BYZANTINE-RESILIENT DISTRIBUTED FINITE-SUM OPTIMIZATION OVER NETWORKS

Zhaoxian Wu[*] *Qing Ling*[*] *Tianyi Chen*[†] *Georgios B. Giannakis*[‡]

[*] School of Data and Computer Science, Sun Yat-Sen University, Guangzhou, Guangdong 510006, China
[†] Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, New York 12180, USA
[‡] Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, Minnesota 55455, USA

ABSTRACT

In this paper, we investigate the problem of distributed finite-sum optimization in presence of malicious attacks from Byzantine workers. Existing Byzantine-resilient algorithms often combine stochastic gradient descent (SGD) with various robust aggregation rules to handle malicious attacks. However, the large gradient noise of SGD brings difficulty to distinguish malicious messages sent by the Byzantine workers from noisy stochastic gradients sent by the honest workers. This fact motivates us to reduce the gradient noise so as to achieve better performance than Byzantine-resilient SGD. Therefore, we propose Byrd-SAGA, a Byzantine-resilient variant of distributed SAGA to deal with the malicious attacks in the distributed finite-sum optimization setting. Byrd-SAGA uses geometric median to aggregate the corrected stochastic gradients sent by the distributed workers, other than uses mean in distributed SAGA. When less than half of the workers are Byzantine, the robustness of geometric median to outliers enables Byrd-SAGA to achieve provable linear convergence to a neighborhood of the optimal solution, where the size of neighborhood is determined by the number of Byzantine workers.

***Index Terms*—** Byzantine-resilience, distributed finite-sum optimization, variance reduction

1. INTRODUCTION

With the fast development of information technologies, the volume of distributed big data increases explosively. Every day, distributed devices (e.g., sensors, cellphones, computers, vehicles, etc) generate a huge amount of data, which are often transmitted to datacenters for processing and learning. However, collecting the data from the distributed devices and storing them in the datacenters raise significant privacy concerns [1–3]. To address this issue, federated learning has been proposed as a new privacy-preserving distributed data processing and machine learning framework [4]. In federated learning, the data are kept privately by and the computation is assigned to the distributed devices. Iteratively, the distributed devices calculate their local variables (e.g., stochastic gradients, corrected stochastic gradients, models, etc) using the private data samples, while the datacenter aggregates the local variables and disseminates the aggregated result to the distributed devices.

Nevertheless, the distributed nature of federated learning makes it vulnerable to errors and attacks. Some of the distributed devices can be unreliable in either computation or communication, while some can be hacked by malicious attackers. These distributed devices may send arbitrary malicious messages to the datacenter, aiming at misleading the learning process [5–7]. We call these arbitrary malicious attacks as Byzantine attacks [8]. It is crucial to develop robust federated learning algorithms to handle these Byzantine attacks for secure processing and learning.

In view of the challenge in Byzantine-resilient federated learning, various robust aggregation rules have been developed in these years, mainly focused on improving the distributed stochastic gradient descent (SGD) method. Through aggregating stochastic gradients with geometric median [9,10], median [11], trimmed mean [12], iterative filtering [13], Krum [14], or RSA [15], Byzantine-resilient distributed SGD is able to tolerate the attacks from a small number of Byzantine devices.

Although these Byzantine-resilient SGD methods are often guaranteed to reach a neighborhood of the Byzantine-free optimal solution, the size of the neighborhood can be large under well-designed Byzantine attacks [16]. Essentially, SGD suffers from large gradient noise in computing the stochastic gradients. This disadvantage leads to the key difficulty in distinguishing the malicious messages sent by the Byzantine attackers from the noisy stochastic gradients sent by the honest devices.

Considering the deficiency of Byzantine-resilient distributed SGD, we ask: *Can we better distinguish the malicious messages from the stochastic gradients through reducing the gradient noise?* Our answer is affirmative. When the gradient noise is small, the malicious messages are easy to be identified (see the illustrative example in Section 2). This observation suggests the combination of variance reduction techniques with robust aggregation rules to handle Byzantine attacks in federated learning.

