Byzantine-Resilient SGD in High Dimensions on Heterogeneous Data

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

***Abstract*—We study distributed stochastic gradient descent (SGD) in the master-worker architecture under Byzantine attacks. We consider the heterogeneous data model, where different workers may have different local datasets, and we do not make any probabilistic assumptions on data generation. At the core of our algorithm, we use the polynomial-time outlier-filtering procedure for robust mean estimation proposed by Steinhardt et al. (ITCS 2018) to filter-out corrupt gradients. In order to be able to apply their filtering procedure in our *heterogeneous* data setting where workers compute *stochastic* gradients, we derive a new matrix concentration result, which may be of independent interest. We provide convergence analyses for smooth strongly-convex and non-convex objectives and show that our convergence rates match that of vanilla SGD in the Byzantine-free setting. In order to bound the heterogeneity, we assume that the gradients at different workers have bounded deviation from each other, and we also provide concrete bounds on this deviation in the statistical heterogeneous data model.**

I. Introduction

Stochastic gradient descent (SGD) [2] is the main workhorse behind the optimization procedure in several modern large-scale learning algorithms [3]. 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 [4]; see Figure 1. This setting naturally arises in the case of *federated learning* [5]–[7], where user devices are recruited to help build machine learning models using their locally generated data. 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 transmitted information to their advantage [8]. These Byzantine adversaries may collaborate and arbitrarily deviate from their pre-specified programs. The importance of this problem motivates us to study Byzantine-resilient optimization algorithms that are suitable for large-scale learning problems.

We consider an empirical risk minimization (ERM) problem, where data is stored at R worker nodes, each having a different dataset (with no probabilistic assumption on data generation); node $r \in [R]$ has dataset $\mathcal{D}_r$. Let $F_r : \mathbb{R}^d \to \mathbb{R}$ denote the local loss function associated with the dataset $\mathcal{D}_r$, which is defined as $F_r(\boldsymbol{x}) \triangleq \mathbb{E}_{i \in_U [n_r]}[F_{r,i}(\boldsymbol{x})]$, where $n_r = |\mathcal{D}_r|$, i is uniformly distributed over $[n_r] \triangleq \{1, 2, \ldots, n_r\}$, and $F_{r,i}(\boldsymbol{x})$ is the loss associated with the i'th data point at node r with respect to (w.r.t.) the model parameters $\boldsymbol{x} \in \mathbb{R}^d$. Our goal is to solve the following minimization problem:

$$\arg\min_{\boldsymbol{x} \in \mathcal{C}} \Big(F(\boldsymbol{x}) \triangleq \frac{1}{R} \sum_{r=1}^{R} \mathbb{E}_{i \in_U [n_r]}[F_{r,i}(\boldsymbol{x})] \Big). \tag{1}$$

Here, $\mathcal{C} \subset \mathbb{R}^d$ denotes the parameter space and is a compact, convex set.

We can minimize (1) using distributed *vanilla* SGD, where in any iteration, master broadcasts the current model parameters to all workers, each of them then samples a stochastic gradient from its local dataset and sends it back to the server, who aggregates the received gradients and updates the global model parameters. However, this simple solution breaks down even with a single malicious node [9]; see Figure 1.

There have been several works in literature [9]–[24] that provide robustness against Byzantine nodes; see also [7, Section 5] for a detailed survey on Byzantine-robustness in federated learning. Among these, [9]–[16] assume homogeneous (either same or i.i.d.) data across all nodes; [17]–[21] use coding across datasets, which is hard to implement in settings such as federated learning; [22] changes the objective function and adds a regularizer term to combat the adversary; [23] effectively reduces the heterogeneous problem to a homogeneous problem by clustering, and then learning happens within each cluster having homogeneous data; and [24] proposed a

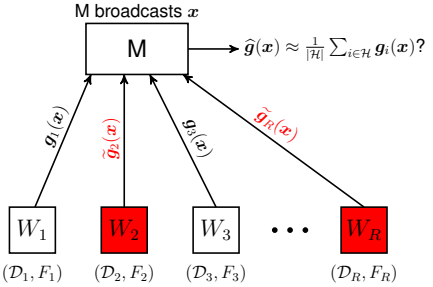

Fig. 1 The training data is distributed across R worker nodes – worker $r \in [R]$ stores dataset $\mathcal{D}_r$ with an associated loss function F_r, and master wants to learn a machine learning model $\boldsymbol{x}^* := \arg\min_{\boldsymbol{x} \in \mathbb{R}^d} \frac{1}{R} \sum_{r=1}^{R} F_r(\boldsymbol{x})$ using SGD in the presence of malicious nodes (denoted in red), who may provide incorrect gradients in each SGD iteration. Filtering out corrupt gradients in heterogeneous data setting and providing convergence analyses for strongly-convex and non-convex objectives is the subject of this paper.

All the proofs and omitted details from this paper can be found in [1].

resampling technique that effectively adapts existing robust algorithms (which might have been designed to work with homogeneous – identical or i.i.d. – datasets) to work with heterogeneous datasets, however, their convergence are only applied to the robust aggregation rule from [9].

