A DOUBLE INCREMENTAL AGGREGATED GRADIENT METHOD WITH LINEAR CONVERGENCE RATE FOR LARGE-SCALE OPTIMIZATION

Aryan Mokhtari[*], *Mert Gürbüzbalaban*[†], *and Alejandro Ribeiro*[*]

[*]Department of Electrical and Systems Engineering, University of Pennsylvania
[†]Department of Management Science and Information Systems, Rutgers University

ABSTRACT

This paper considers the problem of minimizing the average of a finite set of strongly convex functions. We introduce a double incremental aggregated gradient method (DIAG) that computes the gradient of only one function at each iteration, which is chosen based on a cyclic scheme, and uses the aggregated average gradient of all the functions to approximate the full gradient. We prove that not only the proposed DIAG method converges linearly to the optimal solution, but also its linear convergence factor justifies the advantage of incremental methods on full batch gradient descent. In particular, we show theoretically and empirically that one pass of DIAG is more efficient than one iteration of gradient descent.

Index Terms— Incremental methods, gradient descent, linear convergence rate

1. INTRODUCTION

We consider optimization problems where the objective function can be written as the average of a set of strongly convex and smooth functions. Formally, consider variable $\mathbf{x} \in \mathbb{R}^p$ and n objective function summands $f_i : \mathbb{R}^p \to \mathbb{R}$. We aim to find the minimizer of the average function $f(\mathbf{x}) := (1/n)\sum_{i=1}^n f_i(\mathbf{x})$, i.e.,

$$\mathbf{x}^* = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x}) := \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{x}). \tag{1}$$

In this paper, we refer to the functions f_i as the instantaneous functions and the average function f as the global objective function. This class of optimization problems arises in many applications including machine learning [1], estimation [2], and sensor networks [3].

When the number of instantaneous functions f_i is large, it is costly to compute descent directions of the aggregate function f. In particular, this makes the use of gradient descent (GD) in (1) costly because each descent step requires cycling through the whole set of instantaneous functions f_i. A standard solution to this drawback is in the form of the stochastic (S)GD method which evaluates the gradient of only one of the instantaneous functions in each iteration [4]. This algorithm can be shown to converge under mild conditions while incurring a reasonable per-iteration cost. This advantage notwithstanding, the convergence rate of SGD is sublinear, which is slower than the linear convergence rate of GD. Developing alternative stochastic descent algorithms with linear convergence rates has been a very active area in the last few years. A partial list of this consequential literature includes stochastic averaging gradient [5, 6], variance reduction [7, 8], dual coordinate methods [9, 10], hybrid algorithms [11, 12], and majorization-minimization algorithms [13]. All of these stochastic methods are successful in achieving a linear convergence rate in expectation.

A separate alternative to reduce the per-iteration cost of GD is the use of incremental methods [14, 15]. In incremental methods one function is chosen from the set of n functions at each iteration as in GD but the functions are chosen in a *cyclic* order – as opposed from their selection uniformly at random in stochastic methods. As in the case of SGD, cyclic GD exhibits sublinear convergence. This limitation motivated the development of the incremental aggregated gradient (IAG) method that achieves a linear convergence rate [16]. To explain our contribution, we must emphasize that the convergence *constant* of IAG can be smaller than the convergence constant of GD (Section 2). Thus, even though IAG is designed to improve upon GD, the available analyses still make it impossible to assert that IAG outperforms GD under all circumstances. In fact, the question of whether it is possible at all to design a cyclic method that is guaranteed to always outperform GD remains open.

In this paper we introduce the double incremental aggregated gradient (DIAG) method and show that its convergence rate is linear, with a convergence constant that is guaranteed to be smaller than the convergence constant of GD. The main difference between DIAG and IAG methods is that DIAG iterates are computed by using averages of iterates and gradients whereas IAG utilizes gradient averages but does *not* utilize iterate averages. DIAG is the first cyclic incremental gradient method which is guaranteed to improve the performance of GD.

We start the paper by presenting the GD and IAG algorithms and reviewing their convergence rates (Section 2). Then, we present the proposed DIAG method and explain its difference with IAG in approximating the global function f (Section 3). We show that this critical difference leads to an incremental gradient algorithm with a smaller linear convergence factor (Section 4). Moreover, we compare the performances of DIAG, GD, and IAG in solving a quadratic programming and a binary classification problem (Section 5). Finally, we close the paper by concluding remarks (Section 6). Proofs of results in this paper are available in [17].

