Distributed Statistical Machine Learning in Adversarial Settings: Byzantine Gradient Descent

Yudong Chen
Cornell
yudong.chen@cornell.edu

Lili Su
UIUC
lilisu3@illinois.edu

Jiaming Xu
Purdue
xu972@purdue.edu

August 3, 2021

Abstract

We consider the distributed statistical learning problem over decentralized systems that are prone to adversarial attacks. This setup arises in many practical applications, including Google's *Federated Learning*. Formally, we focus on a decentralized system that consists of a parameter server and m working machines; each working machine keeps N/m data samples, where N is the total number of samples. In each iteration, up to q of the m working machines suffer Byzantine faults – a faulty machine in the given iteration behaves arbitrarily badly against the system and has complete knowledge of the system. Additionally, the sets of faulty machines may be different across iterations. Our goal is to design robust algorithms such that the system can learn the underlying true parameter, which is of dimension d, despite the interruption of the Byzantine attacks.

In this paper, based on the *geometric median of means* of the gradients, we propose a simple variant of the classical gradient descent method. We show that our method can tolerate q Byzantine failures up to $2(1+\epsilon)q \leq m$ for an arbitrarily small but fixed constant $\epsilon > 0$. The parameter estimate converges in $O(\log N)$ rounds with an estimation error on the order of $\max\{\sqrt{dq/N},\ \sqrt{d/N}\}$, which is larger than the minimax-optimal error rate $\sqrt{d/N}$ in the centralized and failure-free setting by at most a factor of $\sqrt{q}$. The total computational complexity of our algorithm is of $O((Nd/m)\log N)$ at each working machine and $O(md + kd\log^3 N)$ at the central server, and the total communication cost is of $O(md\log N)$. We further provide an application of our general results to the linear regression problem.

A key challenge arises in the above problem is that Byzantine failures create arbitrary and unspecified dependency among the iterations and the aggregated gradients. To handle this issue in the analysis, we prove that the aggregated gradient, as a function of model parameter, converges *uniformly* to the true gradient function.

1 Introduction

Distributed machine learning has emerged as an attractive solution to large-scale problems and received intensive attention [BPC+11, JLY16, MNSJ15, PH96, DG08, LBG+12]. In this setting, the data samples or/and computation are distributed across multiple machines, which are programmed to collaboratively learn a model. Many efficient distributed machine learning algorithms [BPC+11, JLY16] and system implementations [MNSJ15, PH96, DG08, LBG+12] have been proposed and studied. Prior work mostly focuses on the traditional "training within cloud" framework where the model training process is carried out within the cloud infrastructures. In this framework, distributed machine learning is secured via system architectures, hardware devices, and monitoring [KPS02, PP02, WWRL10]. This framework faces significant privacy risk, as the data has to be collected

from owners and stored within the clouds. Although a variety of privacy-preserving solutions have been developed [AS00, DWJ13], privacy breaches occur frequently, with recent examples including iCloud leaks of celebrity photos and PRISM surveillance program.

To address privacy concerns[1] , a new machine learning paradigm called *Federated Learning* was proposed by Google researchers [KMR15, MR10]. It aims at learning an accurate model without collecting data from owners and storing the data in the cloud. The training data is kept locally on the owners' computing devices, which are recruited to participate directly in the model training process and hence function as working machines. Google has been intensively testing this new paradigm in their recent projects such as *Gboard* [MR10], the Google Keyboard. Compared to "training within cloud", Federated Learning faces the following three key challenges:

- Security: The devices of the recruited data owners can be easily reprogrammed and completely controlled by external attackers, and thus behave adversarially.
- Small local datasets versus high model complexity: While the total number of data samples over all data owners may be large, each individual owner may keep only a small amount of data, which by itself is insufficient for learning a complex model.
- Communication constraints: Data transmission between the recruited devices and the cloud may suffer from high latency and low-throughout. Communication between them is therefore a scarce resource.

In this paper, we address the above challenges by developing a simple variant of the gradient descent method that can (1) tolerate the arbitrary and adversarial failures, (2) accurately learn a highly complex model with low local data volume, and (3) converge exponentially fast using logarithmic communication rounds. Since gradient descent algorithms are well-adopted in existing implementations and applications, our proposed method only requires a small amount of modification of existing codes.

Note that there are many other challenges besides what are listed here, including unevenly distributed training data, intermittent availability of mobile phones, etc. These challenges will be addressed in future work.

1.1 Learning Goals

To formally study the distributed machine learning problem in adversarial settings, we consider a standard statistical learning setup, where the data is generated probabilistically from an unknown distribution and the true model is parameterized by a vector. More specifically, let $X \in \mathcal{X}$ be the input data generated according to some *unknown* distribution μ. Let $\Theta \subset \mathbb{R}^d$ be the set of all choices of model parameters. We consider a loss function $f : \mathcal{X} \times \Theta \to \mathbb{R}$, where $f(x, \theta)$ measures the risk induced by a realization x of the data under the model parameter choice θ. A classical example is linear regression, where $x = (w, y) \in \mathbb{R}^d \times \mathbb{R}$ is the feature-response pair and $f(x, \theta) = \frac{1}{2}(\langle w, \theta \rangle - y)^2$ is the usual squared loss.

We are interested in learning the model choice θ^* that minimizes the *population risk*, i.e.,

$$\theta^* \in \arg\min_{\theta \in \Theta} F(\theta) \triangleq \mathbb{E}\left[f(X, \theta)\right], \tag{1}$$

[1]We would like to characterize the amount of privacy sacrificed in the Federated Learning paradigm. We leave this characterization as one of our future work.

assuming that $\mathbb{E}[f(X,\theta)]$ is well defined over Θ.[2] The model choice θ^* is optimal in the sense that it minimizes the average risk to pay if the model chosen is used for prediction in the future with a fresh random data sample.

When μ—the distribution of X—is known, which is rarely the case in practice, the population risk can be evaluated exactly, and θ^* can be computed by solving the minimization problem in (1). We focus on the more realistic scenario where μ is *unknown* but there exist N independently and identically distributed data samples $X_i \overset{\text{i.i.d.}}{\sim} \mu$ for $i = 1, \ldots, N$. Note that estimating θ^* using finitely many data samples will always have a *statistical error* due to the randomness in the data, even in the centralized, failure-free setting. Our results account for this effect.

1.2 System Model

We focus on solving the above statistical learning problem over decentralized systems that are prone to adversarial attacks. Specifically, the system of interest consists of a parameter server[3] and m working machines. In the example of Federated Learning, the parameter server represents the cloud, and the m working machines correspond to m data owners' computing devices.

