$\mathcal{O}(\mathrm{e}^d)$
NEISWANGER ET AL. (2014)	ONE-SHOT	$\times$	$1/T$	$\times$	$\mathcal{O}(\mathrm{e}^d)$
MINSKER ET AL. (2014)	ONE-SHOT	$\times$	$1/T$	$\surd$	UNKNOWN
SRIVASTAVA ET AL. (2015)	ONE-SHOT	$\times$	$1/T$	$\times$	UNKNOWN
WANG ET AL. (2015)	ONE-SHOT	$\times$	$1/T$	$\surd$	$\mathcal{O}(\mathrm{e}^d)$
SCOTT ET AL. (2016)	ONE-SHOT	$\times$	$1/T$	$\times$	UNKNOWN
NEMETH & SHERLOCK (2018)	ONE-SHOT	$\times$	$1/T$	$\times$	UNKNOWN
JORDAN ET AL. (2019)	ONE-SHOT	$\times$	$1/T$	$\surd$	UNKNOWN
AHN ET AL. (2014)	D-SGLD	$\times$	$1/N$	$\times$	UNKNOWN
CHEN ET AL. (2016)	D-SGLD	$\times$	1	$\surd$	UNKNOWN
EL MEKKAOUI ET AL. (2020)	D-SGLD	$\times$	$1/N$	$\surd$	UNKNOWN
RABINOVICH ET AL. (2015)	G. CONSENSUS	$\times$	$1/N$	$\times$	UNKNOWN
CHOWDHURY & JERMAINE (2018)	G. CONSENSUS	$\surd$	1	N/A	UNKNOWN
RENDELL ET AL. (2020)	G. CONSENSUS	$\times$	$1/N$	$\surd$	UNKNOWN
THIS PAPER	G. CONSENSUS	$\times$	$1/N$	$\surd$	$\mathcal{O}(d^2 \log(d))$

distributed stochastic gradient MCMC and *global consensus* approaches. To ease the understanding, a synthetic overview of their main characteristics is presented in Table 2.

One-shot approaches stand for communication-efficient schemes where workers and master only exchange information at the very beginning and the end of the sampling task; similarly to MapReduce schemes (Dean & Ghemawat, 2004). Most of these methods assume that the posterior density factorises into a product of local posteriors and launch independent Markov chains across workers to target them. The local posterior samples are then combined through the master node using a single final aggregation step. This step turns to be the milestone of one-shot approaches and was the topic of multiple contributions (Wang & Dunson, 2013; Neiswanger et al., 2014; Minsker et al., 2014; Srivastava et al., 2015; Scott et al., 2016; Nemeth & Sherlock, 2018). Unfortunately, the latter are either infeasible in high-dimensional settings or have been shown to yield inaccurate posterior representations empirically, if the posterior is not near-Gaussian, or if the local posteriors differ significantly (Wang et al., 2015; Dai et al., 2019; Rendell et al., 2020). Alternative schemes have been recently proposed to tackle these issues but their theoretical scaling w.r.t. the dimension d is currently unknown (Jordan et al., 2019; Mesquita et al., 2020).

Albeit popular in the machine learning community, distributed stochastic gradient MCMC methods (Ahn et al., 2014) suffer from high variance when the dataset is large because of the use of stochastic gradients (Brosse et al., 2018). Some surrogates have been recently proposed to reduce this variance such as the use of *stale* or *conducive* gradients (Chen et al., 2016; El Mekkaoui et al., 2020). However, these variance reduction methods require an increasing number of workers for the former and come at the price of a prohibitive pre-processing step for the latter. In addition, it is currently unclear whether these methods are able to generate efficiently accurate samples from a given target distribution.

Contrary to aforementioned distributed MCMC approaches, global consensus methods periodically share information between workers by performing a consensus round between the master and the workers (Rabinovich et al., 2015; Chowdhury & Jermaine, 2018; Vono et al., 2019a; Rendell et al., 2020). Again, they have been shown to perform well in practice but their theoretical understanding is currently limited.

