Distributed Approximate Newton's Method Robust to Byzantine Attackers

Xinyang Cao and Lifeng Lai

Abstract—**There is a recent surge of interest in the design of the first-order and the second-order distributed machine learning algorithms. However, distributed algorithms are sensitive to Byzantine attackers who can send falsified information to prevent the convergence of algorithms or lead the algorithms to converge to value of the attackers' choice. Some recent works have proposed algorithms that can defend against Byzantine attackers for the first-order methods. In this paper, we design two algorithms that can deal with Byzantine attackers for the second-order methods. The main idea of the first algorithm, named median-based approximate Newton's method (MNM), is to ask the parameter server to aggregate gradient information and approximate Newton's direction from all workers by geometric median. We show that MNM can converge when up to half of the workers are Byzantine attackers. To deal with the case with an arbitrary number of attackers, we then propose a comparison-based approximate Newton's method (CNM). The main idea of CNM is to ask the server to randomly select a small clean dataset and compute noisy gradient and Newton's direction using this small dataset. These noisy information will then be used as an approximation of the ground truth to filter out bad information from Byzantine attackers. We show that CNM can converge to the neighborhood of the population minimizer even when more than half of the workers are Byzantine workers. We further provide numerical examples to illustrate the performance of the proposed algorithms.**

I. Introduction

The big data problems are arising in science, engineering, Internet, etc, which produce computation and storage challenge in machine learning. Here we list some of them. First, as the amount of data keeps growing at a fast pace, it is challenging to fit all data in one machine [1]–[3]. Second, in certain scenarios, data is naturally collected at different locations, and it is too costly to move all data to a centralized location [4], [5]. To address these challenges and to harness the computing power of multiple machines, there is a growing interest in the design of distributed optimization algorithms [6]–[17]. In a typical distributed optimization setup, there are one parameter server and multiple workers. The whole dataset is divided into small parts and each part is kept in each worker. The server and workers exchange information through a network to collectively compute quantity of interest. However, the network typically has limited bandwidth and high latency. Therefore, there is a need to balance the cost of local computation and communication.

Xinyang Cao and Lifeng Lai are with Department of Electrical and Computer Engineering, University of California, Davis, CA, 95616. Email: {xycao,lflai}@ucdavis.edu. This work was supported by the National Science Foundation under grants CCF-17-17943, ECCS-17-11468, CNS-18-24553 and CCF-19-08258.

Various distributed first-order methods, which use gradient information and are often easy to implement, have been proposed in many existing works, such as distributed stochastic gradient descent (SGD) [9], [18], distributed variance reduced SGD [10], [19], [20], distributed coordinate descent method [1], [2] and dual coordinate ascent algorithms [21], [22] etc. These first-order methods significantly reduce the amount of local computation. But these algorithms may require a far greater number of iterations for communication. Some algorithms also require synchronization in every iteration for parameter updating.

In order to mitigate the negative impact of the large number of iterations for distributed optimization, communication-efficient second-order methods have also been proposed [23]–[28]. Shamir et al. [23] proposed DANE algorithm to minimize a cost function consisting of local loss function, local gradient and global gradient on each worker. AIDE in [24] applies the generic acceleration scheme (catalyst) in InexactDANE to improve the performance of DANE. DiSCO in [25] applies an inexact damped Newton method through preconditioned conjugated gradient method. Smith et al. [26] proposed CoCoA that involves sub-problems which are local quadratic approximations to the dual objective function. Wang [27] proposed GIANT that uses the average of inverse Hessian matrix times global gradient as the approximate Newton's direction. ADN in [28] is built on an adaptive block-separable approximation of the objective function.

Most of the existing works, both the first-order and the second-order methods, assume that these workers behave honestly and follow the protocol. However, in practice, there is a risk that some of the workers are Byzantine attackers. Byzantine attackers can prevent the convergence of the optimization algorithms or lead the algorithms to converge to values chosen by the attackers by modifying or falsifying intermediate results when the server require these intermediate results for updating. For example, as shown in [29], [30], for the first-order methods, the presence of even a single Byzantine worker can prevent the convergence of distributed gradient descent algorithm.

There have been some interesting recent works on designing distributed machine learning algorithms [29]–[47] that can deal with Byzantine attacks. The main idea of these works is to compare information received from all workers, and compute a quantity that is robust to attackers for algorithm update. However, these algorithms only consider the first-order methods.

In this paper, we propose two new robust distributed second-order methods that can converge to the neighborhood of the population minimizer.

The first method, named median-based approximate Newton's method (MNM), can converge to the neighborhood of the population minimizer when less than half of the workers are Byzantine attackers. The main idea is to use geometric median to aggregate information from workers. The geometric median enables the server to mitigate the impact of attackers when up to half of the workers are Byzantine attackers. Using these, we prove that the algorithm can converge to the neighborhood of the population minimizer when q, the number of Byzantine attackers, is less than $m/2$ with m being the total number of workers. We show this result by proving that the distance between the approximate Newton's direction and true Newton's direction can be universally bounded. However, once $q > m/2$, MNM fail to converge.

The second method, named comparison-based approximate Newton's Method (CNM), can converge to the neighborhood of the population minimizer server regardless whether q is larger or smaller than $m/2$. Compared with MNM, CNM requires additional computation at the server. The main idea is to ask server to randomly collect a small clean dataset and compute noisy value as an approximation of the ground truth to filter out information from attackers. In particular, when the server receives gradient from each worker, it will compute noisy gradient using the collected clean dataset, then compute the distance of them. If the distance is small, the server will accept the received gradient. After comparison, the server will build the global gradient by averaging the accepted gradients and its own noisy gradient, then broadcast it to all workers. Then the server will compute a noisy Newton's direction from the Hessian matrix using the collected dataset and global gradient. When receiving Newton's direction from workers, the server will compute the distance between received Newton's direction and the noisy Newton's direction and accept the received Newton's direction if the distance is small. Finally, the server computes the average of all accepted Newton's direction and its own noisy Newton's direction for updating. We prove that CNM can converge to the neighborhood of population minimizer regardless number of Byzantine attackers.

The paper is organized as follows. In Section II, we describe the model. In Section III, we describe the proposed algorithms. In Section IV, we analyze the convergence property of the proposed algorithms. In Section V, we provide numerical examples to validate the theoretic analysis. Finally, we offer several concluding remarks in Section VI. The proofs are collected in Appendix.

II. Model

In this section, we introduce our model. For random variable X with an unknown distribution $\mathcal{D}$, our goal is to infer the model parameter $\theta^* \in \mathbb{R}^d$ of the unknown distribution. It is popular to formulate this inference problem as an optimization problem

$$\theta^* \in \arg\min_{\theta\in\Theta} F(\theta) = \mathbb{E}\{f(X,\theta)\}, \tag{1}$$

in which $f : \mathcal{X} \times \Theta \to \mathbb{R}$ is the loss function, and the expectation is over the distribution $\mathcal{D}$. $F(\theta)$ is called population risk function.

In this paper, we make the following assumption about the population risk function $F(\theta)$.

Assumption 1. *The population risk function $F : \Theta \to \mathbb{R}$ is h-strongly convex, and differentiable over Θ with M-Lipschitz gradient. That is, for all $\theta, \theta' \in \Theta$,*

$$F(\theta') \geq F(\theta) + \langle \nabla F(\theta), \theta' - \theta \rangle + h \parallel \theta' - \theta \parallel^2 /2, \tag{2}$$

and

$$\parallel \nabla F(\theta') - \nabla F(\theta) \parallel \leq M \parallel \theta' - \theta \parallel,$$

in which $\parallel \cdot \parallel$ is the ℓ_2 norm and $0 < h \leq M$.

Since the expectation in (1) is over the unknown distribution $\mathcal{D}$, the population risk function $F(\theta)$ is unknown and hence we cannot solve (1) directly. Instead, one typically aims to minimize the empirical risk:

$$\min_{\theta\in\Theta} \frac{1}{N}\sum_{i=1}^{N} f(X_i, \theta), \tag{3}$$

where $X_1, X_2, ..., X_N$ are data samples generated by the unknown distribution $\mathcal{D}$. By solving (3), we obtain an estimate of the true model parameter θ^*.

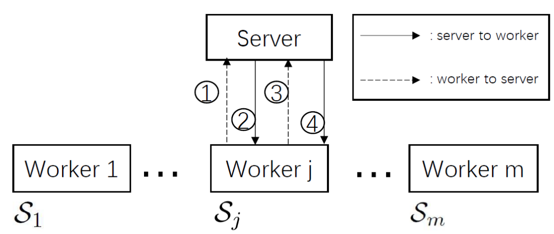

Fig. 1. Information flow of GIANT algorithm in [27]. ①: $\nabla\overline{f}^{(j)}(\theta_{t-1})$; ②: $\nabla\overline{f}(\theta_{t-1})$; ③ : $H^{-1}_{j,t-1}\nabla\overline{f}(\theta_{t-1})$; ④ : θ_t. In this figure we only draw the information flow between machine j and the server, all other machines have similar information flow.

When the number of data points N is large, we can employ distributed optimization methods, where there are one server and m workers in the system, to harness the computing power of multiple machines to solve (3). These N data points are divided and stored in m workers, and the server can communicate with all workers synchronously. Many distributed first-order [1], [2], [9], [10], [18], [19], [21], [22] and second-order optimization methods [23]–[28] have been proposed to solve (3).

In this paper, we focus on an approximate Newton's method, named global improved approximate Newton method (GIANT) proposed in [27]. We will let $\mathcal{S}_j$ be the set of data samples that are kept by the j-th worker. The GIANT algorithm solves (3) using approximate Newton's method by two-steps computing and communication between the server and workers. Figure 1 illustrates information flow between the server and workers during iteration t. In particular, at iteration

Algorithm 1: GIANT algorithm [27]

Parameter server:

Initialize randomly selects $\theta_0 \in \Theta$.

repeat

1: Broadcasts the current model parameter estimator θ_{t-1} to all workers;

2: Waits to receive gradients from the m workers;

3: Computes $\nabla \overline{f}(\theta_{t-1}) = \frac{1}{m}\sum_{j=1}^{m} \nabla \overline{f}^{(j)}(\theta_{t-1})$;

4: Broadcasts the current gradient estimator $\nabla \overline{f}(\theta_{t-1})$ to all workers;

5: Waits to receive estimators from the m workers;

6:Updates $\theta_t = \theta_{t-1} - \eta \frac{1}{m}\sum_{j=1}^{m} H_{j,t-1}^{-1} \nabla \overline{f}(\theta_{t-1})$;

until $\|\theta_t - \theta^*\| \le \epsilon$.