We assume that the N data samples are distributed evenly across the m working machines. In particular, each working machine i keeps a subset $\mathcal{S}_i$ of the data, where $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ and $|\mathcal{S}_i| = N/m$. Note that this is a simplifying assumption of the data imbalance in Federated Learning. Nevertheless, our results can be extended to the heterogeneous data sizes setting when the data sizes are of the same order. We further assume that the parameter server can communicate with all working machines in synchronous communication rounds, and leave the asynchronous setting as future directions.

Among the m working machines, we assume that up to q of them can suffer Byzantine failures and thus behave maliciously; for example, they may be reprogrammed and completely controlled by the system attacker. We assume the parameter server knows q – as q can be estimated from the existing system failures statistics. The set of Byzantine machines can *change* between communication rounds; the system attacker can choose different sets of machines to control across communication rounds. Byzantine faulty machines are assumed to have *complete knowledge* of the system, including the total number of working machines m, all N data samples over the whole system, the programs that the working machines are supposed to run, the program run by the parameter server, and the realization of the random bits generated by the parameter server. Moreover, Byzantine machines can collude [Lyn96]. The only constraint is that these machines cannot corrupt the local data on working machines — but they can lie when communicating with the server. In fact, our main results show that our proposed algorithm still works when at most q different machines with local data corrupted during its execution.

We remark that Byzantine failures are used to capture the unpredictability of extremely large system that consists of heterogeneous processes, as is the case with Federated Learning. The arbitrary behavior of Byzantine machines creates unspecified dependency across communication rounds — a key challenge in our algorithm design and convergence analysis. In this paper, we use *rounds* and *iterations* interchangeably.

[2] For example, if $\mathbb{E}[|f(X,\theta)|]$ is finite for every $\theta \in \Theta$, the population risk $\mathbb{E}[f(X,\theta)]$ is well defined.

[3] Note that, due to communication bandwidth constraints, practical systems use multiple networked parameter servers. In this paper, for ease of explanation, we assume there is only one parameter server. Fortunately, as can be seen from our algorithm descriptions and our detailed correctness analysis, the proposed algorithm also works for the multi-server setting.

1.3 Existing Distributed Machine Learning Algorithms

There are three popular classes of existing distributed machine learning algorithms in terms of their communication rounds.

SGD: On one end of the spectrum lies the *Stochastic Gradient Descent (SGD)* algorithm. Using this algorithm, the parameter server receives, in each iteration, a gradient computed at a single data sample from one working machine, and uses it to perform one gradient descent step. Even when the population risk F is strongly convex, the convergence rate of SGD is only $O(1/t)$ with t iterations. This is much slower than the exponential (geometric) convergence of standard gradient descent. Therefore, SGD requires a large number of communication rounds, which could be costly. Indeed, it has been demonstrated in [MR10] that SGD has 10-100 times higher communication cost than standard gradient descent, and is therefore inadequate for scenarios with scarce communication bandwidth.

One-Shot Aggregation: On the other end of the spectrum, using a *One-Shot Aggregation* method, each working machine computes an estimate of the model parameter using only its local data and reports it to the server, which then aggregates all the estimates reported to obtain a final estimate [ZDW13, ZDW15]. One-shot aggregation method only needs a single round of communication from the working machines to the parameter server, and thus is communication-efficient. However, it requires $N/m \gg d$ so that a coarse parameter estimate can be obtained at each machine. This algorithm is therefore not applicable in scenarios where local data is small in size but the model to learn is of high dimension.

BGD: *Batch Gradient Descent (BGD)* lies in between the above two extremes. At each iteration, the parameter server sends the current model parameter estimate to all working machines. Each working machine computes the gradient based on all locally available data, and then sends the gradient back to the parameter server. The parameter server averages the received gradients and performs a gradient descent step. When the population risk F is strongly convex, BGD converges exponentially fast, and hence requires only a few rounds of communication. BGD also works in the scenarios with limited local data, i.e., $N/m = O(d)$, making it an ideal candidate in Federated Learning. However, it is sensitive to Byzantine failures; a single Byzantine failure at a working machine can completely skew the average value of the gradients received by the parameter server, and thus foils the algorithm.

1.4 Contributions

In this paper, we propose a Byzantine gradient descent method. Specifically, the parameter server aggregates the local gradients reported by the working machines in three steps: (1) it partitions all the received local gradients into k batches and computes the mean for each batch, (2) it computes the *geometric median* of the k batch means, and (3) it performs a gradient descent step using the geometric median.

We prove that the proposed algorithm can tolerate q Byzantine failures up to $2(1+\epsilon)q \leq m$ for an arbitrarily small but fixed constant $\epsilon > 0$. Moreover, the error in estimating the target model parameter θ^* converges in $\log(N)$ communication rounds to the order of $\max\{\sqrt{dq/N},\ \sqrt{d/N}\}$, whereas the minimax-optimal estimation error rate in the centralized and failure-free setting is $\sqrt{d/N}$.[4] Even in the scarce *local* data regime where $N/m = O(d)$, the estimator of our proposed

[4]Note that $\sqrt{d/N}$ is the minimax optimal estimation error rate even in the centralized, failure-free setting when

algorithm is still consistent as long as $N/q = \omega(d)$. The total computational complexity of our algorithm is of $O((N/m)d \log N)$ at each worker and $O(md + qd \log^3 N)$ at the parameter server, and the total communication cost is of $O(md \log N)$. Note that the $\sqrt{q}$ factor in our estimation error rate $\max\{\sqrt{dq/N}, \sqrt{d/N}\}$ may not be fundamental to the problem of learning in adversarial settings. Thus, it may possibly be improved with better algorithms or finer analysis.

A key challenge in our analysis is that there exists complicated probabilistic dependency among the iterates and the aggregated gradients. Even worse, such dependency cannot be specified due to the arbitrary behavior of the Byzantine machines. We overcome this challenge by proving that the geometric median of means of gradients *uniformly* converges to the true gradient function $\nabla F(\cdot)$.

1.5 Outline

The origination of the paper is as follows. In Section 2, we present our algorithm, named *Byzantine Gradient Descent Method*, and summarize our convergence results. Detailed convergence analysis can be found in Section 3. To illustrate the applicability of our convergence results, we provide a linear regression example in Section 4. Related work is discussed in Section 5. Section 6 concludes the paper, and presents several interesting future directions.

2 Algorithms and Summary of Convergence Results

In this section, we present our distributed statistical machine learning algorithm, named *Byzantine Gradient Descent Method*, and briefly summarize our convergence results on its performance.