The most relevant work to ours are [13], [15], [16], in the sense that they also applied the same robust gradient aggregation rule as ours, which is the robust mean estimation (RME) algorithm from [25], however, there are major differences. [16] study SGD for ERM and assume the *same* data across all nodes. [13], [15] analyze *full-batch* gradient descent for minimizing *population* risk assuming i.i.d. data across nodes. In order to use the decoding algorithm of [25], both [13], [15] 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 do need to prove a matrix concentration bound (which is of a very different nature than theirs, and we use entirely different tools to prove that), the need of which arises because of heterogeneity in datasets and that the gradients are *stochastic* due to SGD – if we work with *full-batch* deterministic gradients, we would not need any of such concentration bounds. See Theorem 2 for our new matrix concentration result. Note that [13] left a few problems open, including analyzing the *stochastic* gradient descent in Byzantine settings. In this paper, we resolve this (while minimizing the empirical risk) in a more general *heterogeneous* data setting, and provide comprehensive analyses of Byzantine SGD and prove its convergence for both strongly-convex and non-convex objectives. See [1] for a detailed discussion on related work.

The reason for applying RME algorithms for gradient aggregation is that its error guarantee has a much better dependence on the dimension d than the more traditional approaches based on median or trimmed-mean; see Section III and [1] for more details. So, in high-dimensional problems, decoding based on RME algorithms performs better.

Our contributions. We provide convergence analyses of our Byzantine-resilient SGD algorithm (see Algorithm 1) for smooth strongly-convex and non-convex objectives under the assumption of bounded variance for stochastic gradients (Assumption 1) and the bounded gradient dissimilarity (Assumption 2), which is a *deterministic* condition on datasets for bounding heterogeneity. We also provide concrete upper bounds on the gradient dissimilarity as well as the local variances in the statistical heterogeneous model under different distributional assumptions (sub-exponential and sub-Gaussian) on local gradients.

Our algorithm can tolerate $\epsilon < \frac{1}{4}$ fraction of corrupt worker nodes. In the strongly-convex case, our algorithm can find optimal parameters within an approximation error of $\mathcal{O}(\kappa^2 + \frac{\sigma^2}{bR} + \frac{\sigma^2 d}{bR}\frac{(\epsilon+\epsilon')}{\epsilon'})$ (where κ^2 is the gradient dissimilarity bound, σ^2 is the variance bound, b is the mini-batch size for stochastic gradients, and $\epsilon' > 0$ is any constant such that $\epsilon+\epsilon' \leq \frac{1}{4}$) "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 $\frac{1}{T}$; see Theorem 1. The $\frac{\sigma^2}{bR}$ term in the approximation error is the standard SGD variance and the $\frac{\sigma^2 d}{bR}\frac{(\epsilon+\epsilon')}{\epsilon'}$ term is due to Byzantine attacks. Note that both these terms can be made small by taking a sufficiently large mini-batch size of stochastic gradients. Note that when workers compute full-batch gradients (i.e., $\sigma = 0$), the approximation error becomes $\mathcal{O}(\kappa^2)$.[1] See Section II-B for a detailed discussion on several aspects of our results.

As mentioned earlier, for filtering corrupt gradients, we employ the robust mean estimation algorithm from [25]. In order to apply that in our *heterogeneous* data setting where workers sample *stochastic* gradients from their local datasets, we derive a new matrix concentration bound (stated in Theorem 2). See Section III for more details.

We also extend these results to the case where workers send *compressed* gradients to the master, and the corresponding results can be found in [1, Section 5].

Paper organization. We describe our algorithm and the main convergence results in Section II. We describe our main technical tool, a new matrix concentration result for heterogeneous data with stochastic gradients in Section III. Omitted details/proofs can be found in [1], which is an extended version of this work.

II. Main Convergence Results

In this section, we state our assumptions, describe the adversary model and our algorithm, and state our main convergence results, together-with some important remarks on the results.

Assumption 1 (Bounded local variances)**.** *The stochastic gradient sampled from any local dataset is uniformly bounded over $\mathcal{C}$ for all workers, i.e., there exists a finite σ, such that for every $\boldsymbol{x} \in \mathcal{C}, r \in [R]$, $\mathbb{E}_{i\in_U[n_r]}\|\nabla F_{r,i}(\boldsymbol{x}) - \nabla F_r(\boldsymbol{x})\|^2 \leq \sigma^2$.*

It will be helpful to formally define mini-batch stochastic gradients, where instead of computing stochastic gradients based on just one data point, each worker samples $b \geq 1$ data points (without replacement) from its local dataset and computes the average of b gradients. For any $\boldsymbol{x} \in \mathbb{R}^d, r \in [R], b \in [n_r]$, consider the following set

$$\mathcal{F}_r^{\otimes b}(\boldsymbol{x}) := \left\{ \frac{1}{b}\sum_{i\in\mathcal{H}_b} \nabla F_{r,i}(\boldsymbol{x}) : \mathcal{H}_b \in \binom{[n_r]}{b} \right\}. \quad (2)$$

Note that $\boldsymbol{g}_r(\boldsymbol{x}) \in_U \mathcal{F}_r^{\otimes b}(\boldsymbol{x})$ is a mini-batch stochastic gradient with batch size b at worker r. It is not hard to see the following hold for every $\boldsymbol{x} \in \mathbb{R}^d, r \in [R]$:[2]

$$\mathbb{E}\left[\boldsymbol{g}_r(\boldsymbol{x})\right] = \nabla F_r(\boldsymbol{x}), \quad (3)$$

[1] It is not surprising that when $\kappa = 0$, we reach to an exact optimum in full-batch GD – when $\kappa = 0$, all workers have the same data, and master can decode the correct gradient by simply taking the majority vote of the received gradients.