Worker j**:**

1: Receives model parameter estimator θ_{t-1}, computes the gradient $\nabla \overline{f}^{(j)}(\theta_{t-1})$, sends it back;

2: Receives gradient estimator $\nabla \overline{f}(\theta_{t-1})$, computes the parameter $H_{j,t-1}^{-1} \nabla \overline{f}(\theta_{t-1})$, sends it back;

t, each worker $j \in [1, m]$ first calculates $\nabla \overline{f}^{(j)}(\theta_{t-1})$ based on local data:

$$\nabla \overline{f}^{(j)}(\theta_{t-1}) = \frac{1}{|\mathcal{S}_j|}\sum_{i \in \mathcal{S}_j} \nabla f(X_i, \theta_{t-1}), \tag{4}$$

and sends it to the server, where $|\mathcal{S}_j|$ is the size of data in the j-th worker and we assume the size of data in each worker is equal. After receiving information from all workers, the server computes the gradient information using

$$\nabla \overline{f}(\theta_{t-1}) = \frac{1}{m}\sum_{j=1}^{m} \nabla \overline{f}^{(j)}(\theta_{t-1}), \tag{5}$$

and broadcasts the $\nabla \overline{f}(\theta_{t-1})$ to workers. After receiving $\nabla \overline{f}(\theta_{t-1})$, each worker $j \in [1, m]$ calculates $H_{j,t-1}^{-1} \nabla \overline{f}(\theta_{t-1})$ based on local data and $\nabla \overline{f}(\theta_{t-1})$ where

$$H_{j,t-1} = \nabla^2 \overline{f}^{(j)}(\theta_{t-1}) = \frac{1}{|\mathcal{S}_j|}\sum_{i \in \mathcal{S}_j} \nabla^2 f(X_i, \theta) \tag{6}$$

and sends it back to the server. After receiving Newton's direction information from all workers, the server updates model parameter by

$$\theta_t = \theta_{t-1} - \frac{1}{m}\sum_{j=1}^{m} H_{j,t-1}^{-1} \nabla \overline{f}(\theta_{t-1}), \tag{7}$$

and broadcasts the updated parameters to the workers. This process continues until a certain stop criteria is satisfied. Algorithm 1 summarizes steps involves in the GIANT algorithm.

When all workers are honest, this algorithm can converge fast [27]. However, taking average as in (5) and (7) has no ability to defend against attacks. In particular, even a single Byzantine worker can completely change the average value of gradient and Newton's direction, and thus foil the algorithm.

In this paper, we consider a system with Byzantine workers, in which an unknown subset of workers might be compromised. Furthermore, the set of compromised workers might change over time. In each iteration, if a worker is compromised, it can send arbitrary information to the server when sending gradient information and Newton's direction. In particular, let $\mathcal{B}_t$ denote the set of compromised workers at iteration t, the server receives data $g_1^{(j)}(\theta_{t-1})$ from the j-th worker with

$$g_1^{(j)}(\theta_{t-1}) = \begin{cases} \nabla \overline{f}^{(j)}(\theta_{t-1}) & j \notin \mathcal{B}_t \\ \star & j \in \mathcal{B}_t \end{cases}, \tag{8}$$

in which $\star$ denotes an arbitrary vector chosen by the attacker. After receiving $g_1^{(j)}$ from workers, the server computes and broadcasts

$$g(\theta_{t-1}) = Aggre_1(g_1^{(1)}(\theta_{t-1}), ..., g_1^{(m)}(\theta_{t-1})), \tag{9}$$

in which $Aggre_1(\cdot)$ depends on how the server aggregates gradient information from workers. Each worker then computes Newton's direction based on $g(\theta_{t-1})$. After workers send Newton's direction, the server receives data $g_2^{(j)}(\theta_{t-1})$ from j-th worker

$$g_2^{(j)}(\theta_{t-1}) = \begin{cases} H_{j,t-1}^{-1} g(\theta_{t-1}) & j \notin \mathcal{B}_t \\ \star & j \in \mathcal{B}_t \end{cases}. \tag{10}$$

The server finally computes the final update direction using

$$G(\theta_{t-1}) = Aggre_2(g_2^{(1)}(\theta_{t-1}), ..., g_2^{(m)}(\theta_{t-1})), \tag{11}$$

in which in which $Aggre_2(\cdot)$ depends on how the server processes $g_2^{(j)}(\theta_{t-1})$ from workers.

Note that if both $Aggre_1(\cdot)$ and $Aggre_2(\cdot)$ are mean functions, the algorithm is the same as the GIANT algorithm [27]. But as discussed above, the GIANT algorithm is not robust to adversary attacks. The goal of our paper is to design robust Newton's method algorithms, by designing proper $Aggre_1(\cdot)$ and $Aggre_2(\cdot)$, that can tolerate Byzantine attacks.

III. Algorithms

In this section, we describe two algorithms that can handle different number of Byzantine attackers. Let q be the number of attackers in the system. We will first describe our algorithm that can deal with $q < m/2$, i.e., up to half of the total number of workers are Byzantine attackers. We then describe our algorithm to deal with an arbitrary number of Byzantine attackers, i.e., there is no restrict on q. This algorithm requires additional computations at the server.

A. The Case with $q < m/2$

In the first scenario, we consider the case where there are at most $q < m/2$ Byzantine attackers. We propose a median-based approximate Newton's method (MNM). Main steps of the algorithm are listed in Algorithm 2.

Instead of computing the average, the main idea of our algorithm is to use geometric median of the received information as the aggregation function $aggre_1(\cdot)$ and $aggre_2(\cdot)$.

Algorithm 2: Median-based Approximate Newton's Method (MNM) Algorithm

Parameter server:
Initialize randomly selects $\theta_0 \in \Theta$.
repeat
1: Broadcasts the current model parameter estimator θ_{t-1} to all workers;
2: Waits to receive gradients from the m workers; $g^{(j)}(\theta_{t-1})$ denote the value received from worker j;
3: Computes $g(\theta_{t-1}) = med\{g_1^{(1)}(\theta_{t-1}), ..., g_1^{(m)}(\theta_{t-1})\}$;
4: Broadcasts the current gradient estimator $g(\theta_{t-1})$ to all workers;
5: Waits to receive estimators from the m workers; $g_2^{(j)}(\theta_{t-1})$ denote the value received from worker j;
6: Computes $G(\theta_{t-1}) = med\{g_2^{(1)}(\theta_{t-1}), ..., g_2^{(m)}(\theta_{t-1})\}$;
7:Updates $\theta_t = \theta_{t-1} - \eta G(\theta_{t-1})$;
until $\|\theta_t - \theta^*\| \leq \epsilon$.
Worker j**:**
1: Receives model parameter estimator θ_{t-1}, computes the gradient $\nabla \overline{f}^{(j)}(\theta_{t-1})$;
2: If worker j is honest, it sends $\nabla \overline{f}^{(j)}(\theta_{t-1})$; If not, it sends the value determined by the attacker;
3: Receives gradient estimator $g(\theta_{t-1})$, computes the parameter $H_{j,t-1}^{-1} g(\theta_{t-1})$;
4: If worker j is honest, it sends $H_{i,t-1}^{-1} g(\theta_{t-1})$ back to the server; If not, it sends the value determined by the attacker;

In particular, after receiving the gradient information from workers, the server computes

$$g(\theta_{t-1}) = med\{g_1^{(1)}(\theta_{t-1}), ..., g_1^{(m)}(\theta_{t-1})\}, \tag{12}$$

in which $med\{\cdot\}$ is the geometric median of the vectors. Geometric median is a generalization of median in one-dimension to multiple dimensions, and has been widely used in robust statistics. In particular, let $x_i \in \mathbb{R}^d, i = 1, \cdots, n$, then the geometric median of the set $\{x_1, x_2, ..., x_n\}$ is define as

$$med\{x_1, x_2, ..., x_n\} := \arg\min_x \sum_{i=1}^{n} \|x_i - x\|. \tag{13}$$

Then the server broadcasts value $g(\theta_{t-1})$ to all workers. After receiving the Newton's direction information, the server compute the final Newton's direction information by geometric median again,

$$G(\theta_{t-1}) = med\{g_2^{(1)}(\theta_{t-1}), ..., g_2^{(m)}(\theta_{t-1})\}. \tag{14}$$

Finally, the server uses $G(\theta_{t-1})$ to update parameter θ_{t-1},

$$\theta_t = \theta_{t-1} - \eta G(\theta_{t-1}). \tag{15}$$

Algorithm 3: Comparison-based approximate Newton's Method (CNM) Algorithm

Parameter server:
Initialize randomly selects $\theta_0 \in \Theta$.
repeat
1: Broadcasts the current model parameter estimator θ_{t-1} to all workers;
2: Waits to receive gradients from the m workers; $g^{(j)}(\theta_{t-1})$ denote the value received from worker j;
3: Accepts $g_1^{(j)}(\theta_{t-1})$ which pass test $\|g_1^{(j)}(\theta_{t-1}) - \nabla \overline{f}^{(0)}(\theta_{t-1})\| \leq \xi_1 \|\nabla \overline{f}^{(0)}(\theta_{t-1})\|$, consider them in set $\mathcal{A}^{(1)}$.
4: Computes $g(\theta_{t-1}) = \frac{1}{1+|\mathcal{A}^{(1)}|}(\sum_{i \in \mathcal{A}^{(1)}} g_1^{(i)}(\theta_{t-1}) + \nabla \overline{f}^{(0)}(\theta_{t-1}))$;
4: Broadcasts the current gradient estimator $g(\theta_{t-1})$ to all workers;
5: Accepts $g_2^{(j)}(\theta_{t-1})$ which pass test $\|g_2^{(j)}(\theta_{t-1}) - \tilde{H}_0^{-1} g(\theta_{t-1})\| \leq \xi_2 \|\tilde{H}_0^{-1} g(\theta_{t-1})\|$, consider them in set $\mathcal{A}^{(2)}$.
6: Computes $G(\theta_{t-1}) = \frac{1}{1+|\mathcal{A}^{(2)}|}(\sum_{i \in \mathcal{A}^{(2)}} g_2^{(i)}(\theta_{t-1}) + \tilde{H}_0^{-1} g(\theta_{t-1}))$;
7: Update model parameter $\theta_t = \theta_{t-1} - G(\theta_{t-1})$;
until $\|\theta_t - \theta^*\| \leq \epsilon$.
Worker j**:**
1: Receives model parameter estimator θ_{t-1}, computes the gradient $\nabla \overline{f}^{(j)}(\theta_{t-1})$;
2: If worker j is honest, it sends $\nabla \overline{f}^{(j)}(\theta_{t-1})$; If not, it sends the value determined by the attacker;
3: Receives gradient estimator $g(\theta_{t-1})$, computes the parameter $\tilde{H}_{j,t-1}^{-1} g(\theta_{t-1})$;
4: If worker j is honest, it sends $\tilde{H}_{i,t-1}^{-1} g(\theta_{t-1})$ back to the server; If not, it sends the value determined by the attacker;

B. The Case with an Arbitrary Number of Byzantine Attackers

The MNM algorithm described in Section III-A will converge if $q < m/2$, which will be shown in Section III. However, it will fail to converge once $q > m/2$. In this subsection, we propose another algorithm, named comparison-based approximate Newton (CNM) method, that converges for an arbitrary value of q, regardless whether q is larger or smaller than $m/2$. Compared with the MNM algorithm, the CNM algorithm needs additional computation at the server side. In particular, we assume that the server keep a small set of clear data to compute a noisy gradient and a noisy Newton's direction. These information, which are noisy version of the ground truth, will help the server make decision to whether accept information from each worker or not. Main steps of the algorithm are listed in Algorithm 3.

