Communications in Information and Systems
Vol. 12, No. 3, pp. 221-232, 2012

003

HYBRID DETERMINISTIC-STOCHASTIC GRADIENT LANGEVIN DYNAMICS FOR BAYESIAN LEARNING

QI HE* AND JACK XIN†

Abstract. We propose a new algorithm to obtain Bayesian posterior distribution by a hybrid deterministic-stochastic gradient Langevin dynamics. To speed up convergence and reduce computational costs, it is common to use stochastic gradient method to approximate the full gradient by sampling a subset of the large dataset. Stochastic gradient methods make progress fast initially, however, they often become slow in the late stage as the iterations approach the desired solution. The conventional gradient methods converge better eventually however at the expense of evaluating the full gradient at each iteration. Our hybrid method has the advantages of both approaches for constructing the Bayesian posterior distribution. We prove that our algorithm converges based on the weak convergence methods, and illustrate numerically its effectiveness and improved accuracy.

Key words: Langevin dynamics, Stochastic gradient, Bayesian learning.
AMS Subject Classification. 60J27, 93E20, 34E05.

1. Introduction.

This work focuses on Bayesian learning based on a hybrid deterministic-stochastic gradient descent Langevin dynamics. There has been increasing interest in large scale datasets for machine learning, ranging from network data, signal processing and data mining to bioinformatics. The large scale data significantly increase the computational complexity of the underlying optimization algorithm. One of the successful methods to overcome this difficulty is stochastic gradient method, which is efficient and simple to implement.

In the literature, stochastic gradient methods have been widely used for large scale machine learning. For example, [1, section 3.2] and [2] studied incremental-gradient method in which each iteration only evaluates the gradient along one single index; [3] applied stochastic gradient algorithm to large scale linear prediction problems; [4] explored hybrid deterministic-stochastic methods for large scale optimization. For more applications of stochastic gradient methods in large scale machine learning, we refer to [5] and references therein. The convergence of stochastic gradient algorithm has also been extensively studied in stochastic approximation literature, such as[2, 6], where convergence holds under some mild conditions including the case where the loss function is not everywhere differentiable; see also [7] for the convergence of a simultaneous perturbation gradient approximation and [8] for the convergence of stochastic gradient expectation maximization (EM) algorithm.

Along another line, there are few studies for large scale Bayesian learning by stochastic gradient method. [9] seems to be the first to propose stochastic gradient method for Bayesian learning by Langevin dynamics. Later, [10] improved the method

*Department of Mathematics, University of California, Irvine 92697, qhe2@uci.edu.
†Department of Mathematics, University of California, Irvine 92697, jxin@math.uci.edu

by adding preconditioner using stochastic Fisher scoring. Although the algorithm works efficiently in numerical experiments, there is no theoretical analysis for the convergence of the stochastic gradient Bayesian learning yet. To bridge this gap, in this paper, we study Bayesian learning by hybrid deterministic-stochastic gradient method motivated by [4] and provide a convergence proof. It covers the model in [9] as a special case and so gives the proof of the convergence for algorithm in [9]. This is the first work, to the best of our knowledge, that presents the theoretical analysis of the algorithm for stochastic gradient Bayesian learning.

It is worth mentioning that the convergence analysis of the hybrid deterministic-stochastic method for data fitting in [4] does not apply directly to the Bayesian learning problem here. The algorithm for Bayesian learning converges to a probability distribution rather than to a deterministic number. The traditional error analysis is not suitable for Bayesian learning. Instead we employ weak convergence method [11] to treat both data fitting and Bayesian learning algorithms. The weak convergence method also requires weaker conditions for convergence.

The rest of the paper is organized as follows. Section 2 begins with certain preliminary results. Section 3 provides the proof of convergence for the algorithm. Section 4 shows advantages of our proposed method by numerical simulations. Finally, concluding remarks are given in Section 5.

The work was partially supported by NSF grants DMS-1222507, and DMS-1211179. The authors thank Mr. Sungjin Ahn for helpful conversations on Bayesian learning and related data sets.

2. Preliminary Results. The Markov chain Monte Carlo (MCMC) method is a very popular tool for computational problems in Bayesian statistics, see [12] and [13]. To sample from the target density, MCMC methods construct a Markov chain whose stationary distribution is the target density. Under some suitable ergodic condition of the Markov chain, statistical quantities based on the target density can be calculated by simulating the Markov chain and computing time averaged quantities.