[2] Since clients sample data points *without* replacement, we can in fact show a stronger variance bound of $\mathbb{E}\left\|\boldsymbol{g}_r(\boldsymbol{x}) - \nabla F_r(\boldsymbol{x})\right\|^2 \leq \frac{(n_r-b)}{b(n_r-1)}\sigma^2$. However, for simplicity, we only use the weaker bound (4) in this paper.

$$\mathbb{E}\left\|\boldsymbol{g}_r(\boldsymbol{x}) - \nabla F_r(\boldsymbol{x})\right\|^2 \leq \frac{\sigma^2}{b}. \tag{4}$$

Assumption 2 (Bounded gradient dissimilarity)**.** *The variance of the local gradients $\nabla F_r(\boldsymbol{x}), r \in [R]$ from the global gradient $\nabla F(\boldsymbol{x}) = \frac{1}{R}\sum_{r=1}^{R} \nabla F_r(\boldsymbol{x})$ is uniformly bounded over $\mathcal{C}$ for all workers, i.e., there exists a finite κ, such that*

$$\|\nabla F_r(\boldsymbol{x}) - \nabla F(\boldsymbol{x})\|^2 \leq \kappa^2, \quad \forall \boldsymbol{x} \in \mathcal{C}, r \in [R]. \tag{5}$$

Assumption 1 has been standard in SGD literature. Assumption 2 has also been used earlier to bound heterogeneity in datasets; see, for example, [26], [27], which study decentralized SGD with momentum (without Byzantine workers). Note that when workers compute full-batch gradients, we have $\sigma = 0$ in Assumption 1; similarly, when all workers have access to the same dataset as in [9], [12], [16], we have $\kappa = 0$ in Assumption 2. Note that (5) can be seen as a *deterministic* condition on local datasets, under which we derive our results.

A note on Assumption 2. In the presence of Byzantine adversaries, since we do not know which ϵR workers are corrupt, we have to make some structural assumption on the data that can provide relationships among gradients sampled at different nodes for reliable decoding, and Assumption 2 is a natural way to achieve that. There are many alternatives to establish this relationship, e.g., by assuming homogeneous (same or i.i.d.) data across workers [9]–[13], [15], [16] or by explicitly introducing redundancy in the system via coding-theoretic solutions [17], [18], [21]; however, these approaches fall short of in a distributed setup such as federated learning.

Note that assuming bounded gradients of local functions (i.e., $\|\nabla F_r(\boldsymbol{x})\| \leq G$ for some finite G) is a common assumption in literature with heterogeneous data; see, for example, [28], [29] (without adversaries) and [14] (with adversaries). Note that under this assumption, we can trivially bound the heterogeneity among local datasets by $\|\nabla F_r(\boldsymbol{x}) - \nabla F_s(\boldsymbol{x})\| \leq 2G$. So, assuming bounded gradients not only simplifies the analysis but also obscures the effect of heterogeneity on the convergence bounds, which Assumption 2 clearly brings out.

Bounds on σ^2 and κ^2 in the statistical heterogeneous model. Since all results (matrix concentration and convergence) in this paper are given in terms of σ and κ, to show the clear dependence of our results on the dimensionality of the problem, we bound these quantities in the statistical *heterogeneous* data model under different distributional assumptions on local gradients; see [1] for details. For the SGD variance bound, we show that if local gradients have sub-Gaussian distribution, then $\sigma = \mathcal{O}\left(\sqrt{d\log(d)}\right)$. For the gradient dissimilarity bound, we show that if either the local gradients have sub-exponential distribution and each worker has at least $n = \Omega\left(d\log(nd)\right)$ data points or local gradients have sub-Gaussian distribution and $n \in \mathbb{N}$ is arbitrary, then $\kappa \leq \kappa_{\text{mean}} + \mathcal{O}\left(\sqrt{\frac{d\log(nd)}{n}}\right)$, where κ_{mean} denotes the distance of the expected local gradients from the global gradient. Note that we make distributional assumptions on data generation *only* to derive bounds on σ, κ. Other than that, we do not make any distributional assumption on the data and all results in this paper hold for arbitrary datasets satisfying (4), (5).