Existing variance reduction techniques in stochastic optimization include mini-batch [17], SAG [18], SVRG [19], SAGA [20], SDCA [21], SARAH [22], Katyusha [23], to name a few. Among these algorithms, we are particularly interest in SAGA, which has been proven to be effective in finite-sum optimization. SAGA can also be implemented in a distributed manner [24–26], and is hence fit for the federated learning applications where every distributed device has a finite number of data samples.

In this paper, we propose Byrd-SAGA, which combines the variance reduction technique of SAGA with robust aggregation to deal with the malicious attacks in the distributed finite-sum optimization setting. In Byrd-SAGA, the datacenter uses geometric median to aggregate the corrected stochastic gradients sent by the distributed devices, other than mean in distributed SAGA. Through reducing the gradient noise, Byrd-SAGA is able to achieve better performance than Byzantine-resilient distributed SGD.

2. PROBLEM STATEMENT

Consider a distributed network with one master node (datacenter) and W workers (devices), among which B workers are Byzantine but their identities are unknown to the master node. Denote the set of all workers as $\mathcal{W}$ and the set of Byzantine workers as $\mathcal{B}$ (hence $|\mathcal{W}| = W$ and $|\mathcal{B}| = B$). The data samples are evenly distributed across the honest workers $w \notin \mathcal{B}$. Every honest worker has J data

samples, and we use $f_{w,j}(x)$ to denote the loss function of the j-th data sample on the honest worker w with respect to the model $x \in \mathbb{R}^p$. We are interested in the finite-sum optimization problem

$$x^* = \arg\min_x \ f(x) := \frac{1}{W-B} \sum_{w \notin B} f_w(x), \tag{1}$$

where $f_w(x) := \frac{1}{J} \sum_{j=1}^{J} f_{w,j}(x)$.

Distributed SGD. When all the workers are honest, one of the most popular algorithms to solve the distributed finite-sum optimization problem is SGD [27]. At time k, the master node sends x^k to the workers. Upon receiving x^k, every worker w uniformly at random chooses a local data sample with index i_w^k to calculate the stochastic gradient $f'_{w,i_w^k}(x^k)$ and sends back to the master node. After collecting all the stochastic gradients from the workers, the master node updates the model by $x^{k+1} = x^k - \gamma^k \cdot \frac{1}{W} \sum_{w=1}^{W} f'_{w,i_w^k}(x^k)$, where γ^k is the non-negative step size.

While the honest workers send true stochastic gradients to the master node, the Byzantine workers may not do so. The Byzantine workers can send arbitrary malicious messages to the master node, aiming at biasing the optimization process. We use m_w^k to denote the message sent from worker w to the master node at time k, such that $m_w^k = f'_{w,i_w^k}(x^k)$ for all $w \notin \mathcal{B}$ and $m_w^k = *$ for all $w \in \mathcal{B}$, where $*$ represents an arbitrary p-dimensional vector. Then, the distributed SGD update becomes $x^{k+1} = x^k - \gamma^k \cdot \frac{1}{W} \sum_{w=1}^{W} m_w^k$.

Even when only one Byzantine worker is present, the distributed SGD may fail. Let w_b be the Byzantine worker. It can send to the master node $m_{w_b}^k = -\sum_{w \neq w_b} m_w^k$ which makes $x^{k+1} = x^k$. In practice, the Byzantine workers can send more tricky messages to fool the master node, and bias the optimization process.

Byzantine-resilient distributed SGD. Recent works often robustify the distributed SGD by incorporating the robust aggregation rules when the master node receives messages from the workers. In particular, we will focus on the application and analysis of geometric median, while other robust aggregation rules are also viable [9, 10].

Let $\{z, z \in \mathcal{Z}\}$ be a subset in a normed space. The geometric median of $\{z, z \in \mathcal{Z}\}$ is defined as

$$\underset{z \in \mathcal{Z}}{\text{geomed}}\{z\} = \arg\min_y \sum_{z \in \mathcal{Z}} \|y - z\|. \tag{2}$$

With (2), distributed SGD can be modified to its Byzantine-resilient form as $x^{k+1} = x^k - \gamma^k \cdot \text{geomed}_{w \in \mathcal{W}}\{m_w^k\}$. When the number of Byzantine workers $B < \frac{W}{2}$, the geometric median provides a reasonable approximate to the mean of $\{m_w^k, w \notin \mathcal{B}\}$. This property enables the Byzantine-resilient distributed SGD to converge to a neighborhood of the optimal solution [9, 10].

