On Byzantine-Resilient High-Dimensional Stochastic Gradient Descent

Deepesh Data and Suhas Diggavi
University of California, Los Angeles, USA
Email: {deepesh.data@gmail.com, suhas@ee.ucla.edu}

***Abstract*—We study stochastic gradient descent (SGD) in the master-worker architecture under Byzantine attacks. Building upon the recent advances in algorithmic high-dimensional robust statistics, in each SGD iteration, master employs a non-trivial decoding to estimate the true gradient from the unbiased stochastic gradients received from workers, some of which may be corrupt.**

We provide convergence analyses for both strongly-convex and non-convex smooth objectives under standard SGD assumptions. We can control the approximation error of our solution in both these settings by the mini-batch size of stochastic gradients; and we can make the approximation error as small as we want, provided that workers use a sufficiently large mini-batch size. Our algorithm can tolerate less than $\frac{1}{3}$ fraction of Byzantine workers. It can approximately find the optimal parameters in the strongly-convex setting *exponentially fast*, and reaches to an approximate stationary point in the non-convex setting with *linear speed*, i.e., with a rate of $\frac{1}{T}$, thus, matching the convergence rates of vanilla SGD in the Byzantine-free setting.

I. Introduction

Stochastic gradient descent (SGD) [1] is the main workhorse behind the optimization procedure in several modern large-scale learning algorithms [2]. In this paper, we consider a master-worker architecture, where the training data is distributed across several machines (workers) and a central node (master) wants to learn a machine learning model using SGD [3]. This setting naturally arises in the case of *federated learning* [4], [5], where user devices are recruited to help build machine learning models. This can also arise in a distributed setup, where data is partitioned and stored in many servers to speed up the computation. In such scenarios, the recruited worker nodes may not be trusted with their computation, either because of non-Byzantine failures, such as software bugs, noisy training data, etc., or because of Byzantine attacks, where corrupt nodes may manipulate the information to their advantage [6]. These Byzantine adversaries may collaborate and arbitrarily deviate from their pre-specified programs. Training machine learning models in the presence of Byzantine attacks has received attention lately [7]–[18] and also in the context of the Internet of Battlefield Things (IoBT) [19]. The importance of this problem motivates us to study Byzantine-resilient optimization algorithms that are suitable for large-scale learning problems. See Section I-A where we put our work in context.

In this paper, we study empirical risk minimization using parallel mini-batch SGD in the presence of Byzantine adversaries, where all workers can access the data, and master iteratively builds a machine learning model using the gradients computed at the workers. We do not make any probabilisitic assumption on data generation. In our setup, an ϵ-fraction of workers may be under Byzantine attacks (where $\epsilon > 0$ is a constant), and corrupt workers may collaborate and report adversarially chosen gradients to the master. We propose a method using tools from high-dimensional robust mean estimation [20]–[23] that can tolerate less than $\frac{1}{3}$ fraction of corrupt workers. In particular, we use the outlier-filtering procedure from [22] to filter-out corrupt gradients in each SGD iteration.

Our contributions. We provide convergence analyses for both strongly-convex and non-convex smooth objectives under standard SGD assumptions; see Theorem 1. In the strongly-convex case, our algorithm can find optimal parameters within an approximation error of $\mathcal{O}(\frac{\sigma^2}{bR} + \frac{\sigma^2 d\hat{\epsilon}}{bR})$ (where $\hat{\epsilon} > \epsilon$ is any constant and b is the mini-batch size for stochastic gradients) "exponentially fast"; and in the non-convex case, it can find an approximate stationary point within the same error with "linear speed", i.e., with a rate of $\frac{1}{T}$. The first term $\frac{\sigma^2}{bR}$ in the approximation error is the standard SGD variance term and the second term $\frac{\sigma^2 d\epsilon'}{bR}$ is due to Byzantine attacks. Note that both these terms can be made small by taking a large batch size b. Also note that, in order to use the decoding algorithm of [22], we need to prove a concentration result (as stated in Theorem 2), which we show by building upon some tools in [22]. To the best of our knowledge, this is the first paper that studies Byzantine-resilient SGD under standard assumptions and provides convergence analyses for both strongly-convex and non-convex smooth objectives.

A. Related work

Byzantine-resilient distributed computing has a long history [6] and is a very well studied topic, which has received recent attention in the context of distributed learning [7]–[18]. The approach taken to tackle the Byzantine attacks in literature can be broadly divided into two categories: [7]–[12], [18] make statistical assumptions, either on the data (e.g., i.i.d. data) or on the algorithm (e.g., SGD), together with a non-trivial decoding at the master; [13]–[17] employ coding-theoretic/redundancy-based techniques to mitigate Byzantine attacks. The setting of [9], [12], [18] is similar to ours: Using martingale-based methods, [9] shows convergence under the assumption that the stochastic error in gradients is bounded with probability 1 (instead of assuming bounded variance); [12] considers non-convex objectives and shows an almost sure convergence of

gradients under stringent conditions; and [18] studies linear regression only – it removes corrupt nodes based on norm-filtering and achieves an error that scales with the number of data points. In this paper, we consider mini-batch SGD (without making probabilistic assumptions on the data), and unlike previous works, we can control the approximation error by the mini-batch size of stochastic gradients – larger the batch size, better the accuracy.