A basic sampling method in MCMC is Langevin dynamics [13]. Assume that π is a target density on $\mathbb{R}^r$. Then the Langevin diffusion θ_t is defined by the stochastic differential equation

$$d\theta_t = \frac{1}{2}\nabla \log \pi(\theta_t)dt + dW_t, \tag{2.1}$$

where W_t is a standard Brownian motion. When π is suitably smooth, it can be shown that θ_t has π as a stationary distribution [14]. Denote the density function of the diffusion θ_t by $\rho(t,x)$, then $\lim_{t\to\infty}\rho(t,x) = \pi(x)$. A variety of MCMC algorithms are based on Langevin dynamics. For example, the simplest way is to discretize Langevin dynamics (2.1) by the Euler-Maruyama method [15]. In addition, the Metropolis step can be added to remove bias if the step size is such that bias is significant. There are also some variants of the method, for example, pre-conditioning the dynamic by a positive definite matrix A to obtain

$$d\theta_t = \frac{1}{2}A\nabla \log \pi(\theta_t)dt + A^{1/2}dW_t. \tag{2.2}$$

This dynamic also has π as its stationary distribution.

To apply Langevin dynamics of MCMC method to Bayesian learning, we consider the following model. Let θ denote a parameter vector, with $p(\theta)$ a prior distribution, and $p(x|\theta)$ is the conditional distribution density of data x given θ. The posterior distribution of parameter θ given a set of data $X = (x_i, 1 \le i \le N)$ is

$$p(\theta|X) \propto p(\theta)\Pi_i^N p(x_i|\theta).$$

Replacing π in (2.1) by $p(\theta|X)$, we have the following Langevin dynamic for θ

$$\begin{aligned} d\theta_t &= \frac{1}{2}\nabla \log p(\theta_t|X)dt + dW(t), \\ &= \frac{1}{2}\Big(\nabla \log p(\theta_t) + \sum_{i=1}^{N} \nabla \log p(x_i|\theta_t)\Big)dt + dW(t). \end{aligned} \tag{2.3}$$

Hence, we have that the limit distribution of θ_t is $p(\theta|X)$, [14].

In order to approach the posterior distribution by Langevin dynamics, we use Euler-Maruyama method for discretization [15]. The algorithm is as follows

$$\theta_{k+1} = \frac{\varepsilon_k}{2}\Big(\nabla \log p(\theta_k) + \sum_{i=1}^{N} \nabla \log p(x_i|\theta_k)\Big) + \sqrt{\varepsilon_k}\eta_k, \tag{2.4}$$

where η_k is an i.i.d random variable sequence with normal distribution, and step size ε_k satisfies $\sum_{k=1}^{\infty} \varepsilon_k = \infty$ and $\sum_{k=1}^{\infty} \varepsilon_k^2 < \infty$. These two conditions for ε_k are very common for stochastic approximation with decreasing step size, see [6], [11] and [16]. Typically, step size $\varepsilon_k = a(b+t)^{-\gamma}$ which decays algebraically with $\gamma \in (0.5, 1]$. To correct for discretization error, one can use (2.4) as a proposal distribution and modify it by the Metropolis-Hasting method [17]. Note that as we decrease the step size ε_k, the discretization error decreases to zero so that the rejection rate approaches zero. Hence, we can simply ignore the Metropolis-Hasting acceptance step.

In the algorithm (2.4), we need to evaluate the gradient of $\log(x|\theta)$ over all the data set, which is time consuming. To speed up convergence, [9] used the stochastic gradient method, i.e., at each step only estimate the gradient over a subset of the data. Assume that the size of the subset of the data is $n < N$, the algorithm is as follows

$$\theta_{k+1} = \frac{\varepsilon_k}{2}\Big(\nabla \log p(\theta_k) + \frac{N}{n}\sum_{i=1}^{n} \nabla \log p(x_i|\theta_k)\Big) + \sqrt{\varepsilon_k}\eta_k.$$

Although it is shown in [9] that this stochastic gradient Bayesian learning works well for some numerical simulations, the convergence improves slowly after a certain number of iterations. The reason for this is that the stochastic gradient method makes good progress initially, but it is slow in improving the accuracy as the iterates approach the limiting solution. To overcome this disadvantage, [4] proposed a hybrid deterministic-stochastic method for optimization, in which the size of the sampling subset is increasing after each iteration. It is shown that this method exhibits benefits of both the stochastic gradient method and full gradient method.

Motivated by [4], we propose a hybrid deterministic-stochastic gradient method for Bayesian learning. The new ingredient is to inject additional noise according to Langevin dynamics into the algorithm so that convergence to the full posterior distribution holds.