2. BACKGROUND AND RELATED WORKS

Since the objective function in (1) is convex, descent methods can be used to find the optimal argument $\mathbf{x}^*$. In this paper, we are interested in studying methods that converge to the optimal argument of $f(\mathbf{x})$ at a linear rate. It is customary for the linear convergence analysis of first-order methods to assume that the functions are smooth and strongly convex. We formalize these conditions in the following assumption.

Assumption 1 *The functions f_i are differentiable and strongly convex with constant μ, and the gradients ∇f_i are Lipschitz continuous with constant L, i.e., for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$ we have*

$$\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|. \tag{2}$$

The strong convexity of the functions f_i with constant μ implies that the global function f is also strongly convex with constant μ. Likewise, the condition in (2) yields Lipschitz continuity of the global function gradients ∇f with constant L. Note that the conditions in Assumption 1 are mild, and they hold for most machine learning applications.

The optimization problem in (1) can be solved using the gradient descent (GD) method. In GD, the variable $\mathbf{x}^k$ is updated by descending

Work in this paper is supported by ONR N00014-12-1-0997.

through the negative direction of the gradient $\nabla f(\mathbf{x}^k)$, i.e.,

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \epsilon \nabla f(\mathbf{x}^k) = \mathbf{x}^k - \frac{\epsilon}{n} \sum_{i=1}^{n} \nabla f_i(\mathbf{x}^k), \tag{3}$$

where ϵ is the stepsize. According to the convergence analysis of GD in [18], the sequence of iterates $\mathbf{x}^k$ converges linearly to the optimal argument if the stepsize satisifes $\epsilon < 2/L$. The fastest convergence rate belongs to the stepsize $\epsilon = 2/(\mu + L)$ which leads to the inequality

$$\|\mathbf{x}^k - \mathbf{x}^*\| \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^k \|\mathbf{x}^0 - \mathbf{x}^*\|, \tag{4}$$

where $\kappa = L/\mu$ is the condition number of the objective function. The result in (4) shows that GD reduces the difference between the iterate $\mathbf{x}^k$ and the optimal argument $\mathbf{x}^*$ by the factor $(\kappa - 1)/(\kappa + 1)$ after one iteration or equivalently after one pass over the dataset.

The IAG method reduces the computational complexity of GD by computing only one gradient at each iteration. In IAG, at each iteration the gradient of only one function, which is chosen in a cyclic order, is updated and the average of gradients is used as an approximation for the exact gradient. In particular, if we define $\mathbf{y}_i^k$ as the copy of the variable $\mathbf{x}$ for the last time that the function f_i's gradient is updated up to step k, we can write the update of IAG as

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \frac{\epsilon}{n} \sum_{i=1}^{n} \nabla f_i(\mathbf{y}_i^k). \tag{5}$$

It has been shown that IAG is linearly convergent for strongly convex functions with Lipschitz continuous gradients [16]. In particular, the sequence of iterates $\mathbf{x}^k$ generated by IAG satisfies the inequality

$$\|\mathbf{x}^k - \mathbf{x}^*\| \leq \left(1 - \frac{2}{25n(2n+1)(\kappa+1)^2}\right)^k \|\mathbf{x}^0 - \mathbf{x}^*\|. \tag{6}$$

Comparing the decrement factors of GD in (4) and IAG after n gradient evaluations in (6) shows that for some values of n and κ the GD method is preferable to IAG in terms of upper bounds. In particular, there exists n and κ such that the decrement factor of one iteration of GD is smaller than the decrement factor of IAG after n iterations, i.e., if the inequality

$$\left(\frac{\kappa - 1}{\kappa + 1}\right) < \left(1 - \frac{2}{25n(2n+1)(\kappa+1)^2}\right)^n, \tag{7}$$

is satisfied, the convergence rate of GD in (4) is better than the convergence rate of IAG in (6). This is more likely to happen when the condition number κ is relatively large. In the following section, we propose a new incremental gradient method that is preferable with respect to GD for all values of n and κ.