More specifically, in our algorithm, the server will randomly select a small set of data points $\mathcal{S}_0$ at the very beginning, where $|\mathcal{S}_0| \leq |\mathcal{S}_j|$ and $j \in [1, m]$. Once $\mathcal{S}_0$ is selected, it is fixed throughout the algorithm. Then at each iteration t, the server calculates a noisy gradient using data points in $\mathcal{S}_0$:

$$\nabla\overline{f}^{(0)}(\theta_{t-1}) = \frac{1}{|\mathcal{S}_0|}\sum_{j\in\mathcal{S}_0}\nabla f(X_j,\theta_{t-1}). \tag{16}$$

After computing $\nabla\overline{f}^{(0)}(\theta_{t-1})$, the server compares $g_1^{(j)}(\theta_{t-1})$ received from worker j with $\nabla\overline{f}^{(0)}(\theta_{t-1})$. The server will accept $g_1^{(j)}(\theta_{t-1})$ as authentic value and use it for further processing, if

$$\|g_1^{(j)}(\theta_{t-1}) - \nabla\overline{f}^{(0)}(\theta_{t-1})\| \le \xi_1\|\nabla\overline{f}^{(0)}(\theta_{t-1})\| \tag{17}$$

where ξ_1 is a constant. The server will collect all accepted $g_1^{(j)}(\theta_{t-1})$ in set $\mathcal{A}^{(1)}$. The main enabling observation is that, even though $\nabla\overline{f}^{(0)}(\theta_{t-1})$ is noisy, it is an approximation of the ground truth and hence can be used to filter out bad information from Byzantine workers as done in (17).

Then the server computes $g(\theta_{t-1})$ based on the accepted gradient information in set $\mathcal{A}^{(1)}$:

$$g(\theta_{t-1}) = \frac{1}{1+|\mathcal{A}^{(1)}|}\left(\sum_{i\in\mathcal{A}^{(1)}} g_1^{(i)}(\theta_{t-1}) + \nabla\overline{f}^{(0)}(\theta_{t-1})\right).$$

The server will broadcast $g(\theta_{t-1})$ to all workers, each of which will compute $\tilde{H}_{j,t-1}^{-1}g(\theta_{t-1})$, where $\tilde{H}_{j,t-1} = H_{j,t-1} + \mathbf{I}$ with $\mu \ge 0$ and $\mathbf{I}$ being the identity matrix. Here $\mu\mathbf{I}$ is added to make sure the matrix is invertible. The server also computes a noisy Newton's direction $\tilde{H}_0^{-1}g(\theta_{t-1})$, in which $\tilde{H}_0$ is computed using data points in $\mathcal{S}_0$:

$$\tilde{H}_0 = \frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0}\nabla^2 f(X_i,\theta_{t-1}) + \mu\mathbf{I}. \tag{18}$$

Then the server compares $g_2^{(j)}(\theta_{t-1})$ received from worker j with $\tilde{H}_0^{-1}g(\theta_{t-1})$. If the following condition is satisfied

$$\|g_2^{(j)}(\theta_{t-1}) - \tilde{H}_0^{-1}g(\theta_{t-1})\| \le \xi_2\|\tilde{H}_0^{-1}g(\theta_{t-1})\| \tag{19}$$

the server will collect $g_2^{(j)}(\theta_{t-1})$ in set $\mathcal{A}^{(2)}$. Here, ξ_2 is a constant. Then the server computes the final update direction:

$$G(\theta_{t-1}) = \frac{1}{1+|\mathcal{A}^{(2)}|}\left(\sum_{i\in\mathcal{A}^{(2)}} g_2^{(i)}(\theta_{t-1}) + H_0^{-1}g(\theta_{t-1})\right). \tag{20}$$

IV. Convergence Analysis

In this section, we analyze the convergence property of the proposed algorithms.

A. Convergence of MNM algorithm

In this section, we will prove results that hold simultaneously for all $\theta \in \Theta$ with a high probability. Hence, in the following, we will drop subscript $t-1$. Before presenting detailed analysis, here we describe the high level ideas. If $H^{-1}\nabla F(\theta)$ is available, where $H = \nabla^2 F(\theta)$, the Newton's method will converge to θ^*. The main idea of our proof is to show that the distance between $G(\theta)$ computed in (14) and $H^{-1}\nabla F(\theta)$ is universally bounded in Θ when the number of attackers is at most $m/2$. Hence, $G(\theta)$ is a good estimate of $H^{-1}\nabla F(\theta)$. As the result, we can then show that the proposed algorithm converge to the neighborhood of the population minimizer.

We first show that $\|G(\theta) - H^{-1}\nabla F(\theta)\|$ is universally bounded in Θ. To start with, we first write

$$\begin{aligned}
&\|H^{-1}\nabla F(\theta) - G(\theta)\| \\
&= \|H^{-1}\nabla F(\theta) - med\{g_2^{(1)}(\theta), ..., g_2^{(m)}(\theta)\}\| \\
&= \|Z(\theta) - H^{-1}g(\theta) + H^{-1}\nabla F(\theta)\| \\
&\le \|Z(\theta)\| + \|H^{-1}(g(\theta) - \nabla F(\theta))\|, \\
&\le \|Z(\theta)\| + \|H^{-1}J(\theta)\|,
\end{aligned} \tag{21}$$

where

$$\begin{aligned}
Z(\theta) &= med\{H^{-1}g(\theta) - g_2^{(1)}(\theta), ..., H^{-1}g(\theta) - g_2^{(m)}(\theta)\} \\
&= med\{Z_1(\theta), ..., Z_m(\theta)\},
\end{aligned} \tag{22}$$

and

$$\begin{aligned}
J(\theta) &= med\{\nabla F(\theta) - g_1^{(1)}(\theta), ..., \nabla F(\theta) - g_1^{(m)}(\theta)\} \\
&= med\{J_1(\theta), ..., J_m(\theta)\}.
\end{aligned} \tag{23}$$

To further bound the terms in (21), we need to present several assumptions and intermediate results. These assumptions are similar to those used in [29], [33], [36], and proofs of some lemmas follow closely that of [29].

Assumption 2. *There exist positive constants σ_1 and α_1 such that for any unit vector $v \in B$, $\langle\nabla f(X,\theta^*), v\rangle$ is sub-exponential with σ_1 and α_1, that is,*

$$\sup_{v\in B}\mathbf{E}[\exp(\lambda\langle\nabla f(X,\theta^*), v\rangle)] \le e^{\sigma_1^2\lambda^2/2}, \forall|\lambda| \le 1/\alpha_1,$$

where B denotes the unit sphere $\{v : \|v\|_2 = 1\}$.

Second, we define gradient difference $w(x,\theta) = \nabla f(x,\theta) - \nabla f(x,\theta^*)$ and assume that for every θ, $w(x,\theta)$ normalized by $\|\theta - \theta^*\|$ is also sub-exponential.

Assumption 3. *There exist positive constants σ_2 and α_2 such that for any $\theta \in \Theta$ with $\theta \ne \theta^*$ and any unit vector $v \in B$, $\langle w(X,\theta) - \mathbf{E}[w(X,\theta)], v\rangle / \|\theta - \theta^*\|$ is sub-exponential with σ_2 and α_2, that is,*

$$\begin{aligned}
&\sup_{\theta\in\Theta, v\in B}\mathbf{E}\left[\exp\left(\frac{\lambda\langle w(X,\theta) - \mathbf{E}[w(X,\theta)], v\rangle}{\|\theta-\theta^*\|}\right)\right] \\
&\le e^{\sigma_2^2\lambda^2/2}, \ \forall|\lambda| \le \frac{1}{\alpha_2}.
\end{aligned} \tag{24}$$

This allows us to show that $\frac{1}{|\mathcal{S}_0|}\sum_{i\in\mathcal{S}_i} w(X_i,\theta)$ concentrates on $\mathbf{E}[w(X,\theta)]$ for every fixed θ.

Assumptions 2 and 3 ensure that random gradient $\nabla f(\theta)$ has good concentration properties, i.e., an average of $|\mathcal{S}_i|$ $i.i.d$ random gradients $\frac{1}{|\mathcal{S}_i|}\sum_{i\in\mathcal{S}_i}\nabla f(X_i,\theta)$ sharply concentrates on $\nabla F(\theta)$ for every fixed θ.

We also assume data in each worker $j \in [1,m]$ has following assumption

Assumption 4. *For any $\delta \in (0, 1/|\mathcal{S}_j|)$, there exists an $M' = M'(\delta)$ and $h' = h'(\delta)$ such that*

$$\mathbf{Pr}\left\{\forall \theta, \theta' \in \Theta, h' \leq \frac{\|\nabla f(X,\theta) - \nabla f(X,\theta')\|}{\|\theta - \theta'\|} \leq M'\right\} \geq 1 - \frac{\delta}{3}. \quad (25)$$

Assumption 4 ensures that $\nabla f(X,\theta)$ in each worker is M'-Lipschitz and $f(X,\theta)$ is h' strongly convex with high probability.

Now, we make another standard assumption in analyzing Newton's method for population risk.

Assumption 5. *The Hessian matrix $\nabla^2 F(\theta)$ is L-Lipschitz continuous,i.e, there exists an L such that for $\theta, \theta' \in \Theta$*

$$\|\nabla^2 F(\theta) - \nabla^2 F(\theta)\|_2 \leq L\|\theta - \theta'\|,$$

in which $\|\cdot\|_2$ is the matrix spectral norm.

With these assumptions, we are ready to state our universal bound for $\|Z(\theta)\|$ and $\|H^{-1}J(\theta)\|$.

From (22), we need to bound the geometric median $Z(\theta)$ of $Z_1(\theta), ..., Z_m(\theta)$. We will use the following property of the geometric median from [48].

Lemma 1. *[48] Let $x_1, x_2, ..., x_n$ be n points in a Hilbert space. Let x^* denote the geometric median of these points. For any $\alpha \in (0, 1/2)$, and given $r > 0$, if $\sum_{i=1}^n \mathbf{1}_{\{\|x_i\| \leq r\}} \geq (1-\alpha)n$, then*

$$\|x^*\| \leq \mathcal{C}_\alpha r, \quad (26)$$

where

$$\mathcal{C}_\alpha = \frac{2(1-\alpha)}{1-2\alpha}. \quad (27)$$

From Lemma 1, we can see that, if majority of points are inside the Euclidean ball of radius r centered at origin, then the geometric median must be inside the Euclidean ball of radius $\mathcal{C}_\alpha r$. To use this lemma, we need to show more than half of information received by the server are bounded by some quantity.

We first have the following lemma regarding the spectral norm of $H_i - H$.

Lemma 2. *If Assumption 4 holds, for any $\delta \in (0,1)$, with probability at least $1 - \frac{\delta}{3}$, $|\mathcal{S}_j|$ data satisfy*

$$h' \leq \|\nabla^2 f(X,\theta)\|_2 \leq M', \quad (28)$$

for any $\delta'_3 \in (0,1)$, let

$$\Delta_3 = \sqrt{\frac{14(M \vee M')^2 \log(2d/\delta'_3)}{3|\mathcal{S}_i|}}, \quad (29)$$

then

$$\mathbf{Pr}\left\{\|H_i - H\|_2 \leq \Delta_3\right\} \geq 1 - \delta_2. \quad (30)$$

with $\delta_2 = \delta'_3 + \frac{\delta}{3}$ and $\delta_2 \in (0,1)$.

Proof. Please see Appendix A for detail. □

Now, with these lemmas and assumptions, when worker i is honest, we can show the bound for Z_i.

