A Simple and Fast Distributed Accelerated Gradient Method

Chhavi Sharma CHHAVISHARMA@IITB.AC.IN
Vishnu Narayanan VISHNU@IITB.AC.IN
P Balamurugan BALAMURUGAN.PALANIAPPAN@IITB.AC.IN
Indian Institute of Technology Bombay

Abstract

Accelerated gradient (AG) methods have been instrumental in significantly improving the training time and efficiency of several machine learning algorithms. In this work, we propose fast distributed accelerated gradient (DAG) method for big-data based convex loss minimization problems in machine learning, particularly in a synchronous distributed setting. We offer a new analysis by characterizing the proposed method as a variant of Nesterov's scheme. We demonstrate using the proposed analysis for the first time, the $o(1/k)$ rate of convergence of iterates of an accelerated gradient method in a distributed setting. We also establish the optimal convergence of objective function values with rate $o(k^{-2})$ using the proposed DAG method.

1. Introduction

In this paper, we consider the task of optimizing a convex objective function of the following form:

$$\min_{x\in\mathbb{R}^n} \mathcal{F}(x) = \sum_{i=1}^{m} f_i(x). \tag{1}$$

We assume that m agents collaborate to solve problem (1) and the constituent functions $f_i : \mathbb{R}^n \to \mathbb{R}$, of $\mathcal{F}$ are convex *cost* functions local to the i-th agent and are not visible to other agents. We assume also that agent i possesses sufficient memory storage and processing capabilities to process f_i. Without additional information other than f_i, it is in general not possible for the m agents to collaboratively solve problem (1). To allow for this flow of additional information among agents, we assume that communication links exist between pairs of agents. Since the f_i functions are assumed to be literally distributed over the m agents, the problem (1) is correspondingly called a *distributed optimization* problem. The description of the distributed optimization problem (1) leads us to a general graph model $G = (V, E)$ where $V = \{1, \dots, m\}$ represents the node-set of graph G corresponding to the set of agent identifiers, and $E \subseteq V \times V$ captures the communication links between the agents in terms of edges of the graph G. In this work, we are interested in designing an efficient algorithm to solve problem (1) in a fully decentralized setting. We list below the main contributions of our work.

Contributions: We design a simple accelerated gradient-based algorithm (where the acceleration is in the sense of generalized Nesterov's acceleration scheme) called Distributed Accelerated Gradient (DAG) method, to solve problem (1) in a distributed manner. We provide a novel analysis of the proposed algorithm based on a Lyapunov energy function approach which provides further insights into the behavior of convergence of function values and iterates. We derive optimal convergence rates for objective function values and iterates generated by DAG. Numerical experiments validating our theoretical guarantees are also illustrated.

Paper organization: In the next section, we introduce the underlying distributed setup and the assumptions to be used for our algorithm design and analysis. We illustrate the proposed DAG method in Section 3. Section 4 provides the results on convergence rates of objective function values and iterates along with a sketch of our convergence analysis. Empirical experiments using the proposed DAG method are illustrated in Section 5.

Notations: $\langle \cdot, \cdot \rangle$ represents the standard inner product in $\mathbb{R}^n$ and the Euclidean norm (respectively Frobenius norm) is denoted by $\|\cdot\|_2$ (respectively $\|\cdot\|_F$). The iterate information in agent i at time k is represented using $x_k^i \in \mathbb{R}^n$. Other notations to be introduced in the paper will be made clear according to the context.

2. Model and Assumptions

We assume a synchronous communication model where the agents communicate with their neighbors only at the ticks of a global clock, and these clock ticks are associated with corresponding discrete time steps $k \in \mathbb{N} = \{1, 2, \ldots\}$. At each time step k, the set of neighbors of each agent can vary according to k; this is captured by the notation $G_k{=}(V, E_k)$ (note however that the node set V is fixed over time). We state below the other standing assumptions to be used throughout the paper:

1. The set of optimal solutions $X^\star = \arg\min_{x \in \mathbb{R}^n} \mathcal{F}(x)$ of problem (1) is non-empty.
2. Each function $f_i : \mathbb{R}^n \longrightarrow \mathbb{R}$ is continuously differentiable, convex and has L-Lipschitz continuous gradients ∇f_i.
3. $\forall i \in \{1, \cdots, m\}$, there exists a constant $B > 0$ such that $\|\nabla f_i(x)\|_2 \leq B, \forall x \in \mathbb{R}^n$.
4. Graph $G_k{=}(V, E_k)$ is undirected and simple (without self-edges) $\forall k \in \mathbb{N}$.
5. $\forall k \in \mathbb{N}$, assume weights $W_{ij}(k) \in [0, 1]$ associated with $(i, j) \in V \times V$. Collecting these weights into a matrix $W(k)$ of size $m \times m$ corresponding to time step k, we assume that $W(k)$ satisfies the following conditions:
   (a) Each weight matrix $W(k)$ is doubly stochastic.
   (b) $W_{ij}(k) > 0$ if and only if $(i, j) \in E_k$ and $W_{ii}(k) > 0$ for all $i \in \{1, \cdots, m\}$.
   (c) There exist positive scalars $C_1{>}0$ and $\gamma{<}1$ such that the following inequality holds (assuming $\mathbf{1}$ to be a column vector of one of size m):