The hybrid deterministic-stochastic gradient Bayesian learning algorithm is given by

$$\theta_{k+1} = \frac{\varepsilon_k}{2}\Big(\nabla \log p(\theta_k) + \frac{N}{n_k}\sum_{i=1}^{n_k} \nabla \log p(x_i|\theta_k)\Big) + \sqrt{\varepsilon_k}\eta_k, \tag{2.5}$$

where n_k is the number of size of a sample subset at the k-iteration. It is nondecreasing and $\lim_{k\to\infty} n_k \le N$.

3. Weak convergence method. In this section, we prove that the distribution of the algorithm (2.5) converges to the posterior distribution $p(\theta|X)$. We introduce the following conditions.

Assumption A. Assume that $\nabla \log p(\theta)$ and $\nabla \log p(\theta|x)$ satisfy linear growth and Lipschitz conditions, and that $p(x)$ is continuously differentiable.

REMARK 3.1. *The linear growth and Lipschitz conditions in assumption A can be extended to local linear growth and local Lipschitz conditions by applying truncation techniques, for more details we refer to [6].*

Define $t_k = \sum_{i=0}^{k-1} \varepsilon_i, m(t) = \max\{k : t_k \le t\}$, and continuous-time interpolations

$$\begin{aligned} &\theta^0(t) = \theta_k, \quad \text{for} \quad t \in [t_k, t_{k+1}), \\ &\theta^k(t) = \theta^0(t + t_k). \end{aligned} \tag{3.1}$$

We first obtain an estimate on the pth moment of $\{\theta_k\}$. This is stated as follows.

LEMMA 3.2. *Under assumption A, for any fixed $p \ge 2$ and $T > 0$,*

$$\sup_{0\le k\le m(T)} E|\theta_k|^p \le (|\theta_0|^p + KT)\exp(KT) < \infty, \tag{3.2}$$

for some constant $K > 0$.

Proof. Define $U(\theta) = |\theta|^p$ and use E_k to denote the conditional expectation with respect to the σ-algebra $\mathcal{G}_k$, where $\mathcal{G}_k = \sigma(\theta_1, \dots, \theta_k)$. In the following, we use notation $'$ for the transpose operation. Thus

$$
\begin{aligned}
E_k U(\theta_{k+1}) - U(\theta_k) &= E_k \nabla U'(\theta_k)[\theta_{k+1} - \theta_k] \\
&\quad + E_k(\theta_{k+1} - \theta_k)' \nabla^2 U(\theta_k^+)(\theta_{k+1} - \theta_k) \\
&\le \varepsilon_k \nabla U'(\theta_k)\Big(\nabla \log p(\theta_k) + \frac{N}{n_k}\sum_{i=1}^{n_k} \nabla \log p(x_{ki}|\theta_k)\Big) \\
&\quad + K\varepsilon_k |\theta_k|^{p-2}(1 + |\theta_k|^2) \\
&\le K\varepsilon_k(1 + |\theta_k|^p),
\end{aligned} \tag{3.3}
$$

where ∇U and $\nabla^2 U$ denote the gradient and the Hessian of U w.r.t. to x, and θ_k^+ denotes a vector on the line segment joining θ_k and θ_{k+1}. Note that we have used the linear growth in θ for both $\nabla \log p(\theta)$ and $\nabla \log p(\theta|x)$ in the last line of (3.3). Since $U(\theta_k) = |\theta_k|^p$, we obtain $E_k|\theta_{k+1}|^p \le |\theta_k|^p + K\varepsilon_k + K\varepsilon_k|\theta_k|^p$. Taking the expectation on both sides and iterating on the resulting recursion, we have

$$
E|\theta_{n+1}|^p \le |\theta_0|^p + K\varepsilon_k n + K\varepsilon_k \sum_{k=0}^{n} E|\theta_k|^p.
$$

An application of the Gronwall's inequality yields $E|\theta_{n+1}|^p \le (|\theta_0|^p + KT)\exp(KT)$ as desired. □

To proceed, let us recall the definition of tightness. Assume that B is a metric space and that $\mathcal{B}$ is a σ algebra of subsets of B. A set of probability measures $\{P_\alpha\}$ on $(B, \mathcal{B})$ is tight if for each $\varepsilon > 0$ there is a compact set $B_\varepsilon \subset \mathcal{B}$ such that $\inf_\alpha P_\alpha\{B_\alpha\} \ge 1 - \varepsilon$. A set of random variable $\{x_\alpha\}$ is tight if its corresponding set of measures $\{P_\alpha\}$ is tight. For more details of tightness, we refer readers to the book [11].

