PARALLELIZED STOCHASTIC GRADIENT MARKOV CHAIN MONTE CARLO ALGORITHMS FOR NON-NEGATIVE MATRIX FACTORIZATION

Umut Şimşekli[1]*, Alain Durmus*[1]*, Roland Badeau*[1]*, Gaël Richard*[1]*, Éric Moulines*[2]*, A. Taylan Cemgil*[3]

1: LTCI, Télécom ParisTech, Université Paris-Saclay, 75013, Paris, France
2: Center of Applied Mathematics, École Polytechnique, France
3: Department of Computer Engineering, Boğaziçi University, 34342, Bebek, İstanbul, Turkey

ABSTRACT

Stochastic Gradient Markov Chain Monte Carlo (SG-MCMC) methods have become popular in modern data analysis problems due to their computational efficiency. Even though they have proved useful for many statistical models, the application of SG-MCMC to non-negative matrix factorization (NMF) models has not yet been extensively explored. In this study, we develop two parallel SG-MCMC algorithms for a broad range of NMF models. We exploit the conditional independence structure of the NMF models and utilize a stratified sub-sampling approach for enabling parallelization. We illustrate the proposed algorithms on an image restoration task and report encouraging results.

Index Terms— Stochastic Gradient MCMC, Non-Negative Matrix Factorization, Tweedie Distribution, Beta Divergence, Richardson-Romberg Extrapolation.

1. INTRODUCTION

Non-negative matrix factorization (NMF) models [1] have been widely used in data analysis and have been shown to be useful in various domains, such as recommender systems, audio processing, finance, computer vision, and bioinformatics [2–4]. The aim of an NMF model is to decompose an observed data matrix $\mathbf{V} \in \mathbb{R}_+^{I\times J}$ in the form: $\mathbf{V} \approx \mathbf{WH}$, where $\mathbf{W} \in \mathbb{R}_+^{I\times K}$ and $\mathbf{H} \in \mathbb{R}_+^{K\times J}$ are the factor matrices, known typically as the dictionary and the weight matrices respectively, to be estimated by minimizing a cost function. The typical cost functions used in NMF can be listed as the Euclidean distance, the Kullback-Leibler divergence [1], and the Itakura-Saito divergence [5].

More general noise models and regularization methods can be developed. A popular approach for this task is to use a probabilistic model having the following hierarchical generative structure: $p(\mathbf{W}) = \prod_{ik} p(w_{ik})$, $p(\mathbf{H}) = \prod_{kj} p(h_{kj})$, $p(\mathbf{V}|\mathbf{WH}) = \prod_{ij} p(v_{ij}|\mathbf{W},\mathbf{H})$, where v_{ij}, w_{ik}, and h_{kj} denote the elements of $\mathbf{V}$, $\mathbf{W}$, and $\mathbf{H}$, respectively. Within this probabilistic context, we are interested in the posterior distribution of the latent factors $\mathbf{W}$ and $\mathbf{H}$. By using the Bayes' theorem, the posterior distribution has a density given by

$$p(\mathbf{W},\mathbf{H}|\mathbf{V}) \propto p(\mathbf{V}|\mathbf{W},\mathbf{H})p(\mathbf{W})p(\mathbf{H}), \tag{1}$$

where $\propto$ denotes proportionality up to a multiplicative constant.

This work is partly supported by the French National Research Agency (ANR) as a part of the EDISON 3D (ANR-13-CORD-0008-02) and FBIMATRIX projects.

The majority of the current literature on NMF focuses on obtaining point estimates such as the Maximum a-posteriori (MAP) estimate, defined as follows:

$$(\mathbf{W},\mathbf{H})^\star = \arg\max_{\mathbf{W},\mathbf{H}} \log p(\mathbf{W},\mathbf{H}|\mathbf{V}) \ . \tag{2}$$

The point estimates are obtained via optimization methods applied to the posterior density Eq. 1. The optimization procedures used to solve Eq. 2 have various theoretical guarantees which hold under appropriate conditions on the prior and likelihood functions [1, 5–8]. On the other hand, we might be interested in other quantities based on the posterior distribution, such as the moments or the normalizing constants. These quantities are useful in various applications such as model selection [9] (i.e. estimating the 'rank' K of the model) or estimating the Bayesian predictive densities that would be useful for active learning. Markov Chain Monte Carlo (MCMC) algorithms, which aim to generate samples from the posterior distribution of interest, are one of the most popular approaches for estimating these quantities. However, these methods have received less attention, mainly due to their computational complexity and rather slow convergence of the standard methods, e.g. the Gibbs sampler.