3. ALGORITHM DEFINITION

The update of IAG in (5) can be written as the minimization of a first order approximation of the objective function $f(\mathbf{x})$ where each instantaneous function $f_i(\mathbf{x})$ is approximated by the following approximation

$$f_i(\mathbf{x}) \approx f_i(\mathbf{x}^k) + \nabla f_i(\mathbf{y}_i^k)^T(\mathbf{x} - \mathbf{x}^k) + \frac{1}{2\epsilon}\|\mathbf{x} - \mathbf{x}^k\|^2. \tag{8}$$

Notice that the first two terms $f_i(\mathbf{x}^k) + \nabla f_i(\mathbf{y}_i^k)^T(\mathbf{x} - \mathbf{x}^k)$ correspond to the first order approximation of the function f_i around the iterate $\mathbf{y}_i^k$. The last term, which is $1/(2\epsilon)\|\mathbf{x}-\mathbf{x}^k\|^2$, is a proximal term that is added to the first order approximation. This approximation is different from the customary approximation that is used in first-order methods since the first-order approximation of the function $f_i(\mathbf{x})$ is evaluated at $\mathbf{y}_i^k$, while the iterate $\mathbf{x}^k$ is used in the proximal term. This observation verifies that the IAG method performs well when the delayed variables $\mathbf{y}_i^k$ are close to the current iterate $\mathbf{x}^k$.

We resolve this issue by replacing the approximation of IAG in (8) with the approximation that uses $\mathbf{y}_i^k$ both in first-order approximation and proximity condition. In particular, we propose a novel cyclic incremental method called double incremental aggregated gradient method (DIAG) which approximates the instantaneous function $f_i(\mathbf{x})$ as

$$f_i(\mathbf{x}) \approx f_i(\mathbf{y}_i^k) + \nabla f_i(\mathbf{y}_i^k)^T(\mathbf{x} - \mathbf{y}_i^k) + \frac{1}{2\epsilon}\|\mathbf{x} - \mathbf{y}_i^k\|^2. \tag{9}$$

In general, the approximation in (9) is more accurate than the one in (8) since the first order approximation $f_i(\mathbf{y}_i^k) + \nabla f_i(\mathbf{y}_i^k)^T(\mathbf{x} - \mathbf{y}_i^k)$ and the proximal term $(1/2\epsilon)\|\mathbf{x} - \mathbf{y}_i^k\|^2$ are both evaluated using the same point which is $\mathbf{y}_i^k$. Considering this approximation the update of the DIAG method is given by

$$\mathbf{x}^{k+1} = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^p} \Big\{ \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{y}_i^k) + \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\mathbf{y}_i^k)^T(\mathbf{x} - \mathbf{y}_i^k) + \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2\epsilon}\|\mathbf{x} - \mathbf{y}_i^k\|^2 \Big\}. \tag{10}$$

The update in (10) minimizes the first-order approximation of the global objective function $f(\mathbf{x})$ which is the outcome of the instantaneous functions approximation in (9). Considering the convex programming in (10) we can derive a closed form solution for the variable $\mathbf{x}^{k+1}$ as

$$\mathbf{x}^{k+1} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{y}_i^k - \frac{\epsilon}{n}\sum_{i=1}^{n} \nabla f_i(\mathbf{y}_i^k). \tag{11}$$

The DIAG update in (11) requires the *incremented aggregate* of *both* variables and gradients and only uses *gradient* (first-order) information. Hence, we call it the *double incremental aggregated gradient* method.

Since we use a cyclic scheme, the set of variables $\{\mathbf{y}_1^k, \mathbf{y}_2^k, \ldots, \mathbf{y}_n^k\}$ is equal to the set $\{\mathbf{x}^k, \mathbf{x}^{k-1}, \ldots, \mathbf{x}^{k-n+1}\}$. Hence, the update for the proposed cyclic incremental aggregated gradient method with the cyclic order $f_1, f_2, \ldots, f_n$ can be written as

$$\mathbf{x}^{k+1} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}^{k-n+i} - \frac{\epsilon}{n}\sum_{i=1}^{n} \nabla f_i(\mathbf{x}^{k-n+i}). \tag{12}$$

The update in (12) shows that we use the first-order information of the functions f_i around the last n iterates to evaluate the new update $\mathbf{x}^{k+1}$. In other words, $\mathbf{x}^{k+1}$ is a function of the last n iterates $\{\mathbf{x}^k, \mathbf{x}^{k-1}, \ldots, \mathbf{x}^{k-n+1}\}$. This observation is very fundamental in the analysis of the DIAG method as we study in Section 4.

