Byzantine Fault-Tolerance of Local Gradient-Descent in Federated Model under $2f$-Redundancy

Nirupam Gupta [1] **Thinh T. Doan** [2] **Nitin Vaidya** [3]

Abstract

We study the problem of Byzantine fault-tolerance in a federated optimization setting, where there is a group of agents communicating with a centralized coordinator. We allow up to f Byzantine-faulty agents, which may not follow a prescribed algorithm correctly, and may share arbitrary incorrect information with the coordinator. Associated with each non-faulty agent is a local cost function. The goal of the non-faulty agents is to compute a minimizer of their aggregate cost function. For solving this problem, we propose a local gradient-descent (GD) algorithm that incorporates a novel *comparative elimination* (CE) filter (aka. aggregation scheme) to provably mitigate the detrimental impact of Byzantine faults. In the deterministic setting, when the agents can compute their local gradients accurately, our algorithm guarantees *exact* fault-tolerance against a bounded fraction of Byzantine agents, provided the non-faulty agents satisfy the known necessary condition of $2f$-*redundancy*. In the stochastic setting, when the agents can only compute stochastic estimates of their gradients, our algorithm guarantees *approximate* fault-tolerance where the approximation error is proportional to the variance of stochastic gradients and the fraction of Byzantine agents.

1. Introduction

We consider a distributed optimization framework where there are N agents communicating with a single coordinator. Associated with each agent i is a function $q^i : \mathbb{R}^d \to \mathbb{R}$. The goal of the agents is to find $x^\star$ such that

$$x^* \in \arg\min_x \sum_{i=1}^{N} q^i(x), \tag{1}$$

where each q^i is given as

$$q^i(x) \triangleq \mathbb{E}_{\pi^i}\left[Q^i(x; X^i)\right], \tag{2}$$

for some random variable X^i defined over a compact sample set $\mathcal{X}^i$ with a distribution π^i. We assume that each agent i only has access to the sequence $\{G^i(\cdot)\}$, which can be either the *actual* gradients $\{\nabla q^i(\cdot)\}$ or *stochastic* gradients $\{\nabla Q^i(\cdot, X^i)\}$. This is a common distributed machine learning setting, where there is a large number of data distributed to different agents (or machines). The goal is to design an algorithm that allow these agents is to jointly minimize a loss function defined over their data, i.e., solve optimization problem (1).

For solving (1), we consider the local gradient-descent (GD) method, which has recently received significant attention due to its application in federated learning (Kairouz & McMahan, 2021; Li et al., 2020). In this method, the coordinator maintains an estimate of a solution defined in (1). This estimate is broadcast to all the agents, and each agent updates its copy of the estimate by running a number of local GD steps. The agents send back to the coordinator their local updated estimates. Finally, the coordinator averages the received estimates to obtain new *global* estimate of $x^\star$. Eventually, if all the agents are non-faulty, the sequence of global estimates converges to a solution (1). Since the agents only share their local estimates and not their data, local GD is widely used in federated learning where data privacy is a major concern (Kairouz & McMahan, 2021).

Our interest is to study the performance of the local GD method in the presence of up to f Byzantine faulty agents (Lamport et al., 1982). Such faulty agents may behave arbitrarily, and their identity is a priori unknown. In particular, Byzantine faulty agents may collude and share incorrect information with the coordinator in order to corrupt the output of the algorithm, e.g., see (Xie et al., 2019). We aim to design a new local GD method that allows all the

[1] School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland [2] Department of Electrical and Computer Engineering, Virginia Tech, Virginia, USA [3] Department of Computer Science, Georgetown University, Washington DC, USA. Correspondence to: Nirupam Gupta <nirupam.gupta@epfl.ch>.

This work was presented at the International Workshop on Federated Learning for User Privacy and Data Confidentiality in Conjunction with ICML 2021 (FL-ICML'21). This workshop does not have official proceedings and this paper is non-archival.

non-faulty agents to compute an exact minimum of the aggregate cost of the non-faulty agents, despite the presence of Byzantine agents. In particular, we consider the **exact fault-tolerance** problem defined below. For a set $\mathcal{H}$, $|\mathcal{H}|$ denotes its cardinality.

Definition 1 (**Exact fault-tolerance**). *Let $\mathcal{H}$ with $|\mathcal{H}| \geq N - f$ be the set of non-faulty agents. A distributed optimization algorithm is said to have exact fault-tolerance if it returns*

$$x^\star_{\mathcal{H}} \in \arg\min_x \sum_{i \in \mathcal{H}} q^i(x). \tag{3}$$

Since the identity of the Byzantine faulty agents is a priori unknown, in general, **exact fault-tolerance** is unachievable (Su & Vaidya, 2016). In particular, exact fault-tolerance is impossible unless the non-faulty agents satisfy the property of $2f$-**redundancy** defined as follows (Gupta & Vaidya, 2020a;b).