Algorithm 1 Byzantine-Resilient SGD

Initialize. Set $\boldsymbol{x}^0 := \mathbf{0}$, a fixed learning rate η, and mini-batch size b for stochastic gradients.
for $t = 0$ **to** $T-1$ **do**
  At workers:
  for $r = 1$ **to** R **do**
    Receive $\boldsymbol{x}^t$ from master. Take a mini-batch stochastic gradient $\boldsymbol{g}_r(\boldsymbol{x}^t) \in_U \mathcal{F}_r^{\otimes b}(\boldsymbol{x}^t)$.
    $$\widetilde{\boldsymbol{g}}_r(\boldsymbol{x}^t) = \begin{cases} \boldsymbol{g}_r(\boldsymbol{x}^t) & \text{if worker } r \text{ is honest,} \\ \divideontimes & \text{if worker } r \text{ is corrupt,} \end{cases}$$
    where $\divideontimes$ is an arbitrary vector in $\mathbb{R}^d$.
    Send $\widetilde{\boldsymbol{g}}_r(\boldsymbol{x}^t)$ to master.
  end for
  At master:
  Receive $\{\widetilde{\boldsymbol{g}}_r(\boldsymbol{x}^t)\}_{r=1}^R$ from the R workers.
  Apply the decoding algorithm RGE (described in [1, Appendix E]) on $\{\widetilde{\boldsymbol{g}}_r(\boldsymbol{x}^t)\}_{r=1}^R$.
  Let $\widehat{\boldsymbol{g}}(\boldsymbol{x}^t) = \text{RGE}(\widetilde{\boldsymbol{g}}_1(\boldsymbol{x}^t), \ldots, \widetilde{\boldsymbol{g}}_R(\boldsymbol{x}^t))$.
  Update the parameter vector:
  $$\boldsymbol{x}^{t+1} = \Pi_{\mathcal{C}}\left(\boldsymbol{x}^t - \eta\widehat{\boldsymbol{g}}(\boldsymbol{x}^t)\right),$$
  where $\Pi_{\mathcal{C}}$ is the projection operator onto the set $\mathcal{C}$.
  Broadcast $\boldsymbol{x}^{t+1}$ to all workers.
end for

Adversary model. We assume that an ϵ fraction of R workers are corrupt; as we see later, we can tolerate $\epsilon < \frac{1}{4}$. The corrupt workers can collaborate and arbitrarily deviate from their pre-specified programs: In any SGD iteration, instead of sending the true gradients, corrupt workers can send adversarially chosen vectors (they may not even send anything if they wish, in which case, the master can treat them as *erasures* and replace them with a fixed value). Note that, in the erasure case, master knows which workers are corrupt; whereas, in the Byzantine problem, master does not have this information.

A. Our Algorithm and the Convergence Results

We present our Byzantine-resilient SGD algorithm in Algorithm 1. Our convergence results are for both strongly-convex and non-convex smooth functions.

Theorem 1 (Strongly-convex and Non-convex)**.** *Suppose an $\epsilon > 0$ fraction of R workers are adversarially corrupt. For an L-smooth[3] global objective function $F : \mathcal{C} \to \mathbb{R}$, let Algorithm 1 generate a sequence of iterates $\{\boldsymbol{x}^t\}_{t=0}^T$ when running with a fixed learning rate η, where in the t'th iteration, every honest worker $r \in [R]$ samples a mini-batch stochastic gradient from $\mathcal{F}_r^{\otimes b}(\boldsymbol{x}^t)$, satisfying* (3) *and* (4) *(corrupt workers may send arbitrary vectors). Fix any $\epsilon' > 0$. If $\epsilon \leq \frac{1}{4} - \epsilon'$,*

[3] A function $F : \mathcal{C} \to \mathbb{R}$ is called L-smooth over $\mathcal{C} \subseteq \mathbb{R}^d$, if for every $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$, we have $\|\nabla F(\boldsymbol{x}) - \nabla F(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$ (this property is also known as L-Lipschitz gradients). This is also equivalent to $F(\boldsymbol{y}) \leq F(\boldsymbol{x}) + \langle \nabla F(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2$.

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*[4] *and* $\eta = \frac{\mu}{L^2}$*, then we have*

$$\mathbb{E}\|\boldsymbol{x}^T - \boldsymbol{x}^*\|^2 \leq \left(1 - \frac{\mu^2}{2L^2}\right)^T \|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2 + \frac{2L^2}{\mu^4}\Gamma.$$

  If we take $T = \log\left(\frac{\mu^4}{L^2\Gamma}\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2\right)/\log(\frac{1}{1-\mu^2/2L^2})$*, we get* $\mathbb{E}\|\boldsymbol{x}^T - \boldsymbol{x}^*\|^2 \leq \frac{3L^2}{\mu^4}\Gamma$*.*

- **Non-convex:** *If* $\eta = \frac{1}{4L}$*, then we have*

$$\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\|\nabla F(\boldsymbol{x}^t)\|^2 \leq \frac{8L^2}{T}\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2 + \Gamma,$$

  If we take $T = \frac{8L^2\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2}{\Gamma}$*, we get* $\frac{1}{T}\sum_{t=0}^{T}\mathbb{E}\|\nabla F(\boldsymbol{x}^t)\|^2 \leq 2\Gamma$*.*

In both the bounds, expectation is taken over the sampling of mini-batch stochastic gradients. Here, $\Gamma = \frac{9\sigma^2}{(1-(\epsilon+\epsilon'))bR} + 9\kappa^2 + 9\Upsilon^2$ *with* $\Upsilon = \mathcal{O}\left(\sigma_0\sqrt{\epsilon+\epsilon'}\right)$*, where* $\sigma_0^2 = \frac{24\sigma^2}{b\epsilon'}\left(1 + \frac{d}{(1-(\epsilon+\epsilon'))R}\right) + 16\kappa^2$*.*

Due to lack of space, Theorem 1 is proved in [1].