4.2. Comparison with the proposed methodology

Table 2 compares Algorithm 1 with existing approaches detailed previously. In addition to having a simple implementation and guidelines, it is worth noticing that DG-LMC appears to benefit from favorable convergence properties compared to the other considered methodologies.

We complement this comparison with an informal discussion on the computational and communication complexities of Algorithm 1. Recall that the dataset is assumed to be partitioned into S shards and stored on S workers among a collection of b computing nodes. Suppose that the s-th shard has size n_s, and let T be the number of total MCMC iterations and c_{com} the communication cost. In addition, denote by $c_{\mathrm{eval}}^{(i)}$ the approximate wall-clock time required to evaluate U_i or its gradient. For the ease of exposition, we do not discuss the additional overhead due to bandwidth restrictions and assume similar computation costs, *i.e.*, $Nc_{\text{eval}} \simeq N_i c_{\text{eval}}^{(i)}$, to perform each local LMC step at each iteration of Algorithm 1. Under these assumptions, the total complexity of Algorithm 1 is $\mathcal{O}(T[2c_{\text{com}} + Nc_{\text{eval}}])$. Following the same reasoning, distributed stochastic gradient Langevin dynamics (D-SGLD) and one-shot approaches admit complexities of the order $\mathcal{O}(T[2c_{\text{com}} + Nc_{\text{eval}} n_{\text{mb}}/n_s])$ and $\mathcal{O}(Tc_{\text{eval}} + 2c_{\text{com}})$, respectively. The integer n_{mb} stands for the mini-batch size used in D-SGLD. Despite their very low communication overhead, existing one-shot approaches are rarely reliable and therefore not necessarily efficient to sample from π given a prescribed computational budget, see Rendell et al. (2020) for a recent overview. D-SGLD seems to enjoy a lower complexity than Algorithm 1 when n_{mb} is small. Unfortunately, this choice comes with two main shortcomings: (i) a larger number of iterations T to achieve the same precision because of higher variance of gradient estimators, and (ii) a smaller amount of time spent on exploration compared to communication latency. By falling into the global consensus class of methods, the proposed methodology hence appears as a good compromise between one-shot and D-SGLD algorithms in terms of both computational complexity and accuracy. Section 5 will enhance the benefits of Algorithm 1 by showing experimentally better convergence properties and posterior approximation.

5. Experiments

This section compares numerically DG-LMC with the most popular and recent centralised distributed MCMC approaches namely D-SGLD and the global consensus Monte Carlo (GCMC) algorithm proposed in Rendell et al. (2020). Since all these approaches share the same communication latency, this feature is not discussed here.

5.1. Toy Gaussian Example

In this toy example, we first illustrate the behavior of DG-LMC w.r.t. the number of local iterations which drives the communication overhead. We consider the conjugate Gaussian model $\pi(\boldsymbol{\theta}|\mathbf{y}_{1:n}) \propto \mathrm{N}(\boldsymbol{\theta}|\mathbf{0}_d, \boldsymbol{\Sigma}_{\mathbf{0}}) \prod_{i=1}^{n} \mathrm{N}(\mathbf{y}_i|\boldsymbol{\theta}, \boldsymbol{\Sigma}_{\mathbf{1}})$, with positive definite matrices $\boldsymbol{\Sigma}_{\mathbf{0}}, \boldsymbol{\Sigma}_{\mathbf{1}}$. We set $d = 2$, allocate $n = 20,000$ observations to a cluster made of $b = 10$ workers and compare DG-LMC with D-SGLD. Both MCMC algorithms have been run using the same number of local iterations N per worker and for a fixed budget of $T = 100,000$ iterations including a burn-in period equal to $T_{\text{bi}} = 10,000$. Regarding DG-LMC, we follow the guidelines in Section 3.2.1 and set for all $i \in [b]$, $\mathbf{A}_i = \mathbf{I}_d$, $\rho_i = 1/(5M_i)$ and $\gamma_i = 0.25\rho_i/(\rho_i M_i + 1)$. On the other hand, D-SGLD has been run with batch-size $n/(10b)$ and