In the recent years, alternative approaches based on stochastic optimization have been proposed for scaling up MCMC algorithms. These so-called Stochastic Gradient MCMC (SG-MCMC) methods require to 'see' only a small subset of the data per iteration similarly to the stochastic optimization algorithms and they are well adapted to modern parallel and distributed architectures. Even though these methods have proved useful for many statistical models, their applications on NMF models have not yet been extensively explored. In a more general matrix factorization context, a distributed SG-MCMC algorithm for the matrix factorization framework has been proposed by Ahn et al. [10], where the authors focused on a particular probabilistic model, called the Bayesian probabilistic matrix factorization (BPMF) [11].

In this study, we develop two parallel SG-MCMC algorithms for sampling from the full posterior of a broad range of NMF models, including models that are not easily tackled by using standard methods such as the Gibbs sampler. Our methods are carefully designed for NMF models; they exploit the conditional independence structure of NMF in order to enable parallelism. Our first algorithm builds upon the generic framework that was proposed in [10], whereas the second algorithm is a novel parallel variant of a recently proposed SG-MCMC algorithm [12] that can achieve faster convergence rates. Both of the proposed algorithms have favorable scaling properties and are computationally efficient due to their inherent parallelism. We illustrate the proposed algorithms on an image restoration task.

2. TWEEDIE NON-NEGATIVE MATRIX FACTORIZATION

In this study, we consider the following probabilistic model in order to be able to cover a wide range of likelihood functions:

$$p(\mathbf{W}) = \prod_{ik} \mathcal{E}(w_{ik}; \lambda_w), \qquad p(\mathbf{H}) = \prod_{kj} \mathcal{E}(h_{kj}; \lambda_h)$$
$$p(\mathbf{V}|\mathbf{W}, \mathbf{H}) = \prod_{ij} \mathcal{TW}(v_{ij}; \sum_k w_{ik} h_{kj}, \phi, \beta) \tag{3}$$

where $\mathcal{E}$ and $\mathcal{TW}$ denote the exponential and Tweedie distributions, respectively. The Tweedie distributions belong to the exponential dispersion models [13] and has shown to be useful for factorization-based modeling [5, 14–17]. The Tweedie distributions have densities which can be written in the following form:

$$\mathcal{TW}(v; \mu, \phi, \beta) = \frac{1}{K(x, \phi, \beta)} \exp\Big(-\frac{1}{\phi} d_\beta(v||\mu)\Big) \tag{4}$$

where μ is the mean, ϕ is the dispersion (related to the variance), β is the power parameter, $K(\cdot)$ is the normalizing constant, and $d_\beta(\cdot)$ denotes the β-divergence that is defined as follows:

$$d_\beta(v||\mu) = \frac{v^\beta}{\beta(\beta-1)} - \frac{v\mu^{\beta-1}}{\beta-1} + \frac{\mu^\beta}{\beta}. \tag{5}$$

The β-divergence generalizes several divergence functions that are commonly used in practice. As special cases, we obtain the Itakura-Saito divergence, Kullback-Leibler divergence, and the squared Euclidean distance, for $\beta = 0, 1, 2$, respectively. From the probabilistic perspective, different choices of β yield important distributions such as gamma ($\beta = 0$), Poisson ($\beta = 1$), Gaussian ($\beta = 2$), compound Poisson ($0 < \beta < 1$) [13, 14], and inverse Gaussian ($\beta = -1$) distributions. Due to a technical condition, no Tweedie model exists for the interval $1 < \beta < 2$, but for all other values of β, one obtains the very rich family of Tweedie stable distributions [13]. Thanks to the flexibility of this class of models, we are able to choose an observation model by changing a single parameter β.

3. STOCHASTIC GRADIENT MCMC FOR NMF

In Bayesian machine learning, we are often interested in approximating posterior expectations of a test function f, given as follows:

$$\bar{f} = \int f(\mathbf{\Theta}) \pi(d\mathbf{\Theta}) \approx \hat{f} = \frac{1}{T} \sum_{t=1}^{T} f(\mathbf{\Theta}^{(t)}) \tag{6}$$

where $\mathbf{\Theta} \equiv \{\mathbf{W}, \mathbf{H}\}$, $\pi(\mathbf{\Theta}) = p(\mathbf{W}, \mathbf{H}|\mathbf{V})$ is the posterior distribution, and $\mathbf{\Theta}^{(t)}$ are samples that are ideally drawn from the target distribution π. However, sampling directly from π is intractable. In this section, we will describe two SG-MCMC algorithms that generate approximate samples from the target distribution π.