$$\left\| W_{ij}(k) - \frac{1}{m}\mathbf{1}\mathbf{1}^\top \right\| \leq C_1 \gamma^k \;\; \forall\, i, j \in \{1, \cdots, m\}. \tag{2}$$

Assumption 5c plays a crucial role in controlling the consensus error accumulated due to the time-varying topolgy of the network. Many existing distributed accelerated gradient methods, for example [3], [5], [4], [7] require fully-consensus step at each iteration to control the consensus error which gives rise to high communication cost. Assumption 5c takes away the fully-consensus step as demanded by most of the distributed accelerated methods and thus amortized the communication cost incurred at each iteration.

3. Distributed Accelerated Gradient (DAG) algorithm

Our Distributed Accelerated Gradient (DAG) scheme is described in Algorithm 1. The DAG scheme is similar to the algorithm proposed in [3], except that we have a general momentum coefficient and a sequence of connected graphs converging to a fully connected graph.

```
  Algorithm 1: Distributed Accelerated Gradient (DAG) Algorithm (Behavior at node $i$)
Input: Lipschitz parameter $L > 0,\ 0 < s \leq 1/L$, $m \geq 1$, $\alpha > 0, x_0^i = y_0^i = \mathbf{0}$.
for $k = 1, 2, \ldots$ do
    $x_k^i = y_{k-1}^i - s\nabla f_i(y_{k-1}^i)$.
    $v_k^i = \sum_{j=1}^{m} W_{ij}(k) x_k^j$.
    $y_k^i = v_k^i + \frac{k-1}{k+\alpha-1}\left(v_k^i - v_{k-1}^i\right)$.
end
```

In the next section, we give a short overview of our analysis of DAG scheme (done along the lines of the analysis in [1]) and state the major results on the convergence of objective function values and iterates generated by Algorithm 1.

4. Convergence analysis

Our analysis is similar in spirit to the analysis done in [1], however the extension is non-trivial. We define a function $F : \mathbb{R}^{mn} \longrightarrow \mathbb{R}$ such that $F(x) \triangleq \sum_{i=1}^{m} f_i(x^i)$. Using notations $x_k = \left((x_k^1)^\top \ldots (x_k^m)^\top\right)^\top$, $y_k = \left((y_k^1)^\top \ldots (y_k^m)^\top\right)^\top$, $v_k = \left((v_k^1)^\top \ldots (v_k^m)^\top\right)^\top$, we can write the update steps in Algorithm 1 as:

$$x_{k+1} = y_k - s\nabla F(y_k), \tag{3}$$

$$v_{k+1} = (W(k+1) \otimes I_n) x_{k+1}, \tag{4}$$

$$y_{k+1} = v_{k+1} + \frac{k}{k+\alpha}\left(v_{k+1} - v_k\right), \tag{5}$$

where $\otimes$ denotes the Kronecker product and I_n denotes the identity matrix of size n. We define a function $h : \mathbb{R}^{mn} \longrightarrow \mathbb{R}$ and a perturbation term g_k as follows:

$$h(x_k) = \textstyle\sum_{j=1}^{m} f_j(\bar{x}_k), \tag{6}$$

$$g_k \coloneqq \tfrac{1}{s}(W(k+1) \otimes I_n - I_{mn}) y_k - (W(k+1) \otimes I_n)\nabla F(y_k) + \nabla h(y_k). \tag{7}$$

Using assumption 5c, we can prove that $\sum_{k=1}^{\infty} \|g_k\| < \infty$. The summability of the norm of error term g_k leads us to the following Lemma 1.

Lemma 1 *Under Assumptions 1–5, let $\left(x_k^1\right)_{k\in\mathbb{N}}, \cdots, \left(x_k^m\right)_{k\in\mathbb{N}}$ be the sequences generated by Algorithm 1. Let $\tilde{x} \in X^\star$ be an optimal solution of (1). Then,*

$$\mathcal{F}(\bar{x}_k) - \mathcal{F}(\tilde{x}) \leq \frac{C(\alpha-1)}{2s(k+\alpha-2)^2} \text{ for all } \alpha \geq 3, \tag{8}$$

$$\textit{where } C = \frac{\mathcal{G}(0)}{(\alpha-1)} + 2sM\sum_{r=0}^{\infty}(r+\alpha-1)\left\|g_r\right\|, \; M = \frac{\mathcal{G}(0)}{(\alpha-1)} + \sum_{j=1}^{\infty}\|g_j\|.$$

$$\sum_{k=1}^{\infty} k\left(\mathcal{F}(\bar{x}_k) - \mathcal{F}(\tilde{x})\right) < \infty \textit{ for all } \alpha > 3. \tag{9}$$

Using Lemma 1, we are ready to present our main result, stated below in Theorem 2, on the convergence of objective function values and iterates.

