On Convex Stochastic Variance Reduced Gradient for Adversarial Machine Learning

Saikiran Bulusu
EECS Department
Syracuse University
Syracuse, USA
sabulusu@syr.edu

Qunwei Li
EECS Department
Syracuse University
Syracuse, USA
qli33@syr.edu

Pramod K. Varshney
EECS Department
Syracuse University
Syracuse, USA
varshney@syr.edu

***Abstract*—We study the finite-sum problem in an adversarial setting using stochastic variance reduced gradient (SVRG) optimization in a distributed setting. Here, a fraction of the workers are assumed to be Byzantine that exhibit adversarial behavior by providing arbitrary data. We propose a robust scheme to combat the actions of Byzantine adversaries in this setting, and provide rates of convergence for the convex case. This is the first study of SVRG in an adversarial setting.**

***Index Terms*—Byzantines, Stochastic Gradient Descent (SGD), Stochastic variance reduced gradient (SVRG), Distributed optimization, Adversarial machine learning**

I. Introduction

Many contemporary machine learning tasks require the solution of optimization problems over large datasets. Therefore, it is essential to consider distributed versions of machine learning algorithms to solve the optimization problems. In the distributed setting, data is distributed across many workers with the objective of optimizing a global function. However, due to the distributed nature of the algorithms and involvement of many workers, robustness is an important concern. For example, consider a scenario where data is collected from a number of workers and sent to a central node, with some nodes exhibiting adversarial behavior. These adversarial workers intentionally send arbitrary vectors that are not the correct vectors to be sent according to the algorithm. This can significantly degrade the performance of the optimization algorithm. For example, optimization algorithms based on simple aggregation of workers' computed results can be significantly affected by the appearance of even a single adversarial worker. Hence, designing robust distributed machine learning algorithms to solve optimization problems that can withstand adversarial attacks is of crucial importance [2].

A typical algorithm to solve optimization problems posed in the machine learning context with large datasets is stochastic gradient descent (SGD). Unlike the traditional gradient descent algorithm, SGD computes a single stochastic gradient per iteration resulting in faster computations over large datasets. Due to its popularity and advantages, several works on Byzantine machine learning have considered SGD [3]–[12], [16]. However, a major drawback of SGD is that the convergence is slower compared to gradient descent due to the large variance introduced by stochasticity.

To reduce the variance of the update rule in SGD, the stochastic variance reduced gradient (SVRG) algorithm was proposed in [1]. SVRG ensures convergence by employing a relatively large step size. The advantages of SVRG over SGD have motivated us to consider SVRG in the adversarial setting, propose a robust variant, and analyze its robustness. A major challenge while considering the adversarial setting is the arbitrary nature of the vectors sent by the adversarial workers which are not the correct vectors to be sent according to the algorithm. This arbitrary vector does not follow any parametric or bounded form, further exacerbating the analysis of the optimization algorithm. To the best of our knowledge, ours is the first work that proposes a robust variant of SVRG in an adversarial setting. Here, we consider a distributed setting with n workers where some of them are adversarial in nature and one central node.

Related Work: Several of the recent works have considered a distributed optimization framework [17]–[20] utilizing SGD without the presence of adversarial workers in the system. Furthermore, adversarial attacks have been considered in a distributed setting using SGD with various aggregation rules with the goal to enhance the robustness in [3], [8], [9], [16]. Moreover, in [3], the central node monitors the behavioral state of the workers based on their gradient values, and establishes a set containing the workers that it believes to be non-Byzantine. An asynchronous distributed SGD algorithm is proposed in [12] for an adversarial setting where the workers maintain their own respective local model parameters without the need for a central node.

Although there has been a considerable amount of interest in Byzantine machine learning [3]–[12], [16], a robust scheme to combat the adversarial nature of Byzantine workers with variance reduction of SGD has not been explored. The previously proposed algorithms either consider SGD in an adversarial setting or consider SVRG in the distributed setting in the absence of any adversarial workers. To ensure robustness of SVRG, we compute the average of the intermediate gradients across the number of iterations for each worker and compute the median across all workers as the aggregation rule [3].