3.1. Stochastic Gradient Langevin Dynamics

In the last decade, the Stochastic Gradient Descent (SGD) algorithm [18] has become very popular due to its low computational requirements and convergence guarantee. In [19], Welling and Teh proposed a scalable MCMC framework called the Stochastic Gradient Langevin Dynamics (SGLD), that brings the ideas of SGD and Langevin Monte Carlo [20] together in order to generate samples from the posterior distribution in a computationally efficient way. In an algorithmic sense, SGLD is identical to SGD except that it injects a Gaussian noise at each iteration. For NMF models, SGLD iteratively applies the following update rules in order to obtain the samples $\mathbf{W}^{(t)}$ and $\mathbf{H}^{(t)}$: $\mathbf{W}^{(t)} = \mathbf{W}^{(t-1)} + \Delta\mathbf{W}^{(t)}$ and $\mathbf{H}^{(t)} = \mathbf{H}^{(t-1)} + \Delta\mathbf{H}^{(t)}$, where

$$\Delta\mathbf{W}^{(t)} = \epsilon\Big(\frac{N}{|\Omega^{(t)}|} \sum_{(i,j)\in\Omega^{(t)}} \nabla_{\mathbf{W}} \log p(v_{ij}|\mathbf{W}^{(t-1)}, \mathbf{H}^{(t-1)}) + \nabla_{\mathbf{W}} \log p(\mathbf{W}^{(t-1)})\Big) + \mathbf{\Psi}^{(t)}, \tag{7}$$

$$\Delta\mathbf{H}^{(t)} = \epsilon\Big(\frac{N}{|\Omega^{(t)}|} \sum_{(i,j)\in\Omega^{(t)}} \nabla_{\mathbf{H}} \log p(v_{ij}|\mathbf{W}^{(t-1)}, \mathbf{H}^{(t-1)}) + \nabla_{\mathbf{H}} \log p(\mathbf{H}^{(t-1)})\Big) + \mathbf{\Xi}^{(t)}, \tag{8}$$

for all $t \in 1, \ldots, T$, T is the number of iterations. Here, ϵ is the step size, N is the number of elements in $\mathbf{V}$, $\Omega^{(t)} \subset [I] \times [J]$ is the sub-sample that is drawn at iteration t, the set $[I]$ is defined as $[I] = \{1, \ldots, I\}$, ∇ denotes the gradients, and $|\Omega^{(t)}|$ denotes the number of elements in $\Omega^{(t)}$. The elements of the noise matrices $\mathbf{\Psi}^{(t)}$ and $\mathbf{\Xi}^{(t)}$ are independently Gaussian distributed:

$$\psi_{ik}^{(t)} \sim \mathcal{N}(0, 2\epsilon), \qquad \xi_{kj}^{(t)} \sim \mathcal{N}(0, 2\epsilon).$$

The convergence properties of SGLD has been studied in [21, 22]. It has been shown that under certain assumptions and with sufficiently a large number of iterations, the bias $|\mathbb{E}[\hat{f} - f]|$ and the mean-squared-error (MSE) $\mathbb{E}[(\hat{f} - f)^2]$ of SGLD can be bounded as $\mathcal{O}(\epsilon)$ and $\mathcal{O}(\epsilon^2)$, respectively [23]. Several extensions of SGLD have been proposed [12, 23–31].

3.2. Stochastic Gradient Richardson-Romberg Langevin Dynamics

Even though SGLD has proved useful in several applications, its performance is often limited by its bias. In a recent study, [12] aimed at addressing this issue and proposed a new SG-MCMC algorithm, referred to as Stochastic Gradient Richardson-Romberg Langevin Dynamics (SGRRLD), whose asymptotic bias and MSE can be bounded as $\mathcal{O}(\epsilon^2)$ and $\mathcal{O}(\epsilon^4)$, respectively.

The SGRRLD algorithm is based on a numerical sequence acceleration method, called the Richardson-Romberg extrapolation, which simply boils down to running two SGLD chains in parallel with different step sizes. For the first chain, we use a step size ϵ and for the second chain, we use $\epsilon/2$ as the step size. These two chains are started from the same initial points and are run accordingly to Eqs. 7-8, except that the chain with the smaller step size is run twice more often than the other one. To be more precise, in the first chain we have the following update equation for $\mathbf{W}$:

$$\Delta\mathbf{W}^{(t,1)} = \epsilon\Big(\frac{N}{|\Omega^{(t,1)}|} \sum_{(i,j)\in\Omega^{(t,1)}} \nabla_{\mathbf{W}} \log p(v_{ij}|\mathbf{W}^{(t-1,1)}, \mathbf{H}^{(t-1,1)}) + \nabla_{\mathbf{W}} \log p(\mathbf{W}^{(t-1,1)})\Big) + \mathbf{\Psi}^{(t,1)}, \ \forall t \in [T] \tag{9}$$

and in the second chain we have the following update equation:

$$\Delta\mathbf{W}^{(t,2)} = \frac{\epsilon}{2}\Big(\frac{N}{|\Omega^{(t,2)}|} \sum_{(i,j)\in\Omega^{(t,2)}} \nabla_{\mathbf{W}} \log p(v_{ij}|\mathbf{W}^{(t-1,2)}, \mathbf{H}^{(t-1,2)}) + \nabla_{\mathbf{W}} \log p(\mathbf{W}^{(t-1,2)})\Big) + \mathbf{\Psi}^{(t,2)}, \ \forall t \in [2T]. \tag{10}$$

