AIMS Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 7(1): 1445–1459.
DOI: 10.3934/math.2022085
Received: 14 June 2021
Accepted: 17 October 2021
Published: 26 October 2021

Research article

On stochastic accelerated gradient with non-strongly convexity

Yiyuan Cheng[1], Yongquan Zhang[2,*], Xingxing Zha[1] and Dongyin Wang[1]

[1] School of Mathematics and Statistics, Chaohu University, 238024 Hefei, China
[2] School of Data Sciences, Zhejiang University of Finance & Economics, 310018 Hangzhou, China

* **Correspondence:** Email: zyqmath@163.com.

Abstract: In this paper, we consider stochastic approximation algorithms for least-square and logistic regression with no strong-convexity assumption on the convex loss functions. We develop two algorithms with varied step-size motivated by the accelerated gradient algorithm which is initiated for convex stochastic programming. We analyse the developed algorithms that achieve a rate of $O(1/n^2)$ where n is the number of samples, which is tighter than the best convergence rate $O(1/n)$ achieved so far on non-strongly-convex stochastic approximation with constant-step-size, for classic supervised learning problems. Our analysis is based on a non-asymptotic analysis of the empirical risk (in expectation) with less assumptions that existing analysis results. It does not require the finite-dimensionality assumption and the Lipschitz condition. We carry out controlled experiments on synthetic and some standard machine learning data sets. Empirical results justify our theoretical analysis and show a faster convergence rate than existing other methods.

Keywords: least-square regression; logistic regression; accelerated stochastic approximation; convergence rate
Mathematics Subject Classification: 68Q19, 68Q25, 68Q30

1. Introduction

In the era of 'big data', machine learning algorithms that only need to process each observation only once, or a few times, are desirable. Stochastic approximation algorithms such as stochastic gradient descent (SGD) and stochastic proximal gradient descent (SPGD), have been widely studied for this specific task and they have been successfully applied in various scenarios [2–16]. Regression and classification are effective analysis methods in machine learning, and have been successfully applied in practical problems. With the wide application of deep learning, these two methods are often used to train the parameters. In this paper, we consider stochastic approximation algorithms that consider minimizing a convex function where only the unbiased estimates of its gradients at the observations

are assumed available.

The convex function defined on a closed convex set in Euclidean space is usually given by $f(\theta) = \frac{1}{2}\mathbb{E}\left[\ell\left(y_i, \langle\theta, x_i\rangle\right)\right]$, where $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ denotes the sample data which is assumed to be i.i.d., ℓ denotes a loss function that is convex and $\mathbb{E}$ represents the expectation under the second variable. This loss function includes such as the least square and logistic regression. In the stochastic approximation framework, the samples appear sequentially according to an unknown probability measure ρ and the predictor defined by θ is updated after each pair is seen.

Robbins and Monro [1] were the first authors who proposed the stochastic approximation (SA) on the gradient descent method. From then on, algorithms based on SA have been widely used in stochastic optimisation and machine learning. Polyak [2] and Polyak and Juditsky [3] developed an important improvement of SA by using longer step-sizes with consequent averaging of the obtained iterates. The mirror-descent SA was developed by Nemirovski et al. [6] who showed that the mirror-descent SA exhibited an un-improvable expected rate for solving non-strongly convex programming problems. Shalev-Shwartz et al. [5] and Nemirovski et al. [6] studied an averaged stochastic gradient descent method for the least-square regression.

Theoretical studies on SGD for the least-square and logistic regression have shown that the convexity and smoothness of the loss function and the step-size policy play a critical role on the convergence rate. It was found that under strong-convexity assumption (i.e., the loss function is twice differentiable, the Hessians of the loss function is lower bounded by a constant c), the convergence rate of averaged SGD with proper step-size is of $O(1/cn)$ [5, 6], while it is only of $O(1/\sqrt{n})$ in non-strongly-convex case [6]. By using the smoothness property of the loss function, it was shown in [10] that the averaged SGD with constant-step-size can achieve a convergence rate of $O(1/n)$ without requiring the strong-convexity assumption.

D. P. Kingma [24] propose Adam, a method for efficient stochastic optimization that only requires first-order gradients with little memory requirement. The method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients; the name Adam is derived from adaptive moment estimation.

Z. A. Zhu [25] introduced Katyusha, a direct, primal-only stochastic gradient method to fix this issue. It has a provably accelerated convergence rate in convex (off-line) stochastic optimization. It can be incorporated into a variance-reduction based algorithm and speed it up, both in terms of sequential and parallel performance.

In this paper, we develop two stochastic accelerated gradient algorithms for the consider least-square and logistic regressions aiming to improve the convergence rate without assuming strong-convexity. Our development is inspired by the work in the area of accelerated gradient method for general stochastic non-linear programming (NLP) [17–23]. In the stochastic NLP setting, different from the consider problem where the gradient can be estimated unbiased at certain points, the gradient is noisy and available through a stochastic oracle. That is, random vectors with unknown distribution are associated with each gradient at certain points. For such problem, recently developed stochastic accelerated gradient method proposed by Ghadimi and Lan [22] showed that for convex smooth function with Lipschitz continuous gradients achieves a convergence rate of $O(1/n^2)$.

In this paper, we prove that without the Lipschitz continuous gradient and strong-convexity assumption, the developed algorithms achieve a convergence rate of $O(1/n^2)$ by using non-asymptotic analysis. Experimental studies on synthetic data and benchmarks justify our theoretical results and

show a faster convergence rate than classical SGD and constant-step-size averaged SGD [10].

The rest of the paper is organised as follows. In Section 2, we present the accelerated gradient algorithm for the least square regression. In Section 3, we study the accelerated gradient algorithm for the logistic regression. Section 4 empirically verify the obtained theoretical results. Section 5 concludes the paper.

2. Stochastic accelerated gradient algorithm for least square regression

In this section, we consider the least square regression. Let (X, d) be a compact metric space and $Y = \mathbb{R}$. Assume ρ be a probability distribution on $Z = \mathcal{X} \times \mathcal{Y}$ and (X, Y) be corresponding random variable. We further assume:

(a) The training data $(x_k, y_k), k \geq 1$ are i.i.d. sampled from ρ.
(b) $\mathbb{E}\|x_k\|^2$ is finite, i.e., $\mathbb{E}\|x_k\|^2 \leq M$ for any $k \geq 1$.
(c) The global minimum of $f(\theta) = \frac{1}{2}\mathbb{E}[\langle\theta, x_k\rangle^2 - 2y_k\langle\theta, x_k\rangle]$ is attainable at a certain point $\theta^* \in \mathbb{R}^d$.
(d) In the following, we denote $\xi_k = \left(y_k - \langle\theta^k, x_k\rangle\right) x_k$ as the residual. We assume that $\mathbb{E}\|\xi_k\|^2 \leq M_1$ for every k and $\overline{\xi}_k = \frac{1}{k}\sum_{i=1}^{k} \xi_i$.