Impact of gradient noise on robust aggregation. In distributed SGD, the stochastic gradients calculated by the honest workers are noisy because of the randomness in choosing data samples. Due to the existence of gradient noise, it is not always easy to distinguish the malicious messages from the stochastic gradients using the robust aggregation rules, such as geometric median.

Fig. 1 depicts the impact of gradient noise on geometric median-based robust aggregation. When the variance of the stochastic gradients sent by the honest workers is smaller, the gap between the true mean and the aggregated value is also smaller. That is to say, the same Byzantine attacks are less effective. We will theoretically justify this statement in the theoretical analysis.

Motivated by this fact, we propose to reduce the gradient noise in Byzantine-resilient SGD so as to achieve better robustness to Byzantine attacks. In the Byzantine-free case, an effective approach to alleviate the gradient noise of SGD is variance reduction. Through correcting the noise in the stochastic gradients, variance reduction techniques enable the algorithms converge faster than SGD. In this paper, we focus on SAGA, which reduces gradient noise for finite-sum optimization [20], and will show that this technique effectively helps robust aggregation against Byzantine attacks.

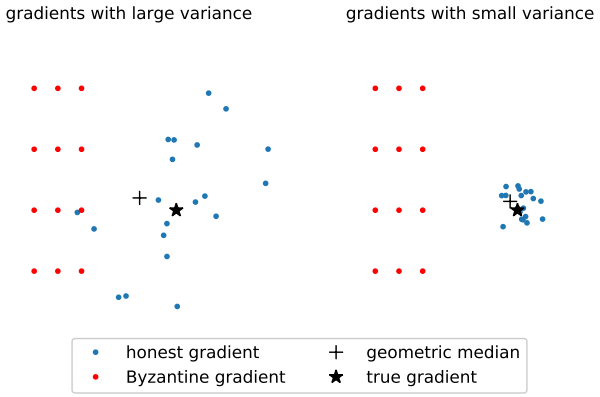

Fig. 1. Impact of gradient noise on geometric median-based robust aggregation. Blue dots denote stochastic gradients sent by the honest workers. Red dots denote malicious messages sent by the Byzantine workers. Plus signs denote the outputs of geometric median-based robust aggregation. Pentagrams denote the means of the stochastic gradients sent by the honest workers.

3. ALGORITHM DEVELOPMENT

Distributed SAGA with mean aggregation. In distributed SAGA, every worker maintains a table of stochastic gradients for all of its local data samples [24, 25]. Like the distributed SGD, at time k, the master node sends x^k to the workers and every worker w uniformly at random chooses a local data sample with index i_w^k to calculate a stochastic gradient $f'_{w,i_w^k}(x^k)$. However, worker w does not send back $f'_{w,i_w^k}(x^k)$ to the master node. Instead, it corrects $f'_{w,i_w^k}(x^k)$ by first subtracting the previously stored stochastic gradient of the i_w^k-th data sample, and then adding the average of the stored stochastic gradients of all the local data samples. Then, worker w sends the corrected stochastic gradient to the master node, and stores $f'_{w,i_w^k}(x^k)$ as the stochastic gradient of the i_w^k-th data sample in the table. After collecting all of the corrected stochastic gradients from the workers, the master node updates the model x^{k+1}. To better describe distributed SAGA, define $\phi_{w,j}^{k+1} = \phi_{w,j}^k$ when $j \neq i_w^k$ and $\phi_{w,j}^{k+1} = x^k$ when $j = i_w^k$. Then, $f'_{w,j}(\phi_{w,j}^k)$ refers to the the previously stored stochastic gradient of the j-th data sample prior to time k on worker w, and $g_w^k := f'_{w,i_w^k}(x^k) - f'_{w,i_w^k}(\phi_{w,i_w^k}^k) + \frac{1}{J} \sum_{j=1}^{J} f'_{w,j}(\phi_{w,j}^k)$ is the corrected stochastic gradient of worker w at time k. The model update of SAGA is hence

$$x^{k+1} = x^k - \gamma \cdot \frac{1}{W} \sum_{w=1}^{W} g_w^k, \tag{3}$$

where $\gamma > 0$ is the constant step size.