Remark 1 One may consider the proposed DIAG method as a cyclic version of the stochastic MISO algorithm in [13]. This is a valid interpretation; however, the convergence analysis of MISO cannot guarantee that for all choices of n and κ it outperforms GD, while we establish theoretical results in Section 4 which guarantee the advantages of DIAG on GD for any n and κ. Moreover, the proposed DIAG method is designed based on the new interpretation in (9) that leads to a novel proof technique; see Lemma 1. This new analysis is different from the analysis of MISO in [13] and provides stronger convergence results.

3.1. Implementation Details

Naive implementation of the update in (11) requires computation of sum of n vectors per iteration which is computationally costly. This unnecessary computation can be avoided by tracking the sums in (11) over time. To be more precise, the first sum in (11) which is the sum of the variables can be updated as

$$\sum_{i=1}^{n} \mathbf{y}_i^{k+1} = \sum_{i=1}^{n} \mathbf{y}_i^k + \mathbf{x}^{k+1} - \mathbf{y}_{i^k}^k, \tag{13}$$

Algorithm 1 The proposed DIAG method

1: **Require**: $\{\mathbf{y}_i^0\}_{i=1}^n = \mathbf{x}^0$, and $\{\nabla f_i(\mathbf{y}_i^0)\}_{i=1}^n$
2: Set the function index as $i^0 = 1$
3: **for** $k = 0, 1, \dots$ **do**
4: Update variable $\mathbf{x}^{k+1} = \frac{1}{n}\sum_{i=1}^n \mathbf{y}_i^k - \frac{\epsilon}{n}\sum_{i=1}^n \nabla f_i(\mathbf{y}_i^k)$.
5: Update the sum of variables $\sum_{i=1}^n \mathbf{y}_i^{k+1} = \sum_{i=1}^n \mathbf{y}_i^k + \mathbf{x}^{k+1} - \mathbf{y}_{i^k}^k$.
6: Compute $\nabla f_{i^k}(\mathbf{x}^{k+1})$ and update the sum of gradients $\sum_{i=1}^n \nabla f_i(\mathbf{y}_i^{k+1}) = \nabla f_{i^k}(\mathbf{x}^{k+1}) - \nabla f_{i^k}(\mathbf{y}_{i^k}^k) + \sum_{i=1}^n \nabla f_i(\mathbf{y}_i^k)$.
7: Replace $\mathbf{y}_{i^k}^k$ and $\nabla f_{i^k}(\mathbf{y}_{i^k}^k)$ in the table by $\nabla f_{i^k}(\mathbf{x}^{k+1})$ and $\mathbf{x}^{k+1}$, respectively. The rest remain unchanged. i.e., $\mathbf{y}_i^{k+1} = \mathbf{y}_i^k$ and $\nabla f_i(\mathbf{y}_i^{k+1}) = \nabla f_i(\mathbf{y}_i^k)$ for $i \neq i^k$.
8: Update the function index $i^{k+1} = mod(i^k, n) + 1$.
9: **end for**

where i^k is the index of the function that is chosen at step k. Likewise, the sum of gradients in (11) can be updated as

$$\sum_{i=1}^n \nabla f_i(\mathbf{y}_i^{k+1}) = \sum_{i=1}^n \nabla f_i(\mathbf{y}_i^k) + \nabla f_{i^k}(\mathbf{x}^{k+1}) - \nabla f_{i^k}(\mathbf{y}_{i^k}^k). \quad (14)$$

Note that the implementation of DIAG requires a memory of the order $\mathcal{O}(np)$ to store the variables $\mathbf{y}_i^k$ and gradients $\nabla f_i(\mathbf{y}_i^k)$.

The proposed DIAG method is summarized in Algorithm 1. The variables for all copies of the vector $\mathbf{x}$ are initialized by vector $\mathbf{0}$, i.e., $\mathbf{y}_1^0 = \cdots = \mathbf{y}_n^0 = \mathbf{x}^0 = \mathbf{0}$, and their corresponding gradients are stored in the memory. At each iteration k, the updated variable $\mathbf{x}^{k+1}$ is computed in Step 4 using the update in (11). The sums of variables and gradients are updated in Step 5 and 6, respectively, following the recursions in (13) and (14). In Step 7, the old variable and gradient of the updated function f_{i^k} are replaced with their updated versions and other components of the the variable and gradient tables remain unchanged. Finally, in Step 8, the function index is updated in a cyclic manner by increasing the index i^k by 1. If the current value of the index i^k is n, we set $i^{k+1} = 1$ for the next iteration.