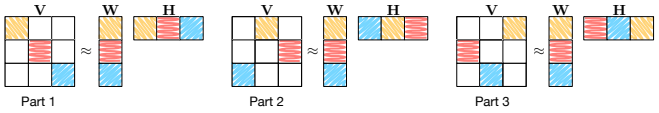

Fig. 1. Illustration of the *parts* and the *blocks*. Given the blocks in a part, the corresponding blocks in $\mathbf{W}$ and $\mathbf{H}$ become conditionally independent, as illustrated in different textures.

Then, for estimating the posterior expectation of a test function f (see Eq. 6), we apply a Richardson-Romberg extrapolation to the estimates that are obtained from the two chains:

$$\hat{f} = \frac{1}{T}\sum_{t=1}^{2T} f(\mathbf{\Theta}^{(t,2)}) - \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{\Theta}^{(t,1)}), \tag{11}$$

where $\mathbf{\Theta}^{(t,1)} \equiv \{\mathbf{W}^{(t,1)}, \mathbf{H}^{(t,1)}\}$ and $\mathbf{\Theta}^{(t,2)} \equiv \{\mathbf{W}^{(t,2)}, \mathbf{H}^{(t,2)}\}$, respectively. For the Richardson-Roomberg extrapolation to be effective, the random subsets $\Omega^{(t,1)}$ and $\Omega^{(t,2)}$ should have identical distributions, implying that $|\Omega^{(t,1)}| = |\Omega^{(t',2)}|$ for all $t \in [T], t' \in [2T]$. The update rules of $\mathbf{H}$ have a similar form. Besides, to minimize the variance of the estimator $\hat{f}$ defined by Eq. 11, it has been shown in [12] that the crucial part in this algorithm is that the injected Gaussian noises should be perfectly correlated. Formally, each element of $\mathbf{\Psi}^{(t,2)}$ is distributed as $\psi_{ik}^{(t,2)} \sim \mathcal{N}(0, \epsilon)$, whereas each element of $\mathbf{\Psi}^{(t,1)}$ is obtained as follows: $\psi_{ik}^{(t,1)} = \psi_{ik}^{(2t-1,2)} + \psi_{ik}^{(2t,2)}$, instead of being independently drawn.

An important property of SGRRLD is that the two SGLD chains can be run in an completely parallel fashion by using the same seed in the pseudo random number generators.

4. PARALLEL SG-MCMC ALGORITHMS FOR NMF

In the SGLD updates given in Eqs. 7-8, the sub-sample $\Omega^{(t)}$ can be drawn with or without replacement [21]. However, since we are dealing with NMF models, instead of sub-sampling the data arbitrarily, one might come up with more clever sub-sampling schemas that could reduce the computational burden drastically by enabling parallelism. In this section, using the conditional independence structure of NMF models, we will develop two parallel SG-MCMC algorithms for NMF, namely P-SGLD and P-SGRRLD, where 'P' stands for 'parallel'.

Inspired by [6–8], we utilize a stratified sub-sampling schema where the observed data is carefully partitioned into mutually disjoint blocks and the latent factors are also partitioned accordingly. An illustration of this approach is depicted in Fig. 1. In this particular example, the observed matrix $\mathbf{V}$ is partitioned into 3×3 disjoint blocks and the latent factors $\mathbf{W}$ and $\mathbf{H}$ are partitioned accordingly into 3×1 and 1×3 blocks. At each iteration, P-SGLD and P-SGRRLD sub-sample 3 blocks from $\mathbf{V}$, called the *parts*, in such a way that the blocks in a part will not 'touch' each other in any dimension of $\mathbf{V}$, as illustrated in Fig. 1. This sub-sampling schema enables parallelism, since given a part, the SGLD updates can be applied to different blocks of the latent factors in parallel. In the general case, the observed matrix $\mathbf{V}$ will be partitioned into $B \times B = B^2$ blocks and these blocks can be formed in a data-dependent manner, instead of using simple grids.

