

Linear Speedup of Incremental Aggregated Gradient Methods on Streaming Data

Xiaolu Wang¹, Cheng Jin², Hoi-To Wai¹, Yuantao Gu²

Abstract—This paper considers a type of incremental aggregated gradient (IAG) method for large-scale distributed optimization. The IAG method is well suited for the parameter server architecture as the latter can easily aggregate potentially staled gradients contributed by workers. Although the convergence of IAG in the case of deterministic gradient is well known, there are only a few results for the case of its stochastic variant based on streaming data. Considering strongly convex optimization, this paper shows that the streaming IAG method achieves linear speedup when the workers are updating frequently enough, even if the data sample distribution across workers are heterogeneous. We show that the expected squared distance to optimal solution decays at $\mathcal{O}((1+T)/(nt))$, where n is the number of workers, t is the iteration number, and T/n is the update frequency of workers. Our analysis involves careful treatments of the conditional expectations with staled gradients and a recursive system with both delayed and noise terms, which are new to the analysis of IAG-type algorithms. Numerical results are presented to verify our findings.

I. INTRODUCTION

Distributed optimization is an important algorithmic paradigm that has received immense attention due to its wide applicability in machine learning, signal processing, control, etc [1]–[3]. It is suitable for a broad range of circumstances where data are dispersed across multiple entities, e.g., CPU cores, computing clusters, wireless sensors, and wearable devices [1]. Classical distributed optimization deals with the case when each worker holds a fixed set of local data samples that is available at any time, which is also referred to as the *batch data learning* setting. However, with the growing scenarios including federated learning [3] where the data are acquired in an online fashion (e.g., online review and social network platforms) and each data sample is allowed to be used only once [2], it is important to adapt the distributed optimization algorithms to the streaming data setting.

This paper is concerned with the following stochastic optimization problem:

$$\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^n F_i(\mathbf{w}), \quad F_i(\mathbf{w}) := \mathbb{E}_{\xi_i \sim \mathcal{D}_i} [f_i(\mathbf{w}; \xi_i)], \quad (1)$$

where \mathcal{D}_i represents the data distribution supported on the sample space Ξ_i accessible by worker i . The optimization problem shall be solved cooperatively by n workers. For

¹Xiaolu Wang and Hoi-To Wai are with System Engineering & Engineering Management, Faculty of Engineering, The Chinese University of Hong Kong, Hong Kong SAR. Emails: xwangcu@gmail.com, htwai@se.cuhk.edu.hk. ²Cheng Jin and Yuantao Gu are with Department of Electronic Engineering, Tsinghua University, Beijing. Emails: jinc21@mails.tsinghua.edu.cn, gyt@tsinghua.edu.cn. This work is partly supported by CUHK Direct Grant #4055208.

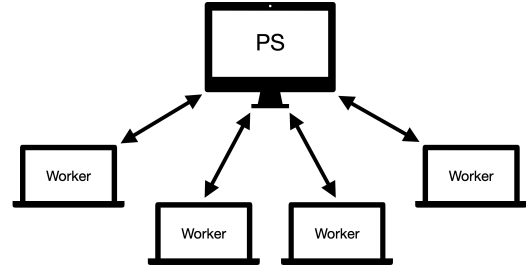


Fig. 1: PS architecture. In this architecture, the PS keeps a buffer of size $n \times d$ that stores the latest gradient computed and sent by the workers and distributes the latest \mathbf{w}^t to the workers.

$i \in [n]$, $\xi_i \in \Xi_i$, the sampled local loss function $f_i(\cdot; \xi_i)$ is continuously differentiable and is known to the i th worker. We denote by $F(\mathbf{w}) := (1/n) \sum_{i=1}^n F_i(\mathbf{w})$ the global objective function and assume that $F(\mathbf{w})$ is strongly convex.

Consider solving (1) in a distributed fashion under the coordination of a central server communicating with the n workers. Each of the worker has access to an independent streaming data source \mathcal{D}_i and the latest iterate \mathbf{w} stored at the server, with which it computes stochastic estimates of the gradient $\nabla F_i(\mathbf{w})$. We concentrate on an asynchronous setting where workers can be idle in some iterations, due to, e.g., network connection failure. Notice that using direct average of the stochastic gradients may result in a non-converging algorithm unless the local loss satisfies some form of similarity conditions; see the related studies on FedAvg in [4], [5]. Remedies such as designing stochastic control variate have been proposed, e.g., [5], [6].

To deal with the worker asynchrony issue over heterogeneous data, this paper utilizes a parameter server (PS) architecture [7], [8] for distributed optimization of (1). In this setting, the PS maintains a buffer that stores gradient information reported from the workers. The coordinating server then aggregates the information stored in the buffer to update its iterate. While performing gradient aggregation seems to alleviate the reliance on data similarity, we note the aggregated information may contain staled gradients due to worker asynchrony that can affect convergence. To this end, this work inquires the following questions:

Does the above distributed algorithm solve (1)? Can it achieve linear speedup in convergence rate compared to centralized/sequential SGD solving (1) taking one sample per iteration?

Our Contributions. In this paper, we provide an affirmative

answer to the above questions. We study a streaming incremental aggregated gradient (sIAG) method adapted from the incremental aggregated gradient (IAG) method [9] to handle streaming data. Our key results are as follows:

- We show that the sIAG method converges in expectation to the global optimum solution of (1) under mild conditions and *without requiring an explicit similarity condition* between $F_i(\mathbf{w})$.
- Suppose that the maximum staleness of the aggregated gradient is T , we show that the expected squared error between the t th iterate and optimal solution to (1) is $\mathcal{O}(\frac{1}{t} \frac{\sigma^2(1+T)}{n})$ for the sIAG method, where σ^2 is the variance of a gradient sample. As such, the sIAG method achieves linear speedup despite that it utilizes staled gradient information with asynchronous workers.