Contributions: We consider the finite-sum problem and

This work was supported in part by National Science Foundation under Grant ENG 60064237.

978-1-7281-2723-1/19/$31.00 ©2019 IEEE

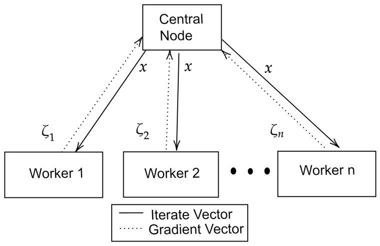

Fig. 1. System Model

propose a robust variant of SVRG in an adversarial setting for the convex case and show its convergence. Importantly, we show that if α-fraction of workers are Byzantine and the objective function f is convex then our proposed algorithm finds a point x within $T = \tilde{O}(\frac{1}{\gamma} + \frac{1}{n\gamma^2} + \frac{\alpha^2}{\gamma^2})$ iterations such that $\mathbb{E}[f(x) - f(x^*)] \leq \gamma$ is satisfied, where the notation $\tilde{O}$ subsumes all the logarithmic factors and x^* is the optimal point. Note that the third term captures the loss in terms of the computational effort due to the presence of Byzantine workers.

II. System Model

We assume that the network has n workers where α-fraction, ($\alpha < 1/2$), of the workers are Byzantine as illustrated in Fig.1. All the workers communicate with a central node synchronously. Also, the Byzantine workers may communicate among themselves.

Problem Formulation: We consider the finite-sum problem

$$\min_{x \in \mathbb{R}^d} \Big\{ f(x) = \frac{1}{n} \sum\nolimits_{i=1}^{n} f_i(x, \xi_i) \Big\}, \tag{1}$$

where $f(x)$ is assumed to be convex. Each worker has access to a set of M sample functions. Here, the sample function is $f_i(\cdot, \xi_i)$ and ξ_i indicates the index of the sample function chosen randomly from the set of sample functions of worker i, $i \in [n]$. Note that the number of workers are less than the total number of sample functions available. Each worker locally computes and sends its intermediate gradient to the central node which after processing the received data sends the updated iterate value to all the workers. This process continues for a number of iterations after which the central node obtains the solution to (1), namely the minimum. In particular, for every iteration the central node broadcasts the current update value and the workers return their respective intermediate gradients at the update value using their respective sample functions. Here, a worker returns the intermediate gradient vector to the central node given by

$$\zeta_{i,t}^{s+1} = \begin{cases} \nabla f_i(x_t^{s+1}, \xi_i) - \nabla f_i(\tilde{x}^s, , \xi_i) + \nabla f_i(\tilde{x}^s), i \in \mathcal{G} \\ *, i \notin \mathcal{G}, \end{cases} \tag{2}$$

where $*$ denotes an arbitrary vector that may or may not be the correct gradient vector to be sent according to the algorithm and the set $\mathcal{G}$ includes only the non-Byzantine workers. Note that the index of the sample function indicated by ξ_i changes for every iteration of the algorithm. Here, $\tilde{x}^s$ is the point where full gradient $\nabla f_i(\tilde{x}^s)$ is computed at the end of epoch s. Also, the stochastic gradient $\nabla f_i(\tilde{x}^s, \xi_i)$ (or $\nabla f_i(x_t^{s+1}, \xi_i)$) is computed at point $\tilde{x}^s$ (or x_t^{s+1}) using the sample function indicated by the index ξ_i for worker i, $i \in [n]$. Note that $\mathbb{E}[\nabla f_i(\tilde{x}^s, \xi_i)] = \nabla f_i(\tilde{x}^s)$, for $i \in [n]$. Furthermore, this intermediate gradient vector $\zeta_{i,t}^{s+1}$ satisfies $||\zeta_{i,t}^{s+1} - \nabla f(x_t^{s+1})|| \leq K$, for $t = \{0, 1, \ldots, m-1\}, s = \{0, 1, \ldots, S-1\}$ where $\nabla f(x_t^{s+1})$ is the gradient of the function f at point x_t^{s+1} and K indicates the variance. Also, note that $||x_0^1 - x^*|| \leq \mu$ where x_0^1 is the initial value and μ is the diameter.