These assumptions are standard in stochastic approximation [9,10]. However, compared with the work in [10], we do not make assumptions on the covariance operator $\mathbb{E}(x_k \bigotimes x_k)$ and $\mathbb{E}[\xi_k \bigotimes \xi_k]$.

In the following, we present the accelerated stochastic gradient algorithm for least square regression learning in Algorithm 1. The algorithm takes a stream of data (x_k, y_k) as input, and an initial guess of the parameter θ_0. The other requirements include $\{\alpha_k\}$ which satisfies $\alpha_1 = 1$ and $\alpha_k > 0$ for any $k \geq 2$, $\beta_k > 0$, and $\lambda_k > 0$. The algorithm involves two intermediate quantities θ^{ag} (which is initialised to be θ_0) and θ^{md}. θ^{md} is updated as a linear combination of θ^{ag} and the current estimation of the parameter θ when a data comes in (line 1), where α_k is the coefficient. The parameter θ is estimated in line 1 taking λ_k as a parameter. The residue and the average residue of previous residues up to the k-th data are computed in line 1. θ^{ag} is then updated by taking β_k as a parameter in line 1. The process continues whenever a new pair of data is seen.

Algorithm 1 The accelerated stochastic gradient algorithm for least square regression.

Require: θ_0 and $\alpha_1 = 1, \alpha_k > 0$ for $k = 2, \cdots$, $\{\beta_k > 0\}$ and $\{\lambda_k > 0\}$

(1) Set $\theta_0^{ag} = \theta_0$, $\overline{\xi}_0 = 0$ and $k = 1$
(2) Set $\theta_k^{md} = (1 - \alpha_k)\theta_{k-1}^{ag} + \alpha_k\theta_{k-1}$,
(3) Set $z_k = \nabla f(\theta_k^{md})/\alpha_k = \left(\langle\theta_k^{md}, x_k\rangle x_k - y_k x_k\right)/\alpha_k$,
(4) Set $\theta_k = \theta_{k-1} - \lambda_k z_k$,
(5) Compute $\xi_k = (y_k - \langle\theta_k, x_k\rangle)\, x_k$ and $\overline{\xi}_k = \overline{\xi}_{k-1} + \frac{1}{k}(\xi_k - \overline{\xi}_{k-1})$;
(6) Set $\theta_k^{ag} = \theta_k^{md} - \beta_k\left(z_k + \frac{1}{k}\overline{\xi}_k\right)$.
(7) Set $k \leftarrow k + 1$, and goto step 2.

2.1. Notes on the algorithm

The unbiased estimate of the gradient, i.e., $\left(\langle\theta_k^{md}, x_k\rangle x_k - y_k x_k\right)$ for each data point (x_k, y_k) is used in line 1. From this perspective, it is seen that the update of θ_k (line 1) is actually the same as in the

stochastic gradient descent (also called least-mean-square, LMS) algorithm if we set $\alpha_k = 1$.

During optimizing, the residue ξ_k is computed (line 1). All the residues up to now are averaged and the averaged residue takes effect on the update of θ_k^{ag} (line 1). It differs from the accelerated stochastic gradient algorithm in [22] where no residue is computed and used in the optimizing.

2.2. Non-asymptotic analysis on convergence rate

This section we establish the convergence rate of the developed algorithm. The goal is to estimate the bound on the expectation $\mathbb{E}[f(\theta_n^{ag}) - f(\theta^*)]$. It turns out that the developed algorithm is able to achieve a convergence rate of $O(1/n^2)$ without strong convexity and Lipschitz continuous gradient assumptions.

To establish the convergence rate of the developed gradient algorithm, we need the following Lemma (see Lemma 1 of [22]).

Lemma 1. *Let α_k be a sequence of step sizes in the accelerated gradient algorithm and the sequence $\{\eta_k\}$ satisfies $\eta_k \le (1-\alpha_k)\eta_{k-1} + \tau_k,\ \ k = 1, 2, \ldots,$ where*

$$\Gamma_k = \begin{cases} 1, & k = 1, \\ (1-\alpha_k)\Gamma_{k-1}, & k \ge 2. \end{cases} \tag{2.1}$$

Then we have $\eta_k \le \Gamma_k \sum_{i=1}^{k} \frac{\tau_i}{\Gamma_i}$ for any $k \ge 1$.

Proof. Noting that $\alpha_1 = 1$ and $\alpha_k \in (0, 1]$, we obtain

$$\frac{\eta_1}{\Gamma_1} = \frac{(1-\alpha_1)\eta_0}{\Gamma_1} + \frac{\tau_1}{\Gamma_1} = \frac{\tau_1}{\Gamma_1},$$

and

$$\frac{\eta_i}{\Gamma_i} \le \frac{(1-\alpha_i)\eta_{i-1}}{\Gamma_i} + \frac{\tau_i}{\Gamma_i} = \frac{\eta_{i-1}}{\Gamma_{i-1}} + \frac{\tau_i}{\Gamma_i},$$

The result then immediately follows by summing up the above inequalities and rearranging the terms. □

Applying Lemma 1, we can obtain the convergence rate of the developed algorithm as explained in Theorem 1.

Theorem 1. *Let $\{\theta_k^{md}, \theta_k^{ag}\}$ be computed by the accelerated gradient algorithm and Γ_k be defined in (2.1). Assume (a–d). If $\{\alpha_k\}, \{\beta_k\}, \{\lambda_k\}$ are chosen such that*

$$1 - \frac{\lambda_k \alpha_k}{2\beta_k} - \frac{\beta_k}{\alpha_k} M \ge 0, \qquad \frac{\alpha_1^2}{\lambda_1 \Gamma_1} \ge \frac{\alpha_2^2}{\lambda_2 \Gamma_2} \ge \cdots, \ \ 2\beta_k^2 M - \beta_k \alpha_k = 0,$$

then for any $n \ge 1$, we have

$$\mathbb{E}\left[f\left(\theta_n^{ag}\right) - f(\theta^*)\right] \ \le \ \frac{\Gamma_n}{2\lambda_1}\|\theta_0 - \theta^*\|^2 + M M_1 \Gamma_n \sum_{k=1}^{n} \frac{\beta_k^2}{k^2 \Gamma_k}.$$