We remark that our analysis utilizes new analysis techniques to yield the tight bound for sIAG; see Sec. III.

Related Works. The studies of IAG method and its variants based on batch data have generated substantial interests after [9]. Notably, [10]–[12] have worked on analyzing the convergence rate of IAG under the bounded delay assumption. Extensions have been considered to speed up convergence, e.g., [13], [14] studied adaptive strategies for gradient aggregation, [15] utilized local Hessian information.

On the other hand, the study of IAG-like methods under the streaming data setting has received attention from the machine learning community. The closest work to ours is [16] that studied an sIAG method with smooth objective function under stronger assumption than ours, e.g., drawing independent samples for staled gradient. As mentioned, federated learning algorithms such as FedAvg [4], FedProx [17], SCAFFOLD [5] adopted similar aggregation technique for the local information reported from the workers.

II. STREAMING IAG METHOD

This section introduces the sIAG method and discusses its implementation in a distributed optimization setting with the PS. To motivate the algorithmic idea of sIAG method, below we first briefly review the IAG method for (1) with batch data [9].

Consider a distributed computing architecture with a PS and n workers [7]; see Fig. 1. The IAG method assumes that each worker $i \in [n]$ has *full access* to an oracle that queries the gradient of its local loss function $F_i(\mathbf{w})$ at any point $\mathbf{w} \in \mathbb{R}^d$, e.g., when the local dataset is fixed. The workers send their computed gradient to the PS that keeps a buffer of n gradient vectors storing the most recent copy of the computed gradient from each worker. Note that during most iterations, this buffer may contain *staled gradient* when the workers is idle at the current iteration.

To fix notations, we let $\mathcal{A}_t \subseteq [n]$ to be set of active workers at iteration t and define

$$\tau_i(t) = \max \{ \tau : \tau \leq t, i \in \mathcal{A}_\tau \}. \quad (2)$$

In other words, $\tau_i(t) \leq t$ indicates the iteration number in which the gradient stored in the PS from worker i

Algorithm 1 sIAG Algorithm

- 1: **input:** Initialization \mathbf{w}^0 , step sizes $\{\eta_t\}_{t \geq 0}$.
 - 2: At the PS, initialize the buffer with $\mathbf{g}_i^{-1} = \mathbf{0}$ for $i = 1, \dots, n$.
 - 3: **for** $t = 0, 1, 2, \dots$ **do**
 - 4: A set of workers $\mathcal{A}_t \subseteq [n]$ is selected/active.
 - 5: **for** each worker $i \in \mathcal{A}_t$ **do**
 - 6: Take \mathbf{w}^t from the PS and draw a sample $\xi_i^t \sim \mathcal{D}_i$.
 - 7: Compute the stochastic gradient $\nabla f_i(\mathbf{w}^t; \xi_i^t)$ and send back to the PS.
 - 8: **end for**
 - 9: Update the buffer as

$$\begin{aligned} \mathbf{g}_i^t &\leftarrow \nabla f_i(\mathbf{w}^t; \xi_i^t) \text{ for } i \in \mathcal{A}_t, \\ \mathbf{g}_i^t &\leftarrow \mathbf{g}_i^{t-1} \text{ for } i \notin \mathcal{A}_t \end{aligned}$$
 - 10: Compute sIAG update: $\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t - (\eta_t/n) \sum_{i=1}^n \mathbf{g}_i^t$.
 - 11: **end for**
-

is computed. In iteration $t \geq 0$, the coordinating server performs the update:

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \frac{\eta}{n} \sum_{i=1}^n \nabla F_i(\mathbf{w}^{\tau_i(t)}), \quad (3)$$

where $\eta > 0$ is the step size. Observe that the server directly aggregates all local (possibly staled) gradients in the PS as the descent direction.

Despite that (3) utilizes some staled gradients in the updates, a key result established in [11], [12] is that IAG admits *linear convergence* towards the optimal solution of (1) when $F(\mathbf{w})$ is strongly convex and smooth, similar to a *centralized* gradient method for (1) under the same setting. Especially, this convergence rate holds regardless of the differences between the local loss functions $F_i(\mathbf{w})$. In comparison, without the aggregation step performed with the PS, the distributed algorithm that relies on aggregating only the current gradients reported by \mathcal{A}_t may converge sublinearly and requires further modification such as taking a diminishing step size [18].

Streaming IAG Method. We are interested in a variant of the IAG method utilizing *streaming data*. We consider the generic form for (1) where each of the local loss function is (possibly) stochastic. Unlike the IAG method, the streaming IAG (sIAG) method considers that each worker only has access to a *stochastic oracle* that queries an unbiased and independent estimate for the gradient of local loss function, denoted by

$$\nabla f_i(\mathbf{w}; \xi_i), \quad \xi_i \sim \mathcal{D}_i. \quad (4)$$

Note that $\mathbb{E}[\nabla f_i(\mathbf{w}; \xi_i)] = \nabla F_i(\mathbf{w})$. The sIAG method reads

$$\mathbf{w}^{t+1} = \mathbf{w}^t - (\eta_t/n) \mathbf{g}^t, \quad (5)$$

for any $t \geq 0$, where $\eta_t > 0$ is a (possibly time varying) step

size and

$$\mathbf{g}^t := \sum_{i=1}^n \nabla f_i(\mathbf{w}^{\tau_i(t)}; \boldsymbol{\xi}_i^{\tau_i(t)}) \quad (6)$$

denotes the aggregated *stochastic gradients*. Note that in the above, we adopted the same notations as in the IAG method where the index $\tau_i(t)$ was defined in (2).