A Byzantine worker may intentionally send an arbitrary vector that is not equal to the computed intermediate gradient vector. The Byzantine worker is assumed to have complete knowledge of all the gradient vectors sent by all the workers until the most recent iterate, $\{\{\zeta_{i,t'}^{s'+1})\}_{t' \leq t, s' \leq s}\}_{i \in [n]}$, for $t = \{0, 1, \ldots, m-1\}, s = \{0, 1, \ldots, S-1\}$ where $T = Sm$. The vector sent by the Byzantine worker may use this knowledge to collude even in an iteration by sending an arbitrary value to change the descent direction leading to degradation of the optimization performance. We assume that the function f is L-smooth, where L is Lipschitz constant, so that $||\nabla f(x) - \nabla f(y)|| \leq L||x - y||$. Next, we propose a novel robust variant of SVRG to solve the minimization problem in (1). The pseudocode for proposed algorithm is provided in Algorithm 1.

III. Byzantine SVRG

In Algorithm 1, the outer loop indicates epoch iteration which is tracked by index s, for $s = \{0, 1, \ldots, S-1\}$, and the inner loop is tracked by t, for $t = \{0, 1, \ldots, m-1\}$. At each update value x_t^{s+1}, the workers return their respective intermediate gradients. Then, the central node computes x_{t+1}^{s+1} with respect to the descent direction computed by aggregation of these intermediate gradients and the set $\mathcal{G}_t \subseteq [n]$ which is updated at each iteration. Here, the set $\mathcal{G}_t$ consists of all the workers that the central node has estimated to be non-Byzantine at the current iteration. Note that the update is performed in the direction of $\zeta_{i,t}^{s+1} = \nabla f_i(x_t^{s+1}, \xi_i) - \nabla f_i(\tilde{x}^s, \xi_i) + \nabla f_i(\tilde{x}^s)$. Consequently, the update value x_{t+1}^{s+1} is computed as

$$x_{t+1}^{s+1} = x_t^{s+1} - \eta_t \sum\nolimits_{i \in \mathcal{G}_t} \zeta_{i,t}^{s+1}, \tag{3}$$

where η_t is the step size.

Updating set $\mathcal{G}_t$: The manner in which the set $\mathcal{G}_t$ is updated is described as the following. The set $\mathcal{G}_t$ consists of all the workers $i \in [n]$ from $\mathcal{G}_{t-1}$ which satisfy the following criterion

$$\begin{aligned} &\mathcal{G}_t \leftarrow \{i \in \mathcal{G}_{t-1} : |\Xi_i - \Xi_{med}| \leq \tau_\Xi \\ &\quad \wedge ||\Theta_i - \Theta_{med}|| \leq \tau_\Theta \wedge ||\zeta_{i,t}^{s+1} - \nabla_{med}|| \leq 4K\}, \end{aligned} \tag{4}$$

where $\mathcal{G}_0 = [n]$. The criterion in (4) consists of three conditions. First, in order for a worker to be in the set $\mathcal{G}_t$, it should satisfy $||\Xi_i - \Xi_{med}|| \leq \tau_\Xi$ which ensures that the value $\Xi_i = \sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\langle \zeta_{i,t}^{s+1}, x_t^{s+1} - x_0^1\rangle$ should be τ_Ξ-close to Ξ_{med} where Ξ_{med} is the median of $\{\Xi_1, \Xi_2, \dots, \Xi_n\}$. Next, a worker should also satisfy $||\Theta_i - \Theta_{med}|| \leq \tau_\Theta$ to be in the set $\mathcal{G}_t$. Here, the vector $\Theta_i = \sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\zeta_{i,t}^{s+1}$ should be τ_Θ-close to Θ_{med} where the vector median Θ_{med} is equal to a vector Θ_i such that $|j \in [n] : ||\Theta_j - \Theta_i|| \leq \tau_\Theta| > n/2$. Furthermore, a worker should also satisfy $||\zeta_{i,t}^{s+1} - \nabla_{med}|| \leq 4K$ to be in the set $\mathcal{G}_t$. Here, the vector $\zeta_{i,t}^{s+1}$ should be $4K$-close to ∇_{med} where the vector median ∇_{med} is equal to a vector $\zeta_{i,t}^{s+1}$ such that $|j \in [n] : ||\zeta_{j,t}^{s+1} - \zeta_{i,t}^{s+1}|| \leq 2K| > n/2$.