Proof. By Taylor expansion of the function f, Algorithm1 (line 3) and (line 6), we have:

$$\begin{aligned} f\left(\theta_k^{ag}\right) &= f(\theta_k^{md}) + \left\langle \nabla f(\theta_k^{md}), \theta_k^{ag} - \theta_k^{md} \right\rangle + (\theta_k^{ag} - \theta_k^{md})^{\intercal} \nabla^2 f(\theta_k^{md})(\theta_k^{ag} - \theta_k^{md}) \\ &\leq f(\theta_k^{md}) - \beta_k \alpha_k \|z_k\|^2 - \beta_k \alpha_k \frac{1}{k} \langle z_k, \bar{\xi}_k \rangle + \beta_k^2 \mathbb{E}\|x_k\|^2 \|z_k + \frac{1}{k}\bar{\xi}_k\|^2 \\ &\leq f(\theta_k^{md}) - \beta_k \alpha_k \|z_k\|^2 - \beta_k \alpha_k \frac{1}{k} \langle z_k, \bar{\xi}_k \rangle + \beta_k^2 M_1 \|z_k + \frac{1}{k}\bar{\xi}_k\|^2, \end{aligned}$$

where the last inequality holds due to assumption (b). Since

$$f(\mu) - f(\nu) = \langle \nabla f(\nu), \mu - \nu \rangle + (\mu - \nu)^T \mathbb{E}(x_k x_k^T)(\mu - \nu),$$

we have

$$\begin{aligned} f(\nu) - f(\mu) &= \langle \nabla f(\nu), \nu - \mu \rangle - (\mu - \nu)^T \mathbb{E}(x_k x_k^T)(\mu - \nu) \\ &\leq \langle \nabla f(\nu), \nu - \mu \rangle, \end{aligned} \tag{2.2}$$

where the inequality follows from the positive semi-definition of matrix $\mathbb{E}(x_k x_k^{\intercal})$. By Algorithm 1 (line 2) and (2.2), we have

$$\begin{aligned} f(\theta_k^{md}) - [(1-\alpha_k) f(\theta_{k-1}^{ag}) + \alpha_k f(\theta)] &= \alpha_k [f(\theta_k^{md}) - f(\theta)] + (1-\alpha_k)[f(\theta_k^{md}) - f(\theta_{k-1}^{ag})] \\ &\leq \alpha_k \langle \nabla f(\theta_k^{md}), \theta_k^{md} - \theta \rangle + (1-\alpha_k) \langle \nabla f(\theta_k^{md}), \theta_k^{md} - \theta_{k-1}^{ag} \rangle \\ &= \langle \nabla f(\theta_k^{md}), \alpha_k(\theta_k^{md} - \theta) + (1-\alpha_k)(\theta_k^{md} - \theta_{k-1}^{ag}) \rangle \\ &= \alpha_k \langle \nabla f(\theta_k^{md}), \theta_{k-1} - \theta \rangle \\ &= \alpha_k^2 \langle z_k, \theta_{k-1} - \theta \rangle. \end{aligned}$$

So we obtain

$$\begin{aligned} f\left(\theta_k^{ag}\right) \leq\ & (1-\alpha_k) f(\theta_{k-1}^{ag}) + \alpha_k f(\theta) + \alpha_k^2 \langle z_k, \theta_{k-1} - \theta \rangle \\ & -\beta_k \alpha_k \|z_k\|^2 - \beta_k \alpha_k \frac{1}{k} \langle z_k, \bar{\xi}_k \rangle + \beta_k^2 M \|z_k + \frac{1}{k}\bar{\xi}_k\|^2. \end{aligned}$$

It follows from Algorithm 1 (line 4) that

$$\|\theta_k - \theta\|^2 = \|\theta_{k-1} - \lambda_k z_k - \theta\|^2 \tag{2.3}$$

$$= \|\theta_{k-1} - \theta\|^2 - 2\lambda_k \langle z_k, \theta_{k-1} - \theta \rangle + \lambda_k^2 \|z_k\|^2. \tag{2.4}$$

Then we have

$$\langle z_k, \theta_{k-1} - \theta \rangle = \frac{1}{2\lambda_k} \left[\|\theta_{k-1} - \theta\|^2 - \|\theta_k - \theta\|^2 \right] + \frac{\lambda_k}{2} \|z_k\|^2.$$

While

$$\left\| z_k + \frac{1}{k}\bar{\xi}_k \right\|^2 = \|z_k\|^2 + \frac{1}{k^2} \left\|\bar{\xi}_k\right\|^2 + 2\frac{1}{k} \left\langle z_k, \bar{\xi}_k \right\rangle. \tag{2.5}$$

Combining (2.4) and (2.5), we obtain

$$\begin{aligned} f\left(\theta_k^{ag}\right) &\leq (1-\alpha_k)f(\theta_{k-1}^{ag}) + \alpha_k f(\theta) + \frac{\alpha_k^2}{2\lambda_k}\left[\|\theta_{k-1}-\theta\|^2 - \|\theta_k-\theta\|^2\right] \\ &\quad -\beta_k\alpha_k\left(1-\frac{\lambda_k\alpha_k}{2\beta_k}-\frac{\beta_k}{\alpha_k}M\right)\|z_k\|^2 + M\beta_k^2\frac{1}{k^2}\left\|\bar{\xi}_k\right\|^2 + \left\langle\bar{\xi}_k, \frac{1}{k}(2\beta_k^2M-\beta_k\alpha_k)z_k\right\rangle. \end{aligned}$$

The above inequality is equal to

$$\begin{aligned} f\left(\theta_k^{ag}\right) - f(\theta) &\leq (1-\alpha_k)[f(\theta_{k-1}^{ag}) - f(\theta)] + \frac{\alpha_k^2}{2\lambda_k}\left[\|\theta_{k-1}-\theta\|^2 - \|\theta_k-\theta\|^2\right] \\ &\quad -\beta_k\alpha_k\left(1-\frac{\lambda_k\alpha_k}{2\beta_k}-\frac{\beta_k}{\alpha_k}M\right)\|z_k\|^2 + M\beta_k^2\frac{1}{k^2}\left\|\bar{\xi}_k\right\|^2 + \left\langle\bar{\xi}_k, \frac{1}{k}(2\beta_k^2M-\beta_k\alpha_k)z_k\right\rangle. \end{aligned}$$