2.1 Byzantine Gradient Descent Method

Recall that our fundamental goal is to learn the optimal model choice θ^* defined in (1). We make the following standard assumption [BPC+11] so that the minimization problem in (1) can be solved efficiently (exponentially fast) in the ideal case when the population risk function F is known exactly, i.e., the distribution μ is known.

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

$$F(\theta') \geq F(\theta) + \langle \nabla F(\theta), \theta' - \theta \rangle + \frac{L}{2}\|\theta' - \theta\|^2,$$

and

$$\|\nabla F(\theta) - \nabla F(\theta')\| \leq M\|\theta - \theta'\|.$$

Under Assumption 1, it is well-known [BV04] that using the standard gradient descent update

$$\theta_t = \theta_{t-1} - \eta \times \nabla F(\theta_{t-1}), \tag{2}$$

where η is some fixed stepsize, θ_t approaches θ^* exponentially fast. In particular, choosing $\eta = L/(2M^2)$, it holds that

$$\|\theta_t - \theta^*\| \leq \left(1 - \left(\frac{L}{2M}\right)^2\right)^{t/2} \|\theta_0 - \theta^*\|.$$

we would like to estimate a d-dimensional unknown parameter without any additional structure from N i.i.d. samples, see e.g., [Wu17, Section 3.2] for a proof in the special case of Gaussian mean estimation. When there is additional structure, say sparsity, then the $\sqrt{d}$ factor can possibly be improved.

Nevertheless, when the distribution μ is unknown, as assumed in this paper, the population gradient ∇F can only be approximated using sample gradients, if they exist.

Recall that each working machine j (can possibly be Byzantine) keeps a very small set of data $\mathcal{S}_j$ with $|\mathcal{S}_j| = N/m$. Define the local empirical risk function, denoted by $\bar{f}^{(j)} : \Theta \to \mathbb{R}$, as follows:

$$\bar{f}^{(j)}(\theta) \triangleq \frac{1}{|\mathcal{S}_j|} \sum_{i \in \mathcal{S}_j} f(X_i, \theta), \quad \forall\, \theta \in \Theta. \tag{3}$$

Notice that $\bar{f}^{(j)}(\cdot)$ is a function of data samples $\mathcal{S}_j$ stored at machine j. Hence $\bar{f}^{(j)}(\cdot)$ is random. Although Byzantine machines can send arbitrarily malicious messages to the parameter server, they are unable to corrupt the local stored data. Thus, the local risk function $\bar{f}^{(j)}(\cdot)$ is well-defined for all j, including the Byzantine machines. With a bit of abuse of notation, we let

$$\bar{\boldsymbol{f}}(\theta) \triangleq \left(\bar{f}^{(1)}(\theta), \ldots, \bar{f}^{(m)}(\theta)\right)$$

be the vector that stacks the values of the m local functions evaluated at θ. For any $x \in \mathcal{X}$, we assume that $f(x, .) : \Theta \to \mathbb{R}$ is differentiable. When there is no confusion, we write $\nabla_\theta f(x, \theta)$ – the gradient of function $f(x, \cdot)$ evaluated at θ – simply as $\nabla f(x, \theta)$.

It is well-known that the average of the local gradients can be viewed as an approximation of the population gradient $\nabla F(\cdot)$. In particular, for a fixed θ, as $N \to \infty$

$$\frac{1}{m} \sum_{j=1}^{m} \nabla \bar{f}^{(j)}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \nabla f(X_i, \theta) \xrightarrow{\text{a.s.}} \nabla F(\theta). \tag{4}$$

Batch Gradient Descent relies on this observation. However, this method is sensitive to Byzantine failures as we explain next.

Batch Gradient Descent We describe the *Batch Gradient Descent (BGD)* in Algorithm 1. We initialize θ_0 to be some arbitrary value in Θ for simplicity. In practice, there are standard guides in choosing the initial point [SB98]. In round $t \geq 1$, the parameter server sends the current model parameter estimator θ_{t-1} to all working machines. Each working machine j computes the gradient $\nabla \bar{f}^{(j)}(\theta_{t-1})$ and sends $\nabla \bar{f}^{(j)}(\theta_{t-1})$ back to the parameter server. Note that Byzantine machines may not follow the codes in Algorithm 1. Instead of the true local gradients, Byzantine machines can report arbitrarily malicious messages or no message to the server. If the server does not receive any message from a working machine, then that machine must be Byzantine faulty. In that case, the server sets $g_t^{(j)}(\theta_{t-1})$ to some arbitrary value. Precisely, let $\mathcal{B}_t$ denote the set of Byzantine machines at round t in a given execution. The message received from machine j, denoted by $g_t^{(j)}(\theta_{t-1})$, can be described as

$$g_t^{(j)}(\theta_{t-1}) = \begin{cases} \nabla \bar{f}^{(j)}(\theta_{t-1}) & \text{if } j \notin \mathcal{B}_t \\ \star & \text{o.w. ,} \end{cases} \tag{5}$$

where, with a bit of abuse of notation, $\star$ denotes the arbitrary message whose value may be different across Byzantine machines, iterations, executions, etc. In step 3, the parameter server averages the received $g_t^{(j)}(\theta_{t-1})$ and updates θ_t using a gradient descent step.

Under Assumption 1, when there are no Byzantine machines, it is well-known that BGD converges exponentially fast. However, a single Byzantine failure can completely skew the average value of the gradients received by the parameter server, and thus foils the algorithm. It is still

Algorithm 1 Standard Gradient Descent: Iteration $t \geq 1$

Parameter server:

1: *Initialize:* Let θ_0 be an arbitrary point in Θ.
2: Broadcast the current model parameter estimator θ_{t-1} to all working machines;
3: Wait to receive all the gradients reported by the m machines; Let $g_t^{(j)}(\theta_{t-1})$ denote the value received from machine j.
If no message from machine j is received, set $g_t^{(j)}(\theta_{t-1})$ to be some arbitrary value;
4: Update: $\theta_t \leftarrow \theta_{t-1} - \eta \times \left(\frac{1}{m}\sum_{j=1}^m g_t^{(j)}(\theta_{t-1})\right)$;

Working machine j:

1: Compute the gradient $\nabla \bar{f}^{(j)}(\theta_{t-1})$;
2: Send $\nabla \bar{f}^{(j)}(\theta_{t-1})$ back to the parameter server;

the case even if the parameter server takes an average of a randomly selected subset of received gradients. This is because a Byzantine machine is assumed to have complete knowledge of the system, including the gradients reported by other machines, and the realization of the random bits generated by the parameter server.