Let us formally define a *block* and a *part*. First, we need to define a partition of a set $\mathcal{S}$. Let $\{0,1\}^{\mathcal{S}}$ be the power set of $\mathcal{S}$ and let $B \geq 1$. $\mathcal{I} = (\mathcal{I}_b)_{b\in[B]} \subset \{0,1\}^{\mathcal{S}}$ is a partition of $\mathcal{S}$ if it is a family of non-empty disjoint subsets of $\mathcal{S}$, whose union is equal to $\mathcal{S}$. Denote by $\mathcal{P}_B(\mathcal{S})$ the set of all the partitions of $\mathcal{S}$ of size B.

Definition 1 *Let $B \geq 1$. A part Π of size B is a subset of $\{0,1\}^{[I]\times[J]}$ of the form $\Pi = (\mathcal{I}_b \times \mathcal{J}_b)_{b\in[B]}$ where $(\mathcal{I}_b)_{b\in[B]}$ and $(\mathcal{J}_b)_{b\in[B]}$ are partitions of $[I]$ and $[J]$ respectively. For all $b \in [B]$, the subset $\mathcal{I}_b \times \mathcal{J}_b$ of $[I] \times [J]$ is said to be a block associated with the part Π.*

Suppose we observe a part $\Pi^{(t)} = (\mathcal{I}_b^{(t)} \times \mathcal{J}_b^{(t)})_{b\in[B]}$ at iteration t. Then the SGLD updates for $\mathbf{W}$ can be written as follows:

$$\Delta\mathbf{W}^{(t)} = \epsilon\Big(\frac{N}{|\Pi^{(t)}|}\sum_{b=1}^{B}\sum_{(i,j)\in\mathcal{I}_b^{(t)}\times\mathcal{J}_b^{(t)}} \nabla_{\mathbf{W}} \log p(v_{ij}|\mathbf{W}^{(t-1)}, \mathbf{H}^{(t-1)}) + \nabla_{\mathbf{W}} \log p(\mathbf{W}^{(t-1)})\Big) + \mathbf{\Psi}^{(t)} \tag{12}$$

Since by definition the familly of sets $(\mathcal{I}_b^{(t)} \times \mathcal{J}_b^{(t)})_{b\in[B]}$ are mutually disjoint, we can decompose Eq. 12 into B *interchangeable* updates (i.e., they can be applied in any order), that are given as follows: $\mathbf{W}_b^{(t)} = \mathbf{W}_b^{(t-1)} + \Delta\mathbf{W}_b^{(t)}$, where

$$\Delta\mathbf{W}_b^{(t)} = \epsilon^{(t)}\Big(\frac{N}{|\Pi^{(t)}|}\sum_{(i,j)\in\mathcal{I}_b^{(t)}\times\mathcal{J}_b^{(t)}} \nabla_{\mathbf{W}_b} \log p(v_{ij}|\mathbf{W}_b^{(t-1)}, \mathbf{H}_b^{(t-1)}) + \nabla_{\mathbf{W}_b} \log p(\mathbf{W}_b^{(t-1)})\Big) + \mathbf{\Psi}_b^{(t)} \tag{13}$$

for all $b = 1, \ldots, B$. Here, $\mathbf{W}_b^{(t)}$ and $\mathbf{H}_b^{(t)}$ are the latent factor blocks at iteration t, that are determined by the current data block $\mathcal{I}_b^{(t)} \times \mathcal{J}_b^{(t)}$ and are formally defined as follows: $\mathbf{W}_b^{(t)} \equiv \{w_{ik}^{(t)} | i \in \mathcal{I}_b^{(t)}, k \in [K]\}$ and $\mathbf{H}_b^{(t)} \equiv \{h_{kj}^{(t)} | j \in \mathcal{J}_b^{(t)}, k \in [K]\}$. The noise matrix $\mathbf{\Psi}_b^{(t)}$ is of the same size as $\mathbf{W}_b$ and its entries are independently Gaussian distributed with mean 0 and variance 2ϵ. The parallelism comes from the fact that these update equations can be applied in parallel. By following a similar approach, we obtain B interchangeable update rules for $\mathbf{H}$. A more detailed explanation of P-SGLD can be found in the technical report by Şimşekli et al. [32].

Finally, we apply the stratified sub-sampling approach (Eq. 13) to the two SGLD chains in SGRRLD (Eqs. 9-10) and obtain our new algorithm P-SGRRLD. Note that the two chains in SGRRLD can already be run in parallel whereas in P-SGRRLD we further increase this parallelism and run the update equations of each chain in parallel, as in P-SGLD. We also note that P-SGRRLD requires a careful implementation since the Gaussian noises should be correlated.