There have been works in *full-batch* gradient descent against Byzantine attacks, where data at workers is drawn i.i.d. from a probability distribution, and the goal is to minimize the *population risk* [7], [8], [10], [11]. In the following, n denotes the number of data points that each worker has and R denotes the total number of workers. [7] employed coordinate-wise median and trimmed median, and gave an approximation error of $\widetilde{\mathcal{O}}(\frac{d^2}{nR})$[1] for both convex and non-convex objectives, which could be prohibitive in high-dimensional problems; [11] and [10] considered only strongly-convex objectives, where [11] used decoding based on median-of-means and gave an error of $\widetilde{\mathcal{O}}(\frac{\epsilon d}{n})$, and [10] improved it to $\widetilde{\mathcal{O}}(\frac{d}{nR})$ for constant ϵ – observe that these papers only study strongly-convex objectives, whereas, in addition, we study non-convex objectives too. The decoding algorithm in [10] is taken from [22], which is based on robust mean estimation, and we also use that algorithm in our decoding. [8] proposed and analyzed an algorithm to avoid saddle-point attacks in non-convex problems and provided *second-order* convergence guarantees, and we believe that our results can also be extended to combat the saddle-point attacks in non-convex problems, which we leave as part of the future work. In the high-dimensional setting, they also used the decoding algorithm of [22] and gave an approximation error of $\widetilde{\mathcal{O}}(\frac{d}{nR})$. Note that these results are not directly comparable to ours, as they study *full-batch* gradient descent and assume that the workers' training data come i.i.d. from a probability distribution and minimize *population risk*, which is different from our parallel SGD setup, where there is no data distribution and we minimize *empirical risk*; and instead of taking full gradients, we work with a relaxed mini-batch stochastic gradients. As it turns out, an advantage of doing mini-batch SGD (over full-batch GD) is that it allows a tradeoff between the mini-batch size and the approximation error.

Though, similar to [8], [10], we also use the same decoding algorithm of [22] to combat Byzantine attacks, however, there are technical differences between ours and these works. In order to use the decoding algorithm of [22], both these works derive a matrix concentration bound, the need of which arises because they minimize the *population* risk. In this paper, since we minimize the *empirical* risk, we do not need such a result. However, we also need to prove a concentration bound (which is of a very different nature than theirs and is stated in Theorem 2), the need of which arises because the gradients in our work are *stochastic* due to SGD – if we work with *full-batch* deterministic gradients as in [8], [10], we would not need any such concentration bounds.

[1]The $\widetilde{\mathcal{O}}$ and $\widetilde{\Omega}$ notations hide logarithmic factors.

Paper organization. We describe the problem setup in Section II and state our main convergence results for strongly-convex and non-convex objectives in Section III. We describe how we use the robust mean estimation result of [22] in our SGD setting in Section IV. The omitted proofs from this paper can be derived as special cases from the full version [24], which studies Byzantine-resilient SGD in a more general *heterogeneous* data setting.

II. Problem Setup

We are given n training samples $\{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$ in $\mathbb{R}^d$, and we want to learn a model $\mathbf{x} \in \mathbb{R}^d$ that minimizes the average empirical risk $\frac{1}{n}\sum_{i=1}^n F_i(\mathbf{x})$, where $F_i : \mathbb{R}^d \to \mathbb{R}$ is the risk associated with the i'th sample $\mathbf{s}_i$. In other words, we want to solve the following unconstrained minimization problem:

$$\arg\min_{\mathbf{x}\in\mathbb{R}^d} \left(F(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^n F_i(\mathbf{x}) \right). \tag{1}$$

When F is strongly-convex, let the minimization in (1) be attained at $\mathbf{x}^*$. In the case of non-convex F, as standard in literature, we find a stationary point where the gradient becomes zero.

We can minimize (1) using *stochastic gradient descent* (SGD), which is an iterative algorithm that updates the parameters according to the following update rule:

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \eta \nabla F_{i_t}(\mathbf{x}^t), \quad t = 1, 2, 3, \ldots \tag{2}$$

where $i_t \in_U [n]$ is sampled uniformly at random from $[n] := \{1, 2, \ldots, n\}$ and η is a constant step-size. Note that $\mathbb{E}_{i\in_U[n]}[\nabla F_i(\mathbf{x})] = \nabla F(\mathbf{x})$ holds for all $\mathbf{x} \in \mathbb{R}^d$. We make a standard assumption about SGD, namely, the bounded variance assumption, which states that $\mathbb{E}_{i\in_U[n]}\|\nabla F_i(\mathbf{x}) - \nabla F(\mathbf{x})\|^2 \leq \sigma^2 < \infty$ holds for every $\mathbf{x} \in \mathbb{R}^d$.

In the master-worker architecture for parallel SGD, based on the current parameter vector, workers send unbiased stochastic gradients to the master, which, upon receiving the gradients, updates the parameter vector iteratively. Concretely, at the t'th iteration, master broadcasts $\mathbf{x}^t$; each worker $r \in [R]$ sends $\mathbf{g}_r(\mathbf{x}^t) := \nabla F_{r_t}(\mathbf{x}^t)$ to the master for a randomly chosen $r_t \in_U [n]$, independent of the choice of other workers; master updates the parameter vector according to $\mathbf{x}^{t+1} = \mathbf{x}^t - \eta\frac{1}{R}\sum_{r=1}^R \nabla \mathbf{g}_r(\mathbf{x}^t)$. Since the variance of the average stochastic gradients reduces by a factor of R, this speeds up the SGD convergence.

This aggregation rule at the master (i.e., the averaging) is vulnerable to Byzantine attacks, where, instead of sending the true stochastic gradients, the corrupt workers may send adversarially chosen vectors to disrupt the computation. It is known that even a single Byzantine worker can prevent the algorithm to convergence, even worse, it can cause the algorithm to converge to an adversarially chosen point [12]. Our adversary model is described next.