Robust Gradient Aggregation Instead of taking the average of the received gradients

$$g_t^{(1)}(\theta_{t-1}), \cdots, g_t^{(m)}(\theta_{t-1}),$$

we propose a robust way to aggregate the collected gradients. Our aggregation rule is based on the notion of *geometric median*.

Geometric median is a generalization of median in one-dimension to multiple dimensions, and has been widely used in robust statistics [MNO+10, MD+87, Kem87, CCZ+13]. Let $\{y_1, \ldots, y_n\} \subseteq \mathbb{R}^d$ be a multi-set of size n. The geometric median of $\{y_1, \ldots, y_n\}$, denoted by $\mathsf{med}\{y_1, \ldots, y_n\}$, is defined as

$$\mathsf{med}\{y_1, \ldots, y_n\} \triangleq \arg\min_{y \in \mathbb{R}^d} \sum_{i=1}^n \|y - y_i\|. \tag{6}$$

Geometric median is NOT required to lie in $\{y_1, \ldots, y_n\}$, and is unique unless all the points in $\{y_1, \ldots, y_n\}$ lie on a line. Note that if the ℓ_2 norm in (6) is replaced by the squared ℓ_2 norm, i.e., $\|\cdot\|^2$, then the minimizer is exactly the average.

In one dimension, median has the following nice robustness property: if strictly more than $\lfloor n/2 \rfloor$ points are in $[-r, r]$ for some $r \in \mathbb{R}$, then the median must be in $[-r, r]$. Likewise, in multiple dimensions, geometric median has similar robust property [M+15, Lemma 2.1] [CLM+16, Lemma 24]. The following lemma shows that a $(1+\gamma)$- approximate geometric median is also robust. Its proof is a simple adaptation of the proof of Lemma 24 in [CLM+16] and presented in Appendix A.

Lemma 1. *Let $z_1, \ldots, z_n$ denote n points in a Hilbert space. Let z_* denote a $(1+\gamma)$-approximation of their geometric median, i.e., $\sum_{i=1}^n \|z_* - z_i\| \leq (1+\gamma) \min_z \sum_{i=1}^n \|z - z_i\|$ for $\gamma > 0$. For any*

$\alpha \in (0, 1/2)$ and given $r \in \mathbb{R}$, if $\sum_{i=1}^n \mathbf{1}_{\{\|z_i\| \le r\}} \ge (1-\alpha)n$, then

$$\|z_*\| \le C_\alpha r + \gamma \frac{\min_z \sum_{i=1}^n \|z - z_i\|}{(1-2\alpha)n} \le C_\alpha r + \gamma \frac{\max_{1 \le i \le n} \|z_i\|}{1-2\alpha},$$

where

$$C_\alpha = \frac{2(1-\alpha)}{1-2\alpha}. \tag{7}$$

The above lemma shows that as long as there are sufficiently many points (majority in terms of fraction) inside the Euclidean ball of radius r centered at origin, then the geometric median ($\gamma = 0$) must lie in the Euclidean ball blowed up by a constant factor only. Intuitively, geometric median can be viewed as an aggregated center of a set based on majority vote. Note that the exact geometric median may not be computed efficiently in practice. The above lemma further shows that $(1+\gamma)$-approximate geometric median also lies in the Euclidean ball blowed up by a constant factor plus a deviation term proportional to γ and $\max_i \|z_i\|$.

Let $\boldsymbol{g}_t(\theta_{t-1}) = \left(g_t^{(1)}(\theta_{t-1}), \ldots, g_t^{(m)}(\theta_{t-1})\right)$ be the vector that stacks the gradients received by the parameter server at iteration t. Let k be an integer which divides m and let $b = m/k$ denote the batch size. In our proposed robust gradient aggregation, the parameter server (1) first divides m working machines into k batches, (2) then takes the average of local gradients in each batch, and (3) finally takes the geometric median of those k batch means. With the aggregated gradient, the parameter server performs a gradient descent update. Notice that when the number of batches

Algorithm 2 Byzantine Gradient Descent: Iteration $t \ge 1$

Parameter server:

1: *Initialize:* Let θ_0 be an arbitrary point in Θ; group the m machines into k batches, with the ℓ-th batch being $\{(\ell-1)b+1, \ldots, \ell b\}$ for $1 \le \ell \le k$.
2: Broadcast the current model parameter estimator θ_{t-1} to all working machines;
3: Wait to receive all the gradients reported by the m machines; If no message from machine j is received, set $\nabla \widetilde{g}_j(\theta_{t-1})$ to be some arbitrary value;
4: *Robust Gradient Aggregation*

$$\mathcal{A}_k(\boldsymbol{g}_t(\theta_{t-1})) \leftarrow \mathsf{med}\left\{\frac{1}{b}\sum_{j=1}^{b} g_t^{(j)}(\theta_{t-1}),\ \cdots,\ \frac{1}{b}\sum_{j=n-b+1}^{n} g_t^{(j)}(\theta_{t-1})\right\}. \tag{8}$$

5: Update: $\theta_t \leftarrow \theta_{t-1} - \eta \times \mathcal{A}_k\big(\boldsymbol{g}_t(\theta_{t-1})\big)$;

Working machine j:

1: Compute the gradient $\nabla \bar{f}^{(j)}(\theta_{t-1})$;
2: Send $\nabla \bar{f}^{(j)}(\theta_{t-1})$ back to the parameter server;

$k = 1$, the geometric median of means reduces to the average, i.e.,

$$\mathcal{A}_1\{\boldsymbol{g}_t(\theta_{t-1})\} = \frac{1}{m}\sum_{j=1}^{m} g_t^{(j)}(\theta_{t-1}).$$

When $k = m$, the median of means reduces to the geometric median

$$\mathcal{A}_m\{\boldsymbol{g}_t(\theta_{t-1})\} = \mathsf{med}\{g_t^{(1)}(\theta_{t-1}), \ldots, g_t^{(m)}(\theta_{t-1})\}.$$

Hence, the geometric median of means can be viewed as an interpolation between the mean and the geometric median. Since the parameter server knows q – the upper bound on the number of Byzantine machines q, it can choose k accordingly. We will discuss the choice of k after the statement of our main theorem.

2.2 Summary of Convergence Results

For ease of presentation, we present an informal statement of our main theorem. The precise statement and its proof are given in Section 3.3. Our convergence results hold under some technical assumptions on the sample gradients $\nabla f(X_i, \cdot)$, formally stated in Section 3.3. Roughly speaking, such assumptions mimic Assumption 1 (placed on population risk F), and can be viewed a stochastic version of strong convexity and Lipschitz-continuity conditions.