Handling non-negativity: In an optimization framework, the latent factors can be kept in a constraint set by using projections that apply the minimum force to keep the variables in the constraint set. However, since we are in an MCMC framework, it is not clear that appending a projection step to the P-SGLD and P-SGRRLD updates would still result in a proper MCMC method. Instead, similar to [25, 33], we make use of a simple mirroring trick, where we replace the negative entries of $\mathbf{W}^{(t)}$ and $\mathbf{H}^{(t)}$ with their absolute values. Intuitively, we let w_{ik} and h_{kj} take values in the whole $\mathbb{R}$, however we parametrize the prior and the observation models with the absolute values, $|w_{ik}|$ and $|h_{kj}|$. Since $w_{ik}^{(t)}$ and $-w_{ik}^{(t)}$ (similarly, $h_{kj}^{(t)}$ and $-h_{kj}^{(t)}$) will be equiprobable in this setting, we can replace the negative elements of $\mathbf{W}^{(t)}$ and $\mathbf{H}^{(t)}$ with their absolute values.

5. EXPERIMENTS

In this section, we evaluate the proposed algorithms on an image restoration task. In our experiments we use the AT&T Database of Faces [34]. This dataset contains face images from 40 distinct subjects, where there are 10 images for each subject. In total there are 400 images in the dataset, where the size of each image is 92×112 pixels, with 256 gray levels per pixel. The database can be downloaded from `http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`.

We conduct all the experiments on a standard desktop computer with 3.50GHz Intel Xeon CPU with 8 cores, 32 GB of memory. We have implemented both algorithms on the CPU in C, where we have used the GNU Scientific Library and BLAS for the matrix operations (`https://www.gnu.org/software/gsl`) and OpenMP (`http://openmp.org`) for parallel computations. The source codes for P-SGLD and P-SGRRLD can be obtained from `http://perso.telecom-paristech.fr/~simsekli/nmf_sgmcmc/`.

In our experiments, we vectorize all the images in the dataset and concatenate these vectors in order to represent the dataset as a matrix. Finally, we obtain an observed matrix $\mathbf{V}$ of dimensions $I = 92 \times 112$ and $J = 400$. In each experiment, we randomly erase some entries of the data matrix $\mathbf{V}$, which will be reconstructed later on. We evaluate and compare the performance of P-SGLD and P-SGRRLD by measuring the root-normalized-mean-squared error (RNMSE) [35] between the true and the reconstructed data, given as follows:

$$\text{RNMSE}(\mathbf{V}||\hat{\mathbf{V}}) = \sqrt{\frac{\sum_{ij}(v_{ij}-\hat{v}_{ij})^2}{\sum_{ij} v_{ij}^2}}, \tag{14}$$

where $\hat{\mathbf{V}}$ denotes the restored matrix that is obtained via P-SGLD or P-SGRRLD, defined as follows:

$$\hat{v}_{ij} = \begin{cases} v_{ij}, & \text{if } v_{ij} \text{ is observed,} \\ \frac{1}{T}\sum_{t=1}^{T}\sum_k w_{ik}^{(t)} h_{kj}^{(t)}, & \text{if } v_{ij} \text{ is missing.} \end{cases} \tag{15}$$

In all the experiments, we set the latent dimension $K = 100$, we partition the sets $[I]$ and $[J]$ into $B = 8$ pieces, and choose a random part at each iteration. For P-SGLD we launch 8 parallel threads, whereas for P-SGRRLD we launch 16 parallel threads since we run two parallel chains in P-SGRRLD. We generate $T = 1000$ samples with P-SGLD where we discard the first 500 samples as burn-in and we set $T = 500$ for P-SGRRLD where we discard the first half of the samples, in order to keep the computational needs comparable since the second chain in P-SGRRLD requires $2T$ iterations. For P-SGLD, we tried several values for the parameters, chose the best performing ones, and we used the same parameters for P-SGRRLD. We evaluate our algorithms under different missing data percentages where we repeat each experiment 5 times for each missing data percentage and report the average results.

In our first experiment, we consider the Poisson-NMF model [1, 36], where we use the $\mathcal{TW}(v; \mu, \phi = 1, \beta = 1)$ observation model. We set $\epsilon = 10^{-5}$ and $\lambda = 1/5000$. Fig. 2 shows the performance of our algorithms on this problem. The results show that both of our methods yield a significant improvement in the RNMSE. The methods perform similarly when the percentage of the missing data is low, whereas the performance gap between P-SGRRLD and P-SGLD increases along with the increasing missing data percentage. The main advantage of our methods is their computational efficiency: P-SGLD finishes its computations nearly in 85 seconds,

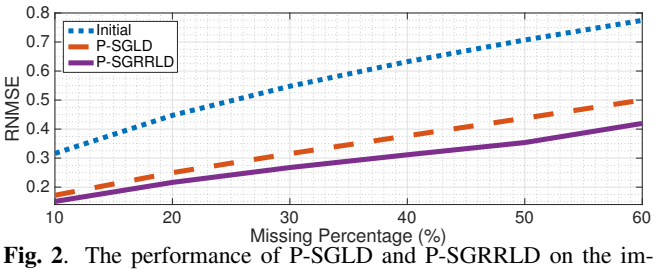

Fig. 2. The performance of P-SGLD and P-SGRRLD on the image restoration problem under the Poisson-NMF model. The initial RNMSE is computed by substituting 0 for the missing values.