In view of the estimate above, $\{\theta_k : 0 \le k \le m(T)\}$ is tight in $\mathbb{R}^r$ by applying the well-known Chebyshev inequality. That is, for each $\delta > 0$, there is a K_δ satisfying $K_\delta > \sqrt{(1/\delta)}$ such that

$$
P(|\theta_k| > K_\delta) \le \frac{\sup\limits_{0 \le n \le m(T)} E|\theta_k|^2}{K_\delta^2} \le K\delta.
$$

Next, we show that $\{\theta^k(\cdot)\}$ is tight in suitable function spaces.

LEMMA 3.3. *Under assumption A, $\{\theta^k(\cdot)\}$ is tight in $D^r[0, \infty)$, the space of functions that are right continuous and have left limits, endowed with the Skorohod topology.*

Proof. For any $\eta > 0$, $t \geq 0$, $0 \leq s \leq \eta$, we have

$$\begin{aligned}
&E|\theta^k(t+s) - \theta^k(t)|^2 \\
&= E\left|\frac{1}{2}\sum_{k=m(t)}^{m(t+s)-1} \varepsilon_k\Big(\nabla \log p(\theta_k) + \sum_{i=1}^{n_k} \frac{N}{n_k}\nabla \log p(\theta_k|x_{ki})\Big) + \sum_{k=m(t)}^{m(t+s)-1} \sqrt{\varepsilon}_k \eta_k\right|^2 \\
&\leq K \sum_{k=m(t)}^{m(t+s)-1} \varepsilon_k^2(1 + E|\theta_k|^2) + K \sum_{k=m(t)}^{m(t+s)-1} \varepsilon_k E|\eta_k|^2 \\
&\leq K \sum_{k=m(t)}^{m(t+s)-1} \varepsilon_k^2(1 + \sup_{m(t)\leq k \leq m(t+s)-1} E|\theta_k|^2) + K \sum_{k=m(t)}^{m(t+s)-1} \varepsilon_k \\
&\leq O\left(t+s-t\right) = O(s).
\end{aligned} \tag{3.4}$$

In the above, we have used Lemma 3.2 to ensure that $\sup_{m(t)\leq k \leq m(t+s)} E|\theta_k|^2 < \infty$ and the conditions of the stepsize $\sum_{i=1}^{\infty} \varepsilon_k^2 < \infty$ and $\sum_{i=1}^{\infty} \varepsilon_k = \infty$. Therefore, (3.4) leads to

$$\lim_{\eta\to 0} \limsup_{k\to\infty} E|\theta^k(t+s) - \theta^k(t)|^2 = 0.$$

The tightness of $\{\theta^k(\cdot)\}$ then follows from [11, p. 47]. □

3.1. Weak Convergence. Since $\{\theta^k(\cdot)\}$ is tight, by Prohorov's Theorem (see [6]), we may select a convergent subsequence. For simplicity, we still denote the subsequence by $\{\theta^k(\cdot)\}$ with limit denoted by $\{\widetilde{\theta}(\cdot)\}$.

THEOREM 3.4. *Under assumption A, the sequence $\{\theta^k(\cdot)\}$ converges weakly to $\theta(\cdot)$, which is the solution of Langevin dynamic given by* (2.3).

Proof. By Skorohod representation (see [6]), without loss of generality and without changing notation, we may assume that $\{\theta^k(\cdot)\}$ converges to $\widetilde{\theta}(\cdot)$ almost surely, and the convergence is uniform on each bounded interval. We proceed to characterize the limiting process.

Step 1: We show that the algorithm converges to the solution of (2.3). Define the operator

$$\mathcal{L}g(\theta) = \frac{1}{2}\Big\langle \nabla g(\theta), \nabla \log p(\theta) + \sum_{i=1}^{n} \nabla \log p(\theta|x_i)\Big\rangle + \frac{1}{2}tr[\nabla^2 g(\theta)], \tag{3.5}$$

for any suitable function g. For each $t > 0$ and $s > 0$, each positive integer κ, each $0 \leq t_\iota \leq t$ with $\iota \leq \kappa$, each bounded and continuous function $\rho_0(\cdot)$, and for each twice continuously differentiable function $h(\cdot)$ with compact support, we shall show that

$$E\rho(\widetilde{\theta}(t_\iota); \iota \leq \kappa)\Big[h(\widetilde{\theta}(t+s)) - h(\widetilde{\theta}_t) - \int_t^{t+s} \mathcal{L}h(\widetilde{\theta}(u))du\Big] = 0. \tag{3.6}$$

This yields that

$$h(\widetilde{\theta}_t) - \int_0^t \mathcal{L}h(\widetilde{\theta}(u))du \text{ is a continuous-time Martingale,}$$

which in turn implies that $\widetilde{\theta}(\cdot)$ is a solution of the Martingale problem with operator $\mathcal{L}$ defined in (3.5).