The sIAG method (5), (6) is motivated by an instantaneous gradient computation model that can be readily implemented on the parameter server architecture; as summarized in Algorithm 1. Observe that the algorithm requires active workers to return the stochastic gradient before the current iteration concludes. This restriction, while mild as only stochastic gradients are required, might be relaxed by allowing further delays between the sampled gradient and the iterate in which it is computed. However, in the interest of space, we focus on a current simplified setting.

We recall that as $\tau_i(t) = t$ for $i \in \mathcal{A}_t$ and $\tau_i(t) < t$ for $i \notin \mathcal{A}_t$, we may express the aggregated gradient (6) alternatively as an incremental update via

$$\mathbf{g}^t = \mathbf{g}^{t-1} - \sum_{i \in \mathcal{A}_t} \nabla f_i(\mathbf{w}^{\tau_i(t-1)}; \boldsymbol{\xi}_i^{\tau_i(t-1)}) + \sum_{i \in \mathcal{A}_t} \nabla f_i(\mathbf{w}^t; \boldsymbol{\xi}_i^t).$$

We remark that the above update recursion for \mathbf{g}^t is related to the famous SAG method [19]. It is reduced to the SAG method when $f_i \equiv f$, $\boldsymbol{\xi}_i^t = \boldsymbol{\xi}^t$, and $\mathcal{D}_i \equiv \mathcal{D}$ has a finite support for all $i \in [n]$.

While convergence guarantees for the IAG method (3) are well known, the stochastic variant, sIAG, considered in (5), (6) has received less attention. As the first step towards understanding the behavior of sIAG, the next section analyzes the convergence rate of sIAG under the standard setting when $F(\mathbf{w})$ is strongly convex and smooth.

III. CONVERGENCE ANALYSIS

Missing proofs in this section can be found in the online appendix¹. We shall begin by stating some assumptions on (1) and the sIAG method that are necessary for our analysis. First,

Assumption 1. *Problem (1) satisfies: (i) $F(\mathbf{w})$ is μ -strongly convex with $\mu > 0$; (ii) for any $i \in [n]$, the gradient $\nabla f_i(\mathbf{w})$ is L -Lipschitz continuous.*

The above specifies the function class of interest for (1). We define $\mathbf{w}^* = \arg \min_{\mathbf{w} \in \mathbb{R}^d} F(\mathbf{w})$. We also assume:

Assumption 2. *There exists a constant $\sigma \geq 0$ such that*

$$\mathbb{E} \|\nabla f_i(\mathbf{w}; \boldsymbol{\xi}_i) - \nabla F_i(\mathbf{w})\|_2^2 \leq \sigma^2(1 + \|\mathbf{w} - \mathbf{w}^*\|^2), \quad (7)$$

for all $i \in [n]$, $\mathbf{w} \in \mathbb{R}^d$.

Assumption 3. *There exists $T \geq 0$ such that*

$$\tau_i(t) \geq t - T, \quad \forall i \in [n], t \geq 0. \quad (8)$$

The above assumptions state that the stochastic gradients computed in sIAG has bounded variance, and the staled

gradient delay is bounded by T . Notice that n/T corresponds roughly to the number of workers that are active at any iteration.

We define respectively the following notations for the t th suboptimality gap and its delayed version:

$$\mathcal{E}_t := \mathbb{E} \|\mathbf{w}^t - \mathbf{w}^*\|_2^2, \quad \mathcal{E}_t^{\max} := \max_{s \in [(t-2T)_+, t]} \mathcal{E}_s, \quad (9)$$

where $x_+ := \max\{x, 0\}$ for $x \in \mathbb{R}$. Furthermore, it is instrumental to define the following filtration:

$$\mathcal{F}_t := \sigma(\boldsymbol{\xi}_i^s, i \in [n], s = 0, \dots, t-1), \quad (10)$$

where $\sigma(\cdot)$ denotes the sigma algebra generated by the random variables in the operand. Observe that for any $t \geq 0$, \mathbf{w}^t is measurable with respect to (w.r.t.) \mathcal{F}_t . We shall use the shorthand notation $\mathbb{E}_t[\cdot] := \mathbb{E}[\cdot | \mathcal{F}_t]$ for conditional expectation.

From (5), we deduce that for any $t \geq 0$,

$$\mathcal{E}_{t+1} = \mathcal{E}_t - 2\eta_t \mathbb{E} \left[\left\langle \mathbf{w}^t - \mathbf{w}^*, \frac{1}{n} \mathbf{g}^t \right\rangle \right] + \eta_t^2 \mathbb{E} \left[\left\| \frac{1}{n} \mathbf{g}^t \right\|^2 \right]. \quad (11)$$

We shall control the last two terms in the above decomposition. Observe the following lemmas:

Lemma 1. *Suppose that Assumptions 1–3 hold. Then,*

$$\mathbb{E} \|(1/n) \mathbf{g}^t\|_2^2 \leq \frac{2\sigma^2}{n} + C_L \max_{s \in [(t-T)_+, t]} \mathcal{E}_s, \quad (12)$$

for any $t \geq 0$, where $C_L := 20L^2 + \frac{2\sigma^2}{n}$.