4. CONVERGENCE ANALYSIS

In this section, we study the convergence properties of the DIAG method and justify its advantages versus the GD algorithm.

In the following lemma, we characterize an upper bound for the optimality error at step $k+1$ in terms of the optimality errors of its previous n iterations.

Lemma 1 *Consider the proposed DIAG method in* (11)*. If the conditions in Assumption 1 hold, and the stepsize ϵ is chosen as $\epsilon = 2/(\mu + L)$, the sequence of iterates $\mathbf{x}^k$ generated by DIAG satisfies the inequality*

$$\|\mathbf{x}^{k+1} - \mathbf{x}^*\| \leq \frac{\kappa - 1}{\kappa + 1}\left[\frac{\|\mathbf{x}^k - \mathbf{x}^*\| + \cdots + \|\mathbf{x}^{k-n+1} - \mathbf{x}^*\|}{n}\right], \quad (15)$$

where $\kappa = L/\mu$ is the objective function condition number.

The result in Lemma 1 has a significant role in the analysis of the proposed DIAG method. It shows that the error $\|\mathbf{x}^{k+1} - \mathbf{x}^*\|$ at step $k+1$ is smaller than the average of the last n errors. This is true since the ratio $(\kappa - 1)/(\kappa + 1)$ is strictly smaller than 1. Note that the cyclic scheme of DIAG is critical to prove the result in (15), since it allows to replace the sum $\sum_{i=1}^n \|\mathbf{y}_i^k - \mathbf{x}^*\|$ by the sum of the last n errors $\|\mathbf{x}^k - \mathbf{x}^*\| + \cdots + \|\mathbf{x}^{k-n+1} - \mathbf{x}^*\|$. If we pick functions uniformly at random, as in MISO, it is not possible to write the expression in (15), even in expectation. Likewise, for the IAG method, we cannot guarantee that the inequality in (15) holds. This special property distinguishes DIAG from IAG and MISO. In the following theorem, we use the result in Lemma 1 to show that the sequence of errors $\|\mathbf{x}^k - \mathbf{x}^*\|$ is convergent.

Theorem 1 *Consider the proposed DIAG method in* (11)*. If the conditions in Assumption 1 hold, and the stepsize ϵ is chosen as $\epsilon = 2/(\mu + L)$, then the error after m passes over the functions f_i, i.e., $k = mn$ iterations, is bounded above by*

$$\|\mathbf{x}^{mn} - \mathbf{x}^*\| \leq \rho^m \left(1 - \left(\frac{n-1}{n}\right)(1-\rho)\right)\|\mathbf{x}^0 - \mathbf{x}^*\|, \quad (16)$$

where $\rho := (\kappa - 1)/(\kappa + 1)$.

The result in Theorem 1 verifies the advantage of DIAG with respect to GD. The result in (16) shows that the error of DIAG after m passes over the dataset, which is bounded above by $\rho^m(1 - (1-\rho)(n-1)/n)\|\mathbf{x}^0 - \mathbf{x}^*\|$, is strictly smaller than the upper bound for the error of GD after m iterations, which is given by $\rho^m\|\mathbf{x}^0 - \mathbf{x}^*\|$. Hence, the DIAG method outperforms GD for any choice of κ and $n > 1$. Notice that the upper bound for GD is tight, and there exists an optimization problem such that the error of GD satisfies the equality $\|\mathbf{x}^m - \mathbf{x}^*\| = \rho^m\|\mathbf{x}^0 - \mathbf{x}^*\|$.

Although the result in Proposition 1 implies that DIAG is preferable relative to GD, it is cannot show linear convergence of DIAG. To be more precise, the result in Proposition 1 shows that the subsequence of errors $\|\mathbf{x}^{kn} - \mathbf{x}^*\|_{k=1}^\infty$, which are associated with the variables at the end of each pass over the set of functions, is linearly convergent. However, we aim to show that the whole sequence $\|\mathbf{x}^k - \mathbf{x}^*\|$ is linearly convergent. In the following theorem, we show that the sequence of iterates generated by DIAG converges linearly.