To establish the desired result, we work with the sequence $\theta^k(\cdot)$. By virtue of the weak convergence and the Skorohod representation, it is readily seen that

$$\begin{aligned} &E\rho(\theta^k(t_\iota);\iota\le\kappa)[h(\theta^k(t+s))-h(\theta^k(t))] \\ &\qquad\to E\rho(\widetilde{\theta}(t_\iota);\iota\le\kappa)\Big[h(\widetilde{\theta}(t+s))-h(\widetilde{\theta}(t))\Big] \quad \text{as } \varepsilon\to 0. \end{aligned} \tag{3.7}$$

On the other hand, given a small Δ, direct calculation shows that

$$\begin{aligned} &E\rho(\theta^k(t_\iota);\iota\le\kappa)\Big[h(\theta^k(t+s))-h(\theta^k(t))\Big] \\ &\quad= E\rho(\theta^k(t_\iota);\iota\le\kappa)\Big\{\sum_{l=0}^{s/\Delta-1}\big[h(\theta_{m(t+t_k+l\Delta+\Delta)})-h(\theta_{m(t+t_k+l\Delta)})\big]\Big\}. \end{aligned} \tag{3.8}$$

Step 2: For simplicity, we define

$$f(\theta)=\nabla\log p(\theta)+\sum_{i=1}^{N}\nabla\log p(\theta|x_i),\ \text{ and } f_k(\theta)=\nabla\log p(\theta)+\frac{N}{n_k}\sum_{i=1}^{n_k}\nabla\log p(\theta|x_{ki}).$$

For the terms on the last line of (3.8), we have

$$\begin{aligned} &\lim_{k\to\infty}E\rho(\theta^k(t_\iota);\iota\le\kappa)\sum_{l=0}^{s/\Delta-1}\big[h(\theta_{m(t+t_k+l\Delta+\Delta)})-h(\theta_{m(t+t_k+l\Delta)})\big] \\ &\quad=\lim_{k\to\infty}E\rho(\theta^k(t_\iota);\iota\le\kappa)\Big\{\sum_{l=0}^{s/\Delta-1}\Big[\nabla h'(\theta_{m(t+t_k+l\Delta)})\sum_{i=m(t+t_k+l\Delta)}^{m(t+t_k+l\Delta+\Delta)-1}\frac{\varepsilon_i}{2}f_i(\theta_i) \\ &\qquad\qquad+\sum_{i=m(t+t_k+l\Delta)}^{m(t+t_k+l\Delta+\Delta)-1}\frac{\varepsilon_i}{2}\mathrm{tr}[\nabla^2 h(\theta_{m(t+t_k+l\Delta)})]\Big]\Big\}. \end{aligned} \tag{3.9}$$

By the continuity of $f(\cdot)$ and the fact that $Ef(\theta)=Ef_k(\theta)$,

$$\begin{aligned} &\lim_{k\to\infty}E\rho(\theta^k(t_\iota);\iota\le\kappa)\Big\{\sum_{l=0}^{s/\Delta-1}\nabla h'(\theta_{m(t+t_k+l\Delta)}) \\ &\qquad\sum_{i=m(t+t_k+l\Delta)}^{m(t+t_k+l\Delta+\Delta)-1}\frac{\varepsilon_i}{2}\big[f(\theta_i)-f_i(\theta_{m(t+t_k+l\Delta)})\big]\Big\} \\ &=\lim_{k\to\infty}E\rho(\theta^k(t_\iota);\iota\le\kappa)\Big\{\sum_{l=0}^{s/\Delta-1}\nabla h'(\theta_{m(t+t_k+l\Delta)}) \\ &\qquad\sum_{i=m(t+t_k+l\Delta)}^{m(t+t_k+l\Delta+\Delta)-1}\frac{\varepsilon_i}{2}\big[f(\theta_i)-f_i(\theta_i)\big]\Big\} \\ &+\lim_{k\to\infty}E\rho(\theta^k(t_\iota);\iota\le\kappa)\Big\{\sum_{l=0}^{s/\Delta-1}\nabla h'(\theta_{m(t+t_k+l\Delta)}) \\ &\qquad\sum_{i=m(t+t_k+l\Delta)}^{m(t+t_k+l\Delta+\Delta)-1}\frac{\varepsilon_i}{2}\big[f_i(\theta_i)-f_i(\theta_{m(t+t_k+l\Delta)})\big]\Big\} \\ &=0. \end{aligned}$$

Thus, in evaluating the limit, $f_i(\theta_i)$ can be replaced by $f(\theta_{m(t+t_k+l\Delta)})$.