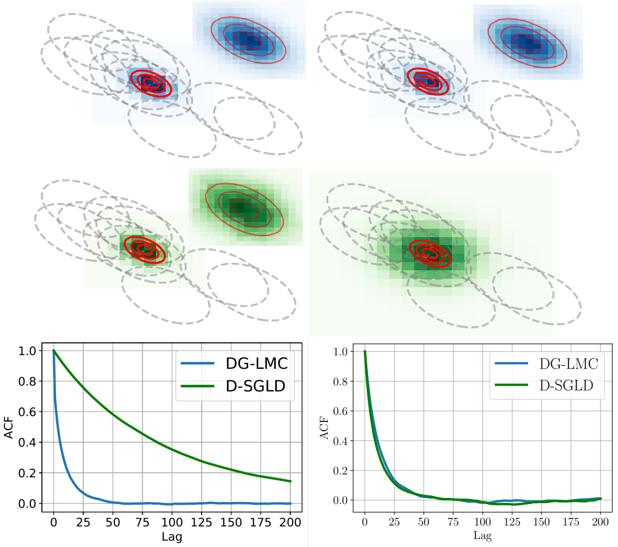

Figure 2. Toy Gaussian experiment. (left) $N = 1$ local iterations and (right) $N = 10$. (top) DG-LMC, (middle) D-SGLD and (bottom) ACF comparison between DG-LMC and D-SGLD.

a step-size chosen such that the resulting posterior approximation is similar to that of DG-LMC for $N = 1$. Figure 2 depicts the results for $N = 1$ and $N = 10$ on the left and right columns, respectively. The top row (resp. middle row) shows the contours of the b local posteriors in dashed grey, the contours of the target posterior in red and the 2D histogram built with DG-LMC (resp. D-SGLD) samples in blue (resp. green). When required, a zoomed version of these figures is depicted at the top right corner. It can be noted that DG-LMC exhibits better mixing properties while achieving similar performances as shown by the autocorrelation function (ACF) on the bottom row. Furthermore, its posterior approximation is robust to the choice of N in contrast to D-SGLD, which needs further tuning of its step-size to yield an accurate posterior representation. This feature is particularly important for distributed computations since N is directly related to communication costs and might often change depending upon the hardware architecture.

5.2. Bayesian Logistic Regression

This second experiment considers a more challenging problem namely Bayesian logistic regression. We use the *covtype*[3] dataset with $d = 54$ and containing n = 581,012 observations partitioned into $b = 16$ shards. We set $N = 10$, $T = 200,000$, $T_{\text{bi}} = T/10$ for all approaches, and again used the guidelines in Section 3.2.1 to tune DG-LMC. Under the Bayesian paradigm, we are interested in performing uncertainty quantification by estimating highest posterior density (HPD) regions. For any $\alpha \in (0,1)$, define $\mathcal{C}_\alpha = \{\boldsymbol{\theta} \in \mathbb{R}^d; -\log \pi(\boldsymbol{\theta}|\mathbf{y}_{1:n}) \leq \eta_\alpha\}$ where $\eta_\alpha \in \mathbb{R}$

[3] www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets

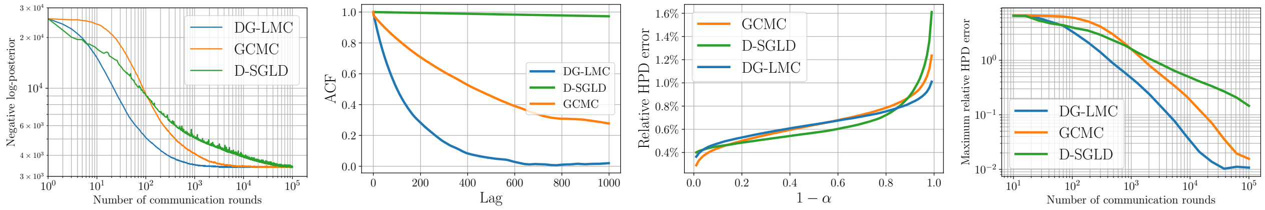

Figure 3. Logistic regression. From left to right: negative log-posterior, ACF, HPD relative error after and during the sampling procedure.