Proposition 1. *Suppose Assumptions 1-4 hold, and $\Theta \subset \{\theta : \|\theta - \theta^*\| \leq r\sqrt{d}\}$ for some $r > 0$. For any $\delta_3 \in (0,1), \alpha \in (q/m, 1/2)$ and $\delta_4 = \delta_2 + e^{-mD(\alpha - q/m\|\delta_3)}$,*

$$\mathbf{Pr}\left\{\forall \theta : \|Z_i(\theta)\| \leq \left(\frac{8\mathcal{C}_\alpha \Delta_3 \Delta_2}{hh'} + \frac{\Delta_3 M}{hh'}\right)\|\theta - \theta^*\| + \frac{4\mathcal{C}_\alpha \Delta_3 \Delta_1}{hh'}\right\} \geq 1 - \delta_4. \quad (31)$$

where $\Delta_1 = \sqrt{2}\sigma_1\sqrt{(d\log 6 + \log(6/\delta_3))/|\mathcal{S}_i|}$, $\Delta_2 = \sqrt{2}\sigma_2\sqrt{(\tau_1+\tau_2)/|\mathcal{S}_i|}$, with $\tau_1 = d\log 18 + d\log(M \vee M'/\sigma_2)$, $\tau_2 = 0.5d\log(|\mathcal{S}_i|/d) + \log(6/\delta_3) + \log(\frac{2r\sigma_2^2\sqrt{|\mathcal{S}_i|}}{\alpha_2\sigma_1})$, $\mathcal{C}_\alpha = \frac{2(1-\alpha)}{1-2\alpha}$ and $D(\delta'\|\delta) = \delta'\log\frac{\delta'}{\delta} + (1-\delta')\log\frac{1-\delta'}{1-\delta}$.

Proof. Please see Appendix B for details. □

Now we have already shown that for honest workers, the local Newton's direction received at the server is uniformly close to the true Newton's direction. Now using Lemma 1, we can show the median $Z(\theta)$ is bounded.

Proposition 2. *Suppose Assumptions 1-4 hold, and $\Theta \subset \{\theta : \|\theta - \theta^*\| \leq r\sqrt{d}\}$ for some $r > 0$. For any $\alpha \in (q/m, 1/2)$ and $0 < \delta_4 < \alpha - q/m$,*

$$\mathbf{Pr}\left\{\forall \theta : \|Z(\theta)\| \leq \left(\frac{8\mathcal{C}_\alpha \Delta_3 \Delta_2}{hh'} + \frac{\Delta_3 M}{hh'}\right)\mathcal{C}_\alpha\|\theta - \theta^*\| + \frac{4\mathcal{C}_\alpha^2 \Delta_3 \Delta_1}{hh'}\right\} \geq 1 - e^{-mD(\alpha - q/m\|\delta_4)}. \quad (32)$$

Proof. Please see Appendix C. □

With Proposition 2, we are ready to show that $G(\theta)$ is a good approximation of $H^{-1}\nabla F(\theta)$ from (21), and show the convergence of the proposed MNM algorithm.

Theorem 1. *If Assumptions 1-4 hold, and $\Theta \subset \{\theta : \|\theta - \theta^*\| \leq r\sqrt{d}\}$ for some $r > 0$, then for any $0 < \eta \leq 1$, $\alpha \in (q/m, 1/2)$, $0 < \delta_3 < \alpha - q/m$ and $0 < \delta_4 < \alpha - q/m$ with probability at least $1 - e^{-mD(\alpha - q/m\|\delta_3)} - e^{-mD(\alpha - q/m\|\delta_4)}$ that*

$$\|\theta_t - \theta^*\| \leq \rho\|\theta_{t-1} - \theta^*\| + \frac{\eta L\|\theta_{t-1} - \theta^*\|^2}{2h} + \frac{4}{hh'}\mathcal{C}_\alpha^2\Delta_3\Delta_1 + \eta\frac{4}{h}\mathcal{C}_\alpha\Delta_1,$$

where

$$\rho = 1 - \eta + \eta\frac{8}{hh'}\mathcal{C}_\alpha^2\Delta_3\Delta_2 + \eta\frac{8}{h}\mathcal{C}_\alpha\Delta_2 + \eta\mathcal{C}_\alpha\frac{\Delta_3 M}{hh'} \quad (33)$$

Proof. Please see Appendix D. □

This theorem shows that under an event that happens with a high probability, the estimated θ can converge to the neighborhood of θ^* with a linear-quadratic rate. Since we consider $\theta^* \in \arg\min_{\theta\in\Theta} F(\theta)$, there is always a gap between estimator θ and θ^*. This gap is due to the approximation error introduced by solving (3), instead of (1).

B. Convergence of CNM algorithm

In this section, we prove the convergence of CNM algorithm regardless the number of Byzantine attackers. In other words, q could be larger than $m/2$. Towards this goal, we will show that the distance between $G(\theta)$ defined in (20) and $H^{-1}\nabla F(\theta)$ is universally bounded in Θ regardless the number of attackers. As the result, $G(\theta)$ is a good estimate of $H^{-1}\nabla F(\theta)$. Finally, we will show that the proposed algorithm converge to the neighborhood of minimizer of the population risk.

Lemma 3. *For an arbitrary number of attackers, the distance between $G(\theta)$ and $H^{-1}\nabla F(\theta)$ is bounded by*

$$\begin{aligned} & \|H^{-1}\nabla F(\theta) - G(\theta)\| \\ < \quad & (1+\xi_2)\|\tilde{H}_0^{-1}\|_2\|g(\theta) - \nabla F(\theta)\| \\ + \quad & \xi_2\|\tilde{H}_0^{-1}\|_2\|\nabla F(\theta)\| + \|H^{-1}\nabla F(\theta) - \tilde{H}_0^{-1}\nabla F(\theta)\|. \end{aligned} \tag{34}$$

Proof. Please see Appendix E. □

Now, in order to bound the distance between $G(\theta)$ and $H^{-1}\nabla F(\theta)$, we need to bound the three terms in the right hand side of (34).

For the second term, from Assumption 1, we have $\|\nabla F(\theta)\| = \|\nabla F(\theta) - \nabla F(\theta^*)\| \leq M\|\theta - \theta^*\|$, since $\nabla F(\theta^*) = 0$.

For the third term, we have $\|(H^{-1} - \tilde{H}_0^{-1})\nabla F(\theta)\| = \|(\mathbf{I} - \tilde{H}_0^{-1}H)H^{-1}\nabla F(\theta)\| \leq \|\mathbf{I} - \tilde{H}_0^{-1}H\|_2\|H^{-1}\|_2\|\nabla F(\theta)\|$.

Then, we use the following lemma to bound $\|\mathbf{I} - \tilde{H}_0^{-1}H\|_2$.

Lemma 4. *If $\|H_0 - H\|_2 \leq \beta$ and $\beta < \frac{h(h+\mu)}{3h+2\mu}$,*

$$\|\mathbf{I} - \tilde{H}_0^{-1}H\|_2 \leq \frac{\mu}{h+\mu} + \frac{2\beta}{h+\mu-\beta} < 1. \tag{35}$$

Proof. Please see Appendix F. □

From this lemma, we have that $\|\mathbf{I} - \tilde{H}_0^{-1}H\|_2$ is bounded by a constant value smaller than 1, when $\|H_0 - H\|_2$ is bounded.

Proposition 3. *Suppose Assumptions 1-4 hold, and $\Theta \subset \{\theta : \| \theta - \theta^* \| \leq r\sqrt{d}\}$ for some $r > 0$, and $\Delta_3 < \frac{h(h+\mu)}{3h+2\mu}$. For any $\delta \in (0,1), \delta_3' \in (0,1), \delta_2 \in (0,1)$ and $\delta_2 = \frac{\delta}{3} + \delta_3'$*

$$\begin{aligned} & \mathbf{Pr}\left\{\forall\theta : \|(H^{-1} - \tilde{H}_0^{-1})\nabla F(\theta)\| \leq \frac{\Delta_4 M}{h}\|\theta - \theta^*\|\right\} \\ \geq \quad & 1 - \delta_2, \end{aligned} \tag{36}$$

with

$$\Delta_4 = \frac{\mu}{h+\mu} + \frac{2\Delta_3}{h+\mu-\Delta_3}, \tag{37}$$

and

$$\Delta_3 = \sqrt{\frac{14M^2\log(2d/\delta_3')}{3|\mathcal{S}_0|}}. \tag{38}$$

Proof. Please see Appendix G. □

Using these intermediate results, we have the following convergence result.

Theorem 2. *If Assumptions 1-5 hold, and $\Theta \subset \{\theta : \| \theta - \theta^* \| \leq r\sqrt{d}\}$ for some $r > 0$, $\mu \geq 0$ and $|\mathcal{S}_0|$ is sufficiently large, then for arbitrary number of attackers with probability at least $1 - \delta_5 - \delta_2$ that*

$$\|\theta_t - \theta^*\| \leq \frac{L}{2h}\|\theta_{t-1} - \theta^*\|^2 + \gamma_1\|\theta_{t-1} - \theta^*\| + \eta\gamma_2. \tag{39}$$

where $\Delta_4 = \frac{\mu}{h+\mu} + \frac{2\Delta_3}{h+\mu-\Delta_3}$, and

$$\gamma_1 = \left[(8\Delta_2 + \xi_1(8\Delta_2 + M))\frac{1+\xi_2}{h'+\mu} + \frac{\xi_2 M}{h'+\mu} + \frac{\Delta_4}{h}\right], \tag{40}$$

and

$$\gamma_2 = (4\Delta_1 + \xi_1 4\Delta_1)\frac{1+\xi_2}{h'+\mu}. \tag{41}$$

Proof. Please see Appendix H. □

This theorem shows that, with high probability, the estimated θ can converge to the neighborhood of θ^* with a linear-quadratic rate when there are arbitrary number of Byzantine attackers. Since we consider $\theta^* \in \arg\min_{\theta\in\Theta} F(\theta)$, we can only use empirical risk to approximate population risk, there is always a gap between estimator θ and θ^*.

V. Numerical results

In this section, we provide numerical examples, with both synthesized data and real data, to illustrate the performance of the proposed algorithms.

A. Synthesized data

We first use synthesized data. In this example, we focus on linear regression, in which

$$Y_i = X_i^T\theta^* + \epsilon_i, i = 1, 2, \cdots, N,$$

where $X_i \in \mathbb{R}^d$, θ^* is a $d \times 1$ vector and ϵ_i is the noise. We set $\mathbf{X} = [X_1, \cdots, X_N]$ as $d \times N$ data matrix.