Consequently, by the weak convergence and the Skorohod representation,

$$\begin{aligned}
&\lim_{k\to\infty} E\rho(\theta^k(t_\iota);\iota\le\kappa)\Big\{\sum_{l=0}^{s/\Delta-1}\nabla h'(\theta_{m(t+t_k+l\Delta)}) \\
&\qquad\qquad\sum_{i=m(t+t_k+l\Delta)}^{m(t+t_k+l\Delta+\Delta)-1}\frac{\varepsilon_i}{2}f(\theta_{m(t+t_k+l\Delta)})\Big\} \\
&= \lim_{k\to\infty} E\rho(\theta^k(t_\iota);\iota\le\kappa)\Big\{\sum_{l=0}^{s/\Delta-1}\nabla h'(\theta_{m(t+t_k+l\Delta)})\frac{\Delta}{2}f(\theta_{m(t+t_k+l\Delta)})\Big\} \\
&= \lim_{k\to\infty} E\rho(\theta^k(t_\iota);\iota\le\kappa)\Big\{\sum_{l=0}^{s/\Delta-1}\frac{\Delta}{2}\nabla h'(\theta^k(t+l\Delta))f(\theta^k(t+l\Delta))\Big\} \\
&= E\rho(\widetilde{\theta}(t_\iota);\iota\le\kappa)\Big\{\int_t^{t+s}\nabla h'(\widetilde{\theta}(u))f(\widetilde{\theta}(u))du\Big\}.
\end{aligned} \tag{3.10}$$

In the above, treating such terms as $f(\theta^k(t+l\delta_\varepsilon))$, we can approximate $\theta^k(\cdot)$ by a process taking finitely many values using a standard approximation argument (see example, [6, p. 169] for more details).

Similar to (3.10), we also obtain

$$\begin{aligned}
&\lim_{k\to\infty} E\rho(\theta^k(t_\iota);\iota\le\kappa)\Big\{\sum_{l=0}^{s/\Delta-1}\sum_{i=m(t+t_k+l\Delta)}^{m(t+t_k+l\Delta+\Delta)-1}\frac{\varepsilon_i}{2}\mathrm{tr}\big[\nabla^2 h(\theta_{t+t_k+l\Delta})\big]\Big\} \\
&\quad = E\rho(\widetilde{\theta}(t_\iota);\iota\le\kappa)\Big\{\int_t^{t+s}\frac{1}{2}\mathrm{tr}\big[\nabla^2 h(\widetilde{\theta}(u))\big]du\Big\}.
\end{aligned} \tag{3.11}$$

Step 3: Combining Steps 1–2, we obtain that $\widetilde{\theta}(\cdot)$, the weak limit of $\theta^k(\cdot)$, is a solution of the Martingale problem with operator $\mathcal{L}$ defined in (3.5). Using characteristic functions, we show as in [18, Lemma 7.18], $\theta(\cdot)$ the solution of the Martingale problem with operator $\mathcal{L}$, is unique in the sense of distribution. Thus $\theta^k(\cdot)$ converges to $\theta(\cdot)$ as desired, which concludes the proof of the theorem. □

REMARK 3.5. *The proof can also be extended to the algorithm* (2.2) *including the preconditioner matrix, which is more common in practice.*

4. Numerical simulations.

In this section, we show the performance of the hybrid deterministic-stochastic gradient method for binary logistic regression and multinomial logistic regression models of data classification, in comparison with the stochastic gradient descent method.

4.1. Binary logistic regression.

We apply our hybrid deterministic-stochastic gradient method to Bayesian logistic regression model and compare with the result of stochastic gradient method. Logistic regression models [20] have been used widely in a vast number of applications for the problem of binary classification. Assume that we have data with N examples of input-output pairs (x_i, y_i), where $x_i \in \mathbb{R}^m$ is a vector of m features, and $y_i \in \{-1, 1\}$ is the binary outcome. The goal is to find a linear classifier that given the features x_i and a vector of parameters β, the sign of the inner product $\beta^T x_i$ gives y_i. The probability of the outcome given features x_i is

$$p_1(y_i|x_i) = \sigma(y_i \beta^T x_i),$$

where $\sigma(z) = \frac{1}{1+\exp(-z)}$. The bias parameter is absorbed into β by including 1 as an entry in x_i. We use standard normal distribution as the prior distribution for β. Then the gradient of the log likelihood is:

$$\frac{\partial}{\partial \beta} \log p_1(y_i|x_i) = \sigma(-y_i \beta^T x_i) y_i x_i,$$