Distributed SAGA with geometric median aggregation. A Byzantine worker may send malicious messages, other than the corrected stochastic gradient, to the master node. We use m_w^k to denote

Algorithm 1 Byzantine-Resilient Distributed SAGA

Master node and honest workers initialize x^0
for all honest worker w **do**
  for $j \in \{1, \ldots, J\}$ **do**
    Initializes gradient storage $f'_{w,j}(\phi_{w,j}) = f'_{w,j}(x^0)$
  end for
  Initializes average gradient $\bar{g}_w^1 = \frac{1}{J}\sum_{j=1}^{J} f'_{w,j}(x^0)$
  Sends $\bar{g}_w^1$ to master node
end for
Master node updates $x^1 = x^0 - \gamma \cdot \text{geomed}_{w\in\mathcal{W}}\{\bar{g}_w^1\}$
for all $k = 1, 2, \cdots$ **do**
  Master node broadcasts x^k to all workers
  for all honest worker node w **do**
    Samples i_w^k from $\{1, \cdots, J\}$ uniformly at random
    Updates $m_w^k = f'_{w,i_w^k}(x^k) - f'_{w,i_w^k}(\phi_{w,i_w^k}) + \bar{g}_w^k$
    Sends m_w^k to master node
    Updates $\bar{g}_w^{k+1} = \bar{g}_w^k + \frac{1}{J}(f'_{w,i_w^k}(x^k) - f'_{w,i_w^k}(\phi_{w,i_w^k}))$
    Stores gradient $f'_{w,i_w^k}(\phi_{w,i_w^k}) = f'_{w,i_w^k}(x^k)$
  end for
  Master updates $x^{k+1} = x^k - \gamma \cdot \text{geomed}_{w\in\mathcal{W}}\{m_w^k\}$
end for

the message sent from worker w to the master node at time k, as

$$m_w^k = \begin{cases} g_w^k, & w \notin \mathcal{B}, \\ *, & w \in \mathcal{B}, \end{cases} \tag{4}$$

where $*$ is an arbitrary p-dimensional vector. Similar to distributed SGD, distributed SAGA is also sensitive to Byzantine attacks. Here we propose to use geometric median as the robust aggregation rule. Thus, distributed SAGA in (3) can be modified to a Byzantine-resilient form of

$$x^{k+1} = x^k - \gamma \cdot \underset{w\in\mathcal{W}}{\text{geomed}}\{m_w^k\}. \tag{5}$$

The Byzantine-resilient distributed SAGA, abbreviated as Byrd-SAGA, is outlined in Algorithm 1.

4. THEORETICAL ANALYSIS

Importance of reducing gradient noise. The influence of gradient noise on the geometric median aggregation can be demonstrated by the following lemma.

Lemma 1. *(Concentration property) Let $\{z, z \in \mathcal{Z}\}$ be a subset of random vectors distributed in a normed vector space. If $\mathcal{Z}' \subseteq \mathcal{Z}$ and $|\mathcal{Z}'| < \frac{|\mathcal{Z}|}{2}$, then it holds*

$$\begin{aligned} &E\|\underset{z\in\mathcal{Z}}{\text{geomed}}\{z\} - \bar{z}\|^2 \\ \le& 2C_\alpha^2 \frac{\sum_{z\notin\mathcal{Z}'} E\|z - Ez\|^2}{|\mathcal{Z}| - |\mathcal{Z}'|} + 2C_\alpha^2 \frac{\sum_{z\notin\mathcal{Z}'} \|Ez - \bar{z}\|^2}{|\mathcal{Z}| - |\mathcal{Z}'|}, \end{aligned} \tag{6}$$

where $\bar{z} := \frac{\sum_{z\notin\mathcal{Z}'} Ez}{|\mathcal{Z}| - |\mathcal{Z}'|}$, $C_\alpha := \frac{2-2\alpha}{1-2\alpha}$ and $\alpha := \frac{|\mathcal{Z}'|}{|\mathcal{Z}|}$.