Adversary model. We assume that an ϵ fraction of R workers are corrupt (where $\epsilon > 0$ is a constant and we will

decide its value later). The corrupt workers can collaborate and arbitrarily deviate from their pre-specified programs: In any SGD iteration, instead of sending the true stochastic gradients, corrupt workers may send adversarially chosen vectors (they may not even send anything if they wish, in which case master can treat them as *erasures* and replace them with a fixed value). Observe that, in the erasure case, master knows which workers are corrupt; whereas, in the Byzantine problem, master does not have this information, which makes solving this problem challenging; see also Section IV for a more detailed discussion on why solving this problem in general is difficult.

III. Our Results

We tackle the Byzantine behavior of corrupt workers by applying a non-trivial decoding algorithm at the master node in each iteration of the parallel SGD algorithm; see Algorithm 1. Our decoding algorithm is inspired by the recent breakthrough results in theoretical computer science for robust mean estimation [20]–[22], and we use the outlier-filtering algorithm from [22], in particular.

Before stating our results, we need to formally define *mini-batch* SGD. Note that we can further speed up the convergence of parallel SGD by having each worker sample many data points (without replacement), say, $b \geq 1$ data points, and send the average gradients on these data points to the master. This is called mini-batch SGD. To formalize this, for any $\mathbf{x} \in \mathbb{R}^d$, consider the following set

$$\mathcal{F}^{\otimes b}(\mathbf{x}) := \left\{ \frac{1}{b} \sum_{i \in \mathcal{H}} \nabla F_i(\mathbf{x}) : \mathcal{H} \in \binom{[n]}{b} \right\}. \tag{3}$$

Note that each element of $\mathcal{F}^{\otimes b}(\mathbf{x})$ is the average of b randomly chosen elements (without replacement) of $\mathcal{F}(\mathbf{x}) := \mathcal{F}^{\otimes 1}(\mathbf{x})$. It is not hard to show that if we pick an element from $\mathcal{F}^{\otimes b}(\mathbf{x})$ uniformly at random, its mean remains unchanged and is equal to $\nabla F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n \nabla F_i(\mathbf{x})$, and the variance reduces by a factor of b, i.e.,

$$\mathbb{E}_{Z \leftarrow \mathcal{F}^{\otimes b}(\mathbf{x})}[Z] = \nabla F(\mathbf{x}), \tag{4}$$

$$\mathbb{E}_{Z \leftarrow \mathcal{F}^{\otimes b}(\mathbf{x})} \|Z - \nabla F(\mathbf{x})\|^2 \leq \frac{\sigma^2}{b}. \tag{5}$$

In this modified setup, we can equivalently describe our parallel mini-batch SGD algorithm as follows: At the t'th iteration, upon receiving the parameter vector $\mathbf{x}^t$ from the master, each worker $r \in [R]$ samples $\mathbf{g}_r(\mathbf{x}^t) \in_U \mathcal{F}^{\otimes b}(\mathbf{x}^t)$ (independent of the other workers) and sends it to the master. Suppose the master receives $\tilde{\mathbf{g}}_1(\mathbf{x}^t), \ldots, \tilde{\mathbf{g}}_R(\mathbf{x}^t)$, where $\tilde{\mathbf{g}}_r(\mathbf{x}^t) = \mathbf{g}_r(\mathbf{x}^t)$ if the r'th worker is honest, otherwise can be arbitrary. Note that $\mathbb{E}[\mathbf{g}_r(\mathbf{x}^t)] = \nabla F(\mathbf{x}^t)$ and $\mathbb{E}\|\mathbf{g}_r(\mathbf{x}^t) - \nabla F(\mathbf{x}^t)\|^2 \leq \frac{\sigma^2}{b}$ holds for every $r \in [R]$. Upon receiving $\{\tilde{\mathbf{g}}_r(\mathbf{x}^t)\}_{r\in[R]}$, master applies a decoding algorithm (from [22]; see also the discussion in Section IV) and outputs $\hat{\mathbf{g}}(\mathbf{x}^t)$ (which is an estimate of $\nabla F(\mathbf{x}^t)$), based on which it updates the parameters according to (6). The goal of the master is to estimate $\nabla F(\mathbf{x}^t)$ as accurately as possible in each SGD iteration. We present our Byzantine-resilient SGD algorithm in Algorithm 1.

Algorithm 1 Byzantine-Resilient SGD

1: **Initialize.** Set $\mathbf{x}^0 := \mathbf{0}$. Fix a constant step-size η and a mini-batch size b.
2: **for** $t = 0$ **to** $T-1$ **do**
3: **On Workers:**
4: **for** $r = 1$ **to** R **do**
5: Receive $\mathbf{x}^t$ from master. Take a mini-batch stochastic gradient $\mathbf{g}_r(\mathbf{x}^t) \in_U \mathcal{F}^{\otimes b}(\mathbf{x}^t)$.
6: $\tilde{\mathbf{g}}_r(\mathbf{x}^t) = \begin{cases} \mathbf{g}_r(\mathbf{x}^t) & \text{if worker } r \text{ is honest,} \\ \divideontimes & \text{if worker } r \text{ is corrupt,} \end{cases}$ where $\divideontimes$ is an arbitrary vector in $\mathbb{R}^d$.
7: Send $\tilde{\mathbf{g}}_r(\mathbf{x}^t)$ to master.
8: **end for**
9: **At Master:**
10: Receive $\{\tilde{\mathbf{g}}_r(\mathbf{x}^t)\}_{r=1}^R$ from the R workers.
11: Apply the decoding algorithm RGE (from [22]) on $\{\tilde{\mathbf{g}}_r(\mathbf{x}^t)\}_{r=1}^R$. Let $\hat{\mathbf{g}}(\mathbf{x}^t) = \text{RGE}(\tilde{\mathbf{g}}_1(\mathbf{x}^t), \ldots, \tilde{\mathbf{g}}_R(\mathbf{x}^t))$.
12: Update the parameter vector:
$$\mathbf{x}^{t+1} = \mathbf{x}^t - \eta \hat{\mathbf{g}}(\mathbf{x}^t). \tag{6}$$
13: Broadcast $\mathbf{x}^{t+1}$ to all workers.
14: **end for**