The above is a natural consequence of (6), where the bound (12) depends on both the stochastic gradient variance σ^2 and the delayed optimality gap $\max_{s \in [(t-T)_+, t]} \mathcal{E}_s \leq \mathcal{E}_t^{\max}$.

The next lemma controls the inner product term in (11):

Lemma 2. *Suppose that Assumptions 1–3 hold. Then*

$$\begin{aligned} & \mathbb{E} \langle \mathbf{w}^t - \mathbf{w}^*, (1/n) \mathbf{g}^t \rangle \\ & \geq \frac{\mu}{4} \mathcal{E}_t - \left[C_L T \eta_{t-T} + \left(\frac{\mu}{4} + \frac{5L^2}{2\mu} \right) C_L T^2 \eta_{t-T}^2 \right] \mathcal{E}_t^{\max} \\ & \quad - \left[2T \eta_{t-T} + \left(\frac{\mu}{2} + \frac{5L^2}{\mu} \right) T^2 \eta_{t-T}^2 \right] \frac{\sigma^2}{n}, \quad \forall t \geq 0. \end{aligned} \quad (13)$$

The above lemma shows that the inner product term is *lower bounded* by $(\mu/4)\mathcal{E}_t$ with (negative) perturbation terms that are controllable by the step sizes.

Before substituting (12), (13) into (11) to derive the convergence rate of sIAG method, we highlight that Lemma 2 deviates from the standard analysis for SGD with strongly convex objective function. In the case of standard SGD where $\mathbf{g}^t = \nabla f(\mathbf{w}^t; \boldsymbol{\xi}^t)$ with $\mathbb{E}_t[\mathbf{g}^t] = \nabla F(\mathbf{w}^t)$, the law of total expectation and the strong convexity of F imply that

$$\begin{aligned} & \mathbb{E} \langle \mathbf{w}^t - \mathbf{w}^*, \mathbf{g}^t \rangle = \mathbb{E} \langle \mathbf{w}^t - \mathbf{w}^*, \mathbb{E}_t[\mathbf{g}^t] \rangle \\ & = \mathbb{E} \langle \mathbf{w}^t - \mathbf{w}^*, \nabla F(\mathbf{w}^t) \rangle \geq \mu \mathcal{E}_t. \end{aligned}$$

However, in the sIAG method, \mathbf{g}^t is a function of $\boldsymbol{\xi}_i^s$ for $i \in [n]$ and $s = 0, 1, \dots, t$. Note that for any $i \notin \mathcal{A}_t$, $\boldsymbol{\xi}_i^{\tau_i(t)}$ is \mathcal{F}_{t-1} -measurable and thus $\mathbb{E}_t[\nabla f_i(\mathbf{w}_i^{\tau_i(t)}; \boldsymbol{\xi}_i^{\tau_i(t)})] \neq \nabla F_i(\mathbf{w}^{\tau_i(t)})$.

¹<https://www1.se.cuhk.edu.hk/~htwai/pdf/sIAG-cdc23.pdf>

This implies that \mathbf{g}^t is *not* independent of \mathcal{F}_t and thus

$$\mathbb{E} \langle \mathbf{w}^t - \mathbf{w}^*, \mathbf{g}^t \rangle \neq \mathbb{E} \left\langle \mathbf{w}^t - \mathbf{w}^*, \sum_{i=1}^n \nabla F_i(\mathbf{w}^{\tau_i(t)}) \right\rangle.$$

Our remedy is to consider² the following decomposition for the inner product:

$$\begin{aligned} & \langle \mathbf{w}^t - \mathbf{w}^*, \nabla f_i(\mathbf{w}^{\tau_i(t)}; \boldsymbol{\xi}_i^{\tau_i(t)}) \rangle \\ &= \langle \mathbf{w}^t - \mathbf{w}^{\tau_i(t)} + \mathbf{w}^{\tau_i(t)} - \mathbf{w}^*, \nabla f_i(\mathbf{w}^{\tau_i(t)}; \boldsymbol{\xi}_i^{\tau_i(t)}) \rangle. \end{aligned} \quad (14)$$

Observe that $\nabla f_i(\mathbf{w}^{\tau_i(t)}; \boldsymbol{\xi}_i^{\tau_i(t)})$ is independent of $\mathbf{w}^{\tau_i(t)}$ and thus the simplification $\mathbb{E}_{\tau_i(t)}[\langle \mathbf{w}^{\tau_i(t)} - \mathbf{w}^*, \nabla f_i(\mathbf{w}^{\tau_i(t)}; \boldsymbol{\xi}_i^{\tau_i(t)}) \rangle] = \langle \mathbf{w}^{\tau_i(t)} - \mathbf{w}^*, \nabla F_i(\mathbf{w}^{\tau_i(t)}) \rangle$.

Furthermore, we develop the following bound to control the size of difference $\|\mathbf{w}^t - \mathbf{w}^{\tau_i(t)}\|_2^2$:

Lemma 3. *Suppose that Assumptions 1 and 2 holds and $\{\eta_t\}_{t \geq 0}$ is a monotonically non-increasing sequence. Then, it holds for all $i \in [n]$ and $t \geq 0$ that*

$$\mathbb{E}[\|\mathbf{w}^t - \mathbf{w}^{\tau_i(t)}\|_2^2] \leq T^2 \eta_{t-T}^2 \left(\frac{2\sigma^2}{n} + C_L \mathcal{E}_t^{\max} \right). \quad (15)$$

Observe that the bound is composed of the stochastic gradient's variance and the (delayed) optimality gap. Note that the proof is obtained as a consequence of Lemma 1.