Using Lemma 1, we have

$$\begin{aligned} f\left(\theta_n^{ag}\right) - f(\theta) &\leq \Gamma_n\sum_{k=1}^n\frac{\alpha_k^2}{2\lambda_k\Gamma_k}\left[\|\theta_{k-1}-\theta\|^2 - \|\theta_k-\theta\|^2\right] + \Gamma_n\sum_{k=1}^n\frac{\beta_k^2M}{\Gamma_k k^2}\left\|\bar{\xi}_k\right\|^2 \\ &\quad -\Gamma_n\sum_{k=1}^n\frac{\beta_k\alpha_k}{\Gamma_k}\left(1-\frac{\lambda_k\alpha_k}{2\beta_k}-\frac{\beta_k}{\alpha_k}M\right)\|z_k\|^2 + \Gamma_n\sum_{k=1}^n\frac{1}{\Gamma_k}\left\langle\bar{\xi}_k, \frac{1}{k}(2\beta_k^2M-\beta_k\alpha_k)z_k)\right\rangle. \end{aligned}$$

Since

$$\frac{\alpha_1^2}{\lambda_1\Gamma_1} \geq \frac{\alpha_2^2}{\lambda_2\Gamma_2} \geq \cdots,\ \alpha_1 = \Gamma_1 = 1,$$

then

$$\sum_{k=1}^n\frac{\alpha_k^2}{2\lambda_k\Gamma_k}\left[\|\theta_{k-1}-\theta\|^2 - \|\theta_k-\theta\|^2\right] \leq \frac{\alpha_1^2}{2\lambda_1\Gamma_1}\left[\|\theta_0-\theta\|^2\right] = \frac{1}{2\lambda_1}\|\theta_0-\theta\|^2.$$

So we obtain

$$f\left(\theta_n^{ag}\right) - f(\theta) \leq \frac{\Gamma_n}{2\lambda_1}\|\theta_0-\theta\|^2 + \Gamma_n\sum_{k=1}^n\frac{\beta_k^2M}{\Gamma_k k^2}\left\|\bar{\xi}_k\right\|^2 + \Gamma_n\sum_{k=1}^n\frac{1}{\Gamma_k}\left\langle\bar{\xi}_k, \frac{1}{k}(2\beta_k^2M-\beta_k\alpha_k)z_k)\right\rangle, \quad (2.6)$$

where the inequality follows from the assumption

$$1-\frac{\lambda_k\alpha_k}{2\beta_k}-\frac{\beta_k}{\alpha_k}M \geq 0,\ \ 2\beta_k^2M-\beta_k\alpha_k = 0.$$

Under assumption (d), we have

$$\mathbb{E}\left\|\bar{\xi}_k\right\|^2 = \mathbb{E}\left\{\frac{1}{k^2}\|\sum_{i=1}^k\xi_i\|^2\right\} \leq \mathbb{E}\left\{\frac{1}{k^2}k\sum_{i=1}^k\|\xi_i\|^2\right\} \leq M_1.$$

Taking expectation on both sides of the inequality (2.6) with respect to (x_i, y_i), we obtain for $\theta \in \mathbb{R}^d$,

$$\mathbb{E}\left[f\left(\theta_n^{ag}\right) - f(\theta)\right] \leq \frac{\Gamma_n}{2\lambda_1}\|\theta_0-\theta\|^2 + MM_1\Gamma_n\sum_{k=1}^n\frac{\beta_k^2}{k^2\Gamma_k}.$$

Now, fixing $\theta = \theta^*$, we have

$$\mathbb{E}\left[f\left(\theta_n^{ag}\right) - f(\theta^*)\right] \leq \frac{\Gamma_n}{2\lambda_1}\|\theta_0 - \theta^*\|^2 + MM_1\Gamma_n \sum_{k=1}^{n} \frac{\beta_k^2}{k^2\Gamma_k}.$$

This finishes the proof of the theorem. □

In the following, we apply the results of Theorem 1 to some particular selections of $\{\alpha_k\}, \{\beta_k\}$ and $\{\lambda_k\}$. We obtain the following Corollary 1.

Corollary 1. *Suppose that α_k and β_k in the accelerated gradient algorithm for regression learning are set to*

$$\alpha_k = \frac{2}{k+1}, \ \beta_k = \frac{1}{M(k+1)} \ \text{and} \ \lambda_k = \frac{k}{2M(k+1)} \quad \forall k \geq 1, \tag{2.7}$$

then for any $n \geq 1$, we have

$$\mathbb{E}\left[f\left(\theta_n^{ag}\right) - f(\theta^*)\right] \leq \frac{4M}{n(n+1)}\|\theta_0 - \theta^*\|^2 + \frac{M_1}{Mn(n+1)}.$$

Proof. In the view (2.1) and (2.7), we have for $k \geq 2$

$$\Gamma_k = (1-\alpha_k)\Gamma_{k-1} = \frac{k-1}{k+1} \times \frac{k-2}{k} \times \frac{k-3}{k-1} \times \cdots \times \frac{2}{4} \times \frac{1}{3} \times \Gamma_1 = \frac{2}{k(k+1)}.$$

It is easy to check

$$1 - \frac{\lambda_k\alpha_k}{2\beta_k} - \frac{\beta_k}{\alpha_k}M \geq 0, \ \ \frac{\alpha_1^2}{\lambda_1\Gamma_1} = \frac{\alpha_2^2}{\lambda_2\Gamma_2} = \cdots = 4M, \ \ 2\beta_k^2 M - \beta_k\alpha_k = 0.$$

Then we obtain

$$M\Gamma_n M_1 \sum_{k=1}^{n} \frac{\beta_k^2}{k^2\Gamma_k} = \frac{2M_1}{n(n+1)} \sum_{k=1}^{n} \frac{\frac{M}{M^2(k+1)^2}}{\frac{2k^2}{k(k+1)}} = \frac{M_1}{Mn(n+1)} \sum_{k=1}^{n} \frac{1}{k(k+1)} \leq \frac{M_1}{Mn(n+1)}.$$

From the result of Theorem 1, we have

$$\mathbb{E}\left[f\left(\theta_n^{ag}\right) - f(\theta^*)\right] \leq \frac{4M}{n(n+1)}\|\theta_0 - \theta^*\|^2 + \frac{M_1}{Mn(n+1)}.$$

This finishes the proof of the Collorary. □

3. The accelerated stochastic gradient algorithm for logistic regression learning

In this section, we develop the accelerated gradient algorithm for logistic regression. For the logistic regression, we consider the logistic loss function: $l(\theta) = \mathbb{E}[\log(1 + \exp(-y\langle x, \theta\rangle))]$. Assume the observations $(x_i, y_i) \in \mathcal{F} \times \{-1, 1\}$ are independent and identically distributed from unknown distribution ρ where $\mathcal{F}$ is a $d-$dimension Euclidean space, with $d \geq 1$. Further, we denote by $\theta^* \in \mathbb{R}^d$ a global minimiser of l and assume its existence. Let $\xi_i = \left(y_i - \langle\theta^k, x_i\rangle\right)x_i$ denote the residual. We denote $\bar{\xi}_k = \frac{1}{k}\sum_{i=1}^{k}\xi_i$ the average residue up until k input data. To analyse the algorithm, we make the following assumptions:

(B1) $\mathbb{E}\|x_i\|^2$ is finite, i.e., $\mathbb{E}\|x_i\|^2 \leq M$ for any $i \geq 1$.
(B2) $\mathbb{E}\|\xi_i\|^2 \leq M_1$ for every i.