Our convergence results are for both strongly-convex and non-convex smooth functions and are stated below.

Theorem 1 (Strongly-convex and Non-convex)**.** *Suppose an $\epsilon > 0$ fraction of R workers are adversarially corrupt. For an L-smooth[2] objective function $F : \mathbb{R}^d \to \mathbb{R}$, let Algorithm 1 generate a sequence of iterates $\{\mathbf{x}^t\}_{t=0}^T$ when run with a constant step-size η. Fix an arbitrary constant $\epsilon' > 0$. If $\epsilon < \frac{1}{3} - \epsilon'$, then with probability at least $1 - T\exp(-\frac{\epsilon'^2(1-\epsilon)R}{16})$, we have the following convergence guarantees:*

- **Strongly-convex:** *If F is also μ-strongly convex[3] and we take $\eta = \frac{1}{2L}$, then $\{\mathbf{x}^t\}_{t=0}^T$ satisfy*
$$\mathbb{E}\|\mathbf{x}^T - \mathbf{x}^*\|^2 \leq \left(1 - \frac{\mu}{4L}\right)^T \|\mathbf{x}^0 - \mathbf{x}^*\|^2 + \frac{5}{\mu^2}\Gamma. \tag{7}$$
If we take $T \geq \log\left(\frac{\mu^2}{\Gamma}\|\mathbf{x}^0 - \mathbf{x}^*\|^2\right)/\log(\frac{1}{1-\mu/4L})$, we get $\mathbb{E}\|\mathbf{x}^T - \mathbf{x}^*\|^2 \leq \frac{6}{\mu^2}\Gamma$.
- **Non-convex:** *If we take $\eta = \frac{1}{4L}$, then $\{\mathbf{x}^t\}_{t=0}^T$ satisfy*
$$\frac{1}{T}\sum_{t=0}^T \mathbb{E}\|\nabla F(\mathbf{x}^t)\|^2 \leq \frac{8L^2}{T}\|\mathbf{x}^0 - \mathbf{x}^*\|^2 + 3\Gamma. \tag{8}$$
If we take $T \geq \frac{8L^2\|\mathbf{x}^0 - \mathbf{x}^*\|^2}{\Gamma}$, we get $\frac{1}{T}\sum_{t=0}^T \mathbb{E}\|\nabla F(\mathbf{x}^t)\|^2 \leq 4\Gamma$.

In both (7) and (8), expectation is taken over the sampling of mini-batch stochastic gradients. Here, $\Gamma = \frac{2\sigma^2}{(1-(\epsilon+\epsilon'))bR} + 2\Upsilon^2$ with $\Upsilon = \mathcal{O}\left(\sigma_0\sqrt{\epsilon+\epsilon'}\right)$, where $\sigma_0^2 = \frac{16\sigma^2}{b\epsilon'}\left(1 + \frac{d}{(1-(\epsilon+\epsilon'))R}\right)$.

Due to lack of space, we omit the proof of Theorem 1: It can be derived as a spacial case from our full version [24,

[2] $F(\mathbf{y}) \leq F(\mathbf{x}) + \langle \nabla F(\mathbf{x}), y - x\rangle + \frac{L}{2}\|y - x\|_2^2, \quad \forall\, \mathbf{x}, \mathbf{y} \in \mathbb{R}^d.$
[3] $F(\mathbf{y}) \geq F(\mathbf{x}) + \langle \nabla F(\mathbf{x}), y - x\rangle + \frac{\mu}{2}\|y - x\|_2^2, \quad \forall\, \mathbf{x}, \mathbf{y} \in \mathbb{R}^d.$

Theorem 1], which studies Byzantine-resilient SGD in a more general heterogeneous data setting.

Analysis of the approximation error. In both parts of Theorem 1, the approximation error Γ consists of two error terms: First is $\Gamma_1 = \mathcal{O}(\frac{\sigma^2}{(1-(\epsilon+\epsilon'))bR})$, which is the standard error arising due to stochastic sampling of gradients; and second is $\Gamma_2 = \mathcal{O}\left(\frac{\sigma^2}{b\epsilon'}\left(1+\frac{d}{(1-(\epsilon+\epsilon'))R}\right)(\epsilon+\epsilon')\right)$, which is due to Byzantine attacks. Observe that both Γ_1 and Γ_2 decrease with the mini-batch size b and the number of workers R, as desired, and we can make them as small as we want by taking a sufficiently large batch size b of stochastic gradients.