and the gradient of the log of the prior is $\frac{\partial}{\partial \beta} \log p(\beta) = -\beta$. We applied our algorithm to the $a9a$ dataset used in [9] from UCI adult dataset. It consists of 32561 observations and 123 features. We used 80% of the data as the training data, i.e. $N = 26049$ and the rest 20% as the test data. For hybrid deterministic-stochastic method, we set the number of the size of the sampling subset as $n_k = \min(1.1 n_{k-1}, N), k = 1, 2, \cdots$ with $n_1 = 1$. While for the stochastic gradient method, we use the constant number of samples with $n = 4$. We run each algorithm 50 times and take the average. Figure 1 shows the average log joint probability per data item. It can be seen that stochastic gradient algorithm leads to large likelihood at the beginning, while the hybrid stochastic-deterministic gradient algorithm dominates the stochastic gradient when the sample size grows.

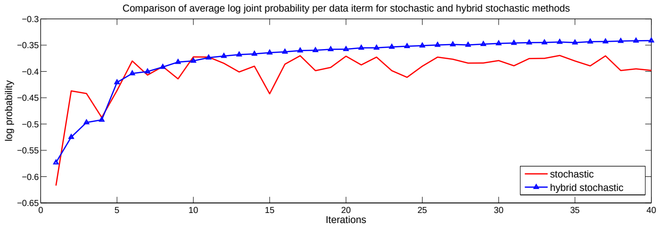

FIG. 1. *Comparison of the log probability of logistic regression for stochastic and hybrid deterministic-stochastic methods.*

Figure 2 shows the accuracy by comparing these two methods on the testing data set. The accuracy is in the sense of data percentages correctly predicted over the testing data set. The hybrid nature can be seen again: the stochastic gradient algorithm has more rapid initial progress, while the hybrid deterministic-stochastic gradient algorithm gains more accuracy when the sample size grows with the iteration.

4.2. Multinomial logistic regression. In this subsection, we apply both algorithms to a more general logistic regression model. Multinomial logistic regression extends the binary requirement to allow each outcome y_i to take any value from a set

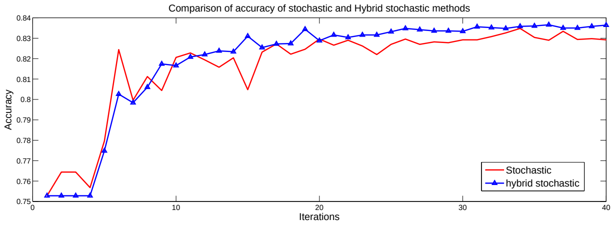

FIG. 2. *Comparison of the accuracy of logistic regression on test set for stochastic and hybrid deterministic-stochastic methods.*

of classes $C = \{1, 2, 3, \cdots, n\}$, [20]. In this model, we have a separate parameter β_j for each class $j \in C$. Given data $x_i \in \mathbb{R}^m$ with m features, we model the probability of the outcome by:

$$p_2(y_i = j|x_i, (\beta_j)_{j\in C}) = \frac{\exp{(\beta_j^T x_i)}}{\sum_{l\in C}\exp{(\beta_l^T x_l)}}.$$

The gradient of the log likelihood is:

$$\frac{\partial}{\partial \beta_j} p_2(y_i = j|x_i, (\beta_j)_{j\in C}) = \sigma(-\sum_{l\in C} y_i \beta_l I_{\{y_i=l\}} x_i) y_i x_i,$$

where $I_{\{\cdot\}}$ is the indicator function.

We run the experiments for multinomial logistic regression on the well-known MNIST data set [4], containing 70000 examples of 28×28 images of digits, where each digit is classified as one of the integers between 0 to 9. We used 60000 examples as the training data and the rest 10000 as the test data. We set the sample size $n = 10$ for stochastic gradient algorithm and $n_k = min(1.1n_k, N)$ for hybrid deterministic-stochastic gradient algorithm. Assume that prior distributions for parameter $\beta_j, j \in C$ are standard normal distributions. Then the gradient of the logarithm of the prior for β_j is $\frac{\partial}{\partial beta_j} \log p(\beta_j) = -\beta_j$. We run each algorithm 20 times and take the average. Figure 3 shows the accuracy of these two methods. As the results of the above experiments, the stochastic gradient algorithm is better initially. However, hybrid deterministic-stochastic gradient algorithm gains more accuracy as the sample size grows.