Definition 2 ($2f$-**redundancy**). *A set of non-faulty agents $\mathcal{H}$, with $|\mathcal{H}| \geq n - f$, is said to have $2f$-redundancy if for any subset $\mathcal{S} \subseteq \mathcal{H}$ with $|\mathcal{S}| \geq N - 2f$,*

$$\arg\min_x \sum_{i \in \mathcal{S}} q_i(x) = \arg\min_x \sum_{i \in \mathcal{H}} q_i(x). \tag{4}$$

The $2f$-redundancy property is critical to our algorithm, presented in Section 2, for solving (3). This property implies that a minimum of the aggregate cost of any $N - 2f$ non-faulty agents is also a minimum of the aggregate cost of all the non-faulty agents, and vice-versa. This seemingly contrived redundancy condition arises naturally with high probability in many practical applications, including distributed sensing (Chong et al., 2015; Gupta & Vaidya, 2019; Mishra et al., 2016; Su & Shahrampour, 2019), and distributed learning (Alistarh et al., 2018; Blanchard et al., 2017; Charikar et al., 2017; Guerraoui et al., 2018). Note that in the context of distributed learning, in the i.i.d. setting, i.e., when all agents have the same data generating distribution, $2f$-redundancy hold true trivially. More generally, in the non-i.i.d. setting, $2f$-redundancy holds true as long as the learning problem can be solved (i.e., computing an optimal model over the collective data of all non-faulty agents) using data of only $n - 2f$ non-faulty agents. For further details on $2f$-redundancy, and a formal proof of its necessity, see (Gupta & Vaidya, 2020a;b; Liu et al., 2021).

As the identity of faulty agents is a priori unknown to the coordinator, solving (3) is nontrivial even under the $2f$-redundancy property, especially in the high dimension case, e.g., see (Gupta & Vaidya, 2020a; Kuwaranancharoen et al., 2020; Su & Shahrampour, 2019). The key element of our algorithm is a filter named **comparative elimination** (CE), which is implemented at the coordinator to aggregate the local updated estimates sent by the agents to mitigate the detrimental impact of potentially *adversarial values* from the Byzantine agents. In particular, instead of simply averaging the agents' estimates, the coordinator eliminates f (out of N) received estimates that are f farthest from the current global estimate. The new global estimate is obtained by averaging the remaining $N - f$ agents' estimates. Details on our scheme are presented in Algorithm 1 in Section 2. In the following, we summarize our main contributions, and then discuss the related literature.

1.1. Main Contributions

We show that our proposed scheme, *comparative elimination* (CE) filter, when coupled with the local GD method formulates a Byzantine robust distributed algorithm for solving (3). Specifically, assuming each non-faulty agent's local cost to be *L-smooth*, the aggregate non-faulty cost to be *μ-strongly convex* (but, the local costs may only be convex) and the necessary condition of *$2f$-redundancy*, we present the following results:

- **In the deterministic setting**, when each non-faulty agent i updates its local estimates using *actual* gradients of its cost function $\{\nabla q^i(\cdot)\}$, the CE filter scheme guarantees *exact fault-tolerance* if $\frac{f}{|H|} \leq \frac{\mu}{3L}$. Moreover, the convergence rate is linear, similar to the fault-free setting under strong convexity.
- **In the stochastic setting**, when a non-faulty agent i can only access stochastic estimates of its local gradients $\{\nabla Q^i(\cdot, X^i)\}$ we guarantee *approximate* fault-tolerance. Specifically, the sequence of global estimates $\{\bar{x}^k\}$ satisfy the following for all k:

  $$\mathbb{E}[\|\bar{x}^k - x^\star_{\mathcal{H}}\|^2] \leq \lambda^k \mathbb{E}[\|\bar{x}_0 - x^\star_{\mathcal{H}}\|^2] + \mathcal{O}\left(\sigma^2\alpha + \frac{\sigma^2 f}{N - f}\right)$$

  for some $\lambda \in (0, 1)$ where α is some constant step-size of the algorithm, and σ is the noise level of the stochastic gradients.

Specific details of our results are given in Section 3.

1.2. Related Work

In recent years, several other aggregation schemes have been proposed for Byzantine fault-tolerance in distributed optimization or learning. Some of the prominent Byzantine robust schemes are *coordinate-wise trimmed mean* (CWTM) (Su & Vaidya, 2016; Su & Shahrampour, 2019; Yang & Bajwa, 2019; Yin et al., 2018), *multi-KRUM* (Blanchard et al., 2017), *geometric median-of-means* (GMoM) (Chen et al., 2017), *coordinate-wise median* (Sundaram & Gharesifard, 2018; Yin et al., 2018), *Bulyan* (Guerraoui et al., 2018), *minimum-diameter averaging* (MDA) (Guerraoui et al., 2018), *phocas* (Xie et al., 2018b), *Byzantine-robust stochastic aggregation* (RSA) (Li et al., 2019), *signSGD* with majority voting (Sohn et al., 2020), and *spectral decomposition* based filters (Diakonikolas et al., 2019; Prasad et al., 2020). Most these works, with the exception of (Su & Vaidya, 2016; Su & Shahrampour, 2019; Sundaram & Gharesifard, 2018; Kuwaranancharoen et al., 2020; Li et al., 2019; Yang & Bajwa, 2019), only consider the distributed GD framework wherein the agents send gradients of their local costs, instead of the federated local GD framework that we consider. Nevertheless, from these works we infer that the above aggregation schemes need not guarantee *exact fault-tolerance* in general, even in the deterministic setting (i.e., when agents can compute actual gradients of their costs) under $2f$-redundancy, unless further assumptions are made on the non-faulty agents' costs.