Convergence rates. Note that, in the strongly-convex case, Algorithm 1 approximately finds the optimal parameters $\mathbf{x}^*$ (within Γ error, which could be a constant) "exponentially fast"; and in the non-convex case, Algorithm 1 approximately finds a stationary point up to the same error with "linear speed", i.e., with a rate of $\frac{1}{T}$. Thus, we recover the convergence rates of vanilla SGD (running in the Byzantine-free setting) for both the objectives.

Corruption threshold. Our proposed algorithm can tolerate up to $\frac{1}{3}$ fraction Byzantine workers, which is away from the information-theoretic optimal $\frac{1}{2}$ fraction. The $\frac{1}{3}$ bound comes from the subroutine of robust mean estimation (RME) that we use for robust gradient estimation (RGE), as explained in Section IV. So, improved algorithms for RME that can be adapted to our setting will directly give an improved corruption threshold for our algorithm.

Failure probability. The failure probability of our algorithm is at most $T\exp(-\frac{\epsilon'^2(1-\epsilon)R}{16})$, which is at most δ, for any $\delta > 0$, provided we run our algorithm for $T \leq \delta\exp(\frac{\epsilon'^2(1-\epsilon)R}{16})$ iterations. Though the error probability scales linearly with T, it also goes down *exponentially* with the number of workers R. As a result, in large-scale distributed settings (e.g., federated learning [5]), where number of workers R could be very large (in tens of thousands, or in millions), we can get a very small probability of error, say, $1/100$, even if run our algorithm for a very long time.

IV. Robust Gradient Estimation (RGE)

We are given R gradient vectors $\tilde{\mathbf{g}}_1(\mathbf{x}^t), \ldots, \tilde{\mathbf{g}}_R(\mathbf{x}^t) \in \mathbb{R}^d$, where, if the r'th worker is honest, then $\tilde{\mathbf{g}}_r(\mathbf{x}^t) = \mathbf{g}_r(\mathbf{x}^t)$ is a uniform sample from $\mathcal{F}^{\otimes b}(\mathbf{x})$; otherwise, if r'th worker is corrupt, then $\tilde{\mathbf{g}}_r(\mathbf{x}^t)$ can be arbitrary. We want to output $\hat{\mathbf{g}}(\mathbf{x}^t)$, an estimate of $\nabla F(\mathbf{x})$, such that $\|\hat{\mathbf{g}}(\mathbf{x}) - \nabla F(\mathbf{x})\|$ is small for all $\mathbf{x} \in \mathbb{R}^d$. Note that $\nabla F(\mathbf{x}^t)$ is the mean of $\mathcal{F}^{\otimes b}(\mathbf{x})$. We use the outlier-filtering algorithm from [22] which was developed for robust mean estimation in high dimensions, and in order to use that we prove a concentration bound.

This problem is related to robust mean estimation (RME) in high dimensions, in which we are given R samples in $\mathbb{R}^d$ (out of which an ϵ-fraction is corrupted) from an unknown distribution with unknown mean, and the goal is to estimate its mean. This is a classic problem in robust statistics [25], [26]. Until recently, all the solutions to this problem were either computationally intractable or were very poor in terms of the quality of the estimator produced. The method of *Tukey median* [27] solves this problem with dimension-independent error guarantees, but it is NP-hard to compute in general [28]. On the other hand, solutions based on *geometric-median*, *coordinate-wise median* are computationally tractable, but can only give dimension-dependent error guarantees, which scales as $\sqrt{d}$ [20].

Why is robust mean estimation in high-dimensions such a difficult problem? To understand this, assume that gradients are distributed according to a high-dimensional Gaussian distribution $\mathcal{N}(\mathbf{0}, I)$. It is a well known fact that samples from $\mathcal{N}(\mathbf{0}, I)$ lie around the annulus at a distance $\sqrt{d}$ from the origin, w.h.p. So, it would not be in the adversary's best interest to put corrupt samples far from the annulus, as they can be trivially filtered out just based on the norm. However, the adversary can put the corrupt samples in a concentrated form around the annulus, which cannot be detected just based on then norm, but can shift the sample mean away from the true mean in an adversarially chosen direction. This makes devising efficient decoding algorithm with good approximation guarantees highly non-trivial.

Recently, Lai et al. [20] and Diakonikolas et al. [21] in their breakthrough papers independently provided *computationally efficient* algorithms for robust mean estimation that give *dimension-independent* error guarantees. Following these papers, there has been a flurry of research improving upon their results in various directions; see [23] and references therein. Most of these papers focus on particular distributions, e.g., Gaussian, which is not applicable in our setting, as we only assume that gradients have bounded variance. It should be noted that the sample complexity (i.e., the number of samples required) for robustly estimating the mean grows at least linearly with the dimension d [20].

To map our problem to RME, note that our R samples come from a uniform distribution over the discrete set $\mathcal{F}^{\otimes b}(\mathbf{x})$, whose mean is equal to $\nabla F(\mathbf{x})$ and it has variance bounded by $\frac{\sigma^2}{b}$; see (4), (5). Our goal is to estimate the mean $\nabla F(\mathbf{x})$.

In view of the sample complexity result in mean estimation, it means that for robustly estimating $\nabla F(\mathbf{x})$, the number of workers R should grow linearly with the dimension d. In a distributed setup, since it is not practical to increase the number of workers with the dimension of the problem, we address this issue by increasing the mini-batch size b. Since the variance of the mini-batch stochastic gradients (which are uniform samples from $\mathcal{F}^{\otimes b}(\mathbf{x})$) reduces as the batch-size b increases. This implies that as we increase b, the resulting gradients from each worker become closer to the mean than earlier; and as we see later, this will cut down the requirement that R grows linearly with d. Observe that it is crucial that increasing the mini-batch size does not change the mean, as we want to estimate $\nabla F(\mathbf{x})$ using $\mathcal{F}^{\otimes b}(\mathbf{x})$.