Projection. Since the parameter space $\mathcal{C}$ is not equal to $\mathbb{R}^d$, our convergence analysis for non-convex objectives requires a mild technical assumption on the size of $\mathcal{C}$. This assumption is only required to ensure that the iterates $\boldsymbol{x}^t$ always stay inside $\mathcal{C}$ without projection. Similar assumption has also been made in [11] for the same purpose. This assumption streamlines our convergence analysis, as our focus in this paper is on Byzantine-resilience.

Assumption 3 (Size of $\mathcal{C}$)**.** *Suppose* $\|\nabla F(\boldsymbol{x})\| \leq M$ *for all* $\boldsymbol{x} \in \mathcal{C}$*. We assume that* $\mathcal{C}$ *contains the* ℓ_2 *ball* $\{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x} - \boldsymbol{x}^0\| \leq \frac{2L}{\Gamma}(M + \Gamma_1)\|\boldsymbol{x}^0 - \boldsymbol{x}^*\|^2\}$*, where* $\Gamma = \frac{9\sigma^2}{(1-(\epsilon+\epsilon'))bR} + 9\kappa^2 + 9\Upsilon^2$ *and* $\Gamma_1 = \frac{n_{\max}\sigma}{b} + \kappa + \Upsilon$*, where* $n_{\max} = \max_{r\in[R]} n_r$ *and other parameters are as defined in Theorem 1 above.*

Note the dependence of the size of $\mathcal{C}$ on $\frac{n_{\max}}{b}$, which is the maximum number of data samples at any worker. This happens because we want a *deterministic* bound on the size of $\mathcal{C}$ (not in expectation) even though we are doing *stochastic* sampling of data points for gradient computation.

B. Important Remarks about Theorem 1

Analysis of the approximation error. In both parts of Theorem 1, the approximation error Γ consists of three error terms: First is $\Gamma_1 = \mathcal{O}(\sigma^2/(1-(\epsilon+\epsilon'))bR)$, which is the standard error arising due to the sampling of stochastic gradients; second is $\Gamma_2 = \mathcal{O}(\kappa^2)$, which is due to dissimilarity in the local datasets; and third is $\Gamma_3 = \mathcal{O}\left(\left(\frac{\sigma^2}{b\epsilon'}\left(1 + \frac{d}{(1-(\epsilon+\epsilon'))R}\right) + \kappa^2\right)(\epsilon+\epsilon')\right)$, which is due to Byzantine attacks. Observe that Γ_1 decreases with the mini-batch size b and the number of workers R, as desired. Note that Γ_3 consists of two terms $\Gamma_{3,1} = \frac{\sigma^2}{b\epsilon'}\left(1 + \frac{d}{(1-(\epsilon+\epsilon'))R}\right)(\epsilon+\epsilon')$ and $\Gamma_{3,2} = \kappa^2(\epsilon+\epsilon')$, where we can make $\Gamma_{3,1}$ small by taking a large mini-batch size b. Note that the presence of $\Gamma_{3,2}$ is inevitable, since κ captures the dissimilarity in different datasets, and that will always show up when bounding the deviation of the true "global" gradient from the decoded one in the presence of Byzantine workers.

Hence, by taking a sufficiently large mini-batch size (or full-batch gradients, which gives $\sigma = 0$), we can reduce the error term to $\mathcal{O}(\kappa^2)$, which, in the statistical heterogeneous model is equal to $\mathcal{O}\left(\kappa_{\text{mean}}^2 + \frac{d\log(nd)}{n}\right)$, where κ_{mean} captures the difference between local and global population means and n is the number of data samples at each worker. In particular, if each worker has $n = \Omega(d\log(nd))$ data points, and they take a sufficiently large mini-batch size in each iteration of Algorithm 1, the approximation error reduces to $\mathcal{O}(\kappa_{\text{mean}}^2)$.

Convergence rates. Note that, in the strongly-convex case, Algorithm 1 approximately finds optimal parameters $\boldsymbol{x}^*$ (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 less than $\frac{1}{4}$ fraction Byzantine workers, which is away from the information-theoretically optimal $\frac{1}{2}$ fraction. The $\frac{1}{4}$ bound comes from the subroutine of robust mean estimation (RME) that we use for robust gradient estimation (RGE), as explained in Section III. 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 settings such as federated learning, where number of workers R could be very large (in tens of thousands or millions), we can get a very small probability of error even if run our algorithm for a long time. Note that the probability of error is due to *stochastic* sampling of gradients, and if we want a "zero" probability of error, we can run full-batch gradient descent; we provide the corresponding results in [1].