5. Remarks. Our work developed hybrid deterministic-stochastic gradient algorithm of Langevin dynamics to Bayesian learning. We further developed the hybrid deterministic-stochastic gradient optimization method [4] for Bayesian learning, and advanced the stochastic Bayesian learning method in [9]. A comprehensive weak convergence analysis of these algorithms is presented based on stochastic approximations method [6]. We showed by numerical simulations that our proposed method has both the advantages of the stochastic gradient Bayesian learning and the full gradient Bayesian learning. More efficient MCMC sampling methods based on hybrid

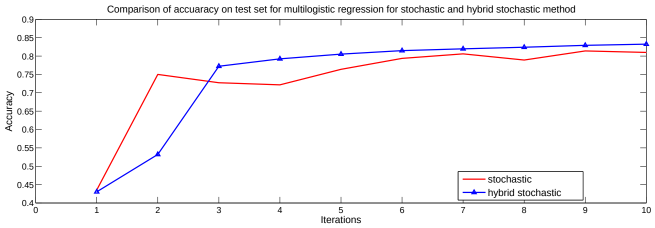

Fig. 3. *Comparison of the accuracy of multi-logistic regression on test dataset for stochastic and hybrid deterministic-stochastic methods.*

deterministic-stochastic gradients are interesting directions in the future, for example, the study of the more sophisticated Hamiltonian Monte Carlo approaches based on the hybrid deterministic-stochastic gradients.

REFERENCES

[1] D.P. Bertsekas and J.N. Tsitsiklis, *Neuro-dynamic programming: An overview*, in: Proceedings of the 34th IEEE Conference on Decision and Control, 1995, vol. 1, pp. 560–564.

[2] L. Bottou, *Online learning and stochastic approximations*, On-line learning in neural networks, 17(1998), pp. 9–42.

[3] T. Zhang, *Solving large scale linear prediction problems using stochastic gradient descent algorithms*, in: Proceedings of the 21th international conference on Machine learning(ICML), 2004, pp. 116–120.

[4] M.P. Friedlander and M. Schmidt, *Hybrid deterministic-stochastic methods for data fitting*, SIAM Journal on Scientific Computing, 34:3(2012), pp. 1380–1405.

[5] L. Bottou, *Large-scale machine learning with stochastic gradient descent*, in: Proceedings of COMPSTAT 2010, 2010 , pp. 177–186.

[6] H.J. Kushner and G. Yin, *Stochastic approximation and recursive algorithms and applications*, vol. 35, Springer, 2003.

[7] J.C. Spall, *Multivariate stochastic approximation using a simultaneous perturbation gradient approximation*, IEEE Transactions on Automatic Control, 37:3(1992), pp. 332–341.

[8] B. Delyon, M. Lavielle, and E. Moulines, *Convergence of a stochastic approximation version of the EM algorithm*, Annals of Statistics, 27(1999), pp. 94–128.

[9] M. Welling and Y.W. Teh, *Bayesian learning via stochastic gradient Langevin dynamics*, in: Proceedings of the 28th International Conference on Machine Learning (ICML), 2011, pp. 681–688.

[10] S. Ahn, A. Korattikara, and M. Welling, *Bayesian posterior sampling via stochastic gradient Fisher scoring*, in: Proceedings of the 29th International Conference on Machine Learning (ICML), 2012, pp. 1591–1598.

[11] H.J. Kushner, *Approximation and Weak Convergence Methods for Random Processes With Application to Stochastics Systems Theory*, vol. 6, MIT press, 1984.

[12] A.M. Stuart, J. Voss, and P. Wilberg, *Conditional path sampling of SDEs and the Langevin MCMC method*, Communications in Mathematical Sciences, 2:4(2004), pp. 685–697.

[13] C.P. Robert and G. Casella, *Monte Carlo statistical methods*, Springer, 2010.

[14] G.O. Roberts and R.L. Tweedie, *Exponential convergence of Langevin distributions and their discrete approximations*, Bernoulli, pp. 341–363, 1996.

[15] P.E. Kloeden and E. Platen, *Numerical solution of stochastic differential equations*, vol. 23,

Springer, 1992.

[16] H. Robbins and D. Siegmund, *A convergence theorem for non negative almost supermartingales and some applications*, in: Herbert Robbins Selected Papers, pp. 111–135. Springer, 1985.

[17] G.O. Roberts and O. Stramer, *Langevin diffusions and Metropolis-Hastings algorithms*, Methodology and computing in applied probability, 4:4(2002), pp. 337–357.

[18] G. Yin and Q. Zhang, *Continuous-time Markov chains and applications*, vol. 37, Springer, 2013.

[19] J. Nocedal and S.J. Wright, *Numerical optimization*, Springer, 2006.

[20] D.W. Hosmer and S. Lemeshow, *Applied logistic regression*, vol. 354, John Wiley & Sons, 2004.