For robustly estimating the mean $\nabla F(\mathbf{x})$, we will use some techniques developed by Steinhardt et al. [22]. First we define a notion called (ϵ, δ)-*resilience* for a given set $\mathcal{S}$, which says that if $\mathcal{S}$ is resilient around a point μ (which need not be its

mean), then dropping some elements from $\mathcal{S}$ does not change the concentration of the resulting set around μ by much.

Definition 1 (Resilience, [22]). *A set $\mathcal{S} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m\}$ of m points, each lying in $\mathbb{R}^d$, is* (ϵ, δ)-resilient *around a point $\mu \in \mathbb{R}^d$, if every subset $\mathcal{T} \subseteq \mathcal{S}$ of cardinality at least $(1-\epsilon)m$ satisfies $\left\| \frac{1}{|\mathcal{T}|} \sum_{\mathbf{y} \in \mathcal{T}} (\mathbf{y} - \mu) \right\|_2 \leq \delta$.*

The notion of resilience is useful for us for robustly estimating the true gradient because (i) it is known that if a set of points contains a resilient set around μ, then we can estimate μ within a bounded error, and (ii) we can show that there exists a large subset of uncorrupt gradients from the received R stochastic gradients (out of which an ϵ fraction is arbitrarily corrupt), that is resilient around its mean. We make these statements precise in the subsequent discussion. Our main result for robust gradient estimation is as follows:

Theorem 2. *Fix an arbitrary $\mathbf{x} \in \mathbb{R}^d$. Suppose we are given R gradients $\tilde{\mathbf{g}}_1(\mathbf{x}), \ldots, \tilde{\mathbf{g}}_R(\mathbf{x}) \in \mathbb{R}^d$, where $\tilde{\mathbf{g}}_r(\mathbf{x}) = \mathbf{g}_r(\mathbf{x})$ is a uniform sample from $\mathcal{F}^{\otimes b}(\mathbf{x})$ if the r'th worker is honest, otherwise can be arbitrary. For any constant $\epsilon' > 0$, we have the following:*

1) *With probability $1 - \exp(-\frac{\epsilon'^2(1-\epsilon)R}{16})$, there exists a subset $\mathcal{S}$ of uncorrupted gradients of size $(1-(\epsilon+\epsilon'))R$ (with $\mathbf{g}_{\mathcal{S}}(\mathbf{x}) := \frac{1}{|\mathcal{S}|}\sum_{i \in |\mathcal{S}|} \mathbf{g}_i(\mathbf{x})$ being its sample mean) such that $\mathcal{S}$ is $(\mathcal{O}(\sigma_0\sqrt{\epsilon''}), \epsilon'')$-resilient around $\mathbf{g}_{\mathcal{S}}(\mathbf{x})$ for all $\epsilon'' < \frac{1}{2}$, where $\sigma_0^2 = \frac{16\sigma^2}{\epsilon'}\left(1 + \frac{d}{(1-(\epsilon+\epsilon'))R}\right)$.*
2) *If $\epsilon < \frac{1}{3} - \epsilon'$, then we can find an estimate $\hat{\mathbf{g}}(\mathbf{x})$ of $\mathbf{g}_{\mathcal{S}}(\mathbf{x})$ such that $\|\hat{\mathbf{g}}(\mathbf{x}) - \mathbf{g}_{\mathcal{S}}(\mathbf{x})\| \leq \mathcal{O}\left(\sigma_0\sqrt{\epsilon+\epsilon'}\right)$.*

We prove Theorem 2 with the help of some techniques developed in [22].

Lemma 1 (Proposition 20 in [22]). *Suppose a distribution p in $\mathbb{R}^d$ has bounded variance in all directions, i.e., $\mathbb{E}_{\mathbf{y}\sim p}[\langle \mathbf{y} - \mu, \mathbf{v}\rangle^2] \leq \sigma_p^2, \forall \mathbf{v} \in \mathbb{R}^d, \|\mathbf{v}\| = 1$. Then, given m samples $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_m \sim p$, with probability $1 - \exp(-\epsilon'^2 m/16)$, there is a subset $\mathcal{S}$ of $(1-\epsilon')m$ points such that $\frac{1}{|\mathcal{S}|}\sum_{\mathbf{y}\in\mathcal{S}}\langle \mathbf{y} - \mu, \mathbf{v}\rangle^2 \leq \frac{4\sigma_p^2}{\epsilon'}\left(1 + \frac{d}{(1-\epsilon')m}\right)$ holds for all unit vectors $\mathbf{v} \in \mathbb{R}^d$.*