Theorem 2 *Consider the proposed DIAG method in* (11)*. Further, recall the definition of the constant $\rho := (\kappa - 1)/(\kappa + 1)$. If the conditions in Assumption 1 hold and $\epsilon = 2/(\mu + L)$, the sequence of iterates $\mathbf{x}^k$ generated by DIAG satisfies*

$$\|\mathbf{x}^k - \mathbf{x}^*\| \leq a_0 \gamma_0^k \|\mathbf{x}^0 - \mathbf{x}^*\|, \quad (17)$$

where γ_0 is the only root of the polynomial equation

$$\gamma^{n+1} - \left(1 + \frac{\rho}{n}\right)\gamma^n + \frac{\rho}{n} = 0 \quad (18)$$

in the interval $[0, 1)$, and a_0 is given by

$$a_0 = \max_{i \in \{1,\dots,n\}} \left(1 - \frac{(i-1)(1-\rho)}{n}\right)\gamma_0^{-i}. \quad (19)$$

The result in Theorem 2 shows that the whole sequence of iterates $\mathbf{x}^k$ generated by DIAG converges linearly to the optimal argument $\mathbf{x}^*$. Note that the polynomial in (18) has only one root in the interval $[0, 1)$. To verify this claim, consider the function $h(\gamma) := \gamma^{n+1} - (1 + \rho/n)\gamma^n + \rho/n$ for $\gamma \in [0, 1)$. The derivative of the function h is given by $(dh/d\gamma) = (n+1)\gamma^n - (n+\rho)\gamma^{n-1}$. Therefore, the only critical point of the function h in the interval $(0, 1)$ is $\gamma^* = (n + \rho)/(n + 1)$. The point γ^* is a local minimum for the function h, since the second derivative of the function h is positive at γ^*. Note that $h(\gamma^*) < 0$, $h(0) > 0$, and $h(1) = 0$. These observation imply that the function h has only one root γ_0 in the interval $[0, 1)$ and this root is between 0 and γ^*.

5. NUMERICAL EXPERIMENTS

In this section, we compare the performances of GD, IAG, and DIAG. First, we apply these methods to solve the quadratic programming

$$\min_{\mathbf{x} \in \mathbb{R}^p} f(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^n \frac{1}{2}\mathbf{x}^T\mathbf{A}_i\mathbf{x} + \mathbf{b}_i^T\mathbf{x}, \quad (20)$$

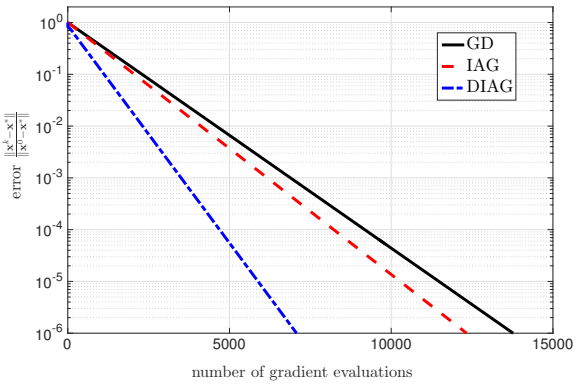

Fig. 1. Convergence paths of GD, IAG, and DIAG for the quadratic programming with $n = 200$ and $\kappa = 10$.

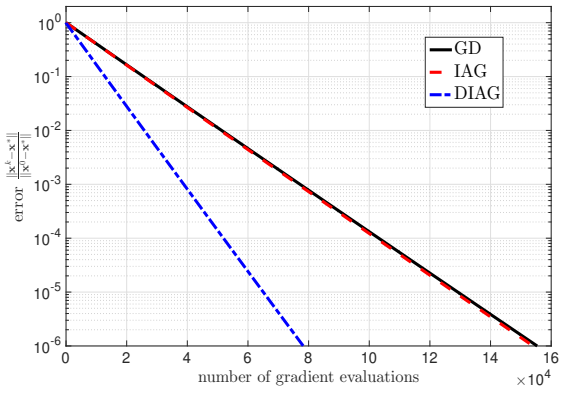

Fig. 2. Convergence paths of GD, IAG, and DIAG for the quadratic programming with $n = 200$ and $\kappa = 117$.