Theorem 1 (Informal)**.** *Suppose some mild technical assumptions hold and* $2(1+\epsilon)q \le k \le m$ *for any arbitrary but fixed constant* $\epsilon > 0$*. Fix any fixed constant* $\alpha \in (\frac{1}{2+2\epsilon}, \frac{1}{2})$ *and any* $\delta > 0$ *such that* $\delta \le \alpha - q/k$*. There exist universal constants* $c_1, c_2 > 0$ *such that if* $N/k \ge c_1 C_\alpha^2 \left(d \log(N/k) + \log(1/\delta)\right)$*, then with probability at least*

$$1 - \exp\left(-kD\left((\alpha - q/k) \| \delta\right)\right),$$

the iterates $\{\theta_t\}$ *given by Algorithm 2 with* $\eta = L/(2M^2)$ *satisfy*

$$\|\theta_t - \theta^*\| \le \left(\frac{1}{2} + \frac{1}{2}\sqrt{1 - \frac{L^2}{4M^2}}\right)^t \|\theta_0 - \theta^*\| + c_2 C_\alpha \sqrt{\frac{k\left(d + \log(1/\delta)\right)}{N}}, \tag{9}$$

for $t \ge 1$*, where* $D(\delta' \| \delta) = \delta' \log \frac{\delta'}{\delta} + (1-\delta') \log \frac{1-\delta'}{1-\delta}$ *denotes the binary divergence.*

The characterization of c_1 and c_2 can be found in Section 3.3. In Theorem 1, in addition to the non-specified "technical assumptions", we also impose assumptions on δ, α, N/k and d. Next we illustrate that these conditions can indeed hold simultaneously.

As can be seen later, δ can be viewed as the expected fraction of batches that are "statistically bad"; the larger the batch sample size N/k (comparing to d), the smaller δ. Additionally, up to q/k fraction of the batches may contain Byzantine machines. In total, we may expect $\delta + q/k$ fraction of the batches to be bad. Theorem 1 says that as long as the total fraction of bad batches is less than $1/2$, we are able to show with high probability, our Byzantine Gradient Descent Method converges exponentially fast.

Remark 1. In this remark, we discuss the choice of k.

When $q = 0$, k can be chosen to be 1 and $\log(1/\delta)$ can be chosen to be d. Thus the geometric median of means reduces to simple averaging. Theorem 1 implies that with probability at least $1 - e^{-\Omega(d)}$, the asymptotic estimation error rate is $\sqrt{d/N}$.

For $q \ge 1$, we can choose k to be $2(1+\epsilon)q$ for an arbitrarily small but fixed constant $\epsilon > 0$ and $\alpha = \frac{2+\epsilon}{4+4\epsilon}$ and $\log(1/\delta) = d$. In this way, $\alpha - q/k = \frac{\epsilon}{4+4\epsilon}$. Using the property that $D(\delta' \| \delta) \ge \delta' \log \frac{\delta'}{e\delta}$, we have that $D\left((\alpha - q/k) \| \delta\right) \ge \Omega(d)$. Hence as long as $N/k \ge c_1 d \log(N/k)$ for a sufficiently large universal constant c_1, with probability at least $1 - e^{-\Omega(qd)}$, the estimation error of θ_t converges exponentially fast to $c_2\sqrt{dq/N}$ for a universal constant c_2.

Based on our analysis, the number of batches k in our Byzantine gradient algorithm provides a trade-off between the statistical estimation error and the Byzantine failures: With a larger k, our algorithm can tolerate more Byzantine failures, but the estimation error gets larger. However, this trade-off may be an artifact of our proof and may not be fundamental.

Remark 2. In this remark, we discuss the practical issues of computing geometric median.

Since exact geometric median may not be computed efficiently in practice, we can use $(1+\gamma)$-approximate geometric median in the robust gradient aggregation step (8). Moreover, in view of Lemma 1, comparing to the exact geometric median, a $(1+\gamma)$-approximate geometric median induces an additional deviation term proportional to γ and the maximum norm of the averaged batch gradients. Therefore, in (8), we also trim away averaged batch gradients of norm larger than a threshold τ before computing the $(1+\gamma)$-approximate geometric median. Since the gradients are of dimension d, one can choose the threshold $\tau = \Theta(d)$ so that with high probability the averaged gradients over Byzantine-free batches will not be trimmed away. Finally by choosing $\gamma = 1/N$, the additional deviation term is ensured to be $O(d/N)$ and thus it will not affect the final estimation error which is on the order of $\max\{\sqrt{dq/N},\ \sqrt{d/N}\}$.

Our algorithm is both computation and communication efficient. Under the choice of k in Remark 1, the computation and communication cost of our proposed algorithm can be summarized as follows. For estimation error converging to $c_2\sqrt{dq/N}$, $O(\log N)$ communication rounds are sufficient. In each round, every working machine transmits a d-dimensional vector to the parameter server. In terms of computation cost, in each round, every working machine computes a gradient based on N/m local data samples, which takes $O(Nd/m)$. The parameter server computes the geometric median of means of gradients. The means of gradients over all batches can be computed in $O(md)$ steps. It is shown in [CLM+16] that a $(1+\gamma)$-approximate geometric median can be computed in $O(qd\log^3(1/\gamma))$ and as we discussed in Remark 2, $\gamma = 1/N$ suffices for our purpose. Therefore, in each round, in total the parameter server needs to take $O(md + qd\log^3(N))$ steps.

3 Convergence Results and Analysis

In this section, we present our main results and their proofs.

Recall that in Algorithm 2, the machines are grouped into k batches beforehand. For each batch of machines $1 \le \ell \le k$, we define a function $Z_\ell : \Theta \to \mathbb{R}^d$ to be the *difference* between the average of the batch sample gradient functions and the population gradient, i.e., $\forall\, \theta \in \Theta$

$$\begin{aligned} Z_\ell(\theta) &\triangleq \frac{1}{b}\sum_{j=(\ell-1)b+1}^{\ell b} \nabla \bar{f}^{(j)}(\theta) - \nabla F(\theta) \\ &= \frac{k}{N}\sum_{j=(\ell-1)b+1}^{\ell b}\sum_{i\in\mathcal{S}_j} \nabla f(X_i,\theta) - \nabla F(\theta), \end{aligned} \tag{10}$$

where the last equality follows from (3) and the fact that batch size $b = m/k$ and local data size $|\mathcal{S}_j| = N/m$. Since each function Z_ℓ depends on the local data at ℓ-batch $\{X_i : i \in \mathcal{S}_j, (\ell-1)b+1 \le j \le \ell b\}$ and X_i's are i.i.d., the functions $Z_\ell(\cdot)$'s are also "independently and identically distributed". For any given positive precision parameters ξ_1 and ξ_2 specified later, and $\alpha \in (0, 1/2)$, define a good event

$$\mathcal{E}_{\alpha,\xi_1,\xi_2} \triangleq \left\{ \sum_{\ell=1}^{k} \mathbf{1}_{\{\forall\theta:\, C_\alpha\|Z_\ell(\theta)\| \le \xi_2\|\theta-\theta^*\|+\xi_1\}} \ge k(1-\alpha)+q \right\}. \tag{11}$$

Informally speaking, on event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$, in at least $k(1-\alpha)+q$ batches, the average of the batch sample gradient functions is uniformly close to the population gradient function.