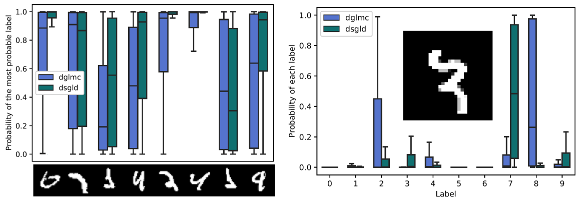

Figure 4. Bayesian neural network. (left) probability of the most probable label for 8 examples and (right) probability of each label for a single example.

is chosen such that $\int_{\mathcal{C}_\alpha} \pi(\boldsymbol{\theta}|\mathbf{y}_{1:n})\mathrm{d}\boldsymbol{\theta} = 1 - \alpha$. For the three approximate MCMC approaches, we computed the relative HPD error based on the scalar summary η_α, *i.e.* $|\eta_\alpha - \eta_\alpha^{\text{true}}|/\eta_\alpha^{\text{true}}$ where $\eta_\alpha^{\text{true}}$ has been estimated using the Metropolis adjusted Langevin algorithm. The parameters of GCMC and D-SGLD have been chosen such that all MCMC algorithms achieve similar HPD error. Figure 3 shows that this error is reasonable and of the order of 1%. Nonetheless, one can denote that DG-LMC achieves this precision level faster than GCMC and D-SGLD due to better mixing properties. This confirms that the proposed methodology is indeed efficient and reliable to perform Bayesian analyses compared to existing popular methodologies.

5.3. Bayesian Neural Network

Up to now, both our theoretical and experimental results focused on the strongly log-concave scenario and showed that even in this case, DG-LMC appeared as a competitive alternative. In this last experiment, we propose to end the study of DG-LMC on an open note without ground truth by tackling the challenging sampling problem associated to Bayesian neural networks. We consider the MNIST training dataset consisting of $n = 60,000$ observations partitioned into $b = 50$ shards and such that for any $i \in [n]$ and $k \in [10]$, $\mathbb{P}(y_i = k|\boldsymbol{\theta}, \mathbf{x}_i) = \beta_k$ where β_k is the k-th element of $\sigma(\sigma(\mathbf{x}_i^\top \mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2)$, $\sigma(\cdot)$ is the sigmoid function, $\mathbf{x}_i$ are covariates, and $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$ and $\mathbf{b}_2$ are matrices of size 784×128, 128 × 10, 1×128 and 1×10, respectively. We set normal priors for each weight matrix and bias vector, $N = 10$ and ran DG-LMC with constant hyperparameters across workers $(\rho, \gamma) = (0.02, 0.005)$ and D-SGLD using a step-size of 10^{-5}. Exact MCMC approaches are too computationally costly to launch for this experiment and therefore no ground truth about the true posterior distribution is available. To this purpose, Figure 4 only compares the credibility regions associated to the posterior predictive distribution. Similarly to previous experiments, we found that D-SGLD was highly sensitive to hyperparameters choices (step-size and mini-batch size). Except for a few testing examples, most of conclusions given by DG-LMC and D-SGLD regarding the predictive uncertainty coincide. In addition, posterior accuracies on the test set given by both algorithms are similar.

6. Conclusion

In this paper, a simple algorithm coined DG-LMC has been introduced for distributed MCMC sampling. In addition, it has been established that this method inherits favorable convergence properties and numerical illustrations support our claims.

Acknowledgements

The authors acknowledge support of the Lagrange Mathematics and Computing Research Center.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pp. 308318, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341394. doi: 10.1145/2976749.2978318.

Ahn, S., Shahbaba, B., and Welling, M. Distributed Stochastic Gradient MCMC. In Xing, E. P. and Jebara, T. (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1044–1052, 2014.