Proof Sketch for $||\zeta_{i,t}^{s+1} - \nabla_{med}|| \leq 4K$: For any $i, j \in \mathcal{G}$, we have $||\zeta_{j,t}^{s+1} - \zeta_{i,t}^{s+1}|| \leq 2K$. Hence, as $\alpha < 1/2$, $\nabla_{med} = \zeta_{i,t}^{s+1}$ is a valid choice for any $i \in \mathcal{G}$. If $||\nabla f(x_t^{s+1}) - \nabla_{med}|| > 3K$ then by triangle inequality, we have $||\zeta_{i,t}^{s+1} - \nabla_{med}|| > 2K$, $\forall i \in \mathcal{G}$ which is a contradiction. Hence, $||\nabla f(x_t^{s+1}) - \nabla_{med}|| \leq 3K$. Adding and subtracting $\zeta_{i,t}^{s+1}$, and applying reverse triangle inequality, we have $||\zeta_{i,t}^{s+1} - \nabla_{med}|| \leq 4K$.

Algorithm 1 Byzantine SVRG Algorithm

```
Input: $\tilde{x}^0 = x_m^0 = x^0 \in \mathbb{R}^d$, epoch length $m$, step sizes $\{\eta_t > 0\}_{t=0}^{m-1}$, $S = \lceil T/m \rceil$, discrete probability distribution $\{p_i\}_{i=0}^{m}$
for s=0 to S-1 do
    $x_0^{s+1} = \tilde{x}^s$;
    Computing full gradient $\nabla f_i(\tilde{x}^s)$;
    for t=0 to m-1 do
        Randomly pick a sample function from the set of $M$ sample functions;
        $\zeta_{i,t}^{s+1} = \nabla f_i(x_t^{s+1}, \xi_i) - \nabla f_i(\tilde{x}^s, \xi_i) + \nabla f_i(\tilde{x}^s)$;
        $\mathcal{G}_t \leftarrow \{i \in \mathcal{G}_{t-1} : |\Xi_i - \Xi_{med}| \leq \tau_\Xi \wedge ||\Theta_i - \Theta_{med}|| \leq \tau_\Theta \wedge ||\zeta_{i,t}^{s+1} - \nabla_{med}|| \leq 4K\}$ ;
        $x_{t+1}^{s+1} = x_t^{s+1} - \eta_t \frac{1}{n}\sum_{i\in\mathcal{G}_t}\zeta_{i,t}^{s+1}$;
    end for
    $\tilde{x}^{s+1} = \sum_{i=0}^{m} p_i x_i^{s+1}$
end for
Value $z$ is average of all the iterates $\{\{x_t^{s+1}\}_{t=0}^{m-1}\}_{s=0}^{S-1}$
```

We define two error terms that correspond to the proposed algorithm given as

$$\epsilon_1 = \sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\mathbb{E}\Big[\sum_{i\in\mathcal{G}_t\setminus\mathcal{G}}\langle(\zeta_{i,t}^{s+1} - \nabla f(x_t^{s+1})), x_t^{s+1} - x^*\rangle\Big], \quad (5)$$

and

$$\epsilon_2 = \mathbb{E}\left[||\frac{1}{n}\sum_{i\in\mathcal{G}_t}(\zeta_{i,t}^{s+1} - \nabla f(x_t^{s+1}))||^2\right], \quad (6)$$

where the expectation is over the stochasticity of the algorithm. We obtain the bounds for the two error terms which are presented in Lemma III.1. Note that the effect of the remaining Byzantine workers in the set $\mathcal{G}_t$ at the end of the algorithm has negligible impact.