where $\mathbf{A}_i \in \mathbb{R}^{p\times p}$ is a diagonal matrix and $\mathbf{b}_i \in \mathbb{R}^p$ is a random vector chosen from the box $[0, 1]^p$. To control the problem condition number, the first $p/2$ diagonal elements of $\mathbf{A}_i$ are chosen uniformly at random from the interval $[1, 10^1, \ldots, 10^{\eta/2}]$ and its last $p/2$ elements chosen from the interval $[1, 10^{-1}, \ldots, 10^{-\eta/2}]$. This selection resulting in the sum $\sum_{i=1}^n \mathbf{A}_i$ having eigenvalues in the range $[n10^{-\eta/2}, n10^{\eta/2}]$. In our simulations, we fix the variable dimension as $p = 20$ and the number of functions as $n = 200$. Moreover, the stepsizes of GD and DIAG are set as their best theoretical stepsize which are $\epsilon_{GD} = 2/(\mu + L)$ and $\epsilon_{DIAG} = 2/(\mu + L)$, respectively. Note that the stepsize suggested in [16] for IAG is $\epsilon_{IAG} = 0.32/((nL)(L + \mu))$; however, this choice of stepsize is very slow in practice. Thus, we use the stepsize $\epsilon_{IAG} = 2/(nL)$ which performs better than the one suggested in [16].

To have a fair comparison, we compare the algorithms in terms of the total number of gradient evaluations. Note that comparison of these methods in terms of the total number of iterations would not be fair since each iteration of GD requires n gradient evaluations, while IAG and DIAG only require one gradient computation per iteration.

We first consider the case that $\eta = 1$ and use the realization with condition number $\kappa = 10$ to have a relatively small condition number. Fig. 1 demonstrates the convergence paths of the normalized error $\|\mathbf{x}^k - \mathbf{x}^*\|/\|\mathbf{x}^0 - \mathbf{x}^*\|$ for IAG, DIAG, and GD when $n = 200$ and $\kappa = 10$. As we observe, IAG performs better than GD, while the best performance belongs to DIAG. In the second experiment, we increase the problem condition number by setting $\eta = 2$ and using the realization with condition number $\kappa = 117$. Fig. 2 illustrates the performances of

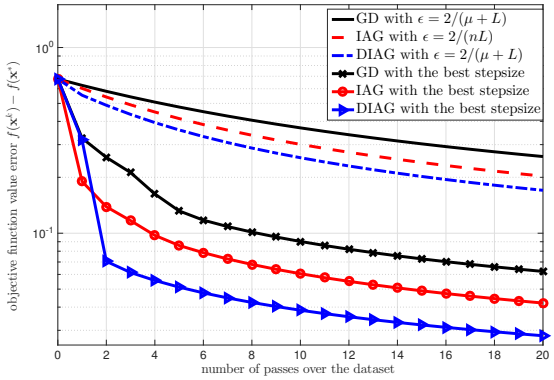

Fig. 3. Convergence paths of GD, IAG, and DIAG for the binary classification application.

these methods for the case that $n = 200$ and $\kappa = 117$. We observe that the convergence path of IAG is almost identical to the one for GD. In this experiment, we also observe that DIAG has the best performance among the three methods. Note that the relative performance of IAG and GD changes for problems with different condition numbers. On the other hand, the relative convergence paths of DIAG and GD does not change in different settings, and DIAG consistently outperforms GD.

We also compare the performances of GD, IAG, and DIAG in solving a binary classification problem. Consider the logistic regression problem where n samples $\{\mathbf{u}_i\}_{i=1}^n$ and their corresponding labels $\{l_i\}_{i=1}^n$ are given. The dimension of samples is p, i.e., $\mathbf{u}_i \in \mathbb{R}^p$, and the labels l_i are either -1 or 1. The goal is to find the optimal classifier $\mathbf{x}^* \in \mathbb{R}^p$ that minimizes the regularized logistic loss which is given by

$$\min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^n \log(1 + \exp(-l_i\mathbf{x}^T\mathbf{u}_i)) + \frac{\lambda}{2}\|\mathbf{x}\|^2. \qquad (21)$$

The objective function f in (21) is strongly convex with constant $\mu = \lambda$ and its gradients are Lipschitz continuous with constant $L = \lambda + \zeta/4$ where $\zeta = \max_i \mathbf{u}_i^T\mathbf{u}_i$. Note that the functions f_i in this case can be defined as $f_i(\mathbf{x}) = \log(1 + \exp(-l_i\mathbf{x}^T\mathbf{u}_i)) + (\lambda/2)\|\mathbf{x}\|^2$. It is easy to verify that the instantaneous functions f_i are also strongly convex with constant $\mu = L$, and their gradients are Lipschitz continuous with constant $L = \lambda + \zeta/4$.