Assume that $\mathcal{Z}$ is the set of messages sent by all the workers in $\mathcal{W}$ and $\mathcal{Z}'$ is the set of malicious messages sent by the Byzantine workers in $\mathcal{B}$. Then, $\bar{z}$ denotes the true gradient (averaged expectation of the stochastic gradients) and the left-hand side of (6) is the variation of the geometric median with respect to the true gradient. The upper bound in the right-hand side of (6) consists of two terms. The first term is determined by the variances of the local stochastic gradients sent by the honest workers (inner variation), while the second term is determined by the variations of the local gradients at the honest workers with respect to the true gradient (outer variation). In Byzantine-resilient SGD, the upper bound can be large due to the large gradient noise of SGD. Through reducing the gradient noise in terms of either inner variation or outer variation, we are able to achieve better accuracy under malicious messages.

Convergence of Byrd-SAGA. Now we show the convergence property of Byrd-SAGA and demonstrate its robustness to Byzantine attacks. We begin with several assumptions on the functions $\{f_{w,j}\}$. Assumptions 1 and 2 are standard in convex analysis. Assumptions 3 and 4 bound the variations of stochastic gradients within and across the honest workers, respectively [28].

Assumption 1. *(Strong convexity and Lipschitz continuous gradients) Each $f_{w,j}$ is μ-strongly convex and has L-Lipschitz continuous gradients. That is, for any $x, y \in \mathbb{R}^p$, we have*

$$f_{w,j}(x) \ge f_{w,j}(y) + \langle f'_{w,j}(y), x - y\rangle + \frac{\mu}{2}\|x - y\|^2,$$

$$\|f'_{w,j}(x) - f'_{w,j}(y)\| \le L\|x - y\|.$$

Assumption 2. *(Bounded gradients) Each $f_{w,j}$ has bounded gradients. That is, for any $x \in \mathbb{R}^p$, we have $\|f'_{w,j}(x)\| \le r$.*

Assumption 3. *(Bounded inner variation) For any honest worker w and any $x \in \mathbb{R}^p$, the variation of its stochastic gradients with respect to its aggregated gradient is upper-bounded as $E_{i_w^k}\|f'_{w,i_w^k}(x) - f'_w(x)\|^2 \le \sigma^2$, $\forall w \notin B$.*

Assumption 4. *(Bounded outer variation) For any $x \in \mathbb{R}^p$, the variation of the aggregated gradients at the honest workers with respect to the overall gradient is upper-bounded as $E_{w\notin\mathcal{B}}\|f'_w(x) - f'(x)\|^2 \le \delta^2$.*

The following theorem shows Byrd-SAGA converges to a neighborhood of the optimal solution x^* at a linear rate, and the size of the neighborhood is determined by the number of Byzantine workers.

Theorem 1. *Under Assumptions 1–4, if the number of Byzantine workers $B < \frac{W}{2}$ and the step size $\gamma < \min\{\frac{2}{n\mu + 32C_\alpha^2 L}, \frac{1}{8LC_\alpha^2}\}$, then for Byrd-SAGA, it holds*

$$E\|x^k - x^*\|^2 \le (1 - \frac{\gamma\mu}{2})^k \Delta_1 + \Delta_2, \tag{7}$$

where $\Delta_1 := \|x^0 - x^*\|^2 + 2\gamma J\left[f(x^0) - f(x^*)\right] - \Delta_2$ and $\Delta_2 := \frac{8C_\alpha^2\delta^2\gamma}{\mu} + \frac{4}{\mu^2}C_\alpha^2(4\sigma^2 + 16r^2 + 2\delta^2)$.