Again, unlike the algorithm by Bach et al. [10], we make no assumption on the Hessian operator at the global optimum θ^*.

The developed accelerated stochastic gradient algorithm for the logistic regression is presented in Algorithm 2. In the algorithm, $\theta_0 \in \mathcal{F}$ is an initial guess, and

$$\nabla l(\theta_k) = \frac{-y_k \exp\{-y_k\langle x_k, \theta_k\rangle\}x_k}{1 + \exp\{-y_k\langle x_k, \theta_k\rangle\}}.$$

It can be seen that the basic framework of Algorithm 2 is the same as Algorithm 1 except that the unbiased estimation to the gradient is different due to the loss functions.

Algorithm 2 The accelerated stochastic gradient approximation algorithm for logistic regression.

Require: θ_0 and $\alpha_1 = 1, \alpha_k > 0$ for $k = 2, \cdots$, $\{\beta_k > 0\}$ and $\{\lambda_k > 0\}$
(1) Set $\theta_0^{ag} = \theta_0$,$\bar{\xi}_0 = 0$ and $k = 1$
(2) Set $\theta_k^{md} = (1 - \alpha_k)\theta_{k-1}^{ag} + \alpha_k\theta_{k-1}$,
(3) Compute $z_k = \frac{1}{\alpha_k}\nabla l(\theta_k^{md})$,
(4) Set $\theta_k = \theta_{k-1} - \lambda_k z_k$,
(5) Compute $\xi_k = (y_k - \langle\theta_k, x_k\rangle)\, x_k$ and $\bar{\xi}_k = \bar{\xi}_{k-1} + \frac{1}{k}(\xi_k - \bar{\xi}_{k-1})$;
(6) Set $\theta_k^{ag} = \theta_k^{md} - \beta_k\left(z_k + \frac{1}{k}\bar{\xi}_k\right)$.
(7) Set $k \leftarrow k + 1$, and goto step 2.

3.1. Non-asymptotic analysis on convergence rate

In this section, we also provide the non-asymptotic analysis on the convergence rate of the developed algorithm in expectation. Theorem 2 describes the convergence rate.

Theorem 2. *Let $\{\theta_k^{md}, \theta_k^{ag}\}$ be computed by the accelerated gradient algorithm and Γ_k be defined in (2.1). Assume (B1 and B2). If $\{\alpha_k\}, \{\beta_k\}, \{\lambda_k\}$ are chosen such that*

$$1 - \frac{\lambda_k\alpha_k}{2\beta_k} - \frac{\beta_k}{\alpha_k}M \geq 0, \quad 2\beta_k^2 M - \beta_k\alpha_k = 0, \quad \frac{\alpha_1^2}{\lambda_1\Gamma_1} \geq \frac{\alpha_2^2}{\lambda_2\Gamma_2} \geq \cdots,$$

then for any $n \geq 1$, we have

$$\mathbb{E}\left[f\left(\theta_n^{ag}\right) - f(\theta^*)\right] \leq \frac{\Gamma_n}{2\lambda_1}\|\theta_0 - \theta^*\|^2 + M\sigma^2\Gamma_n\sum_{k=1}^{n}\frac{\beta_k^2}{k^2\Gamma_k}.$$

Proof. By Taylor expansion of the function l, there exists a ϑ such that

$$\begin{aligned} l\left(\theta_k^{ag}\right) &= l(\theta_k^{md}) + \left\langle\nabla l(\theta_k^{md}), \theta_k^{ag} - \theta_k^{md}\right\rangle + (\theta_k^{ag} - \theta_k^{md})^T\nabla^2 l(\vartheta)(\theta_k^{ag} - \theta_k^{md}) \\ &= l(\theta_k^{md}) - \beta_k\alpha_k\|z_k\|^2 - \beta_k\alpha_k\langle z_k, \frac{1}{k}\bar{\xi}_k\rangle \end{aligned}$$

$$+(\theta_k^{ag}-\theta_k^{md})^T\mathbb{E}\left[\frac{\exp\{-y_k\langle x_k,\vartheta\rangle\}x_kx_k^T}{1+\exp\{-y_k\langle x_k,\vartheta\rangle\}}\right](\theta_k^{ag}-\theta_k^{md}) \tag{3.1}$$

In the equation, we know

$$\frac{\exp\{-y_k\langle x_k,\vartheta\rangle\}}{1+\exp\{-y_k\langle x_k,\vartheta\rangle\}}\leq 1,\quad \lambda_{max}(x_kx_k^T)\leq\|x_k\|^2.$$

It is easy to verify the matrix

$$\mathbb{E}\left[\frac{\exp\{-y_k\langle x_k,\vartheta\rangle\}x_kx_k^T}{1+\exp\{-y_k\langle x_k,\vartheta\rangle\}}\right]$$

is positive semidefinite and its largest eigenvalue satisfies

$$\lambda_{max}\left(\mathbb{E}\left[\frac{\exp\{-y_k\langle x_k,\vartheta\rangle\}x_kx_k^T}{1+\exp\{-y_k\langle x_k,\vartheta\rangle\}}\right]\right)\leq\mathbb{E}\|x_k\|^2\leq M.$$

Combining with Algorithm 2 (line 6) and (3.1), we have

$$l\left(\theta_k^{ag}\right)\leq l(\theta_k^{md})-\beta_k\alpha_k\|z_k\|^2-\beta_k\alpha_k\langle z_k,\frac{1}{k}\bar{\xi}_k\rangle+\beta_k^2M\|z_k+\frac{1}{k}\bar{\xi}_k\|^2.$$