We apply GD, IAG, and DIAG to the logistic regression problem in (21) for the MNIST dataset [19]. We assign label $l_i = 1$ to the samples that correspond to digit 8 and label $l_i = -1$ to those associated with digit 0. We get a total of $n = 11,774$ training examples, each of dimension $p = 784$. The objective function error $f(\mathbf{x}^k) - f(\mathbf{x}^*)$ of the GD, IAG, and DIAG methods versus the number of passes over the dataset are shown in Fig. 3 for the stepsizes $\epsilon_{GD} = 2/(\mu + L)$, $\epsilon_{IAG} = 2/(nL)$, and $\epsilon_{DIAG} = 2/(\mu + L)$. Moreover, we report the convergence paths of these algorithms for their best choice of stepsize in practice. The results verify the advantage of the proposed DIAG method relative to IAG and GD in both scenarios.

6. CONCLUSIONS

In this paper, we proposed a novel cyclic incremental aggregated gradient method (DIAG) for solving the problem of minimizing the average of a set of smooth and strongly convex functions. The proposed method is the first cyclic incremental method that has convergence guarantees better than the gradient descent method. Numerical experiments justify the advantage of the proposed DIAG method relative to gradient descent and other first-order incremental methods.

7. REFERENCES

[1] L. Bottou and Y. Le Cun, "On-line learning for very large data sets," *Applied stochastic models in business and industry*, vol. 21, no. 2, pp. 137–151, 2005.

[2] D. P. Bertsekas, "Incremental least squares methods and the extended kalman filter," *SIAM Journal on Optimization*, vol. 6, no. 3, pp. 807–822, 1996.

[3] S. S. Ram, A. Nedic, and V. Veeravalli, "Stochastic incremental gradient descent for estimation in sensor networks," in *2007 Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers*. IEEE, 2007, pp. 582–586.

[4] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[5] N. L. Roux, M. Schmidt, and F. R. Bach, "A stochastic gradient method with an exponential convergence _rate for finite training sets," in *Advances in Neural Information Processing Systems*, 2012, pp. 2663–2671.

[6] A. Defazio, F. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," in *Advances in Neural Information Processing Systems*, 2014, pp. 1646–1654.

[7] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Advances in Neural Information Processing Systems*, 2013, pp. 315–323.

[8] L. Xiao and T. Zhang, "A proximal stochastic gradient method with progressive variance reduction," *SIAM Journal on Optimization*, vol. 24, no. 4, pp. 2057–2075, 2014.

[9] S. Shalev-Shwartz and T. Zhang, "Stochastic dual coordinate ascent methods for regularized loss," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 567–599, 2013.

[10] ——, "Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization," *Mathematical Programming*, vol. 155, no. 1-2, pp. 105–145, 2016.

[11] L. Zhang, M. Mahdavi, and R. Jin, "Linear convergence with condition number independent access of full gradients," in *Advances in Neural Information Processing Systems*, 2013, pp. 980–988.

[12] J. Konečnỳ and P. Richtárik, "Semi-stochastic gradient descent methods," *arXiv preprint arXiv:1312.1666*, 2013.

[13] J. Mairal, "Incremental majorization-minimization optimization with application to large-scale machine learning," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 829–855, 2015.

[14] D. Blatt, A. O. Hero, and H. Gauchman, "A convergent incremental gradient method with a constant step size," *SIAM Journal on Optimization*, vol. 18, no. 1, pp. 29–51, 2007.

[15] P. Tseng and S. Yun, "Incrementally updated gradient methods for constrained and regularized optimization," *Journal of Optimization Theory and Applications*, vol. 160, no. 3, pp. 832–853, 2014.

[16] M. Gürbüzbalaban, A. Ozdaglar, and P. Parrilo, "On the convergence rate of incremental aggregated gradient algorithms," *arXiv preprint arXiv:1506.02081*, 2015.

[17] A. Mokhtari, M. Gürbüzbalaban, and A. Ribeiro, "On the linear convergence of a cyclic incremental aggregated gradient method," *University of Pennsylvania Technical Report*, 2016. [Online]. Available: https://fling.seas.upenn.edu/~aryanm/wiki/CAG_journal.pdf

[18] Y. Nesterov, *Introductory lectures on convex optimization*. Springer Science & Business Media, 2004, vol. 87.

[19] Y. LeCun, C. Cortes, and C. J. Burges, "The MNIST database of handwritten digits," 1998.