5. NUMERICAL EXPERIMENTS

We conduct numerical experiments on convex and nonconvex learning problems. For each problem, evenly distribute the dataset into $W - B = 50$ honest workers. For the case with Byzantine attacks, we additionally launch $B = 20$ Byzantine workers. We test the performance of the proposed Byrd-SAGA under three typical Byzantine attacks: Gaussian, max-value and zero-gradient attacks [15, 29]. With Gaussian attack, every Byzantine worker $w \in \mathcal{B}$ generates its m_w^k following a Gaussian distribution with mean $\frac{1}{W-B}\sum_{w'\notin\mathcal{B}} m_{w'}^k$ and variance 30. With max-value attack, every Byzantine worker $w \in \mathcal{B}$ sets its message as

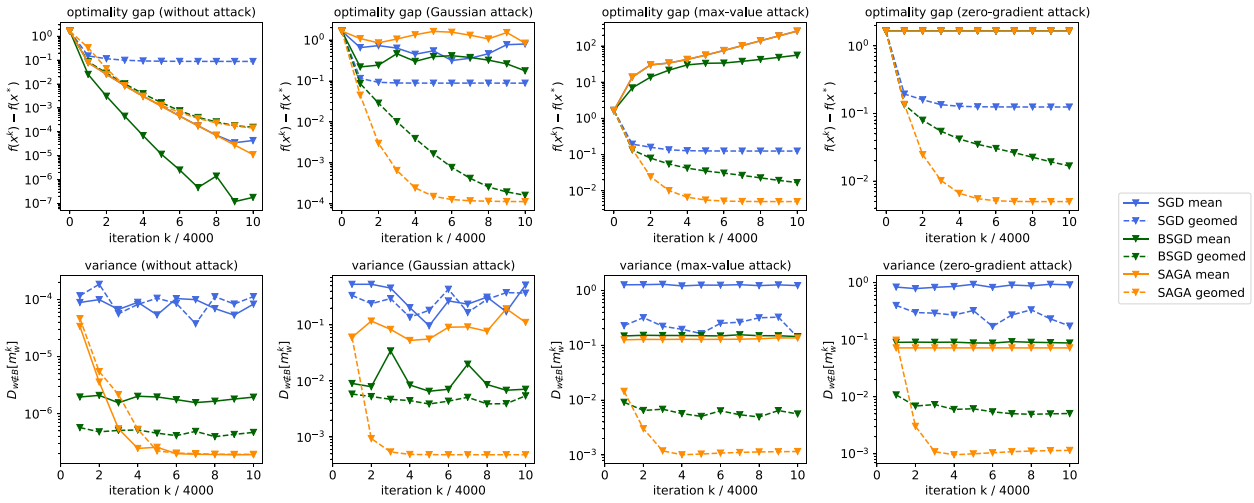

Fig. 2. Performance of the distributed SGD, mini-batch SGD (BSGD) and SAGA, with mean and geometric median (geomed) aggregation rules. Top to Bottom: optimality gap and variance of honest messages. Left to Right: without, Gaussian, max-value and zero-gradient attacks.

$m_w^k = u \cdot \frac{1}{W-B} \sum_{w' \notin \mathcal{B}} m_{w'}^k$, where the magnitude $u = 4$ is used in the numerical experiments. With zero-gradient attack, every Byzantine worker $w \in \mathcal{B}$ sends $m_w^k = -\frac{1}{B} \sum_{w' \notin \mathcal{B}} m_{w'}^k$ such that the messages received by the master node are summed to zero.

Table 1. Accuracy of SGD, mini-batch SGD (BSGD) and SAGA, with mean and geometric median (geomed) aggregation rules.

attack	algorithm	mean acc (%)	geomed acc (%)
without	SGD	97.0	92.3
	BSGD	98.6	98.0
	SAGA	96.5	96.3
Gaussian	SGD	36.3	92.5
	BSGD	36.3	98.0
	SAGA	14.5	96.4
max-value	SGD	0.11	0.03
	BSGD	0.16	90.3
	SAGA	0.12	86.4
zero-gradient	SGD	9.94	26.2
	BSGD	9.89	81.5
	SAGA	9.88	92.4

ℓ_2-regularized logistic regression. Consider the ℓ_2-regularized logistic regression problem, in which every $f_{w,j}(x)$ is in the form of

$$f_{w,j}(x) = \ln\left(1 + \exp\left(-b_{w,j}\langle a_{w,j}, x\rangle\right)\right) + \frac{\rho}{2}||x||^2,$$

where $a_{w,j} \in \mathbb{R}^p$ is the feature vector, $b_{w,j} \in \{-1, 1\}$ is the label, and $\rho = 0.01$ is a constant. We use the IJCNN1 dataset, which contains 49,990 training data samples of $p = 22$ dimensions.