Similar to (3.1), there exists a $\zeta\in\mathbb{R}^d$ satisfying

$$l(\mu)-l(\nu)=\langle\nabla l(\nu),\mu-\nu\rangle+(\mu-\nu)^T\mathbb{E}\left[\frac{\exp\{-y_k\langle x_k,\zeta\rangle\}x_kx_k^T}{1+\exp\{-y_k\langle x_k,\zeta\rangle\}}\right](\mu-\nu),\ \mu,\nu\in\mathbb{R}^d$$

we have

$$\begin{aligned}l(\nu)-l(\mu) &= \langle\nabla l(\nu),\nu-\mu\rangle-(\mu-\nu)^T\mathbb{E}\left[\frac{\exp\{-y_k\langle x_k,\zeta\rangle\}x_kx_k^T}{1+\exp\{-y_k\langle x_k,\zeta\rangle\}}\right](\mu-\nu)\\ &\leq \langle\nabla l(\nu),\nu-\mu\rangle,\end{aligned}$$

where the inequality follows from the positive semi-definition of matrix. Similar to (2.2), we have

$$l(\theta_k^{md})-[(1-\alpha_k)l(\theta_{k-1}^{ag})+\alpha_kl(\theta)]\leq\alpha_k^2\langle z_k,\theta_{k-1}-\theta\rangle.$$

So we obtain

$$\begin{aligned}l\left(\theta_k^{ag}\right) &\leq (1-\alpha_k)l(\theta_{k-1}^{ag})+\alpha_kl(\theta)+\alpha_k^2\langle z_k,\theta_{k-1}-\theta\rangle\\ &\quad-\beta_k\alpha_k\|z_k\|^2-\beta_k\alpha_k\langle z_k,\frac{1}{k}\bar{\xi}_k\rangle+\beta_k^2M\|z_k+\frac{1}{k}\bar{\xi}_k\|^2.\end{aligned}$$

It follows from Algorithm 2 (line 4) that

$$\begin{aligned}\|\theta_k-\theta\|^2 &= \|\theta_{k-1}-\lambda_kz_k-\theta\|^2\\ &= \|\theta_{k-1}-\theta\|^2-2\lambda_k\langle z_k,\theta_{k-1}-\theta\rangle+\|z_k\|^2.\end{aligned}$$

Combining the above two inequalities, we obtain

$$\begin{aligned} l\left(\theta_k^{ag}\right) &\leq (1-\alpha_k) l(\theta_{k-1}^{ag}) + \alpha_k l(\theta) + \frac{\alpha_k^2}{2\lambda_k}\left[\|\theta_{k-1}-\theta\|^2 - \|\theta_k-\theta\|^2\right] \\ &\quad -\beta_k\alpha_k\left(1-\frac{\lambda_k\alpha_k}{2\beta_k}-\frac{\beta_k}{\alpha_k}M\right)\|z_k\|^2 + M\beta_k^2\frac{1}{k^2}\left\|\bar{\xi}_k\right\|^2 + \left\langle\bar{\xi}_k, \frac{1}{k}(2\beta_k^2 M-\beta_k\alpha_k)z_k\right\rangle. \end{aligned}$$

The above inequality is equal to

$$\begin{aligned} l\left(\theta_k^{ag}\right) - l(\theta) &\leq (1-\alpha_k)[l(\theta_{k-1}^{ag}) - l(\theta)] + \frac{\alpha_k^2}{2\lambda_k}\left[\|\theta_{k-1}-\theta\|^2 - \|\theta_k-\theta\|^2\right] \\ &\quad -\beta_k\alpha_k\left(1-\frac{\lambda_k\alpha_k}{2\beta_k}-\frac{\beta_k}{\alpha_k}M\right)\|z_k\|^2 + M\beta_k^2\frac{1}{k^2}\left\|\bar{\xi}_k\right\|^2 + \left\langle\bar{\xi}_k, \frac{1}{k}(2\beta_k^2 M-\beta_k\alpha_k)z_k\right\rangle. \end{aligned}$$

Using Lemma 1, we have

$$\begin{aligned} l\left(\theta_n^{ag}\right) - l(\theta) &\leq \Gamma_n\sum_{k=1}^{n}\frac{\alpha_k^2}{2\lambda_k\Gamma_k}\left[\|\theta_{k-1}-\theta\|^2 - \|\theta_k-\theta\|^2\right] + \Gamma_n\sum_{k=1}^{n}\frac{\beta_k^2 M}{\Gamma_k k^2}\left\|\bar{\xi}_k\right\|^2 \\ &\quad -\Gamma_n\sum_{k=1}^{n}\frac{\beta_k\alpha_k}{\Gamma_k}\left(1-\frac{\lambda_k\alpha_k}{2\beta_k}-\frac{\beta_k}{\alpha_k}M\right)\|z_k\|^2 + \Gamma_n\sum_{k=1}^{n}\frac{1}{\Gamma_k}\left\langle\bar{\xi}_k, \frac{1}{k}(2\beta_k^2 M-\beta_k\alpha_k)z_k\right\rangle. \end{aligned}$$

Since

$$\frac{\alpha_1^2}{\lambda_1\Gamma_1} \geq \frac{\alpha_2^2}{\lambda_2\Gamma_2} \geq \cdots,\ \alpha_1 = \Gamma_1 = 1,$$

then

$$\sum_{k=1}^{n}\frac{\alpha_k^2}{2\lambda_k\Gamma_k}\left[\|\theta_{k-1}-\theta\|^2 - \|\theta_k-\theta\|^2\right] \leq \frac{\alpha_1^2}{2\lambda_1\Gamma_1}\left[\|\theta_0-\theta\|^2\right] = \frac{1}{2\lambda_1}\|\theta_0-\theta\|^2.$$

So we obtain

$$l\left(\theta_n^{ag}\right) - l(\theta) \leq \frac{\Gamma_n}{2\lambda_1}\|\theta_0-\theta\|^2 + \Gamma_n\sum_{k=1}^{n}\frac{\beta_k^2 M}{\Gamma_k k^2}\left\|\bar{\xi}_k\right\|^2, \tag{3.2}$$

where the inequality follows from the assumption

$$1-\frac{\lambda_k\alpha_k}{2\beta_k}-\frac{\beta_k}{\alpha_k}M \geq 0,\ \ 2\beta_k^2 M-\beta_k\alpha_k = 0.$$

Under the assumption (B4), we have

$$\mathbb{E}\left\|\bar{\xi}_k\right\|^2 = \mathbb{E}\left\{\frac{1}{k^2}\|\sum_{i=1}^{k}\xi_i\|^2\right\} \leq \mathbb{E}\left\{\frac{1}{k^2}k\sum_{i=1}^{k}\|\xi_i\|^2\right\} \leq M_1.$$

Taking expectation on both sides of the inequality (3.2) with respect to (x_i, y_i), we obtain for $\theta \in \mathbb{R}^d$,

$$\mathbb{E}\left[l\left(\theta_n^{ag}\right) - l(\theta)\right] \leq \frac{\Gamma_n}{2\lambda_1}\|\theta_0-\theta\|^2 + MM_1\Gamma_n\sum_{k=1}^{n}\frac{\beta_k^2}{k^2\Gamma_k}.$$

Now, fixing $\theta = \theta^*$, we have

$$\mathbb{E}\left[l\left(\theta_n^{ag}\right) - l(\theta^*)\right] \leq \frac{\Gamma_n}{2\lambda_1}\|\theta_0 - \theta^*\|^2 + MM_1\Gamma_n \sum_{k=1}^{n} \frac{\beta_k^2}{k^2\Gamma_k}.$$

This finishes the proof of the theorem. □

Similar to Corollary 1, we specialise the result of Theorem 2 for some particular selections of $\{\alpha_k\}, \{\beta_k\}$ and $\{\lambda_k\}$.