Although (Su & Vaidya, 2016; 2020; Sundaram & Gharesifard, 2018) implicitly show the exact fault-tolerance properties of *trimmed-mean* and *median*, they only consider the scalar case, i.e., when agents' cost functions are univariate. The extension of their results to higher-dimensions is non-trivial and remains poorly understood, e.g., see (Kuwaranancharoen et al., 2020; Su & Vaidya, 2016; Su & Shahrampour, 2019; Yang & Bajwa, 2017). For instance, (Su & Vaidya, 2016) consider a special distributed optimization problem wherein the agents' cost functions have a known common *basis*. The work (Su & Shahrampour, 2019) shows that CWTM can guarantee exact fault-tolerance when solving the distributed linear least squares problem, under $2f$-redundancy, provided the agents' data satisfy an uncommon additional property. In (Yang & Bajwa, 2019), they assume that the agents' costs can be decomposed into independent *scalar strictly convex* functions. Recently, (Kuwaranancharoen et al., 2020) studied the fault-tolerance of coordinate-wise trimmed mean (or median) in a peer-to-peer setting (a generalization of federated model) for generic convex optimization problems; their results suggest that CWTM need not provide exact fault-tolerance even under $2f$-redundancy, in general.

When applied to the federated local GD framework, some of the above aggregation schemes, such as multi-KRUM, Bulyan, CWTM, GMoM and MDA, only operate on the local estimates sent by the agents and disregard the current global estimate maintained by the coordinator (Fang et al., 2020). On the other hand, CE filter exploits the (supposed) closeness between the current global estimate and the non-faulty agents' local updated estimates to obtain improved robustness against Byzantine agents. This similarity exists due to Lipschitz smoothness of non-faulty agents' costs, and is critical to the fault-tolerance property of the algorithm. Other works that exploit this similarity are RSA (Li et al., 2019), and (Muñoz-González et al., 2019). Recently, (Wu et al., 2020) have shown that *geometric median aggregation* scheme provably provides improved Byzantine fault-tolerance compared to other aggregation schemes in federated model. However, computing geometric median is a challenging problem, as there does not exist a closed-form formula (Bajaj, 1988). Moreover, existing numerical algorithms for computing geometric median are only approximate, and computationally quite complex (Cohen et al., 2016). Other schemes, such as the verifiable coding in (So et al., 2020) and manual verification of information sent by agents (Cao et al., 2020), are not directly applicable to the commonly used federated framework where inter-agent communication is absent or there are a large number of agents, and data privacy is a major concern.

Besides federated local GD framework, the proposed CE filter can also guarantee exact fault-tolerance in the distributed GD framework where agents share their gradients instead of estimates (references omitted to preserve authors' anonymity). Also, for the distributed GD framework, recent works have shown that *momentum* helps improve the fault-tolerance of a Byzantine-robust aggregation scheme (Mhamdi et al., 2021; Karimireddy et al., 2020). However, adaptation of their results to federated local GD method is non-trivial and remains to be investigated.

2. Local GD under Byzantine Model

We now present the proposed algorithm for solving (1) in the presence of at most f Byzantine faulty agents. We note that the Byzantine agents can observe the values of other agents and send arbitrarily values to the coordinator. To handle this scenario, the main idea of our approach is a Byzantine robust aggregation rule (or filter), named comparative elimination (CE) filter, which is implemented at the coordinator. This filter together with the local GD formulates our proposed method, formally presented in Algorithm 1 for solving (3).

In Algorithm 1, each agent i maintains a local variable x^i, and the coordinator maintains $\bar{x}$, the average of these x^i. At any iteration $k \geq 0$, agent i receives $\bar{x}_k$ from the coordinator and initializes its iterate $x^i_{k,0} = \bar{x}_k$. Here $x^i_{k,t}$ denotes the iterate at iteration k and local time $t \in [0, \ldots, T-1]$ at agent i. Agent i then runs a number T of local GD steps using time-varying step sizes α_k and its local direction $G^i(x^i_{k,t})$, which can be either the *actual* gradient $\nabla q^i(x^i_{k,t})$ or a *stochastic* estimate $\nabla Q^i(\cdot, X^i)$ of its gradient based on the data $\{X^i_{k,t}\}$ sampled i.i.d from π^i. After T local GD steps, the agents then send their new local updates $x^i_{k,T}$ to the coordinator. However, Byzantine agents may send arbitrary values to disrupt the learning process. The coordinator implements the CE filter (in steps $2(a)$ and $2(b)$ of Algorithm 1) to dilute the impact of "bad" values sent by the Byzantine agents. The main of this filter is to discard