We compare SGD, mini-batch SGD (BSGD) with batch size 50 and SAGA, using mean and geometric median aggregation rules. The step sizes are 0.02, 0.01 and 0.02, respectively. Comparing to SGD, BSGD enjoys smaller gradient noise but suffers from higher computational cost. In comparison, SAGA also reduces gradient noise, but its computational cost is in the same order as that of SGD. For every algorithm, we use the constant step size, which is tuned to achieve the best optimality gap $f(x^k) - f(x^*)$ for the Byzantine-free case. The performance of these algorithms on the IJCNN1 dataset is depicted in Fig. 2. With Byzantine attacks, the three algorithms using mean aggregation all fail. Among the three algorithms using geometric median aggregation, Byrd-SAGA remarkably outperforms the other two, while BSGD is better than SGD. This fact suggests that the importance of variance reduction to handling Byzantine attacks. To be specific, regarding the variance of honest messages, Byrd-SAGA, Byzantine-resilient BSGD and Byzantine-resilient SGD are in the order of 10^{-3}, 10^{-2} and 10^{-1}, respectively.

Neural network training. We carry out a set of numerical experiments on a neural network that has one hidden layer of 50 neurons with Tanh activation functions. We use this neural network for multi-class classification on the MNIST dataset, which has $60,000$ data with dimension $p = 784$. We compare SGD with step size 0.1, mini-batch SGD (BSGD) with step size 0.5 and batch size 50, and SAGA with step size 0.1. We run the algorithms for $15,000$ iterations and record the final accuracy in Table 1. With mean aggregation, all the algorithms yield low accuracy under Byzantine attacks. With the help of geometric median aggregation, BSGD and SAGA are both robust, and outperform SGD. Note that Byrd-SAGA has much lower per-iteration computational cost relative to Byzantine-resilient BSGD.

6. CONCLUSIONS

In this paper, we propose Byrd-SAGA, a Byzantine-resilient distributed SAGA to solve the distributed finite-sum optimization problem with Byzantine attacks. Similar to SAGA, Byrd-SAGA corrects the stochastic gradient through variance reduction. At every iteration, distributed workers calculate their corrected stochastic gradients and send to the master node. But unlike SAGA, in Byrd-SAGA the master node aggregates the received messages using geometric median, other than mean. This robust aggregation rule guarantees the robustness of Byrd-SAGA in presence of Byzantine attacks. Our future work is to develop and analyze Byzantine-resilient algorithms over decentralized networks [30, 31].

Acknowledgement. Qing Ling is supported in part by NSF China Grants 61573331 and 61973324, as well as Fundamental Research Funds for the Central Universities. Georgios B. Giannakis is supported in part by NSF Grants 1509040, 1508993 and 1711471.

7. REFERENCES

[1] R. Agrawal and R. Srikant, "Privacy-preserving data mining," In: Proceedings of SIGMOD, 2000

[2] J. Duchi, M. J. Wainwright, and M. I. Jordan, "Local privacy and minimax bounds: Sharp rates for probability estimation," In: Proceedings of NeurIPS, 2013

[3] L. Zhou, K. Yeh, G. Hancke, Z. Liu, and C. Su, "Security and privacy for the industrial Internet of Things: An overview of approaches to safeguard endpoints," IEEE Signal Processing Magazine, vol. 35, no. 5, pp. 76–87, 2018

[4] J. Konecny, H. B. McMahan, D. Ramage, and P. Richtarik, "Federated optimization: Distributed machine learning for on-device intelligence," arXiv Preprint arXiv:1610.02527, 2016