III. Robust Gradient Estimation (RGE)

In this section, we first briefly describe the main ingredient of Algorithm 1, a method for robust gradient estimation (RGE), and then prove our new matrix concentration inequality. The problem is as follows: We are given R gradient vectors $\widetilde{\boldsymbol{g}}_1(\boldsymbol{x}), \ldots, \widetilde{\boldsymbol{g}}_R(\boldsymbol{x}) \in \mathbb{R}^d$ for an arbitrary $\boldsymbol{x} \in \mathbb{R}^d$, where, $\widetilde{\boldsymbol{g}}_r(\boldsymbol{x}) = \boldsymbol{g}_r(\boldsymbol{x})$ is a uniform sample from $\mathcal{F}_r^{\otimes b}(\boldsymbol{x})$ if the r'th worker is honest, otherwise, $\widetilde{\boldsymbol{g}}_r(\boldsymbol{x})$ can be arbitrary. We want

[4] A function $F : \mathcal{C} \to \mathbb{R}$ is called μ-strongly convex over $\mathcal{C} \subseteq \mathbb{R}^d$, if for every $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$, we have $F(\boldsymbol{y}) \geq F(\boldsymbol{x}) + \langle\nabla F(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2$.

to compute $\widehat{\boldsymbol{g}}(\boldsymbol{x})$, an estimate of $\boldsymbol{g}_{\mathcal{H}}(\boldsymbol{x}) := \frac{1}{|\mathcal{H}|}\sum_{i\in\mathcal{H}}\boldsymbol{g}_i(\boldsymbol{x})$, which is the average of uncorrupted gradients of honest workers, such that $\|\widehat{\boldsymbol{g}}(\boldsymbol{x}) - \boldsymbol{g}_{\mathcal{H}}(\boldsymbol{x})\|$ is small for all $\boldsymbol{x}\in\mathbb{R}^d$.

For RGE, we employ the polynomial-time outlier-filtering procedure for high-dimensional robust mean estimation (RME) from [25]; see also [30], [31]. In the RME problem, the good samples are from the *same* distribution and we want to estimate its mean, which is different from our problem where gradients come from different distributions due to heterogeneity in datasets. For RME, the crucial observation in these methods is that if the empirical mean of the samples is far from their true mean, then the empirical covariance matrix has high largest eigenvalue. So, the idea is to filter out the samples that have large projection on the principal eigenvector of the empirical covariance matrix. This is done via a soft-removal method, where we assign weights (confidence score) to the samples and down-weighting those that have large projection, and remove the samples when their score go below a threshold. In the end, take an average of the surviving samples. The advantage of this aggregation rule over the traditional ones (that are based on median and trimmed-mean) is that the approximation error of the above solution has a much better dependence on the dimension d of the parameter space. Since we apply this subroutine in each SGD iteration, this error eventually translates to the sub-optimality gap in our optimization solution.

Note that the error guarantee of the above procedure is given in terms of the concentration of the good samples around their sample mean. When applied to our setting, where gradients come from *different* distributions, we need to explicitly prove this concentration which is non-trivial. We believe ours is the first matrix concentration result for non-i.i.d. data (in the federated learning setting). Our main result for robust gradient estimation is as follows:

Theorem 2 (Robust Gradient Estimation)**.** *Fix an arbitrary $\boldsymbol{x}\in\mathbb{R}^d$. Suppose an ϵ fraction of workers are corrupt and we are given R gradients $\widetilde{\boldsymbol{g}}_1(\boldsymbol{x}),\ldots,\widetilde{\boldsymbol{g}}_R(\boldsymbol{x})\in\mathbb{R}^d$, where $\widetilde{\boldsymbol{g}}_r(\boldsymbol{x}) = \boldsymbol{g}_r(\boldsymbol{x})$ is a uniform sample from $\mathcal{F}_r^{\otimes b}(\boldsymbol{x})$ satisfying* (3)*,* (4) *if the r'th worker is honest, otherwise can be arbitrary. Let $\widetilde{\boldsymbol{g}}_i := \widetilde{\boldsymbol{g}}_i(\boldsymbol{x})$ for $i\in[R]$. Then, for any constant $\epsilon'>0$, we have the following (where $\boldsymbol{g}_{\mathcal{S}} := \frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\boldsymbol{g}_i$):*

1) **Matrix concentration:** *With probability at least $1-\exp(-\frac{\epsilon'^2(1-\epsilon)R}{16})$, there exists a subset $\mathcal{S}\subset[R]$ of uncorrupted gradients of size $(1-(\epsilon+\epsilon'))R$ such that*

$$\lambda_{\max}\left(\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\left(\boldsymbol{g}_i-\boldsymbol{g}_{\mathcal{S}}\right)\left(\boldsymbol{g}_i-\boldsymbol{g}_{\mathcal{S}}\right)^T\right)\leq \frac{24\sigma^2}{b\epsilon'}\left(1+\frac{d}{(1-(\epsilon+\epsilon'))R}\right)+16\kappa^2, \quad (6)$$

where $\lambda_{\max}$ denotes the largest eigenvalue.