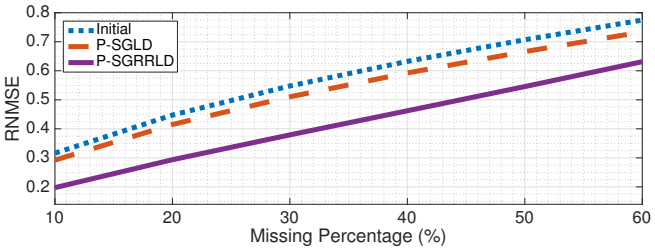

Fig. 3. The performance of P-SGLD and P-SGRRLD on the image restoration problem under the Compound Poisson-NMF model.

whereas this duration is 95 seconds for P-SGRRLD when the percentage of the missing values is 10%. We note that even though a Gibbs sampler (blocked [36] or collapsed [37]) can be developed for this model, it would not be comparable to our algorithms due to its expensive computational requirements.

In our second experiment, we evaluate our algorithms on $\mathcal{TW}(v; \mu, \phi = 1, \beta = 0.5)$ observation model that corresponds to a compound Poisson distribution [13, 14], for which deriving a Gibbs sampler is not straightforward. This distribution is particularly suited to sparse data as it has a non-zero probability mass on $v = 0$ and a continuous density on $v > 0$ [13]. Even though the probability density function of this distribution cannot be written in closed-form analytical expression, we can still generate samples from the posterior distribution by using our methods since we do not need to evaluate the normalizing constant in Eq. 4.

In this experiment, we set $\epsilon = 5 \times 10^{-4}$ and $\lambda = 1/5000$. Fig. 3 visualizes the performance of our algorithms on this model. We obtain qualitatively similar results and P-SGRRLD is clearly more advantageous than P-SGLD. On the other hand, the performance difference between the two methods turns out to be more prominent, while the predictions of both methods are accurate, gracefully degrading from 10% to 60% of missing data.

6. CONCLUSION

In this study, we presented two parallel SG-MCMC algorithms for a general class of NMF models, namely P-SGLD and P-SGRRLD. We built P-SGLD upon the framework that was proposed in [10], whereas P-SGRRLD is a parallel variant of a recently proposed SG-MCMC algorithm [12] that achieves faster convergence rates. Both of the proposed algorithms are inherently parallel and present lower computational complexity compared to conventional MCMC methods. We illustrated our algorithms on an image restoration task where we showed that our algorithms yield accurate results in less than 2 minutes on this particular problem, whereas the conventional approaches would be infeasible due to high computational needs.

7. REFERENCES

[1] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, 1999.

[2] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *WASPAA*, 2003, pp. 177–180.

[3] K. Devarajan, "Nonnegative matrix factorization: An analytical and interpretive tool in computational biology," *PLoS Computational Biology*, vol. 4, 2008.

[4] A. Cichoki, R. Zdunek, A.H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorization*, Wiley, 2009.

[5] C. Févotte, N. Bertin, and J. L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence. with application to music analysis," *Neural Computation*, vol. 21, pp. 793–830, 2009.

[6] C. Liu, H.C. Yang, J. Fan, L.W. He, and Y.M. Wang, "Distributed nonnegative matrix factorization for web-scale dyadic data analysis on MapReduce," in *WWW*, 2010.

[7] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis, "Large-scale matrix factorization with distributed stochastic gradient descent," in *ACM SIGKDD*, 2011.

[8] B. Recht and C. Ré, "Parallel stochastic gradient algorithms for large-scale matrix completion," *Mathematical Programming Computation*, 2013.

[9] U. Şimşekli, R. Badeau, G. Richard, and A. T. Cemgil, "Stochastic thermodynamic integration: efficient Bayesian model selection via stochastic gradient MCMC," in *ICASSP*, 2016.

[10] S. Ahn, A. Korattikara, N. Liu, S. Rajan, and M. Welling, "Large-scale distributed Bayesian matrix factorization using stochastic gradient MCMC," in *KDD*, 2015.

[11] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," in *ICML*, 2008, pp. 880–887.