Importantly, (15) allows us to handle the term $\langle \mathbf{w}^t - \mathbf{w}^{\tau_i(t)}, \nabla f_i(\mathbf{w}^{\tau_i(t)}; \boldsymbol{\xi}_i^{\tau_i(t)}) \rangle$ through applying Cauchy-Schwarz inequality.

Substituting Lemmas 1 and 2 into (11) directly yields the following recursive system: for any $t \geq 0$,

$$\begin{aligned} \mathcal{E}_{t+1} &\leq (1 - (\mu/2) \eta_t) \mathcal{E}_t \\ &+ \left[1 + 2T + \left(\frac{\mu}{2} + \frac{5L^2}{\mu} \right) T^2 \eta_{t-T} \right] C_L \eta_{t-T}^2 \mathcal{E}_t^{\max} \\ &+ \left[1 + 2T + \left(\frac{\mu}{2} + \frac{5L^2}{\mu} \right) T^2 \eta_{t-T} \right] \eta_{t-T}^2 \frac{2\sigma^2}{n}. \end{aligned} \quad (16)$$

If we ignore the last term proportional to σ^2/n , then (16) reduces into a contracting recursion with delays that has been studied by [10]. Subsequently, \mathcal{E}_t converges to zero at a linear rate when a constant step size is used.

The introduction of the noise-related terms in (16) has led to a new recursive system with delayed terms that have not been covered in prior works. As an attempt to derive a tight bound, we fix the step size as

$$\eta_t = \beta / (t + \gamma) \quad (17)$$

for some $\gamma, \beta > 0$ and obtain the following convergence rates for \mathcal{E}_t using induction:

²We remark that [20] considered an algorithm that involve similar staled aggregation property to sIAG but have employed a simplifying assumption that \mathbf{w}^t is independent of \mathbf{g}^t . Leveraging this property, their algorithm achieves linear speedup regardless of T . We conjecture that such linear speedup cannot be obtained when considering the realistic conditions for sIAG method that \mathbf{w}^t is not independent of \mathbf{g}^t .

Theorem 1. *Suppose that $\beta > 2/\mu$ and*

$$\gamma \geq 2T + \max \left\{ \frac{16C_L \beta^2}{\mu\beta - 2} \bar{\rho}(T), \sqrt{\frac{8C_L \beta^2}{\mu\beta - 4} \bar{\rho}(T)} \right\},$$

where $\bar{\rho}(T) := 1 + 2T + \left(\frac{\mu}{2} + \frac{5L^2}{\mu} \right) \beta T$. Let

$$\delta_1 := \frac{32\beta^2 \bar{\rho}(T)}{\mu\beta - 2} + 1, \quad \delta_2 = \gamma^2 \mathcal{E}_0.$$

Then, it holds for all $t \geq 0$ that

$$\mathcal{E}_t \leq \frac{\delta_1}{\gamma + t} \frac{\sigma^2}{n} + \frac{\delta_2}{(\gamma + t)^2}. \quad (18)$$

The detailed proof can be found in Appendix A. Theorem 1 indicates that the expected squared distance to optimal solution decays at $\mathcal{O}((1+T)/(nt))$. We note that in the interest of space, the constants in the bound are not fully optimized. In general, obtaining tight bounds for recursive system of the form (16) is an interesting open problem of independent interest.

The bound (18) shows that the sIAG method converges (in expectation) towards the optimal solution of (1) at the rate of $\mathcal{O}\left(\frac{1}{t} \cdot \frac{(1+T)\sigma^2}{n}\right)$. We notice that: (A) the sublinear rate of $\mathcal{O}(1/t)$ is similar to that of existing analysis with stochastic gradient methods [21] and is in the same order of the minimax lower bound [22], (B) the constant factor $(1+T)\sigma^2/n$ indicates that *linear speedup* can be achieved when the delay satisfies $T = \mathcal{O}(1)$. This rate is reasonable since the number of samples taken per iteration is approximately n/T , the linear speedup ratio should be of the same order. We note that similar slow down due to the delay T is also reported in the analysis for IAG [11], [12].

IV. NUMERICAL SIMULATIONS

We evaluate the empirical performance of the sIAG algorithm on synthetic data. We independently generate n parameters $\mathbf{w}_1^*, \mathbf{w}_2^*, \dots, \mathbf{w}_n^*$ according to the uniform distribution on $[0, 1]^d$. Each data point sampled by worker i takes the form $\boldsymbol{\xi}_i = (\mathbf{A}_i, \mathbf{y}_i)$, where $\mathbf{A}_i \in \mathbb{R}^{p \times d}$ and $\mathbf{y}_i \in \mathbb{R}^p$. The entries of \mathbf{A}_i are independent and follow the Gaussian distribution $\mathcal{N}(0, 1)$ and $\mathbf{y}_i \sim \mathcal{N}(\mathbf{A}_i \mathbf{w}_i^*, \sigma^2 \mathbf{I}_p)$. The loss functions are defined as $f_i(\mathbf{w}; \mathbf{A}_i, \mathbf{y}_i) = \frac{1}{2} \|\mathbf{A}_i \mathbf{w} - \mathbf{y}_i\|_2^2$ for $i \in [n]$. If $[\mathbf{A}_1; \dots; \mathbf{A}_n]$ is full-rank (which holds almost surely when $nd \geq p$), it is obvious that $\mathbf{w}^* = \sum_{i=1}^n \mathbf{w}_i^*/n$.