Lemma III.1. *The bounds for the error terms ϵ_1 and ϵ_2 are given by $|\epsilon_1| \leq 16\alpha nK\mu\sqrt{TR}$ and $\epsilon_2 \leq 32\alpha^2\mu^2 + \frac{4\mu^2 R}{n}$, with probability at least $1-\delta$ where $R = \log(16nT/\delta)$ and $||x_0^1 - x^*|| \leq \mu$.*

Note that the bounds are computed in a manner similar to the method in [3]. Next, we provide the rate of convergence of the robust variant of SVRG algorithm in the adversarial setting for the convex case proved using Lemma III.1 as follows:

Theorem III.2. *For the case with α-fraction Byzantine workers, when f is convex, we have with probability $1-\delta$*

$$\mathbb{E}[f(z) - f(x^*)] \leq \frac{\mu^2}{\eta_t T} + \frac{|\epsilon_1|}{Tn} + \eta_t\epsilon_2, \quad (7)$$

where x^* is the optimal solution, and z is the average of all iterates. As $\eta_t = \eta$ for $t \in \{0, \dots, m-1\}$, substituting the optimal step size $\eta = \sqrt{\frac{1}{T(\frac{4R}{n} + 32\alpha^2)}}$ and the error bounds, we obtain

$$\mathbb{E}[f(z) - f(x^*)] \leq \frac{(\mu^2 + \mu)\sqrt{\frac{4R}{n} + 32\alpha^2} + 32\alpha K\mu\sqrt{R}}{\sqrt{T}}, \quad (8)$$

which ensures convergence.

Note that the distortion caused by the presence of α-fraction of Byzantine workers in the system is captured by the two error terms ϵ_1 and ϵ_2.

Proof. Consider the following

$$\begin{aligned}\frac{1}{T}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\mathbb{E}[\langle\frac{1}{n}\sum_{i\in\mathcal{G}_t}\zeta_{i,t}^{s+1}, x_t^{s+1} - x^*\rangle] &= \frac{1}{T}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\mathbb{E}[\langle\frac{1}{n}\sum_{i\in\mathcal{G}_t}\zeta_{i,t}^{s+1}, x_t^{s+1} - x_{t+1}^{s+1}\rangle \\ &\quad + \langle\frac{1}{n}\sum_{i\in\mathcal{G}_t}\zeta_{i,t}^{s+1}, x_{t+1}^{s+1} - x^*\rangle], \quad (9)\end{aligned}$$

substituting the update equation $x_{t+1}^{s+1} = x_{t+1}^{s+1} - \frac{1}{n}\sum_{i\in\mathcal{G}_t}\zeta_{i,t}^{s+1}$ and expanding the second term , we get

$$\begin{aligned}\frac{1}{T}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\mathbb{E}[\langle\frac{1}{n}\sum_{i\in\mathcal{G}_t}\zeta_{i,t}^{s+1}, x_t^{s+1} - x^*\rangle] &\leq \frac{1}{T}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\mathbb{E}\Bigg[\langle\frac{1}{n}\sum_{i\in\mathcal{G}_t}\zeta_{i,t}^{s+1}, x_t^{s+1} - x_{t+1}^{s+1}\rangle \\ &\quad + \frac{||x_t^{s+1} - x^*||^2}{2\eta_t} - \frac{||x_{t+1}^{s+1} - x^*||^2}{2\eta_t} - \frac{||x_t^{s+1} - x_{t+1}^{s+1}||^2}{2\eta_t}\Bigg]. \quad (10)\end{aligned}$$