We show our convergence results of Algorithm 2 in two steps. The first step is "deterministic", showing that our Byzantine gradient descent algorithm converges exponentially fast on good event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$. The second part is "stochastic", proving that this good event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$ happens with high probability.

3.1 Convergence of Byzantine Gradient Descent on $\mathcal{E}_{\alpha,\xi_1,\xi_2}$

We consider a fixed execution. Recall that $\mathcal{B}_t$ denotes the set of Byzantine machines at iteration t of the given execution, which could change across iterations. Define a vector of functions $\boldsymbol{g}_t(\cdot)$ with respect to $\mathcal{B}_t$ as:

$$\boldsymbol{g}_t(\theta) = (g_t^{(1)}(\theta), \ldots, g_t^{(m)}(\theta)), \ \forall\, \theta$$

such that $\forall\, \theta$,

$$g_t^{(j)}(\theta) = \begin{cases} \nabla \bar{f}^{(j)}(\theta) & \text{if } j \notin \mathcal{B}_t \\ \star & \text{o.w. ,} \end{cases}$$

where $\star$ is arbitrary[5]. That is, $g_t^{(j)}(\cdot)$ is the true gradient function $\bar{f}^{(j)}(\cdot)$ if machine j is not Byzantine at iteration t, and arbitrary otherwise. It is easy to see that the definition of $\boldsymbol{g}_t(\cdot)$ is consistent with the definition of $\boldsymbol{g}_t(\theta_{t-1})$ in (5). Define $\widetilde{Z}_\ell(\cdot)$ for each θ as

$$\widetilde{Z}_\ell(\theta) \triangleq \frac{1}{b} \sum_{j=(\ell-1)b+1}^{\ell b} g_t^{(j)}(\theta) - \nabla F(\theta). \tag{12}$$

By definition of $g_t^{(j)}(\cdot)$, for any ℓ-th batch such that

$$\{b(\ell-1)+1, \ldots, b\ell\} \cap \mathcal{B}_t = \emptyset,$$

i.e., it does not contain any Byzantine machine at iteration t, it holds that $\widetilde{Z}_\ell(\theta) = Z_\ell(\theta)$ for all θ, where $Z_\ell(\cdot)$ is defined in (10).

Lemma 2. *On event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$, for every iteration $t \geq 1$, we have*

$$\|\mathcal{A}_k\left(\boldsymbol{g}_t(\theta)\right) - \nabla F(\theta)\| \leq \xi_2 \|\theta - \theta^*\| + \xi_1, \quad \forall \theta \in \Theta.$$

Proof. By definition of $\mathcal{A}_k$ in (8), for any fixed θ,

$$\mathcal{A}_k(\boldsymbol{g}_t(\theta)) = \mathsf{med}\left\{ \frac{1}{b}\sum_{j=1}^{b} g_t^{(j)}(\theta), \ \cdots, \ \frac{1}{b}\sum_{j=m-b+1}^{m} g_t^{(j)}(\theta) \right\}$$

Since geometric median is invariant with translation, it follows that

$$\mathcal{A}_k(\boldsymbol{g}_t(\theta)) - \nabla F(\theta) = \mathsf{med}\left\{ \widetilde{Z}_1(\theta), \ \cdots, \ \widetilde{Z}_m(\theta) \right\}.$$

On event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$, at least $k(1-\alpha)+q$ of the k batches $\{Z_\ell : 1 \leq \ell \leq k\}$ satisfy $C_\alpha \|Z_\ell(\theta)\| \leq \xi_2\|\theta-\theta^*\| + \xi_1$ uniformly. Moreover, for Byzantine-free batch ℓ, it holds that $\widetilde{Z}_\ell(\cdot) = Z_\ell(\cdot)$. Hence, at least $k(1-\alpha)$ of the k *received* batches $\{\widetilde{Z}_\ell : 1 \leq \ell \leq k\}$ satisfy $C_\alpha\|\widetilde{Z}_\ell(\theta)\| \leq \xi_2\|\theta-\theta^*\| + \xi_1$ uniformly. The conclusion readily follows from Lemma 1 with $\gamma = 0$.

□

[5]By "arbitrary" we mean that $g_t^{(j)}(\cdot)$ cannot be specified.

3.1.1 Convergence of Approximate Gradient Descent

Next, we show a convergence result of an approximate gradient descent, which might be of independent interest. For any $\theta \in \Theta$, define a new θ' as

$$\theta' = \theta - \eta \times \nabla F(\theta). \tag{13}$$

We remark that the above update is one step of population gradient descent given in (2).

Lemma 3. *Suppose Assumption 1 holds. If we choose the step size $\eta = L/(2M^2)$, then θ' defined in (13) satisfies that*

$$\|\theta' - \theta^*\| \le \sqrt{1 - L^2/(4M^2)}\, \|\theta - \theta^*\|. \tag{14}$$

The proof of Lemma 3 is rather standard, and is presented in Section B for completeness. Suppose that for each $t \ge 1$, we have access to gradient function $G_t(\cdot)$, which satisfy the uniform deviation bound:

$$\|G_t(\theta) - \nabla F(\theta)\| \le \xi_1 + \xi_2 \|\theta - \theta^*\|, \quad \forall \theta, \tag{15}$$

for two positive precision parameters ξ_1, ξ_2 that are *independent* of t. Then we perform the following approximate gradient descent as a surrogate for population gradient descent:

$$\theta_t = \theta_{t-1} - \eta \times G_t(\theta_{t-1}). \tag{16}$$

The following lemma establishes the convergence of the approximate gradient descent.