Theorem 2 *Suppose that Assumptions 1–5 hold. Let* $\left(v_k^1\right)_{k\in\mathbb{N}}, \cdots, \left(v_k^m\right)_{k\in\mathbb{N}}$ *be the sequences generated by Algorithm 1. Let* $\tilde{x}$ *be an optimal solution of* (1)*. Then, the following hold for all* $\alpha > 3$.

1. $\mathcal{F}(\bar{v}_k) - \mathcal{F}(\tilde{x}) = o(1/k^2)$
2. $\mathcal{F}(v_k^i) - \mathcal{F}(\tilde{x}) = o(1/k^2)\ \forall\, i \in \{1,\ldots,m\}$
3. $\left\|v_{k+1}^i - v_k^i\right\|_2 = o(1/k)\ \forall\, i \in \{1,\ldots,m\}$
4. $\forall i \in \{1,\ldots,m\}$, *the sequence* $(v_k^i)_{k\in\mathbb{N}}$ *converges to an optimal solution of problem* (1).

Attouch et al. ([2], Example 2.13) showed that there does not exist $p > 2$ such that the rate of convergence is $\mathcal{O}(1/k^p)$ for every convex function. Nesterov [8] gave an example of a convex function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ such that $f(x_k) - f(x^\star) \geq \frac{3\mathrm{L}\|x_0 - x^\star\|^2}{32(k+1)^2}$ as long as $k \leq \frac{(n-1)}{2}$ ([6], equation (3.6)) where x_k is a sequence generated by any first order optimization method and $x^\star$ minimizes $f(\cdot)$. So, Theorem 2 does not contradict the optimal convergence rate of first order optimization methods to solve convex program.

5. Numerical Experiments

In this section, we conduct numerical experiments on binary classification problems over a data set $\mathscr{D}$=$\{(a_i,b_i),\}_{i=1}^N$ with $a_i \in \mathbb{R}^n$ and b_i∈$\{+1,-1\}$. We consider a problem of minimizing the regularized logistic regression function $\min_{x\in\mathbb{R}^n}\frac{1}{N}\sum_{i=1}^N \log\left(1+\exp^{-b_i\langle x,a_i\rangle}\right) + \lambda\left\|x\right\|^2$. For our numerical simulations, we use a4a data set and ijcnn1 data set from https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/, containing $N = 4781$, $n = 123$ features and $N = 49,990$, $n = 22$ features respectively. The weight matrices used in the experiments are generated randomly to satisfy assumption 5. In particular, for $k = 1,\ldots,2000$, we generated random graphs with gradually increasing edge-probability from 0.8 to 1, and after $k = 2000$, we considered fully connected graphs. This resembles the setting $\gamma \approx 0.3$. The data set is divided equally among 20 agents. We compare DAG method with three existing methods namely, Distributed gradient method [9], EXTRA method [10] and mD-NG [4]. We used fixed weight matrix generated at 1200 iteration to implement distributed gradient method, EXTRA and m-DNG. All methods used in our comparison were implemented in Python programming language and were run on a machine with Ubuntu Linux (version 18.04.2 LTS), 8 GB RAM and Intel i7-700 (3.6 GHz, 8 core) processor. The convergence of objective function values for the compared methods is presented in Figure 1, where the relative function value differences $\frac{1}{m}\sum_{i=1}^m\left|\frac{\mathcal{F}(v_k^i)-\mathcal{F}(\tilde{x})}{\mathcal{F}(\tilde{x})}\right|$ are plotted, against iterations. To compute $\tilde{x}$, we ran the algorithms for a large number of iterations first until the condition $\max_{i,j\in V}\|x_k^i - x_k^j\|_2 < 10^{-15}$ was met. We illustrate also the convergence behavior of the average residual $\frac{1}{m}\sum_{i=1}^{m}\left\|v_k^i - \tilde{x}\right\|_2^2$ against iterations in the plot of Figure 2. The convergence of the proposed DAG method is better when compared to the other methods. Also, we plot average relative error and residual behavior against iterations for different values of α in Figure 3 and Figure 4. We observe that smaller α gives better performance in the beginning. The reason for this behavior is that small α (less friction) moves the iterates quickly towards the optimal solution. But for large k, the progress becomes slow because less friction moves the iterate away from the optimal solution. Therefore, for large k, larger α (more friction) enhances stable convergence behavior.