Next, consider again the term $\mathbb{E}[\langle \frac{1}{m}\sum_{i\in\mathcal{G}_t}\zeta_{i,t}^{s+1}, x_t^{s+1} - x^*\rangle]$ which can be expanded as

$$\frac{1}{T}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\mathbb{E}[\langle\frac{1}{n}\sum_{i\in\mathcal{G}_t}\zeta_{i,t}^{s+1}, x_t^{s+1} - x^*\rangle]$$
$$= \frac{1}{T}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\mathbb{E}\Bigg[\langle\frac{1}{n}\sum_{i\in\mathcal{G}}\zeta_{i,t}^{s+1} - \nabla f_i(x_t^{s+1}), x_t^{s+1} - x^*\rangle$$
$$+\langle\frac{1}{n}\sum_{i\in\mathcal{G}}\nabla f_i(x_t^{s+1}), x_t^{s+1} - x^*\rangle$$
$$+\langle\frac{1}{n}\sum_{i\in\mathcal{G}_t\setminus\mathcal{G}}\zeta_{i,t}^{s+1}, x_t^{s+1} - x^*\rangle\Bigg], \quad (11)$$

using the fact that $\mathbb{E}[\zeta_{i,t}^{s+1}] = \nabla f_i(x_t^{s+1})$, for $i \in \mathcal{G}$ which ensures that the first term on the right hand side is zero, and adding and subtracting $\nabla f(x_t^{s+1})$ in the second term, we get

$$\frac{1}{T}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\mathbb{E}[\langle\frac{1}{n}\sum_{i\in\mathcal{G}_t}\zeta_{i,t}^{s+1}, x_t^{s+1} - x^*\rangle]$$
$$= \frac{1}{T}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\mathbb{E}\Bigg[\frac{1}{n}\sum_{i\in\mathcal{G}}\langle\nabla f_i(x_t^{s+1}) - \nabla f(x_t^{s+1}), x_t^{s+1} - x^*\rangle$$
$$+\frac{1}{n}\sum_{i\in\mathcal{G}}\langle\nabla f(x_t^{s+1}), x_t^{s+1} - x^*\rangle$$
$$+\frac{1}{n}\sum_{i\in\mathcal{G}_t\setminus\mathcal{G}}\langle\zeta_{i,t}^{s+1} - \nabla f(x_t^{s+1}), x_t^{s+1} - x^*\rangle$$
$$+\frac{1}{n}\sum_{i\in\mathcal{G}_t\setminus\mathcal{G}}\langle\nabla f(x_t^{s+1}), x_t^{s+1} - x^*\rangle\Bigg]. \quad (12)$$

Note that $\mathbb{E}[\nabla f_i(x_t^{s+1})] = \nabla f(x_t^{s+1})$, for $i \in \mathcal{G}$ which ensures that the first term on the right hand side is zero, rearranging the remaining terms, and using the definition of ϵ_1 along with convexity and L-smoothness properties, we obtain

$$\frac{1}{T}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\mathbb{E}[\langle\frac{1}{n}\sum_{i\in\mathcal{G}_t}\zeta_{i,t}^{s+1}, x_t^{s+1} - x^*\rangle] \geq$$
$$\frac{1}{T}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\mathbb{E}\Bigg[\frac{1}{n}\sum_{i\in\mathcal{G}_t}(f(x_{t+1}^{s+1}) - \langle\nabla f(x_t^{s+1}), x_{t+1}^{s+1} - x_t^{s+1}\rangle$$
$$-\frac{L}{2}||x_t^{s+1} - x_{t+1}^{s+1}||^2 - f(x^*))\Bigg] + \frac{\epsilon_1}{nT}. \quad (13)$$

Substituting the upper bound for $\frac{1}{T}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\mathbb{E}[\langle\frac{1}{m}\sum_{i\in\mathcal{G}_t}\zeta_{i,t}^{s+1}, x_t^{s+1} - x^*\rangle]$ obtained in (13), in (10) yields

$$\frac{1}{T}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\mathbb{E}[f(x_{t+1}^{s+1}) - f(x^*)]$$
$$\leq \frac{1}{T}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\mathbb{E}\Bigg[\langle\frac{1}{n}\sum_{i\in\mathcal{G}_t}\zeta_{i,t}^{s+1} - \nabla f(x_t^{s+1}), x_t^{s+1} - x_{t+1}^{s+1}\rangle$$
$$+\frac{||x_t^{s+1} - x^*||^2}{2\eta_t} - \frac{||x_{t+1}^{s+1} - x^*||^2}{2\eta_t} - (\frac{1}{2\eta_t} - \frac{L}{2})||x_t^{s+1} - x_{t+1}^{s+1}||^2\Bigg]$$
$$-\frac{\epsilon_1}{nT}. \quad (14)$$