We compare sIAG with *non-aggregated SGD*, which uses the following descent direction at the t th iteration:

$$\mathbf{g}_{\text{SGD}}^t := \sum_{i: \tau_i(t)=t} \nabla f(\mathbf{w}^{\tau_i(t)}; \boldsymbol{\xi}_i^{\tau_i(t)}). \quad (19)$$

We simulate three types of worker selection schemes. The first selection scheme chooses one workers at each iteration cyclically, i.e., at iteration t , the $(t \bmod n + 1)$ th worker is active. Notice that $T = n$ in this case and there is no linear speedup according to Theorem 1. The second selection scheme chooses the workers *uniformly at random*. It models the scenario when the workers are equally efficient. Further, the workers will be selected at least once in no more than 15

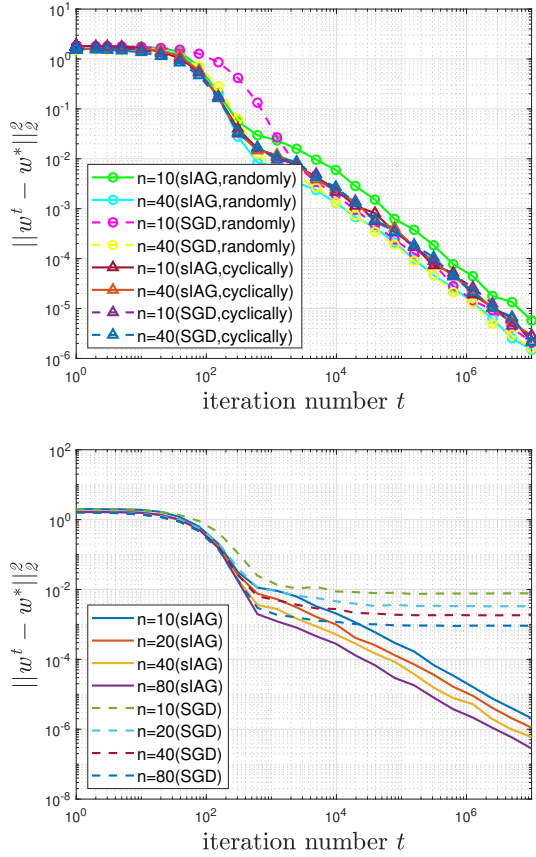


Fig. 2: Convergence of sIAG and SGD with: (Top) uniform worker selection; (Bottom) non-uniform worker selection.

iterations. The third selection scheme adopts *non-uniform selection*. It models a more realistic scenario when the workers are heterogeneous in terms of efficiency. Specifically, worker i is selected at least once in no more than T_i iterations, where T_i is uniformly distributed on $\{10, \dots, 20\}$. The faster workers are selected more frequently. Notice that $T = 15$ in the second and third scheme.

Fig. 2 presents the convergence of two algorithms with $d = 20$, $p = 10$, and $\sigma = 0.1$ under the uniform and non-uniform selection schemes. Observe that under the uniform selection scheme, the sIAG and SGD achieve comparable convergence performance and both exhibit linear speedup as the number of agents increases. We also observe that there is no linear speedup with the cyclical selection scheme. On the other hand, under the non-uniform worker selection scheme, the sIAG still enjoys linear speedup while SGD is not converging to an optimal solution, as the non-uniform selection scheme has led to biased stochastic gradient. This validates the necessity of aggregating (possibly) staled gradients in (5) as opposed to using only the latest gradients in (19) in the presence of system heterogeneity.

V. CONCLUSION

We proposed the sIAG algorithm for distributed optimization over the parameter server architecture with heterogeneous streaming data. The sIAG method is adapted from the

classical IAG method on batch data. We established that sIAG achieves linear speedup compared to the sequential SGD for strongly convex problems. Our analysis relies on careful treatments of the conditional expectations with staled gradients (see (14)) and a new recursive system with both delayed and noise-related terms (see (16)), which can be of independent interest. Numerical results on synthetic data verify our theoretical findings and show significant advantages of sIAG over the non-aggregated SGD method when the workers are not uniformly selected.