Algorithm 1 Local GD with CE Filter

```
Initialization: The coordinator initializes $\bar{x}_0 \in \mathbb{R}^d$. Agent $i$ initializes step sizes $\{\alpha_k\}$ and a positive integer $T$.
Iterations: For $k = 0, 1, 2, ...$
  1. Agent $i$
     (a) Receive $\bar{x}_k$ sent by the server and set $x^i_{k,0} = \bar{x}_k$
     (b) For $t = 0, 1, \ldots, T-1$, implement
             $x^i_{k,t+1} = x^i_{k,t} - \alpha_k G^i(x^i_{k,t}).$        (5)
  2. The coordinator receives $x^i_{k,T}$ from each agent $i$ and implement the CE filter as follows.
     (a) Compute the distances of $x^i_{k,t}$ with its current value $\bar{x}_k$, and sort them in an increasing order
             $\|\bar{x}_k - x^{i_1}_{k,T}\| \leq \ldots \leq \|\bar{x}_k - x^{i_N}_{k,T}\|.$        (6)
     (b) Discard the $f$-largest distances, i.e., it drops $x^{i_{N-f+1}}_{k,T}, \ldots, x^{i_N}_{k,T}$. Let $\mathcal{F}_k = \{i_1, \ldots, i_{N-f}\}$.
     (c) Update its iterate as
             $\bar{x}_{k+1} = \frac{1}{|\mathcal{F}_k|} \sum_{i \in \mathcal{F}_k} x^i_{k,T}.$        (7)
```

f-values (or estimates) that are f-farthest from the current global estimate $\bar{x}_k$. Finally, the coordinator averages the $N-f$ remaining estimates, as shown in (7), to compute the new global estimate. Note that without the CE filter (i.e., without steps $2(a)$ and $2(b)$, and $F_k = [1, N]$), Algorithm 1 reduces to the traditional local GD method.

3. Main Results

In this section, we present the main results of this paper, where we characterize the convergence of Algorithm 1 for solving problem (3). We consider two cases, namely, the deterministic settings (when $G^i(\cdot) = \nabla q^i(\cdot)$) and the stochastic settings (when $G^i(\cdot) = \nabla Q^i(\cdot, X^i)$).[1] In both cases, our theoretical results are derived when the non-faulty agents' cost functions are smooth. Moreover, we also assume the average non-faulty cost function, denoted by $q^{\mathcal{H}}(x)$, to be strongly convex. Specifically,

$$q^{\mathcal{H}}(x) = \frac{1}{|\mathcal{H}|} \sum_{i \in \mathcal{H}} q^i(x). \tag{8}$$

These assumptions are formally stated as follows.

Assumption 1 (Lipschitz smoothness)**.** *The non-faulty agents' functions have Lispchitz continuous gradients, i.e., there exists a positive constant $L < \infty$ such that, $\forall i \in \mathcal{H}$,*

$$\|\nabla q^i(x) - \nabla q^i(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathbb{R}^d.$$

Assumption 2 (Strong convexity)**.** *$q^{\mathcal{H}}$ is strongly convex, i.e., there exists a positive constant $\mu < \infty$ such that*

$$(x-y)^T(\nabla q^{\mathcal{H}}(x) - \nabla q^{\mathcal{H}}(y)) \geq \mu\|x-y\|^2, \; \forall x, y \in \mathbb{R}^d.$$

To this end, we assume that these assumptions and the $2f$-redundancy property always hold true. Note that Assumptions 1 and 2 hold true simultaneously only if $\mu \leq L$.

Remark 1. *Assumption 2 implies that there exists a unique solution $x^\star_{\mathcal{H}}$ of problem* (3)*. However, this assumption does not imply that each local function q^i is strongly convex. Indeed, each q^i can have more than one minimizer. Under the $2f$-redundancy property one can show that*

$$x^\star_{\mathcal{H}} \in \bigcap_{i \in \mathcal{H}} \arg\min_{x \in \mathbb{R}^d} q^i(x). \tag{9}$$

Thus, one can view that Algorithm 1 tries to search one point in the intersection of the minimizer sets of the local functions q^i. However, we do not assume we can compute these sets since this task is intractable in general. Finally, our analysis given later will rely on (9)*, whose proof can be found in (Gupta & Vaidya, 2020a, Appendix B).*

3.1. Deterministic Settings

In this section, we consider the deterministic setting of Algorithm 1, i.e., $G^i(\cdot) = \nabla q^i(\cdot)$. For convenience, we first study the convergence of Algorithm 1 when $T = 1$ in Section 3.1.1 and generalize to the case $T > 1$ in Section 3.1.2.