In the simulation, we set the dimension $d = 20$, the total number of data $N = 50000$. We use $\mathcal{N}(0, 9)$ to independently generate each entry of θ^*. Here $\mathcal{N}(\nu, \sigma^2)$ denotes Gaussian variables with mean ν and variance σ^2. After θ^* is generated, we fix it. The data matrix $\mathbf{X}$ is generated randomly by Gaussian distribution with $\nu = 0$ and fixed known maximal and minimal eigenvalues of the correlation matrix $\mathbf{X}^T\mathbf{X}$. Let $\lambda_{max}(\cdot)$ and $\lambda_{min}(\cdot)$ denote the maximal and minimal eigenvalue of $\mathbf{X}^T\mathbf{X}$ respectively. In the following figures, we use $\lambda_{max}(\mathbf{X}^T\mathbf{X}) = 200$ and $\lambda_{min}(\mathbf{X}^T\mathbf{X}) = 2$ to generate the data matrix $\mathbf{X}$. We set ϵ_i as i.i.d. $\mathcal{N}(0, 1)$ random variable. Finally, we generate Y_i using the linear relationship mentioned above. In the simulation, we set the number of workers $m = 50$, and evenly distribute data among these machines. Furthermore, for robust gradient descent in [40] and proposed algorithm CNM, we set $|\mathcal{S}_0| = 1000$, $\xi = 1.5|\mathcal{S}_0|^{-\frac{1}{4}} = 0.2667, \xi_1 = 0.2667$ and $\xi_2 = 0.2667$. For the GIANT algorithm in [27] and proposed MNM, we set $\eta = 1$. For CNM, we set $\mu = 0.001$. We illustrate our results with 4 different cases: 1) 20 Inverse attack, in which each attacker first calculates the gradient and Newton's direction based on its local data but sends the inverse version of gradient information or vector information to the server; 2) 45 Inverse attack; 3) 20 Random attack, in which the attacker randomly generates gradient value; and 4) 45 Random attack. In our simulation, we compare four algorithms: 1) MNM in Table 2; 2) CNM described in Table 3; 3) Algorithm proposed in [40]; and 4) The GIANT algorithm proposed in [27]. The algorithm proposed in [40] is a first-order method which is robust to Byzantine attackers.

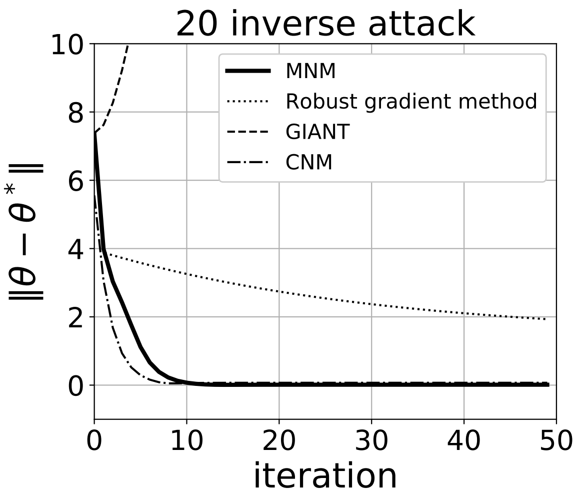

Fig. 2. Synthesized data: 20 Inverse attack. Robust gradient method in [40], GIANT in [27]

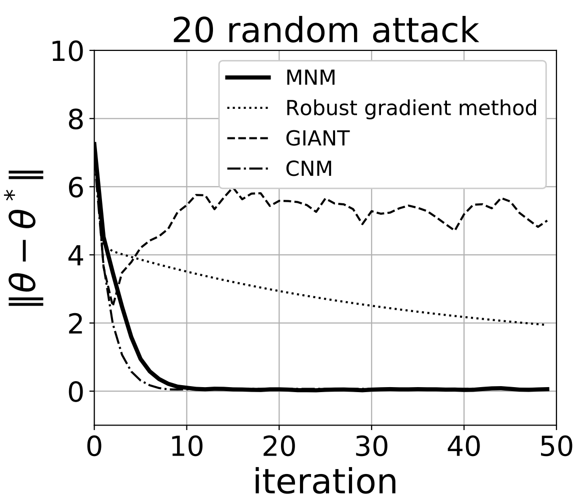

Fig. 3. Synthesized data: 20 Random attack. Robust gradient method in [40], GIANT in [27]

Figures 2 and 3 plot the value of the norm of distance between estimated and the true parameter vs iteration with 20 inverse attacks and 20 random attacks respectively. From Figures 2 and 3, GIANT method does not converge, since computing average cannot defend Byzantine attacks, but the proposed MNM, CNM and robust gradient method can still converge. Furthermore, the proposed two algorithms still perform better than the robust gradient method in [40] in iteration, since our proposed algorithms compute Hessian matrix on each worker, which generate more information in each communication iteration.

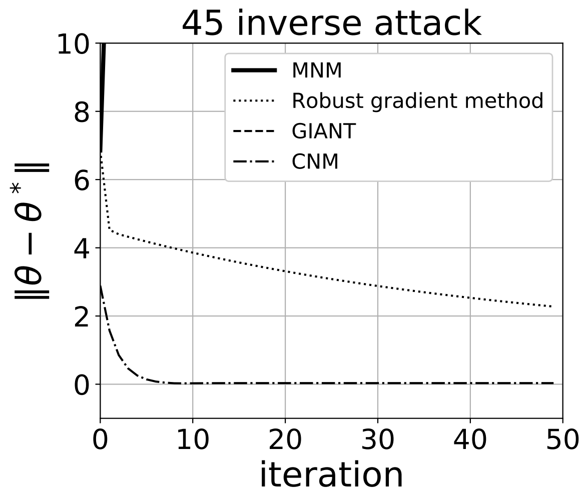

Fig. 4. Synthesized data: 45 Inverse attack. Robust gradient method in [40], GIANT in [27]

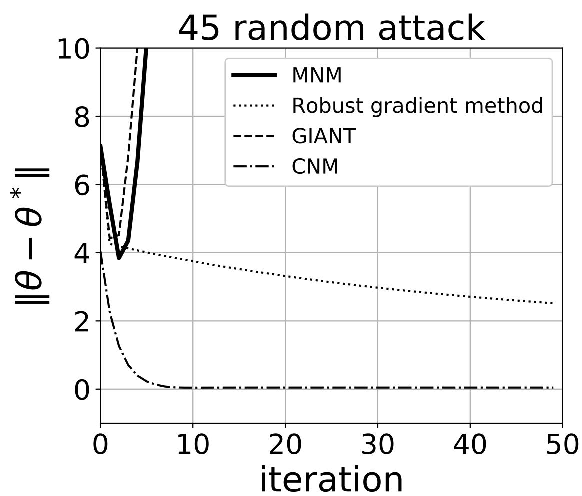

Fig. 5. Synthesized data: 45 Random attack. Robust gradient method in [40], GIANT in [27]

Figures 4 and 5 plot the value of the norm of distance between the estimated and the true parameter vs iteration with 45 inverse attacks and 45 random attacks. From Figures 4 and 5, we can observe that GIANT and MNM do not converge, as more than half of the workers are compromised. However the proposed CNM and robust gradient method can still converge. Furthermore, the proposed CNM can benefit from Newton's direction information and outperforms the robust gradient method in [40] in iteration.

B. Real data

Now we test our algorithms on real datasets MNIST [49] and compare our algorithms with various existing gradient method work [40] and GIANT. MNIST is a widely used computer vision dataset that consists of 70,000 28×28 pixel images of handwritten digits 0 to 9. We use the handwritten images of 3 and 5, which are the most difficult to distinguish in this dataset, to build a logistic regression model. After picking all 3 and 5 images from the dataset, the total number of images is 13454. It is divided into a training subset of size 12000 and a testing subset of size 1454. For the dataset, we set the number of workers to be 50. For algorithm in [40] and algorithm CNM, we random pick 200 images from both subsets to build $\mathcal{S}_0$, For the proposed MNM and GIANT in [27], we set the learning rate $\eta = 1$. For CNM, we set $\mu = 0.0001$. Similar to the synthesized data scenario, we illustrate our results with four cases, namely 20 inverse attack, 20 random attack, 45 inverse attack and 45 random attack, and compare the performance of four algorithms. The following figures show how the testing accuracy varies with training iteration.

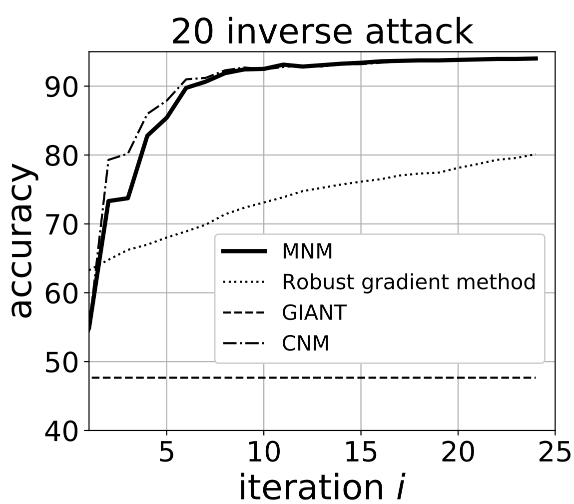

Fig. 6. MNIST: 20 Inverse attack. Robust gradient method in [40], GIANT in [27]

Figures 6 and 7 illustrate the impact of two cases on different algorithms using MNIST respectively. Figures 6 and 7 show the GIANT fails to predict if there are 20 attackers. Our proposed algorithm and robust gradient descent still show high accuracy. Furthermore, the proposed MNM has a better performance than robust gradient descent in [40].

We plot the impact of 45 attacker case on real data in Figures 8 and 9 using MNIST respectively. When there are 45 attackers, which is more than half of the total number of workers, MNM and GIANT can not properly work. CNM and robust gradient descent [40] still perform well, since these algorithm are generated to defend arbitrary number of attackers. Our proposed algorithms outperform the scheme using robust gradient descent in iteration.

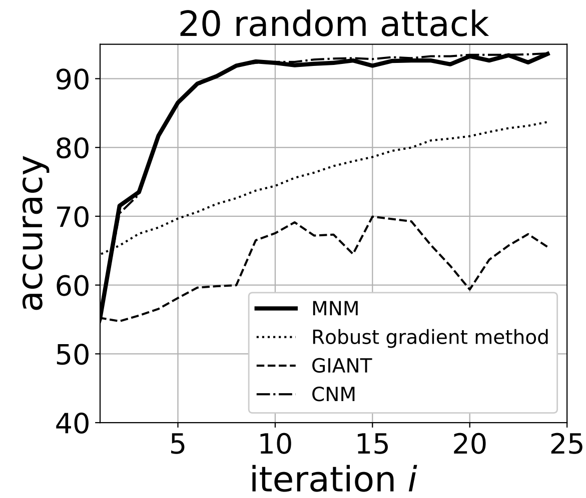

Fig. 7. MNIST: 20 Randome attack. Robust gradient method in [40], GIANT in [27]

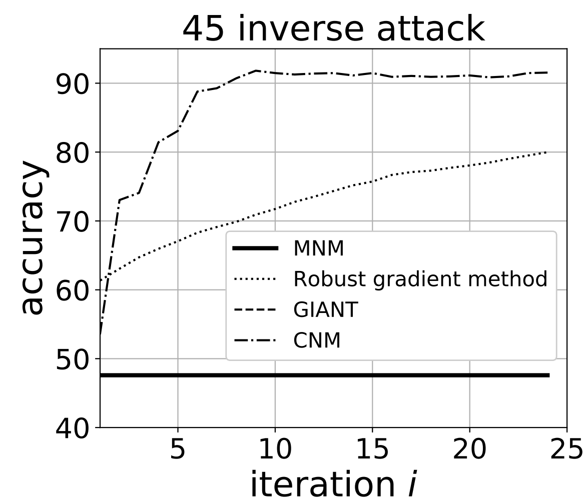

Fig. 8. MNIST: 45 Inverse attack. Robust gradient method in [40], GIANT in [27]

VI. Conclusion

In this paper, we have proposed two robust distributed approximate Newton's method that can tolerant Byzantine attackers. We have shown that the proposed algorithms can converges to the neighborhood of true parameter and have provided numerical examples to illustrate the performance of the proposed algorithm.