2) **Outlier-filtering algorithm:** *If $\epsilon\leq\frac{1}{4}-\epsilon'$, then we can find an estimate $\widehat{\boldsymbol{g}}$ of $\boldsymbol{g}_{\mathcal{S}}$ in polynomial-time with probability 1, such that $\|\widehat{\boldsymbol{g}}-\boldsymbol{g}_{\mathcal{S}}\|\leq\mathcal{O}\left(\sigma_0\sqrt{\epsilon+\epsilon'}\right)$, where $\sigma_0^2 = \frac{24\sigma^2}{b\epsilon'}\left(1+\frac{d}{(1-(\epsilon+\epsilon'))R}\right)+16\kappa^2$.*

The statement of Theorem 2 consists of two parts: First, it shows an existence of a large subset $\mathcal{S}$ of uncorrupted gradients having bounded concentration around their sample mean, which is a matrix concentration result; and second, it efficiently estimates the average of the gradients in $\mathcal{S}$. We provide a proof-sketch for the first part in this section, and for the second part, we use the polynomial-time outlier-filtering procedure from [25], which is described in detail in [1], where we also provide an intuition behind the decoding and its running time analysis; the decoding requires SVD computations of $d\times R$ matrices and hence takes polynomial time.

Now we prove the first part of Theorem 2. In order to show (6), first we prove a separate matrix concentration bound in the following lemma, and then we show how we can use that to prove our desired bound (6).

Lemma 1. *Suppose there are m independent distributions $p_1,p_2,\ldots,p_m$ in $\mathbb{R}^d$ such that $\mathbb{E}_{\boldsymbol{y}\sim p_i}[\boldsymbol{y}]=\boldsymbol{\mu}_i, i\in[m]$ and each p_i has bounded variance in all directions, i.e., $\mathbb{E}_{\boldsymbol{y}\sim p_i}[\langle\boldsymbol{y}-\boldsymbol{\mu}_i,\boldsymbol{v}\rangle^2]\leq\sigma^2$ holds for all unit vectors $\boldsymbol{v}\in\mathbb{R}^d$. Take an arbitrary $\epsilon'\in(0,1]$. Then, given m independent samples $\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_m$, where $\boldsymbol{y}_i\sim p_i$, with probability $1-\exp(-\epsilon'^2m/16)$, there is a subset $\mathcal{S}$ of $(1-\epsilon')m$ points such that $\lambda_{\max}\left(\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\left(\boldsymbol{y}_i-\boldsymbol{\mu}_i\right)\left(\boldsymbol{y}_i-\boldsymbol{\mu}_i\right)^T\right)\leq\frac{4\sigma^2}{\epsilon'}\left(1+\frac{d}{(1-\epsilon')m}\right)$.*

Lemma 1 is a generalization of [32, Proposition B.1], where the m samples $\boldsymbol{y}_1,\ldots,\boldsymbol{y}_m$ are drawn independently from a *single* distribution p with mean $\boldsymbol{\mu}$ and variance bound of σ^2. Note that, in our setting, different $\boldsymbol{y}_i$'s may come from *different* distributions, which may have different means.

Note that we are given R gradients, out of which at least $(1-\epsilon)R$ are according to the correct distribution. Consider only the uncorrupted gradients (i.e., $m=(1-\epsilon)R$) and take p_i to be the uniform distribution over $\mathcal{F}_i^{\otimes b}(\boldsymbol{x})$, which implies, using (3) and (4), that the hypothesis of Lemma 1 is satisfied with $\boldsymbol{y}_i=\boldsymbol{g}_i(\boldsymbol{x}),\boldsymbol{\mu}_i=\nabla F_i(\boldsymbol{x}),\sigma^2=\frac{\sigma^2}{b}$. Now we have from Lemma 1 that there exists a subset $\mathcal{S}$ of R gradients of size $(1-\epsilon')(1-\epsilon)R\geq(1-(\epsilon+\epsilon'))R$ that satisfies $\lambda_{\max}\left(\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\left(\boldsymbol{g}_i(\boldsymbol{x})-\nabla F_i(\boldsymbol{x})\right)\left(\boldsymbol{g}_i(\boldsymbol{x})-\nabla F_i(\boldsymbol{x})\right)^T\right)\leq\frac{4\sigma^2}{b\epsilon'}\left(1+\frac{d}{(1-(\epsilon+\epsilon'))R}\right)$. Note that this bounds the deviation of the points in $\mathcal{S}$ from their respective means $\nabla F_i(\boldsymbol{x})$; however, in (6), we need to bound the deviation of the points in $\mathcal{S}$ from their sample mean $\boldsymbol{g}_{\mathcal{S}}(\boldsymbol{x})=\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\boldsymbol{g}_i(\boldsymbol{x})$. Using the gradient dissimilarity bound (5) together with some algebraic manipulations provided in [1], we show that $\lambda_{\max}\left(\frac{1}{|\mathcal{S}|}\sum_{i\in\mathcal{S}}\left(\boldsymbol{g}_i(\boldsymbol{x})-\boldsymbol{g}_{\mathcal{S}}(\boldsymbol{x})\right)\left(\boldsymbol{g}_i(\boldsymbol{x})-\boldsymbol{g}_{\mathcal{S}}(\boldsymbol{x})\right)^T\right)\leq\frac{24\sigma^2}{b\epsilon'}\left(1+\frac{d}{(1-(\epsilon+\epsilon'))R}\right)+16\kappa^2$, which completes the proof of the first part of Theorem 2.