3.1.1. The case of $T = 1$

When $T = 1$, Algorithm 1 is equivalent to the popular distributed (stochastic) gradient method. Indeed, by (5) we have for any $i \in \mathcal{H}$

$$x^i_{k,1} = x^i_{k,0} - \alpha_k \nabla q^i(x^i_{k,0}) = \bar{x}_k - \alpha_k \nabla q^i(\bar{x}_k). \tag{10}$$

We denote by $\mathcal{B}$ the set of Byzantine agents, i.e., $N = |\mathcal{B}| + |\mathcal{H}|$ and $|\mathcal{B}| \leq f$. Without loss of generality we assume that $|\mathcal{B}| = f$. Similarly, let $\mathcal{B}_k$ be the set of Byzantine agents in $\mathcal{F}_k$ and $\mathcal{H}_k$ be the set of nonfaulty agents in $\mathcal{F}_k$. Then we have $|\mathcal{B}_k| = |\mathcal{F}_k \setminus \mathcal{H}_k| \leq f$, for any $k \geq 0$.

Theorem 1. *Let $\{\bar{x}_k\}$ be generated by Algorithm 1 with $T = 1$. We assume that the following condition holds*

$$\frac{f}{N-f} \leq \frac{\mu}{3L}. \tag{11}$$

Let α_k be chosen as

$$\alpha_k = \alpha \leq \frac{\mu}{4L^2}. \tag{12}$$

[1] Proofs of all the theorems presented in this section are deferred to the appendix attached after the list of references.

Then we have

$$\|\bar{x}^k - \bar{x}^\star_{\mathcal{H}}\|^2 \leq \left(1 - \frac{\mu\alpha}{6}\right)^k \|\bar{x}_0 - x^\star_{\mathcal{H}}\|^2. \tag{13}$$

Remark 2. *In Theorem 1 we show that under the $2f$ redundancy, Algorithm 1 returns an exact solution $x^\star_{\mathcal{H}}$ of problem* (3) *even under of at most f Byzantine agents. Moreover, the convergence is linear, which is the same as what we expect in the non-faulty case (no Byzantine agents).*

3.1.2. The case of $T > 1$

We now generalize Theorem 1 to the case $T > 1$, i.e., each agent implements more than 1 local GD steps. This is indeed a common practice in federated optimization. When $T > 1$, by (5) we have $\forall i \in \mathcal{H}$ and $t \in [0, H)$

$$x^i_{k,t+1} = \bar{x}_k - \alpha_k \sum_{\ell=0}^{t} \nabla q^i(x^i_{k,\ell}), \tag{14}$$

Theorem 2. *Assume that* (11) *hold and let α_k satisfy*

$$\alpha_k = \alpha \leq \frac{\mu}{16TL^2}. \tag{15}$$

Then we have

$$\|\bar{x}^k - \bar{x}^\star_{\mathcal{H}}\|^2 \leq \left(1 - \frac{\mu T\alpha}{6}\right)^k \|\bar{x}_0 - x^\star_{\mathcal{H}}\|^2. \tag{16}$$

3.2. Stochastic Settings

We next consider the setting where each agent only has access to the samples of its gradient, i.e., $G^i(\cdot) = \nabla Q^i(\cdot, X^i)$, where X^i is a sequence of random variables sampled i.i.d from π^i. In the sequel, we denote by

$$\mathcal{P}_{k,t} = \cup_{i\in\mathcal{H}}\{\bar{x}^0, \ldots, \bar{x}_k, x^i_{k,1}, \ldots, x^i_{k,t}\}$$

the filtration containing all the history generated by Algorithm 1 up to time $k + t$. To study the convergence of Algorithm 1 we consider the following assumption, which is often assumed in the literature of stochastic federated optimization (Kairouz & McMahan, 2021).

Assumption 3. *The random variables X^i_k, for all i and k, are generated i.i.d. Moreover, there exists a positive constant σ such that*

$$\mathbb{E}[\nabla Q^i(x, X^i_{k,t}) \,|\, \mathcal{P}_{k,t}] = \nabla q^i(x), \quad \forall x \in \mathbb{R}^d,$$
$$\mathbb{E}[\|\nabla Q^i(x, X^i_{k,t}) - \nabla q^i(x)\|^2 \,|\, \mathcal{P}_{k,t}] \leq \sigma^2, \quad \forall x \in \mathbb{R}^d,$$
$$\|\nabla Q^i(x, X^i) - \nabla Q^i(y, X^i)\| \leq L\|x - y\|, \text{ a.s.,}$$

where the constant L is given in Assumption 1.

Note that $|\mathcal{B}_k| + |\mathcal{H}_k| = |\mathcal{F}_k| = |\mathcal{H}|$, for any $k \geq 0$. Finally, for convenience we denote by

$$\nabla Q^i(x; X) = \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}} \nabla Q^i(x; X^i),$$

where $X = (X^1, \ldots, X^{|\mathcal{H}|})^T$. Due to space limit, we only present the result for the case $T = 1$ in this paper. The result for the case $T > 1$ will be discussed in the longer version of this paper.