Lemma 4. *Suppose Assumption 1 holds, and choose $\eta = L/(2M^2)$. If (15) holds for each $t \ge 1$ and*

$$\rho \triangleq 1 - \sqrt{1 - L^2/(4M^2)} - \xi_2 L/(2M^2) > 0,$$

then the iterates $\{\theta_t\}$ in (16) satisfy

$$\|\theta_t - \theta^*\| \le (1-\rho)^t \|\theta_0 - \theta^*\| + \eta\xi_1/\rho.$$

Remark 3. As it can be seen later, the precision parameter ξ_2 can be chosen to be a function of N/k such that $\xi_2 \to 0$ as $N/k \to \infty$. Thus, there exists ξ_2 for ρ defined in Lemma 4 to be positive.

Proof of Lemma 4. Fix any $t \ge 1$, we have

$$\begin{aligned}
\|\theta_t - \theta^*\| &= \|\theta_{t-1} - \eta G_t(\theta_{t-1}) - \theta^*\| \\
&= \|\theta_{t-1} - \eta \nabla F(\theta_{t-1}) - \theta^* + \eta\left(\nabla F(\theta_{t-1}) - G_t(\theta_{t-1})\right)\| \\
&\le \|\theta_{t-1} - \eta \nabla F(\theta_{t-1}) - \theta^*\| + \eta \left\|\nabla F(\theta_{t-1}) - G_t(\theta_{t-1})\right\|.
\end{aligned}$$

It follows from Lemma 3 that

$$\|\theta_{t-1} - \eta \nabla F(\theta_{t-1}) - \theta^*\| \le \sqrt{1 - L^2/(4M^2)}\, \|\theta_{t-1} - \theta^*\|$$

and from (15) that

$$\|\nabla F(\theta_{t-1}) - G_t(\theta_{t-1})\| \le \xi_1 + \xi_2 \|\theta_{t-1} - \theta^*\|.$$

Hence,

$$\|\theta_t - \theta^*\| \le \left(\sqrt{1 - L^2/(4M^2)} + \eta\xi_2\right) \|\theta_{t-1} - \theta^*\|_2 + \eta\xi_1.$$

A standard telescoping argument then yields that

$$\begin{aligned}
\|\theta_t - \theta^*\| &\le (1-\rho)^t \|\theta_0 - \theta^*\| + \eta\xi_1 \sum_{\tau=0}^{t-1} (1-\rho)^\tau \\
&\le (1-\rho)^t \|\theta_0 - \theta^*\| + \eta\xi_1/\rho,
\end{aligned}$$

where $\rho = 1 - \sqrt{1 - L^2/(4M^2)} - \xi_2 L/(2M^2)$ and $\eta = L/(2M^2)$. ∎

3.1.2 Convergence of Byzantine Gradient Descent on Good Event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$

With Lemma 2 and the convergence of the approximate gradient descent (Lemma 4), we show that Algorithm 2 converges exponentially fast on good event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$.

Theorem 2. *Suppose event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$ holds and iterates $\{\theta_t\}$ are given by Algorithm 2 with $\eta = L/(2M^2)$. If $\rho = 1 - \sqrt{1 - L^2/(4M^2)} - \xi_2 L/(2M^2) > 0$ as defined in Lemma 4, then*

$$\|\theta_t - \theta^*\| \leq (1-\rho)^t \|\theta_0 - \theta^*\| + \eta\xi_1/\rho. \tag{17}$$

Proof. In Algorithm 2, at iteration t, the parameter server updates the model parameter θ_{t-1} using the approximate gradient $\mathcal{A}_k\left(\boldsymbol{g}_t(\theta_{t-1})\right)$ – the value of the approximate gradient function $\mathcal{A}_k\left(\boldsymbol{g}_t(\cdot)\right)$ evaluated at θ_{t-1}. From Lemma 2, we know that on event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$

$$\|\mathcal{A}_k\left(\boldsymbol{g}_t(\theta)\right) - \nabla F(\theta)\| \leq \xi_2\|\theta - \theta^*\| + \xi_1, \quad \forall \theta \in \Theta.$$

The conclusion then follows from Lemma 4 by setting $G_t(\theta)$ to be $\mathcal{A}_k\left(\boldsymbol{g}_t(\theta)\right)$. □

3.2 Bound Probability of Good Event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$

Recall that for each batch ℓ for $1 \leq \ell \leq k$, Z_ℓ is defined in (10) w.r.t. the data samples collectively kept by the machines in this batch. Thus, function Z_ℓ is random. The following lemma gives a lower bound to the probability of good event $\mathcal{E}_{\alpha,\xi_1,\xi_2}$.

Lemma 5. *Suppose for all $1 \leq \ell \leq k$, Z_ℓ satisfies*

$$\mathbb{P}\left\{\forall \theta : C_\alpha\|Z_\ell(\theta)\| \leq \xi_2\|\theta - \theta^*\| + \xi_1\right\} \geq 1 - \delta \tag{18}$$

for any $\alpha \in (q/k, 1/2)$ and $0 < \delta \leq \alpha - q/k$. Then

$$\mathbb{P}\left\{\mathcal{E}_{\alpha,\xi_1,\xi_2}\right\} \geq 1 - e^{-kD(\alpha - q/k\|\delta)}. \tag{19}$$

Proof. Let $T \sim \text{Binom}(k, 1-\delta)$. By assumption (18),

$$\sum_{\ell=1}^{k} \mathbf{1}_{\{\forall\theta : C_\alpha\|Z_\ell(\theta)\|_2 \leq \xi_2\|\theta-\theta^*\|+\xi_1\}}$$

first-order stochastically dominates T, i.e.,

$$\mathbb{P}\left\{\sum_{\ell=1}^{k} \mathbf{1}_{\{\forall\theta : C_\alpha\|Z_\ell(\theta)\|_2 \leq \xi_2\|\theta-\theta^*\|+\xi_1\}} \geq k(1-\alpha) + q\right\} \geq \mathbb{P}\left\{T \geq k(1-\alpha) + q\right\}. \tag{20}$$

By Chernoff's bound for binomial distributions, the following holds:

$$\mathbb{P}\left\{T \geq k(1-\alpha) + q\right\} \geq 1 - e^{-kD(\alpha - q/k\|\delta)}. \tag{21}$$

Combining (20) and (21) together, we conclude (19).

□

It remains to show the uniform convergence of Z_ℓ as required by (18). To this end, we need to impose a few technical assumptions that are rather standard [Ver10]. Recall that gradient $\nabla f(X, \theta)$ is random as the input X is random. We assume gradient $\nabla f(X, \theta^*)$ is sub-exponential. The definition and some related concentration properties of sub-exponential random variables are presented in Section C for completeness.