ACKNOWLEDGMENTS

This work was supported in part by NSF grants # 1740047, #2007714, and UC-NL grant LFR-18-548554.

REFERENCES

[1] D. Data and S. N. Diggavi, "Byzantine-resilient SGD in high dimensions on heterogeneous data," *CoRR*, vol. abs/2005.07866, 2020. [Online]. Available: https://arxiv.org/abs/2005.07866

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

[3] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *COMPSTAT*, 2010, pp. 177–186.

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

[5] J. Konecný, "Stochastic, distributed and federated optimization for machine learning," *CoRR*, vol. abs/1707.01155, 2017.

[6] 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.

[7] P. Kairouz *et al.*, "Advances and open problems in federated learning," *CoRR*, vol. abs/1912.04977, 2019.

[8] 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.

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

[10] 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.

[11] D. Yin, Y. Chen, K. Ramchandran, and P. L. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *ICML*, 2018, pp. 5636–5645.

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

[13] 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.

[14] C. Xie, S. Koyejo, and I. Gupta, "Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance," in *International Conference on Machine Learning (ICML)*, 2019, pp. 6893–6901.

[15] D. Yin, Y. Chen, K. Ramchandran, and P. L. Bartlett, "Defending against saddle point attack in byzantine-robust distributed learning," in *ICML*, 2019, pp. 7074–7084.

[16] D. Data and S. N. Diggavi, "On byzantine-resilient high-dimensional stochastic gradient descent," in *ISIT*, 2020, pp. 2628–2633.

[17] L. Chen, H. Wang, Z. B. Charles, and D. S. Papailiopoulos, "DRACO: byzantine-resilient distributed training via redundant gradients," in *ICML*, 2018, pp. 902–911.

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

[19] D. Data, L. Song, and S. N. Diggavi, "Data encoding methods for byzantine-resilient distributed optimization," in *ISIT*, 2019, pp. 2719–2723.

[20] D. Data and S. N. Diggavi, "Byzantine-tolerant distributed coordinate descent," in *ISIT*, 2019, pp. 2724–2728.

[21] D. Data, L. Song, and S. N. Diggavi, "Data encoding for byzantine-resilient distributed optimization," *IEEE Transactions on Information Theory*, vol. 67, no. 2, pp. 1117–1140, 2021, arXiv: https://arxiv.org/abs/1907.02664.

[22] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," in *Conference on Artificial Intelligence (AAAI)*, 2019, pp. 1544–1551.

[23] A. Ghosh, J. Hong, D. Yin, and K. Ramchandran, "Robust federated learning in a heterogeneous environment," *CoRR*, vol. abs/1906.06629, 2019. [Online]. Available: http://arxiv.org/abs/1906.06629

[24] L. He, S. P. Karimireddy, and M. Jaggi, "Byzantine-robust learning on heterogeneous datasets via resampling," *CoRR*, vol. abs/2006.09365, 2020. [Online]. Available: https://arxiv.org/abs/2006.09365

[25] J. Steinhardt, M. Charikar, and G. Valiant, "Resilience: A criterion for learning in the presence of arbitrary outliers," in *ITCS*, 2018, pp. 45:1–45:21.

[26] H. Yu, R. Jin, and S. Yang, "On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization," in *ICML*, 2019, pp. 7184–7193.

[27] X. Li, W. Yang, S. Wang, and Z. Zhang, "Communication efficient decentralized training with multiple local updates," *CoRR*, vol. abs/1910.09126, 2019.

[28] H. Yu, S. Yang, and S. Zhu, "Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning," in *Conference on Artificial Intelligence (AAAI)*, 2019, pp. 5693–5700.

[29] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *International Conference on Learning Representations (ICLR)*, 2020. [Online]. Available: https://openreview.net/forum?id=HJxNAnVtDS

[30] I. Diakonikolas, G. Kamath, D. Kane, J. Li, A. Moitra, and A. Stewart, "Robust estimators in high-dimensions without the computational intractability," *SIAM J. Comput.*, vol. 48, no. 2, pp. 742–864, 2019.

[31] K. A. Lai, A. B. Rao, and S. S. Vempala, "Agnostic estimation of mean and covariance," in *FOCS*, 2016, pp. 665–674.

[32] M. Charikar, J. Steinhardt, and G. Valiant, "Learning from untrusted data," in *STOC*, 2017, pp. 47–60.