Corollary 2. *Suppose that α_k and β_k in the accelerated gradient algorithm for regression learning are set to*

$$\alpha_k = \frac{2}{k+1}, \ \beta_k = \frac{1}{M(k+1)} \ \text{and} \ \lambda_k = \frac{k}{2M(k+1)}, \quad \forall k \geq 1,$$

then for any $n \geq 1$, we have

$$\mathbb{E}\left[l\left(\theta_n^{ag}\right) - l(\theta^*)\right] \leq \frac{4M}{n(n+1)}\|\theta_0 - \theta^*\|^2 + \frac{M_1}{Mn(n+1)}.$$

4. Experiment results

In this section, we empirically investigate the performance of our algorithms on synthetic data and some benchmarks widely used by the machine learning community.

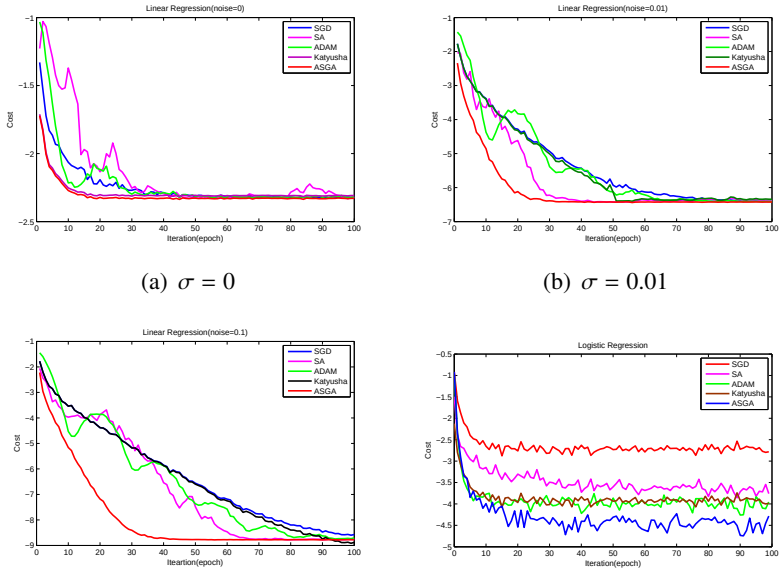
4.1. Least square regression

We consider normally distributed inputs, with covariance matrix H that has random eigenvectors and eigenvalues $1/k, k = 1, \ldots, d$. The outputs are generated from a linear function with homoscedastic noise with various signal to noise-ratio σ. We consider $d = 20$ and 100000 samples using mini-batch size of 100.

We compare SGD, stochastic approximation (SA) with averaging [10], ADAM [24], Katyusha [25] and ASGA on synthetic noisy datasets with different noise levels: $\sigma = 0, 0.1$ and 0.01. For SGD and SA we choose the step size $\rho = 1/2R^2$ and $\gamma_n = 1/2R^2\sqrt{n}$) where $R^2 = trace(H)$, respectively. The loss function is defined as $\log_{10}[f(\theta) - f(\theta^*)]$. The average loss function value over 100 runs on the training data is shown in Figure 1 (a)–(c). It can be seen that ASGA converges much faster than SGD, SA and ADAM. The results also verify our theoretical improvement on the convergence rate of ASGA.

4.2. Logistic regression

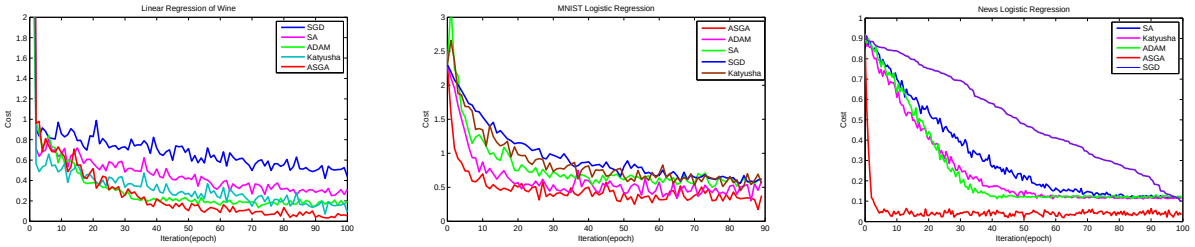
For logistic regression, we consider the same input data as for the least-squares, but outputs are generated from the logistic probabilistic model. Comparison results are shown in Figure 1 (d). A step size $\gamma_n = 1/(2R^2\sqrt{n})$ is chosen for SA with averaging for an optimal performance. For ADAM, its step size α is adjusted by $1/\sqrt{n}$ decay as suggested in [24]. From Figure 1 (d), it is clearly seen that ASGA converges significantly faster than all the compared algorithms.



(a) $\sigma = 0$ (b) $\sigma = 0.01$

(c) $\sigma = 0.1$ (d) logistic regression

Figure 1. (a)–(c): Least square regression training log-likelihood on synthetic data sets with different noise levels (0, 0.01 and 0.1 clockwise in turn); and (d) logistic regression on synthetic data set.



(a) Wine (b) MNIST (c) News

Figure 2. The training procedure of ASGA on Wine, MNIST and News averaged over 50 runs.

4.3. Benchmarks

We evaluate ASGA on the MNIST, Wine dataset and News dataset. In our experiments, the raw features of the datasets are used directly as the input to the classifier. For MNIST, 9000/1000 data points are used as training/test dataset, the numbers are 4000/898 for Wine, while the numbers are 18000/846 for News. We compare SGD, stochastic approximation (SA) with averaging [10], ADAM [24], Katyusha [25] and ASGA using mini-batch size of 90 in MINST, using mini-batch size of 100 in Wine and News. Figure 2 shows the training procedure of ASGA on these three datasets by averaging 50 runs. From Figure 2, it is seen that ASGA obtains better the loss values with faster convergence rate. This clearly demonstrates the effectiveness of ASGA. Table 1 summarizes the prediction accuracies of the obtained optimal parameters of the logistic regression model on the test

datasets by the compared algorithms. The p-values obtained by t-test at 5% significance level are also given in the subscripts of the compared algorithms. From the table, it is seen that ASGA achieves significantly better accuracies that the other algorithms ($p < 0.05$).