3.2.1. The case of $T = 1$

Theorem 3. *Suppose that Assumption 3 and condition* (11) *hold true. Moreover, let α_k be chosen as*

$$\alpha_k = \alpha \leq \frac{\mu}{12L^2}. \tag{17}$$

Then we have

$$\mathbb{E}[\|\bar{x}_k - x^\star_{\mathcal{H}}\|^2] \leq \left(1 - \frac{\mu}{6}\alpha\right)^k \mathbb{E}[\|\bar{x}_0 - x^\star_{\mathcal{H}}\|^2] + \frac{14\sigma^2\alpha}{\mu} + \frac{2\sigma^2 f}{\mu L|\mathcal{H}|}. \tag{18}$$

Remark 3. *Note that in Theorem 3 due to the constant step size, the mean square error converges linearly only to a ball centered at the origin. The size of this ball depends on two terms: 1) one depends on the step size α which often seen in the convergence of local GD with non-faulty agents and 2) the other depends on the level of the gradient noise (or σ). The latter is due to the impact of the Byzantine agents and the stochastic gradient samples. Indeed, our comparative filter is designed to remove the potential bad values sent by the Byzantine agents, but not the variance of their stochastic samples. One potential solution for this issue is to let each agent sample a mini-batch of size m of its gradient. In this case, it is not difficult to see that σ^2 in* (18) *is replaced by σ^2/m. Thus, one can choose m large enough so that the mean square error can get arbitrarily close to zero. Finally, when $\alpha_k \sim 1/k$, one can show that the convergence rate is sublinear $\mathcal{O}(1/k)$.*

4. Experiments

To evaluate the efficacy of our proposed scheme, we simulate the problem of *robust mean estimation* in the federated framework. This problem serves as a test-case to empirically compare our scheme with others of similar computational costs, namely multi-KRUM (Blanchard et al., 2017), CWTM (Su & Shahrampour, 2018; Yin et al., 2018), and coordinate-wise median (Xie et al., 2018a; Yin et al., 2018). For our experiments, we consider $N = 50$ agents and varying number of Byzantine faulty agents. Each non-faulty agent i has 100 noisy observations of a 10-dimensional vector x^* with all elements of unit value. In particular, the sample set $\mathcal{X}^i$ comprises 100 uniformly distributed samples with each sample $X^i = x^* + Z^i$ where $Z^i \sim \mathcal{N}(0, I_d)$, and $Q^i(x; X^i) = (1/2)\|x - X^i\|^2$. In this case, x^* is the unique solution to problem (3) for any set of honest agents $\mathcal{H}$. In our experimental settings, a Byzantine faulty agent j behaves just like an honest agent with 100

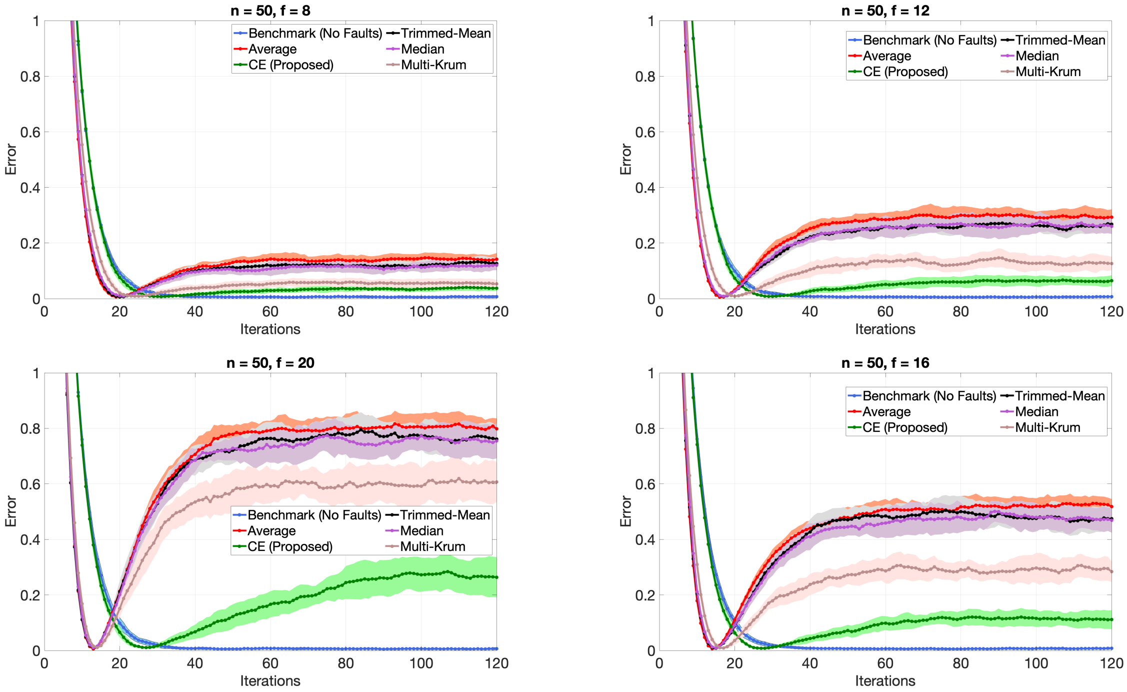

Figure 1. The plots show the *error* $\|\bar{x}_k - x^*\|^2$ in *iteration* $k = 1, \ldots, 120$ of local GD (cf. Algorithm 1) with four different aggregation schemes; *averaging*, *CE*, *multi-KRUM*, *CWTM*, and *coordinate-wise median*. The benchmark corresponds to the fault-free execution of local GD. Solid lines show the mean performances of the schemes, and the shadows show the variance of their performances, observed over 100 runs. In the clockwise order, $f = 8$, 12, 16 and 20. We observe that, expectedly, all schemes obtain improved accuracy in presence of fewer Byzantine agents. We also observe that the performance of CE filter is consistently better than other schemes.