Now we interpret Lemma 1 in our problem setting. Observe that, in our problem, p is a uniform distribution over $\mathcal{F}^{\otimes b}(\mathbf{x})$. It is easy to see that the hypothesis of Lemma 1 is satisfied with $\mathbf{y}_i = \mathbf{g}_i(\mathbf{x})$, $\mu = \nabla F(\mathbf{x})$, and $\sigma_p^2 = \frac{\sigma^2}{b}$:

$$\mathbb{E}_{\mathbf{y}\sim p}[\langle \mathbf{y} - \mu, \mathbf{v}\rangle^2] \overset{(a)}{\leq} \mathbb{E}_{\mathbf{y}\sim p}[\|\mathbf{y} - \mu\|^2] \cdot \|\mathbf{v}\|^2 \overset{(b)}{\leq} \frac{\sigma^2}{b},$$

where (a) follows from the Cauchy-Schwarz inequality and (b) uses (5) and $\|\mathbf{v}\| = 1$. We are given R samples, out of which at least $(1-\epsilon)R$ are according to the correct distribution. Now, by taking $m = (1-\epsilon)R$, Lemma 1 implies that there exists a subset $\mathcal{S}$ of uncorrupted gradients of size $(1-\epsilon')(1-\epsilon)R \geq (1-(\epsilon+\epsilon'))R$, that satisfies

$$\frac{1}{|\mathcal{S}|}\sum_{\mathbf{y}\in\mathcal{S}}\langle \mathbf{y} - \mu, \mathbf{v}\rangle^2 \leq \sigma_0^2, \quad \forall \mathbf{v} \in \mathbb{R}^d, \|\mathbf{v}\| = 1, \tag{9}$$

where $\sigma_0^2 = \frac{4\sigma^2}{b\epsilon'}\left(1 + \frac{d}{(1-(\epsilon+\epsilon'))R}\right)$.

Note that (9) is bounding the deviation of the points in $\mathcal{S}$ from the true mean μ. However, in order to show that the set $\mathcal{S}$ is resilient, we need to bound the deviation from its sample mean. For that, define $\mu_{\mathcal{S}} := \frac{1}{|\mathcal{S}|}\sum_{\mathbf{y}\in\mathcal{S}} \mathbf{y}$ to be the sample mean of $\mathcal{S}$.

$$\begin{aligned}\frac{1}{|\mathcal{S}|}\sum_{\mathbf{y}\in\mathcal{S}}\langle \mathbf{y} - \mu_{\mathcal{S}}, \mathbf{v}\rangle^2 &= \frac{1}{|\mathcal{S}|}\sum_{\mathbf{y}\in\mathcal{S}}\langle \mathbf{y} - \mu + \mu - \mu_{\mathcal{S}}, \mathbf{v}\rangle^2 \\ &\overset{(a)}{\leq} \frac{2}{|\mathcal{S}|}\sum_{\mathbf{y}\in\mathcal{S}}\langle \mathbf{y} - \mu, \mathbf{v}\rangle^2 + \frac{2}{|\mathcal{S}|}\sum_{\mathbf{y}\in\mathcal{S}}\langle \mu - \mu_{\mathcal{S}}, \mathbf{v}\rangle^2 \\ &\leq 2\sigma_0^2 \ + \ 2\langle \mu - \mu_{\mathcal{S}}, \mathbf{v}\rangle^2 \overset{(b)}{\leq} 4\sigma_0^2. \end{aligned} \tag{10}$$

In (a) we used the inequality $(a+b)^2 \leq 2a^2 + 2b^2$. In (b) we used $\langle \mu_{\mathcal{S}} - \mu, \mathbf{v}\rangle^2 \leq \sigma_0^2$, which can be shown as follows: $\langle \mu - \mu_{\mathcal{S}}, \mathbf{v}\rangle^2 = \left[\frac{1}{|\mathcal{S}|}\sum_{\mathbf{y}\in\mathcal{S}}\langle \mathbf{y} - \mu, \mathbf{v}\rangle\right]^2 \overset{(c)}{\leq} \frac{1}{|\mathcal{S}|}\sum_{\mathbf{y}\in\mathcal{S}}\langle \mathbf{y} - \mu, \mathbf{v}\rangle^2 \overset{(d)}{\leq} \sigma_0^2$, where (c) follows from the Jensen's inequality, and (d) follows from (9).

Having established the bound in (10), we can now show that $\mathcal{S}$ is $(\mathcal{O}(\sigma_0\sqrt{\epsilon''}), \epsilon'')$-resilient around its sample mean $\mu_{\mathcal{S}}$ for all $\epsilon'' < \frac{1}{2}$; see [24, Claim 11 in Appendix F] for a detailed proof of this. This proves the first part of Theorem 2.

Now we show that resilience of $\mathcal{S}$ is information-theoretically sufficient for robust recovery of mean $\nabla F(\mathbf{x})$:

Lemma 2 (Proposition 2 in [22]). *Suppose that a set $\widetilde{\mathcal{S}} = \{\mathbf{y}_1, \ldots, \mathbf{y}_R\}$ of R points in $\mathbb{R}^d$ contains a subset $\mathcal{S}$ of size $(1-\tilde{\epsilon})R$ that is $(\tilde{\sigma}, \frac{\tilde{\epsilon}}{1-\tilde{\epsilon}})$-resilient around its sample mean $\mu_{\mathcal{S}}$. Then, if $\tilde{\epsilon} < \frac{1}{2}$, we can recover $\hat{\mu}_{\mathcal{S}}$ such that $\|\hat{\mu}_{\mathcal{S}} - \mu_{\mathcal{S}}\| \leq 2\tilde{\sigma}$.*

In order to use the result of the first part of Theorem 2 in Lemma 2, $\tilde{\epsilon}$ must satisfy $\frac{\tilde{\epsilon}}{1-\tilde{\epsilon}} < \frac{1}{2}$, which is equivalent to requiring that $\tilde{\epsilon} < \frac{1}{3}$. Now, substituting $\epsilon'' = \frac{\epsilon+\epsilon'}{1-\epsilon-\epsilon'}$ and $\tilde{\sigma} = \mathcal{O}\left(\sigma_0\sqrt{\frac{\epsilon+\epsilon'}{1-\epsilon-\epsilon'}}\right) = \mathcal{O}(\sigma_0\sqrt{\epsilon+\epsilon'})$ proves the second part of Theorem 2.