Applying Young's inequality to $\langle\frac{1}{n}\sum_{i\in\mathcal{G}_t}\zeta_{i,t}^{s+1} - \nabla f(x_t^{s+1}), x_t^{s+1} - x_{t+1}^{s+1}\rangle$ in the above inequality, we get

$$\frac{1}{T}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\mathbb{E}[f(x_{t+1}^{s+1}) - f(x^*)]$$
$$\leq \frac{1}{T}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\mathbb{E}\Bigg[\frac{1}{2\beta_t}||\frac{1}{n}\sum_{i\in\mathcal{G}_t}\zeta_{i,t}^{s+1} - \nabla f(x_t^{s+1})||^2$$
$$+\frac{\beta_t}{2}||x_t^{s+1} - x_{t+1}^{s+1}||^2 + \frac{||x_t^{s+1} - x^*||^2}{2\eta_t} - \frac{||x_{t+1}^{s+1} - x^*||^2}{2\eta_t}$$
$$-(\frac{1}{2\eta_t} - \frac{L}{2})||x_t^{s+1} - x_{t+1}^{s+1}||^2\Bigg] - \frac{\epsilon_1}{nT}. \quad (15)$$

Let $\beta_t = \frac{1}{2\eta_t}$. Also, note that $\frac{1}{4\eta_t} \leq \frac{1}{2\eta_t} - \frac{L}{2}$. Simplifying further yields

$$\frac{1}{T}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\mathbb{E}[f(x_{t+1}^{s+1}) - f(x^*)]$$
$$\leq \frac{1}{T}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\mathbb{E}\Bigg[\eta_t||\frac{1}{n}\sum_{i\in\mathcal{G}_t}\zeta_{i,t}^{s+1} - \nabla f(x_t^{s+1})||^2$$
$$+\frac{||x_t^{s+1} - x^*||^2}{2\eta_t} - \frac{||x_{t+1}^{s+1} - x^*||^2}{2\eta_t}\Bigg] - \frac{\epsilon_1}{nT}. \quad (16)$$

summing over the indices s and t, rearranging the terms, using the definition of ϵ_2, and as $\eta_t = \eta$ for $t \in \{0, \ldots, m-1\}$, we obtain

$$\frac{1}{T}\sum_{s=0}^{S-1}\sum_{t=0}^{m-1}\mathbb{E}[f(x_{t+1}^{s+1}) - f(x^*)] \leq \eta\epsilon_2 + \frac{\mu^2}{\eta T} + \frac{|\epsilon_1|}{Tn}, \quad (17)$$

by using Jensen's inequality yields the result in (7). Furthermore, equating the terms containing η, we obtain the value of η as $\eta\epsilon_2 = \frac{1}{T\eta}$, and using Lemma III.1, results in $\eta = \sqrt{\frac{1}{T(\frac{4R}{n}+32\alpha^2)}}$. Substituting the value of η in the above inequality yields the final result. □

IV. Conclusion

We proposed a robust variant of the SVRG algorithm for an adversarial setting where α-fraction of the workers are Byzantine. The Byzantine workers may collude and send arbitrary data to the central node. We analyzed the rate of convergence of the proposed algorithm which provides a solution for the finite-sum problem for the convex case. The result shows its robustness against adversarial attacks and reduced variance of the proposed algorithm. The avenues for future work for the robust variant of SVRG algorithm include analyzing the rates of convergence for the non-convex and strongly-convex cases.

References

[1] R. Johnson, T. Zhang, "Accelerating Stochastic Gradient Descent using Predictive Variance Reduction", Adv. in Neural Inf. Process. Syst. 26, pp. 315–323, 2013.