REFERENCES

- [1] T. Yang, X. Yi, J. Wu, Y. Yuan, D. Wu, Z. Meng, Y. Hong, H. Wang, Z. Lin, and K. H. Johansson, "A survey of distributed optimization," *Annual Reviews in Control*, vol. 47, pp. 278–305, 2019.
- [2] T.-H. Chang, M. Hong, H.-T. Wai, X. Zhang, and S. Lu, "Distributed learning in the nonconvex world: From batch data to streaming and beyond," *IEEE Signal Processing Magazine*, 2020.
- [3] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.
- [4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282, PMLR, 2017.
- [5] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *Proceedings of the 37th International Conference on Machine Learning*, pp. 5132–5143, PMLR, 2020.
- [6] K. Mishchenko, G. Malinovsky, S. Stich, and P. Richtárik, "ProxSkip: Yes! Local gradient steps provably lead to communication acceleration! Finally!," in *Proceedings of the 39th International Conference on Machine Learning*, pp. 15750–15769, PMLR, 2022.
- [7] M. Assran, A. Aytakin, H. R. Feyzmahdavian, M. Johansson, and M. G. Rabbat, "Advances in asynchronous parallel and distributed optimization," *Proceedings of the IEEE*, vol. 108, no. 11, pp. 2013–2031, 2020.
- [8] A. Aytakin, H. R. Feyzmahdavian, and M. Johansson, "Analysis and implementation of an asynchronous optimization algorithm for the parameter server," *arXiv preprint arXiv:1610.05507*, 2016.
- [9] 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.
- [10] H. R. Feyzmahdavian, A. Aytakin, and M. Johansson, "A delayed proximal gradient method with linear convergence rate," in *Proceedings of 2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, IEEE, 2014.
- [11] M. Gurbuzbalaban, A. Ozdaglar, and P. A. Parrilo, "On the convergence rate of incremental aggregated gradient algorithms," *SIAM Journal on Optimization*, vol. 27, no. 2, pp. 1035–1048, 2017.
- [12] N. D. Vanli, M. Gurbuzbalaban, and A. Ozdaglar, "Global convergence rate of proximal incremental aggregated gradient methods," *SIAM Journal on Optimization*, vol. 28, no. 2, pp. 1282–1300, 2018.
- [13] T. Chen, G. Giannakis, T. Sun, and W. Yin, "LAG: Lazily aggregated gradient for communication-efficient distributed learning," *Advances in Neural Information Processing Systems* 31, 2018.
- [14] X. Wu, S. Magnusson, H. R. Feyzmahdavian, and M. Johansson, "Delay-adaptive step-sizes for asynchronous learning," in *Proceedings of the 39th International Conference on Machine Learning*, pp. 24093–24113, PMLR, 2022.
- [15] H.-T. Wai, W. Shi, C. A. Uribe, A. Nedić, and A. Scaglione, "Accelerating incremental gradient optimization with curvature information," *Computational Optimization and Applications*, vol. 76, no. 2, pp. 347–380, 2020.
- [16] X. Lian, Y. Huang, Y. Li, and J. Liu, "Asynchronous parallel stochastic gradient for nonconvex optimization," *Advances in Neural Information Processing Systems* 28, 2015.
- [17] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine Learning and Systems* 2, pp. 429–450, 2020.

- [18] D. P. Bertsekas *et al.*, "Incremental gradient, subgradient, and proximal methods for convex optimization: A survey," *Optimization for Machine Learning*, vol. 2010, no. 1-38, p. 3, 2011.
- [19] M. Schmidt, N. Le Roux, and F. Bach, "Minimizing finite sums with the stochastic average gradient," *Mathematical Programming*, vol. 162, pp. 83–112, 2017.
- [20] X. Lian, W. Zhang, C. Zhang, and J. Liu, "Asynchronous decentralized parallel stochastic gradient descent," in *Proceedings of 35th International Conference on Machine Learning*, pp. 3043–3052, PMLR, 2018.
- [21] E. Moulines and F. Bach, "Non-asymptotic analysis of stochastic approximation algorithms for machine learning," *Advances in Neural Information Processing Systems 24*, 2011.
- [22] A. Agarwal, P. L. Bartlett, P. Ravikumar, and M. J. Wainwright, "Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3235–3249, 2012.

APPENDIX

To simplify notations, throughout this appendix, we denote $\mathbf{g}_i^t := \nabla f_i(\mathbf{w}^t; \xi_i^t)$ and $\mathbf{G}_i^t := \nabla F_i(\mathbf{w}^t)$ as respectively the stochastic gradient and exact gradient of the local loss function for any $i \in [n]$ and $t \geq 0$.

A. Proof of Theorem 1

Let $\rho_t := 1 + 2T + \left(\frac{\mu}{2} + \frac{5L^2}{\mu}\right) T^2 \eta_{t-T}$. Since $\gamma \geq 2T$, we have $\eta_{t-T} = \beta/(\gamma + t - T) \leq \beta/T$, which further implies that

$$\rho_t \leq 1 + 2T + \left(\frac{\mu}{2} + \frac{5L^2}{\mu}\right) \beta T = \bar{\rho}(T).$$

Thus, the recurrence relation (16) implies that

$$\begin{aligned} \mathcal{E}_{t+1} &\leq \left(1 - \frac{\mu}{2} \eta_t\right) \mathcal{E}_t + \bar{\rho}(T) C_L \eta_{t-T}^2 \mathcal{E}_t^{\max} \\ &\quad + \bar{\rho}(T) \eta_{t-T}^2 \frac{2\sigma^2}{n}. \end{aligned} \quad (20)$$

Then, we prove (18) by induction.

i) Base case: Since $\delta_2 = \gamma^2 \mathcal{E}_0$, we have

$$\mathcal{E}_0 \leq \frac{\delta_1}{\gamma} \frac{\sigma^2}{n} + \frac{\delta_2}{\gamma^2}.$$

ii) Induction step: Suppose that for some $t \geq 0$, it holds that

$$\mathcal{E}_s \leq \frac{\delta_1}{\gamma + s} \frac{\sigma^2}{n} + \frac{\delta_2}{(\gamma + t)^2}, \quad s = 0, \dots, t,$$

which implies that

$$\mathcal{E}_t^{\max} = \max_{s \in [(t-2T)_+, t]} \mathcal{E}_s \leq \frac{\delta_1}{\gamma + t - 2T} \frac{\sigma^2}{n} + \frac{\delta_2}{(\gamma + t - 2T)^2}.$$