[12] A. Durmus, U. Şimşekli, E. Moulines, R. Badeau, and G. Richard, "Stochastic gradient Richardson-Romberg Markov chain Monte Carlo," in *NIPS*, 2016.

[13] B. Jørgensen, *The Theory of Dispersion Models*, Chapman & Hall/CRC Monographs on Statistics & Applied Probability, 1997.

[14] U. Şimşekli, A. T. Cemgil, and Y. K Yilmaz, "Learning the beta-divergence in Tweedie compound Poisson matrix factorization models.," in *ICML*, 2013, pp. 1409–1417.

[15] O. Dikmen, Z. Yang, and E. Oja, "Learning the information divergence," *IEEE TPAMI*, vol. 37, no. 7, pp. 1442–1454, 2015.

[16] U. Şimşekli, A. T. Cemgil, and B. Ermiş, "Learning mixed divergences in coupled matrix and tensor factorization models," in *ICASSP*, 2015, pp. 2120–2124.

[17] U. Şimşekli, B. Ermiş, A. T. Cemgil, and E. Acar, "Optimal weight learning for coupled tensor factorization with mixed divergences," in *EUSIPCO*, 2013, pp. 1–5.

[18] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 1951.

[19] M. Welling and Y. W Teh, "Bayesian learning via Stochastic Gradient Langevin Dynamics," in *ICML*, 2011, pp. 681–688.

[20] G. O. Roberts and R. L. Tweedie, "Exponential convergence of Langevin distributions and their discrete approximations," *Bernoulli*, vol. 2, no. 4, pp. 341–363, 1996.

[21] I. Sato and H. Nakagawa, "Approximation analysis of stochastic gradient Langevin dynamics by using Fokker-Planck equation and Ito process," in *ICML*, 2014, pp. 982–990.

[22] Y. W. Teh, A. H. Thiéry, and S. J. Vollmer, "Consistency and fluctuations for stochastic gradient Langevin dynamics," *Journal of Machine Learning Research*, vol. 17, no. 7, pp. 1–33, 2016.

[23] C. Chen, N. Ding, and L. Carin, "On the convergence of stochastic gradient MCMC algorithms with high-order integrators," in *NIPS*, 2015, pp. 2269–2277.

[24] S. Ahn, A. Korattikara, and M. Welling, "Bayesian posterior sampling via stochastic gradient Fisher scoring," in *ICML*, 2012.

[25] S. Patterson and Y. W. Teh, "Stochastic gradient Riemannian Langevin dynamics on the probability simplex," in *NIPS*, 2013.

[26] T. Chen, E. B. Fox, and C. Guestrin, "Stochastic gradient Hamiltonian Monte Carlo," in *ICML*, 2014.

[27] N. Ding, Y. Fang, R. Babbush, C. Chen, R. D. Skeel, and H. Neven, "Bayesian sampling using stochastic gradient thermostats," in *NIPS*, 2014, pp. 3203–3211.

[28] X. Shang, Z. Zhu, B. Leimkuhler, and A. J. Storkey, "Covariance-controlled adaptive Langevin thermostat for large-scale Bayesian sampling," in *NIPS*, 2015, pp. 37–45.

[29] Y. A. Ma, T. Chen, and E. Fox, "A complete recipe for stochastic gradient MCMC," in *NIPS*, 2015, pp. 2899–2907.

[30] C. Li, C. Chen, D. Carlson, and L. Carin, "Preconditioned stochastic gradient Langevin dynamics for deep neural networks," in *AAAI Conference on Artificial Intelligence*, 2016.

[31] U. Şimşekli, R. Badeau, A. T. Cemgil, and G. Richard, "Stochastic quasi-Newton Langevin Monte Carlo," in *ICML*, 2016.

[32] U. Şimşekli, H. Koptagel, H. Güldaş, A. T. Cemgil, F. Öztoprak, and Ş. İ. Birbil, "Parallel stochastic gradient Markov chain Monte Carlo for matrix factorisation models," *arXiv preprint arXiv:1506.01418*, 2015.

[33] R. M. Neal, "MCMC using Hamiltonian dynamics," *Handbook of Markov Chain Monte Carlo*, vol. 54, 2010.

[34] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *WACV*, 1994, pp. 138–142.

[35] J. R. Fienup, "Invariant error metrics for image reconstruction," *Applied optics*, vol. 36, no. 32, pp. 8352–8357, 1997.

[36] A. T. Cemgil, "Bayesian inference in non-negative matrix factorisation models," *Computational Intelligence and Neuroscience*, 2009.

[37] C. Fevotte, O. Cappe, and A. T. Cemgil, "Efficient Markov chain Monte Carlo inference in composite models with space alternating data augmentation," in *SSP*, 2011.