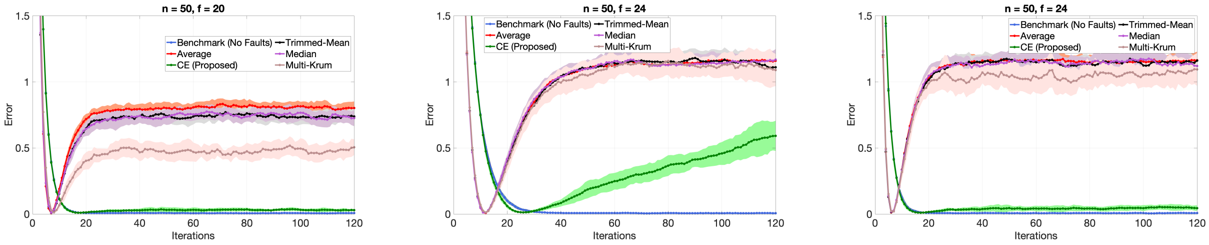

Figure 2. The leftmost plot is for $f = 20$ and $T = 2$, the case with $T = 1$ is shown in Figure 1. The middle and the rightmost plots are for ($f = 24$, $T = 1$) and ($f = 24$, $T = 2$), respectively. We observe that the accuracy of the local GD method with CE filter improves considerably as the number of local GD steps T is increased from 1 to 2. The same cannot be observed for other schemes.

uniformly distributed samples, however each of its sample $X^j = 2 \times x^* + Z^j$ where $Z^j \sim \mathcal{N}(0, I_d)$. That is, honest agents send information corresponding to Gaussian noisy observations of x^* and Byzantine agents send information corresponding to Gaussian noisy observations (with identical variance) of $2 \times x^*$.

We simulate the *stochastic setting* of local GD (cf. Algorithm 1) with different number of faulty agents $f \in \{8, 12, 16, 20, 24\}$, different values of $T \in \{1, 2\}$, and different aggregation schemes in Step 2: *CE*, *mutli-KRUM*, *CWTM*, *coordinate-wise median* and *simple averaging*. The step-size $\alpha_k = 0.1$ for all k. Each setting is run 100 times, and the observed errors $\|\bar{x}_k - x^*\|^2$ for $k = 1, \ldots, 120$ are shown in Figures 1 and 2.

Inference: As suggested from our theoretical results, the final error upon using CE aggregation scheme decreases with the fraction of Byzantine faulty agents. We observe that CE aggregation scheme performs consistently better than multi-KRUM, CWTM and median. Moreover, we also observe that increasing the number of local gradient-descent steps, i.e., T, improves the fault-tolerance of CE filter. However, the same cannot be said for other schemes.

Summary

In this paper, we have considered the problem of Byzantine fault-tolerance in the local gradient-descent method on a federated model. We have proposed a new aggregation scheme, named comparative elimination (CE) filter, and studied its fault-tolerance properties in both deterministic and stochastic settings. In the deterministic setting, we have shown the CE filter guarantees *exact* fault-tolerance against a bounded fraction of Byzantine agents f/N, provided the non-faulty agents' costs satisfy the necessary condition of $2f$-redundancy. In the stochastic setting, we have shown that CE filter obtains *approximate* fault-tolerance where the approximation error is proportional to the variance of the agents' stochastic gradients and the fraction of Byzantine agents.

References

Alistarh, D., Allen-Zhu, Z., and Li, J. Byzantine stochastic gradient descent. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 4618–4628, 2018.

Bajaj, C. The algebraic degree of geometric optimization problems. *Discrete & Computational Geometry*, 3(2): 177–191, 1988.

Blanchard, P., Guerraoui, R., et al. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, pp. 119–129, 2017.

Cao, X., Fang, M., Liu, J., and Gong, N. Z. Fltrust: Byzantine-robust federated learning via trust bootstrapping. *arXiv preprint arXiv:2012.13995*, 2020.

Charikar, M., Steinhardt, J., and Valiant, G. Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 47–60, 2017.

Chen, Y., Su, L., and Xu, J. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):1–25, 2017.

Chong, M. S., Wakaiki, M., and Hespanha, J. P. Observability of linear systems under adversarial attacks. In *American Control Conference*, pp. 2439–2444. IEEE, 2015.

Cohen, M. B., Lee, Y. T., Miller, G., Pachocki, J., and Sidford, A. Geometric median in nearly linear time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pp. 9–21, 2016.

Diakonikolas, I., Kamath, G., Kane, D., Li, J., Steinhardt, J., and Stewart, A. Sever: A robust meta-algorithm for stochastic optimization. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1596–1606. PMLR, 09–15 Jun 2019. URL `http://proceedings.mlr.press/v97/diakonikolas19a.html`.