Table 1. The prediction accuracy of the compared algorithms on test datasets of MNIST and Wine, where the subscript values are the p-values obtained by the t-test between the corresponding algorithm with ASGA.

Method	MNIST	Wine	News
ASGA	0.9056	0.8863	0.9236
ADAM	$0.8812_{1.5\times10^{-6}}$	$0.8520_{2.6\times10^{-8}}$	$0.9120_{3.5\times10^{-6}}$
SA	$0.8913_{4.2\times10^{-5}}$	$0.8612_{2.7\times10^{-7}}$	$0.8875_{4.7\times10^{-5}}$
Katyusha	$0.8845_{1.7\times10^{-4}}$	$0.8624_{3.8\times10^{-3}}$	$0.8943_{6.3\times10^{-4}}$
SGD	$0.8615_{3.7\times10^{-10}}$	$0.8523_{4.7\times10^{-8}}$	$0.8831_{2.4\times10^{-6}}$

5. Conclusions

In this paper, we proposed two accelerated stochastic gradient algorithms (ASGA) for least-square and logistic regression in which the averaged residue is used to adjust the parameter estimation. An asymptotic analysis proved that ASGA can achieve a convergence rate of $O(1/n^2)$ which is much tighter than the state-of-the-art rate under non-strongly convexity assumptions. Experimental results on synthetic data and benchmark datasets justified our theoretical results.

Acknowledgments

The authors acknowledge the financial supports from the National Natural Science Foundation of China [No.61573326], Support project for outstanding young talents in Colleges and universities in Anhui Province [No.gxyq2018076], Natural science research project of colleges and universities in Anhui Province [No.KJ2018A0455], scientific research project of Chaohu University [No.XLY-201903].

Conflict of interest

The authors declare that they have no competing interests.

References

1. H. Robbins, S. Monro, A stochastic approximation method, In: *The annals of mathematical statistics*, Institute of Mathematical Statistics, **22** (1951), 400–407.
2. B. T. Polyak, New stochastic approximation type procedures, *Automat. i Telemekh*, 1990, 98–107.
3. B. T. Polyak, A. B. Juditsky, Acceleration of stochastic approximation by averaging, *SIAM J. Control Optim.*, **30** (1992), 838–855. doi: 10.1137/0330046.
4. L. Bottou, O. Bousquet, The tradeoffs of large scale learning, In: *Advances in neural information processing systems*, **20** (2007), 1–8.

5. S. Shalev-Shwartz, Y. Singer, N. Srebro, A. Cotter, Pegasos: Primal estimated sub-gradient solver for SVM, *Math. Program.*, **127** (2011), 3–30. doi: 10.1007/s10107-010-0420-4.
6. A. Nemirovski, A. Juditsky, G. Lan, A. Shapiro, Robust stochastic approximation approach to stochastic programming, *SIAM J. Optim.*, **19** (2009), 1574–1609. doi: 10.1137/070704277.
7. G. H. Lan, R. D. C. Monteiro, Iteration-complexity of first-order penalty methods for convex programming, *Math. Program.*, **138** (2013), 115–139. doi: 10.1007/s10107-012-0588-x.
8. S. Ghadimi, G. H. Lan, Accelerated gradient methods for nonconvex nonlinear and stochastic programming, *Math. Program.*, **156** (2016), 59–99. doi: 10.1007/s10107-015-0871-8.
9. F. Bach, E. Moulines, Non-asymptotic analysis of stochastic approximation algorithms for machine learning, 2011.
10. F. Bach, E. Moulines, Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$, 2013. Avaliable from: https://proceedings.neurips.cc/paper/2013/file/7fe1f8abaad094e0b5cb1b01d712f708-Paper.pdf.
11. J. C. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, *J. Mach. Learn. Res.*, **12** (2010), 2121–2159.
12. Y. E. Nesterov, A method of solving a convex programming problem with convergence rate $O(1/k^2)$, *Dokl. Akad. Nauk SSSR*, **269** (1983), 543–547.
13. A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, *SIAM J. Imaging Sci.*, **2**, (2009), 183–202. doi: 10.1137/080716542.
14. P. Tseng, On accelerated proximal gradient methods for convex-concave optimization, *SIAM J. Optimiz.*, 2008.
15. Y. Nesterov, Smooth minimization of non-smooth functions, *Math. Program.*, **103** (2005), 127–152. doi: 10.1007/s10107-004-0552-5.
16. Y. Nesterov, Gradient methods for minimizing composite functions, *Math. Program.*, **140** (2013), 125–161. doi: 10.1007/s10107-012-0629-5.
17. G. H. Lan, An optimal method for stochastic composite optimization, *Math. Program.*, **133** (2012), 365–397. doi: 10.1007/s10107-010-0434-y.
18. S. Ghadimi, G. H. Lan, Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization I: A generic algorithmic framework, *SIAM J. Optim.*, **22** (2012), 1469–1492. doi: 10.1137/110848864.
19. S. Ghadimi, G. H. Lan, Stochastic first- and zeroth-order methods for nonconvex stochastic programming, *SIAM J. Optim.*, **23** (2013), 2341–2368. doi: 10.1137/120880811.
20. S. Ghadimi , G. H. Lan, H. C. Zhang, Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization, *Math. Program.*, **155** (2016), 267–305. doi: 10.1007/s10107-014-0846-1.
21. G. H. Lan, Bundle-level type methods uniformly optimal for smooth and nonsmooth convex optimization, *Math. Program.*, **149** (2015), 1–45. doi: 10.1007/s10107-013-0737-x.
22. S. Ghadimi, G. H. Lan, Accelerated gradient methods for nonconvex nonlinear and stochastic programming, *Math. Program.*, **156** (2016), 59–99. doi: 10.1007/s10107-015-0871-8.

23. L. Bottou, Large-scale machine learning with stochastic gradient descent. In: *Proceedings of COMPSTAT'2010*, Physica-Verlag HD, 2010, 177–186. doi: 10.1007/978-3-7908-2604-3_16.

24. D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv: 1412.6980v9, 2012.

25. Z. A. Zhu, Katyusha: The first direct acceleration of stochastic gradient methods, In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2017)*, New York, USA: Association for Computing Machinery. doi: 10.1145/3055399.3055448.

AIMS Press