Bardenet, R., Doucet, A., and Holmes, C. On Markov chain Monte Carlo methods for tall data. *Journal of Machine Learning Research*, 18(47):1–43, 2017.

Bernstein, P. A. and Newcomer, E. *Principles of Transaction*

Processing. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2009. ISBN 1558606238.

Bottou, L., Curtis, F. E., and Nocedal, J. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018. doi: 10.1137/16M1080173.

Brosse, N., Moulines, E., and Durmus, A. The Promises and Pitfalls of Stochastic Gradient Langevin Dynamics. In *Neural Information Processing Systems*, pp. 82788288, 2018.

Bui, T. D., Nguyen, C. V., Swaroop, S., and Turner, R. E. Partitioned Variational Inference: A unified framework encompassing federated and continual learning. *arXiv preprint arXiv:1811.11206*, 2018.

Chen, C., Ding, N., Li, C., Zhang, Y., and Carin, L. Stochastic Gradient MCMC with Stale Gradients. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29, pp. 2937–2945. Curran Associates, Inc., 2016.

Chowdhury, A. and Jermaine, C. Parallel and Distributed MCMC via Shepherding Distributions. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84, pp. 1819–1827, 2018.

Dai, H., Pollock, M., and Roberts, G. Monte Carlo fusion. *Journal of Applied Probability*, 56(1):174191, 2019. doi: 10.1017/jpr.2019.12.

Dalalyan, A. S. and Karagulyan, A. User-friendly guarantees for the langevin monte carlo with inaccurate gradient. *Stochastic Processes and Their Applications*, 129(12): 5278–5311, 2019. doi: 10.1016/j.spa.2019.02.016.

Dean, J. and Ghemawat, S. MapReduce: Simplified Data Processing on Large Clusters. In *OSDI'04: Sixth Symposium on Operating System Design and Implementation*, pp. 137–150, San Francisco, CA, 2004.

Dieuleveut, A. and Patel, K. K. Communication trade-offs for Local-SGD with large step size. In *Advances in Neural Information Processing Systems*, volume 32, pp. 13601–13612, 2019.

Durmus, A. and Moulines, E. Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. *The Annals of Applied Probability*, 27(3):1551–1587, 06 2017. doi: 10.1214/16-AAP1238.

Durmus, A. and Moulines, E. High-dimensional Bayesian inference via the unadjusted Langevin algorithm. *Bernoulli*, 25(4A):2854–2882, 2019. doi: 10.3150/18-BEJ1073.

El Mekkaoui, K., Mesquita, D., Blomstedt, P., and Kaski, S. Distributed stochastic gradient MCMC for federated learning. *arXiv preprint arXiv:2004.11231*, 2020.

Hasenclever, L., Webb, S., Lienart, T., Vollmer, S., Lakshminarayanan, B., Blundell, C., and Teh, Y. W. Distributed bayesian learning with stochastic natural gradient expectation propagation and the posterior server. *Journal of Machine Learning Research*, 18(106):1–37, 2017.

Johnson, M., Saunderson, J., and Willsky, A. Analyzing Hogwild parallel Gaussian Gibbs sampling. In *Neural Information Processing Systems*, pp. 2715–2723, 2013.

Jordan, M. I., Lee, J. D., and Yang, Y. Communication-Efficient Distributed Statistical Inference. *Journal of the American Statistical Association*, 114(526):668–681, 2019. doi: 10.1080/01621459.2018.1429274.

Mesquita, D., Blomstedt, P., and Kaski, S. Embarrassingly Parallel MCMC using Deep Invertible Transformations. In Adams, R. P. and Gogate, V. (eds.), *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pp. 1244–1252, 2020.

Minsker, S., Srivastava, S., Lin, L., and Dunson, D. Scalable and robust Bayesian inference via the median posterior. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Neiswanger, W., Wang, C., and Xing, E. P. Asymptotically exact, embarrassingly parallel MCMC. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence*, 2014.

Nemeth, C. and Sherlock, C. Merging MCMC Subposteriors through Gaussian-Process Approximations. *Bayesian Analysis*, 13(2):507–530, 06 2018. doi: 10.1214/17-BA1063.