Fang, M., Cao, X., Jia, J., and Gong, N. Local model poisoning attacks to byzantine-robust federated learning. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pp. 1605–1622, 2020.

Guerraoui, R., Rouault, S., et al. The hidden vulnerability of distributed learning in byzantium. In *International Conference on Machine Learning*, pp. 3521–3530. PMLR, 2018.

Gupta, N. and Vaidya, N. H. Byzantine fault tolerant distributed linear regression. *arXiv preprint arXiv:1903.08752*, 2019.

Gupta, N. and Vaidya, N. H. Fault-tolerance in distributed optimization: The case of redundancy. In *The 39th Symposium on Principles of Distributed Computing*, pp. 365–374, 2020a.

Gupta, N. and Vaidya, N. H. Resilience in collaborative optimization: Redundant and independent cost functions. *arXiv preprint arXiv:2003.09675*, 2020b.

Kairouz, P. and McMahan, H. B. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1), 2021.

Karimireddy, S. P., He, L., and Jaggi, M. Learning from history for byzantine robust optimization. *CoRR*, abs/2012.10333, 2020. URL `https://arxiv.org/abs/2012.10333`.

Kuwaranancharoen, K., Xin, L., and Sundaram, S. Byzantine-resilient distributed optimization of multi-dimensional functions. In *2020 American Control Conference (ACC)*, pp. 4399–4404. IEEE, 2020.

Lamport, L., Shostak, R., and Pease, M. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4(3):382–401, 1982.

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

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

Liu, S., Gupta, N., and Vaidya, N. H. Approximate byzantine fault-tolerance in distributed optimization. *arXiv preprint arXiv:2101.09337*, 2021.

Mhamdi, E. M. E., Guerraoui, R., and Rouault, S. Distributed momentum for byzantine-resilient stochastic gradient descent. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=H8UHdhWG6A3`.

Mishra, S., Shoukry, Y., Karamchandani, N., Diggavi, S. N., and Tabuada, P. Secure state estimation against sensor attacks in the presence of noise. *IEEE Transactions on Control of Network Systems*, 4(1):49–59, 2016.

Muñoz-González, L., Co, K. T., and Lupu, E. C. Byzantine-robust federated machine learning through adaptive model averaging. *arXiv preprint arXiv:1909.05125*, 2019.

Prasad, A., Suggala, A. S., Balakrishnan, S., and Ravikumar, P. Robust estimation via robust gradient estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(3):601–627, 2020.

So, J., Güler, B., and Avestimehr, A. S. Byzantine-resilient secure federated learning. *IEEE Journal on Selected Areas in Communications*, 2020.

Sohn, J.-y., Han, D.-J., Choi, B., and Moon, J. Election coding for distributed learning: Protecting signsgd against byzantine attacks. *Advances in Neural Information Processing Systems*, 33, 2020.

Su, L. and Shahrampour, S. Finite-time guarantees for Byzantine-resilient distributed state estimation with noisy measurements. *arXiv preprint arXiv:1810.10086*, 2018.

Su, L. and Shahrampour, S. Finite-time guarantees for byzantine-resilient distributed state estimation with noisy measurements. *IEEE Transactions on Automatic Control*, 65(9):3758–3771, 2019.

Su, L. and Vaidya, N. H. Fault-tolerant multi-agent optimization: optimal iterative distributed algorithms. In *Proceedings of the 2016 ACM symposium on principles of distributed computing*, pp. 425–434. ACM, 2016.

Su, L. and Vaidya, N. H. Byzantine-resilient multi-agent optimization. *IEEE Transactions on Automatic Control*, 2020.

Sundaram, S. and Gharesifard, B. Distributed optimization under adversarial nodes. *IEEE Transactions on Automatic Control*, 2018.

Wu, Z., Ling, Q., Chen, T., and Giannakis, G. B. Federated variance-reduced stochastic gradient descent with robustness to byzantine attacks. *IEEE Transactions on Signal Processing*, 68:4583–4596, 2020.

Xie, C., Koyejo, O., and Gupta, I. Generalized Byzantine-tolerant sgd. *arXiv preprint arXiv:1802.10116*, 2018a.

Xie, C., Koyejo, O., and Gupta, I. Phocas: dimensional byzantine-resilient stochastic gradient descent. *CoRR*, abs/1805.09682, 2018b. URL `http://arxiv.org/abs/1805.09682`.

Xie, C., Huang, K., Chen, P.-Y., and Li, B. Dba: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2019.

Yang, Z. and Bajwa, W. U. Byrdie: Byzantine-resilient distributed coordinate descent for decentralized learning, 2017.

Yang, Z. and Bajwa, W. U. Byrdie: Byzantine-resilient distributed coordinate descent for decentralized learning. *IEEE Transactions on Signal and Information Processing over Networks*, 5(4):611–627, 2019.

Yin, D., Chen, Y., Ramchandran, K., and Bartlett, P. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, pp. 5636–5645, 2018.