Appendix A
Proof of Lemma 2

When Assumption 4 holds, from union bound theorem, for any $\delta \in (0,1)$, with probability at least $1-\frac{\delta}{3}$, $|\mathcal{S}_j|$ data satisfy

$$h' \leq \|\nabla^2 f(X,\theta)\|_2 \leq M', \tag{42}$$

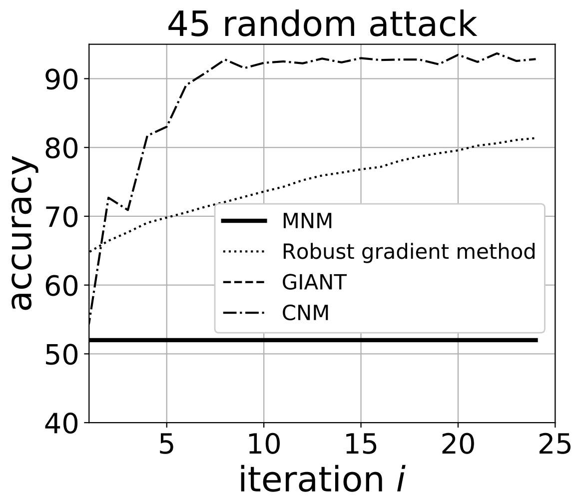

Fig. 9. MNIST: 45 Random attack. Robust gradient method in [40], GIANT in [27]

When $\|\nabla^2 f(X,\theta)\|_2 \le M'$, we have

$$H_i - H = \sum_{j\in\mathcal{S}_i} \frac{1}{|\mathcal{S}_i|}(\nabla^2 f(X_j,\theta) - H), \tag{43}$$

and

$$\begin{aligned}\left\|\frac{1}{|\mathcal{S}_i|}(\nabla^2 f(X_j,\theta) - H)\right\|_2 &\le \frac{1}{|\mathcal{S}_i|}(\|\nabla^2 f(X_j,\theta)\|_2 + \|H\|_2)\\ &\le \frac{2(M\vee M')}{|\mathcal{S}_i|}.\end{aligned} \tag{44}$$

Before proceed further, we define matrix variance statistic $v(Y)$ of a random Hermitian matrix with zero mean Y as

$$v(Y) = \|Var(Y)\|_2 = \|\mathbf{E}[(Y - E[Y])^2]\|_2 = \|\mathbf{E}[Y^2]\|_2.$$

Using this definition, we have

$$\begin{aligned} v(H_i - H) &= \left\|\sum_{j\in\mathcal{S}_i}\mathbf{E}\left[\frac{1}{|\mathcal{S}_i|}(\nabla^2 f(X_j,\theta)-H)^2\right]\right\|_2 \\ &= \left\|\sum_{j\in\mathcal{S}_i}\frac{1}{|\mathcal{S}_i|^2}\mathbf{E}\left[(\nabla^2 f(X_j,\theta)-H)^2\right]\right\|_2 \\ &\le \left\|\sum_{j\in\mathcal{S}_i}\frac{1}{|\mathcal{S}_i|^2}\mathbf{E}[\nabla^2 f(X_j,\theta)^2]\right\|_2 \\ &\le \frac{1}{|\mathcal{S}_i|}\mathbf{E}\left\|\nabla^2 f(X_j,\theta)^2\right\|_2 \\ &= \frac{1}{|\mathcal{S}_i|}\mathbf{E}\left\|\nabla^2 f(X_j,\theta)\right\|_2^2 \\ &\le \frac{(M\vee M')^2}{|\mathcal{S}_i|}. \end{aligned} \tag{45}$$

Since $H = \mathbf{E}[H_i]$, for $0 \le \gamma \le 2(M\vee M')$, we can use Matrix Bernstein inequality from [50] to get

$$\begin{aligned} &\mathbf{Pr}\left\{\|H_i - H\|_2 \ge \gamma\right\} \\ \le\ & 2d\exp\left(\frac{-\gamma^2/2}{v(H_i-H) + 2(M\vee M')\gamma/3|\mathcal{S}_i|}\right) \\ \le\ & 2d\exp\left(\frac{-\gamma^2/2}{(M\vee M')^2/|\mathcal{S}_i| + 2(M\vee M')\gamma/3|\mathcal{S}_i|}\right) \\ \le\ & 2d\exp\left(\frac{-3\gamma^2|\mathcal{S}_i|}{14(M\vee M')^2}\right). \end{aligned} \tag{46}$$

By picking $\Delta_3 = \gamma = \sqrt{\frac{14(M\vee M')^2\log(2d/\delta_3')}{3|\mathcal{S}_i|}}$, we achieve

$$\mathbf{Pr}\left\{\|H_i - H\|_2 \ge \Delta_3\right\} \le \delta_3'. \tag{47}$$

when $\|\nabla^2 f(X,\theta)\|_2 \le M'$.

From union bound theorem, suppose Assumption 4 holds, let $\delta_2 = \frac{\delta}{3} + \delta_3'$ and $\delta_2 \in (0,1)$, we have

$$\mathbf{Pr}\left\{\|H_i - H\|_2 \le \Delta_3\right\} \ge 1 - \delta_2. \tag{48}$$

Appendix B
Proof of Proposition 1

Suppose Assumptions 1-3 hold, and $\Theta \subset \{\theta :\|\theta - \theta^*\| \le r\sqrt{d}\}$ for some $r > 0$. From (22), for an honest worker i, we have

$$\begin{aligned} Z_i(\theta) &= (H^{-1} - H_i^{-1})g(\theta) \\ &= H^{-1}(H_i - H)H_i^{-1}g(\theta) \\ &= H^{-1}(H_i - H)H_i^{-1}(J(\theta) + \nabla F(\theta)). \end{aligned} \tag{49}$$

Using the properties of the spectral norm, we have

$$\begin{aligned} &\|Z_i(\theta)\| \\ \le\ & \|H^{-1}\|_2\|H_i - H\|_2\|H_i^{-1}\|_2(\|J(\theta)\| + \|\nabla F(\theta)\|). \end{aligned}$$

Following similar steps in [29], [40], we can show that, for any $\alpha \in (q/m, 1/2)$ and $0 < \delta_3 < \alpha - q/m$, we have

$$\begin{aligned} &\mathbf{Pr}\{\|J(\theta)\| \le 8\mathcal{C}_\alpha\Delta_2\|\theta - \theta^*\| + 4\mathcal{C}_\alpha\Delta_1\} \\ \ge\ & 1 - e^{-mD(\alpha - q/m\|\delta_3)} \end{aligned} \tag{50}$$

where $\Delta_1 = \sqrt{2}\sigma_1\sqrt{(d\log 6 + \log(6/\delta_3))/|\mathcal{S}_i|}$, $\Delta_2 = \sqrt{2}\sigma_2\sqrt{(\tau_1+\tau_2)/|\mathcal{S}_i|}$, with $\tau_1 = d\log 18 + d\log(M/\sigma_2)$, $\tau_2 = 0.5d\log(|\mathcal{S}_i|/d) + \log(6/\delta_3) + \log(\frac{2r\sigma_2^2\sqrt{|\mathcal{S}_i|}}{\alpha_2\sigma_1})$, $\mathcal{C}_\alpha = \frac{2(1-\alpha)}{1-2\alpha}$ and $D(\delta'\|\delta) = \delta'\log\frac{\delta'}{\delta} + (1-\delta')\log\frac{1-\delta'}{1-\delta}$.

Combining it with Assumption 1, Assumption 4, Lemma 2, we have the following bound

$$\begin{aligned} &\mathbf{Pr}\left\{\forall\theta : \|Z_i(\theta)\| \le \left(\frac{8\mathcal{C}_\alpha\Delta_3\Delta_2}{hh'} + \frac{\Delta_3 M}{hh'}\right)\|\theta - \theta^*\|\right. \\ &\left. + \frac{4\mathcal{C}_\alpha\Delta_3\Delta_1}{hh'}\right\} \ge 1 - \delta_4, \end{aligned} \tag{51}$$

with $1 - \delta_4 = 1 - \delta_2 - e^{-mD(\alpha - q/m\|\delta_3)}$.

Appendix C
Proof of Proposition 2

Suppose Assumptions 1-3 hold, and $\Theta \subset \{\theta :\|\theta - \theta^*\| \le r\sqrt{d}\}$ for some $r > 0$. From Proposition 1, we have the bound

$\|Z_i(\theta)\|$ for honest worker i.

From Lemma 1, in order to bound the geometric median $Z(\theta)$ of $Z_1(\theta),...,Z_m(\theta)$, we need to have more than half of the workers to be honest.

Then we can define a good event $\mathcal{E}_{2,\alpha,\xi_1,\xi_2} = \left\{\sum_{i=1}^m \mathbf{1}_{\{\mathcal{C}_\alpha \|Z_i(\theta)\|_2 \le \xi_3\|\theta-\theta^*\|+\xi_4\}} \ge m(1-\alpha)+q\right\}$, where

$$\xi_3 = \left(\frac{8\mathcal{C}_\alpha\Delta_3\Delta_2}{hh'} + \frac{\Delta_3 M}{hh'}\right)\mathcal{C}_\alpha,$$

and

$$\xi_4 = \frac{4\mathcal{C}_\alpha^2\Delta_3\Delta_1}{hh'}.$$

From proposition 1, for all $1 \le i \le m$, correct Z_i satisfied

$$\mathbf{Pr}\left\{\mathcal{C}_\alpha\|Z_i(\theta)\| \le \xi_3\|\theta-\theta^*\| + \xi_4\right\} \ge 1-\delta_4, \tag{52}$$

for any $\alpha \in (q/m, 1/2)$ and $0 < \delta_4 < \alpha - q/m$. Then following similar steps as in [29], we have

$$\mathbf{Pr}\{\mathcal{E}_{2,\alpha,\xi_1,\xi_2}\} \ge 1 - e^{-mD(\alpha-q/m\|\delta_4)}. \tag{53}$$

Then using Lemma 1, we obtain an bound for norm of geometric median $Z(\theta)$,

$$\mathbf{Pr}\Bigg\{\forall\theta : \|Z(\theta)\| \le \left(\frac{8\mathcal{C}_\alpha\Delta_3\Delta_2}{hh'} + \frac{\Delta_3 M}{hh'}\right)\mathcal{C}_\alpha\|\theta-\theta^*\| + \frac{4\mathcal{C}_\alpha^2\Delta_3\Delta_1}{hh'}\Bigg\} \ge 1 - e^{-mD(\alpha-q/m\|\delta_4)}. \tag{54}$$

Appendix D
Proof of Theorem 1

Suppose Assumptions 1-5 hold, and $\Theta \subset \{\theta : \| \theta - \theta^* \| \le r\sqrt{d}\}$ for some $r > 0$. Following similar steps in [29], [40], we have the following bound for any $\alpha \in (q/m, 1/2)$ and $0 < \delta_3 < \alpha - q/m$,

$$\begin{aligned}&\mathbf{Pr}\{\|J(\theta)\| \le 8\mathcal{C}_\alpha\Delta_2\|\theta-\theta^*\| + 4\mathcal{C}_\alpha\Delta_1\}\\ \ge\ & 1 - e^{-mD(\alpha-q/m\|\delta_3)}\end{aligned} \tag{55}$$