[2] Q. Li, B. Kailkhura, R. Goldhahn, P. Ray, and P. K. Varshney, "Robust Decentralized Learning Using ADMM with Unreliable Agents" , arxiv.org/abs/1710.05241, 2018.

[3] D. Alistarh, Z. Allen-Zhu, and J. Li, "Byzantine Stochastic Gradient Descent", Adv. in Neural Inf. Process. Syst. 31, pp. 4613–4623, 2018.

[4] P. Blanchard, EM. El Mhamdi, R. Guerraoui, and J. Stainer, ”Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent”, Adv. in Neural Inf. Process. Syst. 30, pp. 119–129, 2017.

[5] C. Xie, O. Koyejo, and I. Gupta, ”Generalized Byzantine-tolerant SGD”, arxiv.org/abs/1802.10116, 2018.

[6] G. Damaskinos, EM. El Mhamdi, R. Guerraoui, R. Patra, and M. Taziki, ”Asynchronous Byzantine Machine Learning (the case of SGD)”, Proc. of the 35th International Conference on Mach. Learn., pp. 1145–1154, 2018.

[7] L. Chen, H. Wang, Z. Charles, and D. Papailiopoulos, ”DRACO: Byzantine-resilient Distributed Training via Redundant Gradients”, Proc. of the 35th International Conference on Mach. Learn., pp. 903–912, 2018.

[8] C. Xie, O. Koyejo, and I. Gupta, ”Zeno: Byzantine-suspicious stochastic gradient descent”, arxiv.org/abs/1805.10032, 2018.

[9] C. Xie, O. Koyejo, and I. Gupta, ”Phocas: Dimensional Byzantine-resilient stochastic gradient descent”, arxiv.org/abs/1805.09682, 2018.

[10] N. Konstantinov and C. Lampert, ”Robust Learning from Untrusted Sources”, arxiv.org/abs/1901.10310, 2019.

[11] C. Xie, ”Zeno++: Robust Asynchronous SGD with Arbitrary number of Byzantine workers”, arxiv.org/abs/1903.07020, 2019.

[12] R. Jin, X. He, and H. Dai, ”Distributed Byzantine Tolerant Stochastic Gradient Descent in the Era of Big Data”, arxiv.org/abs/1902.10336, 2019.

[13] M. Li, T. Zhang, Y. Chen, and A. J. Smola, ”Efficient Mini-batch Training for Stochastic Optimization”, Proc. of the 20th ACM SIGKDD International Conference on Knowl. Discovery and Data Mining, pp. 661–670, 2014.

[14] L. Liu, T. Li, and C. Caramanis, ”High Dimensional Robust Estimation of Sparse Models via Trimmed Hard Thresholding”, arxiv.org/abs/1901.08237, 2019.

[15] Q. Yang, Y. Liu, T. Chen, and Y. Tong, ”Federated Machine Learning: Concept and Applications”, ACM Trans. Intell. Syst. Technol., 2019.

[16] L. Su, and J. Xu, ”Securing Distributed Gradient Descent in High Dimensional Statistical Learning”, Proc. ACM Meas. Anal. Comput. Syst., 2019.

[17] M. Zinkevich, M. Weimer, L. Li, A. J. Smola, J.D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, ”Parallelized Stochastic Gradient Descent”, Adv. in Neural Inf. Process. Syst. 23, pp. 2595–2603, 2010.

[18] B. Recht, C. Re, S. Wright, F. Niu, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, ”Hogwild: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent”, Adv. in Neural Inf. Process. Syst. 24, pp. 693–701, 2011.

[19] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, ”Optimal Distributed Online Prediction Using Mini-batches”, J. Mach. Learn. Res., pp. 165–202, 2012.

[20] Q. Ho, J. Cipar, H. Cui, S. Lee, J. K. Kim, P. B. Gibbons, G. A. Gibson, G. Ganger, E. P. Xing, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, ”More Effective Distributed ML via a Stale Synchronous Parallel Parameter Server”, Adv. in Neural Inf. Process. Syst. 26, pp. 1223–1231, 2013.