[5] A. Vempaty, L. Tong, and P. K. Varshney, "Distributed inference with Byzantine data: State-of-the-art review on data falsification attacks," IEEE Signal Processing Magazine, vol. 30, no. 5, pp. 65–75, 2013

[6] Y. Chen, S. Kar, and J. M. F. Moura, "The Internet of Things: Secure distributed inference," IEEE Signal Processing Magazine, vol. 35, no. 5, pp. 64–75, 2018

[7] Z. Yang, A. Gang, and W. U. Bajwa, "Adversary-resilient inference and machine learning: From distributed to decentralized," arXiv Preprint arXiv:1908.08649

[8] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," ACM Transactions on Programming Languages and Systems, vol. 4, no. 3, pp. 382–401, 1982

[9] S. Minsker, "Geometric median and robust estimation in Banach spaces," Bernoulli, vol. 21, no. 4, pp. 2308-C2335, 2015

[10] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," In: Proceedings of SIGMETRICS, 2019

[11] C. Xie, O. Koyejo, and I. Gupta, "Generalized Byzantine-tolerant SGD," arXiv Preprint arXiv:1802.10116, 2018

[12] D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," In Proceedings of ICML, 2018

[13] L. Su and J. Xu, "Securing distributed machine learning in high dimensions," arxiv Preprint arXiv:1804.10140, 2018

[14] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," In Proceedings of NeurIPS, 2017

[15] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," In Proceedings of AAAI, 2019

[16] C. Xie, O. Koyejo, and I. Gupta, "Fall of empires: Breaking Byzantine-tolerant SGD by inner product manipulation," arXiv Preprint arXiv:1903.03936, 2019

[17] P. Goyal, P. Dollar, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch SGD: Training imagenet in 1 hour," arXiv Preprint arXiv:1706.02677, 2017

[18] M. W. Schmidt, N. Le Roux, and F. R. Bach, "Minimizing finite sums with the stochastic average gradient," Mathematical Programming, vol. 162, no. 1–2, pp. 83–112, 2017

[19] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," In Proceedings of NeurIPS, 2013

[20] A. Defazio, F. R. Bach, and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives," In Proceedings of NeurIPS, 2014

[21] S. Shalev-Shwartz and T. Zhang, "Stochastic dual coordinate ascent methods for regularized loss minimization," Journal of Machine Learning Research, vol. 14, no. 2, pp. 567–599, 2013

[22] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takac, "SARAH: A novel method for machine learning problems using stochastic recursive gradient," In Proceedings of ICML, 2017

[23] Z. Allen-Zhu, "Katyusha: The first direct acceleration of stochastic gradient methods," Journal of Machine Learning Research, vol. 18, no. 1, pp. 8194–8244, 2017

[24] C. Calauzenes and N. Le Roux, "Distributed SAGA: Maintaining linear convergence rate with limited communication," arXiv Preprint arXiv:1705.10405, 2017

[25] S. De and T. Goldstein, "Efficient distributed SGD with variance reduction," In Proceedings of ICDM, 2016

[26] S. J. Reddi, A. Hefny, S. Sra, B. Poczos, and A. J. Smola, "On variance reduction in stochastic gradient descent and its asynchronous variants," In Proceedings of NeurIPS, 2015

[27] L. Bottou, "Large-scale machine learning with stochastic gradient descent," In Proceedings of COMPSTAT, 2010

[28] H. Tang, X. Lian, M. Yan, C. Zhang, and J. Liu, "D2: Decentralized training over decentralized data," In Proceedings of ICML, 2018

[29] F. Lin, Q. Ling, and Z. Xiong, "Byzantine-resilient distributed large-scale matrix completion," In Proceedings of ICASSP, 2019

[30] W. Ben-Ameur, P. Bianchi, and J. Jakubowicz, "Robust distributed consensus using total variation," IEEE Transactions on Automatic Control, vol. 61, no. 6, pp. 1550–1564, 2016

[31] Z. Yang and W. U. Bajwa, "BRIDGE: Byzantine-resilient decentralized gradient descent," arXiv Preprint arXiv:1908.08098, 2019

[32] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Springer, 2013