From (21), combined with Proposition 2, we have

$$\begin{aligned}&\|H^{-1}\nabla F(\theta) - G(\theta)\|\\ \le\ & \|Z(\theta)\| + \|H^{-1}J(\theta)\|\\ \le\ & \left(\frac{8}{hh'}\mathcal{C}_\alpha^2\Delta_3\Delta_2 + \frac{8}{h}\mathcal{C}_\alpha\Delta_2 + \mathcal{C}_\alpha\frac{\Delta_3 M}{hh'}\right)\|\theta-\theta^*\|\\ +\ & \frac{4}{hh'}\mathcal{C}_\alpha^2\Delta_3\Delta_1 + \frac{4}{h}\mathcal{C}_\alpha\Delta_1.\end{aligned} \tag{56}$$

Then for any $0 < \eta \le 1$,$\alpha \in (q/m, 1/2)$, $0 < \delta_3 < \alpha - q/m$ and $0 < \delta_4 < \alpha - q/m$ with probability at least $1 - e^{-mD(\alpha-q/m\|\delta_3)} - e^{-mD(\alpha-q/m\|\delta_4)}$, for any $t \ge 1$,

$$\begin{aligned}&\|\theta_t - \theta^*\|\\ =&\|\theta_{t-1} - \eta G(\theta_{t-1}) - \theta^*\|\\ =&\|\theta_{t-1} - \eta H_{t-1}^{-1}\nabla F(\theta_{t-1}) - \theta^* + \eta H_{t-1}^{-1}\nabla F(\theta_{t-1})\\ &\quad -\eta G(\theta_{t-1})\|\\ =&\|\theta_{t-1} - \eta H_{t-1}^{-1}\nabla F(\theta_{t-1}) - \theta^* + \eta Z(\theta_{t-1})\\ &\quad -\eta H_{t-1}^{-1}g(\theta_{t-1}) + \eta H_{t-1}^{-1}\nabla F(\theta_{t-1})\|\\ \le&\|\theta_{t-1} - \eta H_{t-1}^{-1}\nabla F(\theta_{t-1}) - \theta^*\| + \eta\|Z(\theta_{t-1})\|\\ &\quad +\|\eta H_{t-1}^{-1}J(\theta_{t-1})\|\\ \le&\left(1 - \eta + \eta\frac{8}{hh'}\mathcal{C}_\alpha^2\Delta_3\Delta_2 + \eta\frac{8}{h}\mathcal{C}_\alpha\Delta_2 + \eta\mathcal{C}_\alpha\frac{\Delta_3 M}{hh'}\right)\|\theta_{t-1}-\theta^*\|\\ &+\frac{\eta L\|\theta_{t-1}-\theta^*\|^2}{2h} + \eta\frac{4}{hh'}\mathcal{C}_\alpha^2\Delta_3\Delta_1 + \eta\frac{4}{h}\mathcal{C}_\alpha\Delta_1.\end{aligned} \tag{57}$$

Appendix E
Proof of Lemma 3

$$\begin{aligned}&\|H^{-1}\nabla F(\theta) - G(\theta)\|\\ =&\left\|H^{-1}\nabla F(\theta) - \frac{1}{1+|\mathcal{A}^{(2)}|}\Big(\sum_{i\in\mathcal{A}^{(2)}} g_2^{(i)}(\theta) + H_0^{-1}g(\theta)\Big)\right\|\\ \le&\ \frac{1}{1+|\mathcal{A}^{(2)}|}\left\|\Big(\sum_{i\in\mathcal{A}^{(2)}}(g_2^{(i)}(\theta) - H_0^{-1}g(\theta))\Big)\right\|\\ &+\ \|H^{-1}\nabla F(\theta) - H_0^{-1}g(\theta)\|\\ \le&\ \xi_2\frac{|\mathcal{A}^{(2)}|}{1+|\mathcal{A}^{(2)}|}\|H_0^{-1}g(\theta)\|\\ &+\ \|H_0^{-1}g(\theta) - H_0^{-1}\nabla F(\theta)\|\\ &+\ \|H^{-1}\nabla F(\theta) - H_0^{-1}\nabla F(\theta)\|\\ <&\ (1+\xi_2)\|H_0^{-1}\|_2\|g(\theta) - \nabla F(\theta)\|\\ &+\ \xi_2\|H_0^{-1}\|_2\|\nabla F(\theta)\| + \|H^{-1}\nabla F(\theta) - H_0^{-1}\nabla F(\theta)\|.\end{aligned}$$

Appendix F
Proof of Lemma 4

If $\|H_0 - H\|_2 \le \beta$ and $\beta < \frac{h(h+\mu)}{3h+2\mu}$, we have

$$\begin{aligned}&\|\mathbf{I} - \tilde{H}_0^{-1}H\|_2\\ =&\ \|\mathbf{I} - (H+\mu\mathbf{I})^{-1}H + (H+\mu\mathbf{I})^{-1}H - \tilde{H}_0^{-1}H\|_2\\ \le&\ \frac{\mu}{h+\mu} + \|(\tilde{H}_0^{-1} - (H+\mu)^{-1})H\|_2.\end{aligned} \tag{58}$$

Consider $\tilde{H}_0^{-1}$, let $A = H + \mu\mathbf{I}$ and $\Delta_0 = H_0 - H$,noting that $\|A^{-1}\Delta_0\|_2 \le \|A^{-1}\|_2\|\Delta_0\|_2 \le \frac{1}{h+\mu}\frac{h(h+\mu)}{3h+2\mu} < 1$, we have

$$\begin{aligned}\tilde{H}_0^{-1} &= (H + \mu\mathbf{I} + H_0 - H)^{-1}\\ &= (A+\Delta_0)^{-1}\\ &= (A(\mathbf{I} + A^{-1}\Delta_0))^{-1}\\ &= (\mathbf{I} + A^{-1}\Delta_0)^{-1}A^{-1}\\ &= A^{-1} + \sum_{r=1}^{\infty}(-1)^r(A^{-1}\Delta_0)^r A^{-1}.\end{aligned} \tag{59}$$

Then, we can have

$$
\begin{aligned}
& \|(\tilde{H}_0^{-1} - (H+\mu\mathbf{I})^{-1})H\|_2 \\
= \ & \|\sum_{r=1}^{\infty}(-1)^r(A^{-1}\Delta_0)^r A^{-1}(A-\mu\mathbf{I})\|_2 \\
= \ & \|\sum_{r=1}^{\infty}(-1)^r(A^{-1}\Delta_0)^r(\mathbf{I}-\mu A^{-1})\|_2 \\
\leq \ & \sum_{r=1}^{\infty}\|A^{-1}\|_2^r\|\Delta_0\|_2^r\|\mathbf{I}-\mu A^{-1}\|_2 \\
\leq \ & \sum_{r=1}^{\infty}\frac{\beta^r}{(h+\mu)^r}(1+\frac{\mu}{h+\mu}) \\
\leq \ & \frac{2\beta}{h+\mu}\sum_{r=0}^{\infty}\frac{\beta^r}{(h+\mu)^r} \\
= \ & \frac{2\beta}{h+\mu-\beta}.
\end{aligned} \tag{60}
$$

Then, we have

$$
\|\mathbf{I}-\tilde{H}_0^{-1}H\|_2 \leq \frac{\mu}{h+\mu}+\frac{2\beta}{h+\mu-\beta} < 1, \tag{61}
$$

when $\beta < \frac{h(h+\mu)}{3h+2\mu}$.

Appendix G
Proof of Proposition 3

From Lemma 4, we have the bound for $\|\mathbf{I}-\tilde{H}_0^{-1}H\|_2$, if $\|H_0-H\|_2 \leq \beta$ and $\beta < \frac{h(h+\mu)}{3h+2\mu}$. From Lemma 2, we have showned when Assumption 4 holds, that for any $\delta \in (0,1), \delta_2 \in (0,1), \delta_3' \in (0,1)$ and $\delta_2 = \delta_3' + \delta/3$ with probability at least $1-\delta_2$, $\|H_0 - H\|_2 \leq \sqrt{\frac{14M^2\log(2d/\delta_3')}{3|\mathcal{S}_0|}}$. Then if $\|\mathcal{S}_0\|$ is sufficiently large, we have $\Delta_3 = \sqrt{\frac{14M^2\log(2d/\delta_3')}{3|\mathcal{S}_0|}} < \frac{h(h+\mu)}{3h+2\mu}$, and we have the following bound for any $\delta_2 \in (0,1)$ with probability at least $1-\delta_2$

$$
\begin{aligned}
& \|(H^{-1}-\tilde{H}_0^{-1})\nabla F(\theta)\| \\
= \ & \|(\mathbf{I}-\tilde{H}_0^{-1}H)H^{-1}\nabla F(\theta)\| \\
\leq \ & \|\mathbf{I}-\tilde{H}_0^{-1}H\|_2\|H^{-1}\|_2\|\nabla F(\theta)-\nabla F(\theta^*)\| \\
\leq \ & \left(\frac{\mu}{h+\mu}+\frac{2\Delta_3}{h+\mu-\Delta_3}\right)\frac{M}{h}\|\theta-\theta^*\|.
\end{aligned} \tag{62}
$$

Appendix H
Proof of Theorem 2

If Assumptions 1-5 hold, and $\Theta \subset \{\theta : \| \theta - \theta^* \| \leq r\sqrt{d}\}$ for some $r > 0$ and $\Delta_3 < \frac{h(h+\mu)}{3h+2\mu}$ for the first term in Lemma 3, we already have the following bound from [40], regardless of the number of attackers, that with probability at least $1-\delta_5$

$$
\begin{aligned}
& \|g(\theta)-\nabla F(\theta)\| \\
\leq \ & (8\Delta_2+\xi_1(8\Delta_2+M))\|\theta_{t-1}-\theta^*\| + (4\Delta_1+\xi_1 4\Delta_1),
\end{aligned}
$$

in which $\Delta_1 = \sqrt{2}\sigma_1\sqrt{(d\log 6+\log(3/\delta_5))/|\mathcal{S}_0|}$ and $\Delta_2 = \sqrt{2}\sigma_2\sqrt{(\tau_1+\tau_2)/|\mathcal{S}_0|}$, with $\tau_1 = d\log 18 + d\log((M\vee M')/\sigma_2)$, and $\tau_2 = 0.5d\log(n/d)+\log(3/\delta_5)+\log(\frac{2r\sigma_2^2\sqrt{|\mathcal{S}_0|}}{\alpha_2\sigma_1})$.