This completes the proof of Theorem 2.

Remark 1. *Note that the algorithm of Theorem 2 for mean estimation may be inefficient (due to the potential inefficiency of Lemma 2), and an efficient polynomial time algorithm for the same task can be found in [22, Theorem 7], which can tolerate up to $\epsilon \leq \frac{1}{4} - \epsilon'$ fraction of Byzantine workers, for an arbitrary constant $\epsilon' > 0$. The reason for not giving the efficient algorithm here is because, rather than directly giving the algorithm, first we wanted to establish the conceptual connection between Byzantine mini-batch SGD and the setting of [22], and more importantly, how we can leverage their proof technique developed for RME in our SGD framework.*

ACKNOWLEDGEMENT

We would like to thank Navjot Singh for his help in the early stages of this work. This work was supported by the NSF grants #1740047, #1514731, and by the UC-NL grant LFR-18-548554.

REFERENCES

[1] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics. JSTOR*, vol. 22, no. 3, pp. 400–407, 1951.

[2] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *International Conference on Computational Statistics (COMPSTAT)*, 2010, pp. 177–186.

[3] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[4] J. Konecný, "Stochastic, distributed and federated optimization for machine learning," *CoRR*, vol. abs/1707.01155, 2017. [Online]. Available: http://arxiv.org/abs/1707.01155

[5] J. Konecný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *CoRR*, vol. abs/1610.02527, 2016. [Online]. Available: http://arxiv.org/abs/1610.02527

[6] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Trans. Program. Lang. Syst.*, vol. 4, no. 3, pp. 382–401, Jul. 1982.

[7] D. Yin, Y. Chen, K. Ramchandran, and P. L. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning (ICML)*, 2018, pp. 5636–5645.

[8] ——, "Defending against saddle point attack in byzantine-robust distributed learning," in *International Conference on Machine Learning (ICML)*, 2019, pp. 7074–7084.

[9] D. Alistarh, Z. Allen-Zhu, and J. Li, "Byzantine stochastic gradient descent," in *Neural Information Processing Systems (NeurIPS)*, 2018, pp. 4618–4628.

[10] L. Su and J. Xu, "Securing distributed gradient descent in high dimensional statistical learning," *POMACS*, vol. 3, no. 1, pp. 12:1–12:41, 2019.

[11] Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *POMACS*, vol. 1, no. 2, pp. 44:1–44:25, 2017.

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

[13] L. Chen, H. Wang, Z. B. Charles, and D. S. Papailiopoulos, "DRACO: byzantine-resilient distributed training via redundant gradients," in *International Conference on Machine Learning (ICML)*, 2018, pp. 902–911.

[14] S. Rajput, H. Wang, Z. B. Charles, and D. S. Papailiopoulos, "DETOX: A redundancy-based framework for faster and more robust gradient aggregation," in *Neural Information Processing Systems (NeurIPS)*, 2019, pp. 10 320–10 330.

[15] D. Data, L. Song, and S. N. Diggavi, "Data encoding for byzantine-resilient distributed optimization," *CoRR*, vol. abs/1907.02664, 2019. [Online]. Available: http://arxiv.org/abs/1907.02664

[16] ——, "Data encoding methods for byzantine-resilient distributed optimization," in *IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 2719–2723.

[17] D. Data and S. N. Diggavi, "Byzantine-tolerant distributed coordinate descent," in *IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 2724–2728.

[18] N. Gupta and N. H. Vaidya, "Byzantine fault-tolerant parallelized stochastic gradient descent for linear regression," in *Allerton Conference on Communication, Control, and Computing*, 2019, pp. 415–420.

[19] T. F. Abdelzaher *et al.*, "Toward an internet of battlefield things: A resilience perspective," *IEEE Computer*, vol. 51, no. 11, pp. 24–36, 2018.

[20] K. A. Lai, A. B. Rao, and S. S. Vempala, "Agnostic estimation of mean and covariance," in *IEEE Foundations of Computer Science (FOCS)*, 2016, pp. 665–674.

[21] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart, "Robust estimators in high dimensions without the computational intractability," in *IEEE Foundations of Computer Science (FOCS)*, 2016, pp. 655–664.

[22] J. Steinhardt, M. Charikar, and G. Valiant, "Resilience: A criterion for learning in the presence of arbitrary outliers," in *Innovations in Theoretical Computer Science Conference (ITCS)*, 2018, pp. 45:1–45:21.

[23] I. Diakonikolas and D. M. Kane, "Recent advances in algorithmic high-dimensional robust statistics," *CoRR*, vol. abs/1911.05911, 2019. [Online]. Available: http://arxiv.org/abs/1911.05911

[24] D. Data and S. Diggavi, "Byzantine-resilient SGD in high dimensions on heterogeneous data," *CoRR*, 2020, Available on arXiv.

[25] J. W. Tukey, "A survey of sampling from contaminated distributions," *Contributions to probability and statistics*, vol. 2, pp. 448–485, 1960.

[26] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, 03 1964.

[27] J. W. Tukey, "Mathematics and picturing of data," in *Proceedings of ICM*, vol. 6, 1975, pp. 523–531.

[28] D. S. Johnson and F. P. Preparata, "The densest hemisphere problem," *Theor. Comput. Sci.*, vol. 6, pp. 93–107, 1978.