Rabinovich, M., Angelino, E., and Jordan, M. I. Variational Consensus Monte Carlo. In *Advances in Neural Information Processing Systems*, volume 28, pp. 1207–1215, 2015.

Raicu, I., Foster, I., Szalay, A., and Turcu, G. AstroPortal: A Science Gateway for Large-scale Astronomy Data Analysis. In *TeraGrid Conference*, pp. 12–15, 2006.

Rendell, L. J., Johansen, A. M., Lee, A., and Whiteley, N. Global consensus Monte Carlo. *Journal of Computational and Graphical Statistics*, 2020.

Robert, C. P. *The Bayesian Choice: from decision-theoretic foundations to computational implementation*. Springer, New York, 2 edition, 2001.

Roberts, G. O. and Rosenthal, J. S. Harris recurrence of Metropolis-within-Gibbs and trans-dimensional Markov chains. *Annals of Applied Probability*, 16(4):2123–2139, 11 2006. doi: 10.1214/105051606000000510.

Roberts, G. O. and Tweedie, R. L. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 12 1996.

Rossky, P. J., Doll, J. D., and Friedman, H. L. Brownian dynamics as smart Monte Carlo simulation. *The Journal of Chemical Physics*, 69(10):4628–4633, 1978. doi: http://dx.doi.org/10.1063/1.436415.

Scheffé, H. A useful convergence theorem for probability distributions. *The Annals of Mathematical Statistics*, 18 (3):434–438, 1947. doi: 10.1214/aoms/1177730390.

Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and McCulloch, R. E. Bayes and Big Data: The Consensus Monte Carlo Algorithm. *International Journal of Management Science and Engineering Management*, 11:78–88, 2016.

Shokri, R. and Shmatikov, V. Privacy-preserving deep learning. In *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 909–910, 2015. doi: 10.1109/ALLERTON.2015.7447103.

Srivastava, S., Cevher, V., Dinh, Q., and Dunson, D. WASP: Scalable Bayes via barycenters of subset posteriors. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, volume 38, pp. 912–920, 2015.

Vehtari, A., Gelman, A., Sivula, T., Jylanki, P., Tran, D., Sahai, S., Blomstedt, P., Cunningham, J. P., Schiminovich, D., and Robert, C. P. Expectation Propagation as a Way of Life: A Framework for Bayesian Inference on Partitioned Data. *Journal of Machine Learning Research*, 21(17): 1–53, 2020.

Verbraeken, J., Wolting, M., Katzy, J., Kloppenburg, J., Verbelen, T., and Rellermeyer, J. S. A survey on distributed machine learning. *ACM Comput. Surv.*, 53(2), March 2020. ISSN 0360-0300. doi: 10.1145/3377454.

Vono, M., Dobigeon, N., and Chainais, P. Split-and-augmented Gibbs sampler - Application to large-scale inference problems. *IEEE Transactions on Signal Processing*, 67(6):1648–1661, 2019a. doi: 10.1109/TSP.2019.2894825.

Vono, M., Paulin, D., and Doucet, A. Efficient MCMC sampling with dimension-free convergence rate using ADMM-type splitting. *arXiv preprint arXiv:1905.11937*, 2019b.

Vono, M., Dobigeon, N., and Chainais, P. High-dimensional Gaussian sampling: A review and a unifying approach based on a stochastic proximal point algorithm. *arXiv preprint arXiv:2010.01510*, 2020a.

Vono, M., Dobigeon, N., and Chainais, P. Asymptotically exact data augmentation: models, properties and algorithms. *Journal of Computational and Graphical Statistics*, 2020b. doi: 10.1080/10618600.2020.1826954.

Wang, X. and Dunson, D. B. Parallelizing MCMC via Weierstrass sampler. *arXiv preprint arXiv:1312.4605*, 2013.

Wang, X., Guo, F., Heller, K. A., and Dunson, D. B. Parallelizing MCMC with random partition trees. In *Advances in Neural Information Processing Systems*, 2015.