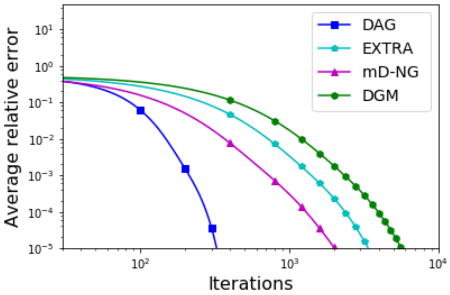

Figure 1: Average relative error convergence ($\lambda = 10^{-2}, \alpha = 20$, ijcnn1 data)

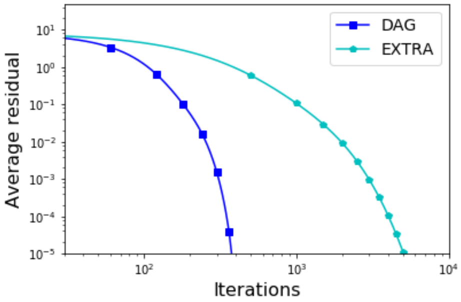

Figure 2: Average residual convergence ($\lambda = 10^{-2}, \alpha = 20$, ijcnn1 data)

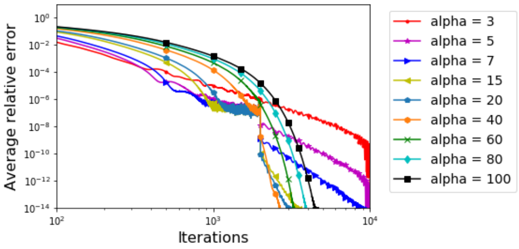

Figure 3: Average relative error behavior with different α values ($\lambda = 10^{-3}$, a4a data)

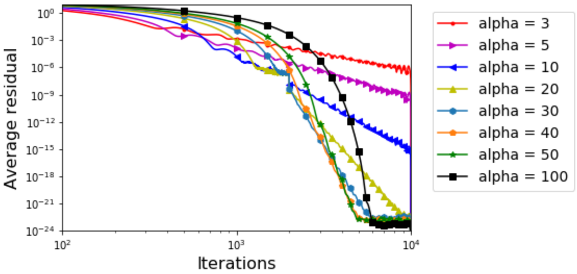

Figure 4: Average residual behavior with different α values ($\lambda = 10^{-3}$, a4a data)

6. Conclusion

In this paper we have proposed a synchronous distributed accelerated gradient (DAG) algorithm. Our novel analysis of the proposed DAG method shows the convergence of objective function values at rate $o(1/k^2)$. Apparently, our work is the first to provide a $o(1/k)$ convergence rate of iterates of an accelerated gradient method in a distributed setting. Despite this, we note that the smoothness assumptions placed on f_i, the asymptotic convergence of graph G_k to a fully connected graph and double stochasticity assumption of weight matrices $W(k)$ are restrictive. Also, the implications of the analysis for strong regularity assumptions on f_i (*e.g.* strong convexity) are not clear.

References

[1] Hedy Attouch and Juan Peypouquet. The rate of convergence of nesterov's accelerated forward-backward method is actually faster than $1/k^2$. *SIAM Journal on Optimization*, 26 (3):1824–1834, 2016.

[2] Hedy Attouch, Zaki Chbani, Juan Peypouquet, and Patrick Redont. Fast convergence of inertial dynamics and algorithms with asymptotic vanishing viscosity. *Mathematical Programming*, 168(1-2):123–175, 2018.

[3] Annie I Chen and Asuman Ozdaglar. A fast distributed proximal-gradient method. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 601–608, 2012.

[4] Dušan Jakovetić, Joao Xavier, and José MF Moura. Distributed nesterov gradient methods for random networks: Convergence in probability and convergence rates. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1508–1511. IEEE, 2014.

[5] Dusan Jakovetic, João Manuel Freitas Xavier, and José M. F. Moura. Fast distributed gradient methods. *IEEE Trans. Automat. Contr.*, 59(5):1131–1146, 2014.

[6] Donghwan Kim and Jeffrey A Fessler. Optimized first-order methods for smooth convex minimization. *Mathematical programming*, 159(1-2):81–107, 2016.

[7] Huan Li, Cong Fang, Wotao Yin, and Zhouchen Lin. A sharp convergence rate analysis for distributed accelerated gradient methods. *arXiv preprint arXiv:1810.01053*, 2018.

[8] Yurii Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Sov. Math. Dokl.*, 27:372–376, 1983.

[9] S. Sundhar Ram, Angelia Nedic, and Venugopal V. Veeravalli. Distributed stochastic subgradient projection algorithms for convex optimization. *J. Optimization Theory and Applications*, 147(3):516–545, 2010.

[10] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM J. on Optimization*, 25(2):944–966, 2015.