Combine it with Assumption 1 and Proposition 3 and Lemma 3, fix any $t \geq 1$, for any $\delta_2 \in (0,1)$, and $\delta_5 \in (0,1)$, with probability at least $1-\delta_2-\delta_5$, the norm of difference between $G_t(\theta)$ and $\nabla F(\theta)$ is

$$
\begin{aligned}
& \|H^{-1}\nabla F(\theta)-G(\theta)\| \\
< \ & (1+\xi_2)\|\tilde{H}_0^{-1}\|_2\|g(\theta)-\nabla F(\theta)\| \\
+ \ & \xi_2\|\tilde{H}_0^{-1}\|_2\|\nabla F(\theta)\| + \|H^{-1}\nabla F(\theta)-\tilde{H}_0^{-1}\nabla F(\theta)\| \\
\leq \ & \gamma_1\|\theta-\theta^*\|+\gamma_2,
\end{aligned} \tag{63}
$$

where $\Delta_4 = \frac{\mu}{h+\mu}+\frac{2\Delta_3}{h+\mu-\Delta_3}$,

$$
\gamma_1 = \left[(8\Delta_2+\xi_1(8\Delta_2+M))\frac{1+\xi_2}{h'+\mu}+\frac{\xi_2 M}{h'+\mu}+\frac{\Delta_4 M}{h}\right], \tag{64}
$$

and

$$
\gamma_2 = (4\Delta_1+\xi_1 4\Delta_1)\frac{1+\xi_2}{h'+\mu}. \tag{65}
$$

Fix any $t \geq 1$,

$$
\begin{aligned}
& \|\theta_t-\theta^*\| \\
= \ & \|\theta_{t-1}-G(\theta_{t-1})-\theta^*\| \\
\leq \ & \|\theta_{t-1}-H^{-1}\nabla F(\theta_{t-1})-\theta^*\| \\
+ \ & \|G(\theta_{t-1})-H^{-1}\nabla F(\theta_{t-1})\| \\
\leq \ & \frac{L}{2h}\|\theta_{t-1}-\theta^*\|^2+\gamma_1\|\theta_{t-1}-\theta^*\|+\eta\gamma_2.
\end{aligned} \tag{66}
$$

References

[1] P. Richtárik and M. Takáč, "Distributed coordinate descent method for learning with big data," *Journal of Machine Learning Research*, vol. 17, pp. 2657–2681, Jan. 2016.

[2] P. Richtárik and M. Takáč, "Parallel coordinate descent methods for big data optimization," *Mathematical Programming*, vol. 156, pp. 433–484, Mar. 2016.

[3] Y. Chi, Y. Eldar, and R. Calderbank, "Petrels: Parallel subspace estimation and tracking by recursive least squares from partial observations," *IEEE Trans. Signal Processing*, vol. 61, pp. 5947–5959, Dec. 2013.

[4] A. Jochems, T. Deist, J. Van Soest, M. Eble, P. Bulens, P. Coucke, W. Dries, P. Lambin, and A. Dekker, "Distributed learning: Developing a predictive model based on data from multiple hospitals without data leaving the hospital–a real life proof of concept," *Radiotherapy and Oncology*, vol. 121, pp. 459–467, Dec. 2016.

[5] Y. Chi and H. Fu, "Subspace learning from bits," *IEEE Trans. Signal Processing*, vol. 65, pp. 4429–4442, Sep. 2017.

[6] A. Crotty, A. Galakatos, and T. Kraska, "Tupleware: Distributed machine learning on small clusters.," *IEEE Data Eng. Bull.*, vol. 37, pp. 63–76, Sept. 2014.

[7] F. Provost and D. Hennessy, "Scaling up: Distributed machine learning with cooperation," in *Proc. National Conf. on Artificial Intelligence*, vol. 1, (Portland, Oregon), pp. 74–79, Aug. 1996.

[8] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, pp. 1–122, Jan. 2011.

[9] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. Intl. Conf. on Computational Statistics*, pp. 177–186, Paris, France: Springer, Aug. 2010.

[10] J. Lee, Q. Lin, T. Ma, and T. Yang, "Distributed stochastic variance reduced gradient methods by sampling extra data with replacement," *Journal of Machine Learning Research*, vol. 18, pp. 4404–4446, Feb. 2017.

[11] D. Friend, R. Thomas, A. MacKenzie, and L. Silva, "Distributed learning and reasoning in cognitive networks: Methods and design decisions," *Cognitive networks: Towards self-aware networks*, pp. 223–246, Jul. 2007.

[12] C. Yu, M. van der Schaar, and A. Sayed, "Distributed learning for stochastic generalized Nash equilibrium problems," *IEEE Trans. Signal Processing*, vol. 65, pp. 3893–3908, Apr. 2017.

[13] P. Mertikopoulos, E. Belmega, R. Negrel, and L. Sanguinetti, "Distributed stochastic optimization via matrix exponential learning," *IEEE Trans. Signal Processing*, vol. 65, pp. 2277–2290, May 2017.

[14] C. Tekin and M. van der Schaar, "Distributed online learning via cooperative contextual bandits," *IEEE Trans. Signal Processing*, vol. 63, pp. 3700–3714, Jul. 2015.

[15] S. Chouvardas, G. Mileounis, N. Kalouptsidis, and S. Theodoridis, "Greedy sparsity-promoting algorithms for distributed learning," *IEEE Trans. Signal Processing*, vol. 63, pp. 1419–1432, Mar. 2015.

[16] B. Swenson, S. Kar, and J. Xavier, "Empirical centroid fictitious play: An approach for distributed learning in multi-agent games," *IEEE Trans. Signal Processing*, vol. 63, pp. 3888–3901, Aug. 2015.

[17] S. Marano, V. Matta, and P. Willett, "Nearest-neighbor distributed learning by ordered transmissions," *IEEE Trans. Signal Processing*, vol. 61, pp. 5217–5230, Nov. 2013.

[18] P. Moritz, R. Nishihara, I. Stoica, and M. Jordan, "Sparknet: Training deep networks in spark," in *Proc. Intl. Conf. on Learning Representations*, (San Juan, Puerto Rico), May 2016.

[19] S. Reddi, A. Hefny, S. Sra, B. Poczos, and A. Smola, "On variance reduction in stochastic gradient descent and its asynchronous variants," in *Advances in Neural Information Processing Systems*, pp. 2647–2655, Dec. 2015.

[20] S. Cen, H. Zhang, Y. Chi, W. Chen, and T.-Y. Liu, "Convergence of distributed stochastic variance reduced methods without sampling extra data," 2020. https://arxiv.org/pdf/1905.12648.pdf.

[21] C. Hsieh, K. Chang, C. Lin, S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear SVM," in *Proc. Intl. Conf. on Machine Learning*, (Helsinki, Finland), pp. 408–415, Jul. 2008.

[22] S. Shalev-Shwartz and T. Zhang, "Stochastic dual coordinate ascent methods for regularized loss minimization," *Journal of Machine Learning Research*, vol. 14, pp. 567–599, Feb. 2013.

[23] O. Shamir, N. Srebro, and T. Zhang, "Communication-efficient distributed optimization using an approximate newton-type method," in *Proc. Intl. Conf. on Machine Learning*, (Beijing, China), pp. 1000–1008, Jun. 2014.

[24] S. Reddi, J. Konečnỳ, P. Richtárik, B. Póczós, and A. Smola, "Aide: Fast and communication efficient distributed optimization," *arXiv preprint arXiv:1608.06879*, Aug. 2016.

[25] Y. Zhang and X. Lin, "Disco: Distributed optimization for self-concordant empirical loss," in *Proc. Intl. Conf. on Machine Learning*, (Lille, France), pp. 362–370, Jul. 2015.

[26] V. Smith, S. Forte, C. Ma, M. Takac, M. Jordan, and M. Jaggi, "Cocoa: A general framework for communication-efficient distributed optimization," *Journal of Machine Learning Research*, vol. 18, pp. 8590–8638, Jan. 2017.

[27] S. Wang, F. Roosta-Khorasani, P. Xu, and M. Mahoney, "Giant: Globally improved approximate newton method for distributed optimization," in *Advances in Neural Information Processing Systems*, pp. 2332–2342, Dec. 2018.

[28] C. Dünner, A. Lucchi, M. Gargiani, A. Bian, T. Hofmann, and M. Jaggi, "A distributed second-order algorithm you can trust," *arXiv preprint arXiv:1806.07569*, Jun. 2018.

[29] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, p. 44, Dec. 2017.

[30] P. Blanchard, E. Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Advances in Neural Information Processing Systems*, pp. 119–129, Dec. 2017.

[31] D. Alistarh, Z. Allen-Zhu, and J. Li, "Byzantine stochastic gradient descent," in *Advances in Neural Information Processing Systems*, pp. 4613–4623, Dec. 2018.

[32] C. Xie, O. Koyejo, and I. Gupta, "Phocas: dimensional Byzantine-resilient stochastic gradient descent," *arXiv preprint arXiv:1805.09682*, May 2018.

[33] D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," *arXiv preprint arXiv:1803.01498*, Mar. 2018.

[34] L. Chen, Z. Charles, D. Papailiopoulos, *et al.*, "Draco: Robust distributed training via redundant gradients," *arXiv preprint arXiv:1803.09877*, Jun. 2018.

[35] G. Damaskinos, E. Mhamdi, R. Guerraoui, R. Patra, and M. Taziki, "Asynchronous Byzantine machine learning," *arXiv preprint arXiv:1802.07928*, Jul. 2018.

[36] L. Su and J. Xu, "Securing distributed gradient descent in high dimensional statistical learning," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 3, p. 12, Mar. 2019.

[37] D. Yin, Y. Chen, K. Ramchandran, and P. Bartlett, "Defending against saddle point attack in Byzantine-robust distributed learning," *arXiv preprint arXiv:1806.05358*, Sep. 2018.

[38] C. Xie, O. Koyejo, and I. Gupta, "Zeno: Byzantine-suspicious stochastic gradient descent," *arXiv preprint arXiv:1805.10032*, Sep. 2018.

[39] X. Cao and L. Lai, "Robust distributed gradient descent with arbitrary number of Byzantine attackers," in *Proc. IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, (Calgary, Canada), pp. 6373–6377, Apr. 2018.

[40] X. Cao and L. Lai, "Distributed gradient descent algorithm robust to an arbitrary number of Byzantine attackers," *IEEE Trans. Signal Processing*, vol. 67, pp. 5850–5864, Nov. 2019.

[41] D. Alistarh, C. De Sa, and N. Konstantinov, "The convergence of stochastic gradient descent in asynchronous shared memory," in *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*, pp. 169–178, Jul. 2018.

[42] L. Li, W. Xu, T. Chen, G. Giannakis, and Q. Ling, "Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 1544–1551, Jul. 2019.

[43] A. Juditsky, A. Nazin, A. Nemirovsky, and A. Tsybakov, "Algorithms of robust stochastic optimization based on mirror descent method," *arXiv preprint arXiv:1907.02707*, Jul. 2019.

[44] L. Su and S. Shahrampour, "Finite-time guarantees for Byzantine-resilient distributed state estimation with noisy measurements," *IEEE Trans. Automatic Control*, Nov. 2019.

[45] R. Jin, Y. Huang, X. He, H. Dai, and T. Wu, "Stochastic-sign SGD for federated learning with theoretical guarantees," *arXiv preprint arXiv:2002.10940*, Feb. 2020.

[46] Z. Yang, A. Gang, and W. Bajwa, "Adversary-resilient inference and machine learning: From distributed to decentralized," *arXiv preprint arXiv:1908.08649*, Feb. 2020.

[47] R. Jin, X. He, and H. Dai, "Distributed Byzantine tolerant stochastic gradient descent in the era of big data," in *Proc. IEEE Intl. Conf. on Communication*, (Shanghai, China), pp. 1–6, May 2019.

[48] S. Minsker *et al.*, "Geometric median and robust estimation in banach spaces," *Bernoulli*, vol. 21, pp. 2308–2335, Mar. 2015.

[49] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, vol. 2, 2010.

[50] J. Tropp *et al.*, "An introduction to matrix concentration inequalities," *Foundations and Trends® in Machine Learning*, vol. 8, pp. 1–230, May 2015.