Together with (20), this implies that

$$\begin{aligned} \mathcal{E}_{t+1} &\leq \left(1 - \frac{\mu\beta/2}{\gamma + t}\right) \frac{\delta_1}{\gamma + t} \frac{\sigma^2}{n} \\ &\quad + \frac{\bar{\rho}(T) C_L \beta^2 \delta_1}{(\gamma + t - T)^2 (\gamma + t - 2T)} \frac{\sigma^2}{n} + \frac{2\bar{\rho}(T) \beta^2}{(\gamma + t - T)^2} \frac{\sigma^2}{n} \\ &\quad + \left(1 - \frac{\mu\beta/2}{\gamma + t}\right) \frac{\delta_2}{(\gamma + t)^2} + \frac{C_L \beta^2 \bar{\rho}(T) \delta_2}{(\gamma + t - T)^2 (\gamma + t - 2T)^2}. \end{aligned} \quad (21)$$

We note that

$$\left(1 - \frac{\mu\beta/2}{\gamma + t}\right) \frac{1}{\gamma + t} = \frac{\gamma + t - 1}{(\gamma + t)^2} - \frac{\mu\beta/2 - 1}{(\gamma + t)^2}$$

$$\leq \frac{1}{\gamma + t + 1} - \frac{\mu\beta/2 - 1}{(\gamma + t)^2}, \quad (22)$$

where the inequality holds because

$$\frac{\gamma + t - 1}{(\gamma + t)^2} \leq \frac{1}{\gamma + t + 1} \Leftrightarrow (\gamma + t)^2 \geq (\gamma + t)^2 - 1.$$

We also note that

$$\begin{aligned} \left(1 - \frac{\mu\beta/2}{\gamma + t}\right) \frac{1}{(\gamma + t)^2} &= \frac{\gamma + t - 2}{(\gamma + t)^3} - \frac{\mu\beta/2 - 2}{(\gamma + t)^3} \\ &\leq \frac{1}{(\gamma + t + 1)^2} - \frac{\mu\beta/2 - 2}{(\gamma + t)^3}, \end{aligned} \quad (23)$$

where the inequality holds because

$$\frac{\gamma + t - 2}{(\gamma + t)^3} \leq \frac{1}{(\gamma + t + 1)^2} \Leftrightarrow -3(\gamma + t) \leq 2.$$

Thus, Plugging (22) and (23) into (21) gives

$$\begin{aligned} \mathcal{E}_{t+1} &\leq \frac{\sigma^2}{n} \left[\frac{\delta_1}{\gamma + t + 1} - \frac{(\mu\beta/2 - 1)\delta_1}{(\gamma + t)^2} \right. \\ &\quad \left. + \frac{C_L \beta^2 \bar{\rho}(T) \delta_1}{(\gamma + t - T)^2 (\gamma + t - 2T)} + \frac{2\beta^2 \bar{\rho}(T)}{(\gamma + t - T)^2} \right] \\ &\quad + \frac{\delta_2}{(\gamma + t + 1)^2} - \frac{(\mu\beta/2 - 2)\delta_2}{(\gamma + t)^2} + \frac{C_L \beta^2 \bar{\rho}(T) \delta_2}{(\gamma + t - T)^2 (\gamma + t - 2T)^2}. \end{aligned} \quad (24)$$

Moreover, $\gamma \geq 2T$ implies that

$$\frac{1}{(\gamma + t - T)^2} \leq \frac{4}{(\gamma + t)^2}. \quad (25)$$

Since $\beta > 2/\mu$ and $\gamma \geq 2T + \frac{16C_L \beta^2}{\mu\beta - 2} \bar{\rho}(T)$, we have

$$\frac{C_L \beta^2 \bar{\rho}(T) \delta_1}{\gamma + t - 2T} \leq \left(\frac{\mu\beta}{2} - 1\right) \frac{\delta_1}{8}. \quad (26)$$

Besides, it follows from $\delta_1 = 32\beta^2 \bar{\rho}(T)/(\mu\beta - 2)$ that

$$2\beta^2 \bar{\rho}(T) = \left(\frac{\mu\beta}{2} - 1\right) \frac{\delta_1}{8}. \quad (27)$$

Combing (25), (26), and (27) gives

$$\frac{C_L \beta^2 \bar{\rho}(T) \delta_1}{(\gamma + t - T)^2 (\gamma + t - 2T)} + \frac{2\beta^2 \bar{\rho}(T)}{(\gamma + t - T)^2} \leq \frac{(\mu\beta/2 - 1)\delta_1}{(\gamma + t)^2}. \quad (28)$$

Since $\beta > 4/\mu$ and $\gamma \geq 2T + \sqrt{\frac{8C_L \beta^2}{\mu\beta - 4} \bar{\rho}(T)}$, we have

$$\frac{C_L \beta^2 \bar{\rho}(T)}{(\gamma + t - 2T)^2} \leq \frac{1}{4} \left(\frac{\mu\beta}{2} - 2\right). \quad (29)$$

Combining (25) and (29) gives

$$\frac{C_L \beta^2 \bar{\rho}(T) \delta_2}{(\gamma + t - T)^2 (\gamma + t - 2T)^2} \leq \frac{(\mu\beta/2 - 2)\delta_2}{(\gamma + t)^2}. \quad (30)$$

Plugging (28) and (30) back into (24) yields

$$\mathcal{E}_{t+1} \leq \frac{\delta_1}{\gamma + t + 1} \frac{\sigma^2}{n} + \frac{\delta_2}{(\gamma + t + 1)^2}, \quad (31)$$

which completes the proof of the induction step.