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

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

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

Intuitively speaking, Assumption 2 is placed to ensure that, with high probability, using the *true* sample gradient for individual batches, we are able to "identify" the optimal model θ^*. That is, $(1/n)\sum_{i=1}^n \nabla f(X_i, \theta^*)$ concentrates around its mean $\nabla F(\theta^*) = 0$.

Lemma 6. *Suppose Assumption 2 holds. For any $\delta \in (0,1)$ and any positive integer n, let*

$$\Delta_1(n, d, \delta, \sigma_1) = \sqrt{2}\sigma_1 \sqrt{\frac{d \log 6 + \log(3/\delta)}{n}}. \tag{22}$$

If $\Delta_1(n, d, \delta, \sigma_1) \leq \sigma_1^2/\alpha_1$, then

$$\mathbb{P}\left\{ \left\| \frac{1}{n} \sum_{i=1}^n \nabla f(X_i, \theta^*) - \nabla F(\theta^*) \right\| \geq 2\Delta_1(n, d, \delta, \sigma_1) \right\} \leq \frac{\delta}{3}.$$

Remark 4. By definition of $\Delta_1(n, d, \delta, \sigma_1)$, $\Delta_1(n, d, \delta, \sigma_1)$ is a non-increasing function of n. In particular, for fixed δ and σ_1, if $d = o(n)$,

$$\Delta_1(n, d, \delta, \sigma_1) = \sqrt{2}\sigma_1 \sqrt{\frac{d \log 6 + \log(3/\delta)}{n}} \to 0 \qquad \text{as} \quad n \to \infty.$$

Thus, if in addition α_1 is assumed to be fixed, then for sufficiently large n, $\Delta_1(n, d, \delta, \sigma_1) \leq \sigma_1^2/\alpha_1$ holds.

With a little abuse of notation, we write $\Delta_1(n, d, \delta, \sigma_1)$ as Δ_1 or $\Delta_1(n)$ for short when its meaning is clear from the context. Also, we let $\nabla \bar{f}_n(\theta)$ denote $\frac{1}{n}\sum_{i=1}^n \nabla f(X_i, \theta)$ for ease of exposition.

Proof of Lemma 6. Let $\mathcal{V} = \{v_1, \ldots, v_{N_{1/2}}\}$ denote an $\frac{1}{2}$-cover of unit sphere B. It is shown in [Ver10, Lemma 5.2, Lemma 5.3] that $\log N_{1/2} \leq d \log 6$, and

$$\left\| \nabla \bar{f}_n(\theta^*) - \nabla F(\theta^*) \right\| \leq 2 \sup_{v \in \mathcal{V}} \left\{ \left\langle \nabla \bar{f}_n(\theta^*) - \nabla F(\theta^*), v \right\rangle \right\}.$$

Note that since $\nabla F(\theta^*) = 0$, it holds that $\nabla f(X_i, \theta^*) - \nabla F(\theta^*) = \nabla f(X_i, \theta^*)$. By Assumption 2 and the condition that $\Delta_1 \leq \sigma_1^2/\alpha_1$, it follows from concentration inequalities for sub-exponential random variables given in Theorem 6 that, for $v \in \mathcal{V}$

$$\mathbb{P}\left\{ \left\langle \nabla \bar{f}_n(\theta^*) - \nabla F(\theta^*), v \right\rangle \geq \Delta_1 \right\} \leq \exp\left(-n\Delta_1^2/(2\sigma_1^2)\right).$$

Recall that in $\mathcal{V}$ contains at most 6^d vectors. In view of the union bound, it further yields that

$$\begin{aligned} \mathbb{P}\left\{ 2 \sup_{v \in \mathcal{V}} \left\{ \left\langle \nabla \bar{f}_n(\theta^*) - \nabla F(\theta^*), v \right\rangle \right\} \geq 2\Delta_1 \right\} &\leq 6^d \exp\left(-n\Delta_1^2/(2\sigma_1^2)\right) \\ &= \exp\left(-n\Delta_1^2/(2\sigma_1^2) + d \log 6\right). \end{aligned}$$

Therefore,

$$\mathbb{P}\left\{ \left\| \nabla \bar{f}_n(\theta^*) - \nabla F(\theta^*) \right\| \geq 2\Delta_1 \right\} \leq \exp\left(-n\Delta_1^2/(2\sigma_1^2) + d \log 6\right).$$

□

In addition to the "identifiability" of the optimal θ^* using sample gradients $\nabla f(X, \theta^*)$, similar to the smoothness requirements of the population gradient $\nabla F(\cdot)$ stated in Assumption 1, some smoothness properties (in stochastic sense) of the sample gradients $\nabla f(X, \cdot)$ are also desired. Next, we define gradient difference

$$h(x, \theta) \triangleq \nabla f(x, \theta) - \nabla f(x, \theta^*), \tag{23}$$

which characterizes the deviation of random gradient $\nabla f(x, \theta)$ from $\nabla f(x, \theta^*)$. Note that

$$\mathbb{E}\left[h(X, \theta)\right] = \nabla F(\theta) - \nabla F(\theta^*) \tag{24}$$

for each θ. The following assumptions ensure that for every θ, $h(x, \theta)$ normalized by $\|\theta - \theta^*\|$ is also sub-exponential.

Assumption 3. There exist positive constants σ_2 and α_2 such that for any $\theta \in \Theta$ with $\theta \neq \theta^*$ and unit vector $v \in B$, $\langle h(X, \theta) - \mathbb{E}\left[h(X, \theta)\right], v\rangle / \|\theta - \theta^*\|$ is sub-exponential with scaling parameters (σ_2, α_2), i.e., for all $|\lambda| \leq \frac{1}{\alpha_2}$,

$$\sup_{\theta \in \Theta, v \in B} \mathbb{E}\left[\exp\left(\frac{\lambda \langle h(X, \theta) - \mathbb{E}\left[h(X, \theta)\right], v\rangle}{\|\theta - \theta^*\|}\right)\right] \leq e^{\sigma_2^2 \lambda^2 / 2}.$$

The following lemma bounds the deviation of $(1/n)\sum_{i=1}^n h(X_i, \theta)$ from $\mathbb{E}\left[h(X, \theta)\right]$ for every $\theta \in \Theta$ under Assumption 3. Its proof is similar to that of Lemma 6 and thus is omitted.

Lemma 7. *Suppose Assumption 3 holds and fix any $\theta \in \Theta$. Let*

$$\Delta_1'(n, d, \delta, \sigma_2) = \sqrt{2}\sigma_2 \sqrt{\frac{d \log 6 + \log(3/\delta)}{n}}. \tag{25}$$

If $\Delta_1'(n, d, \delta, \sigma_2) \leq \sigma_2^2/\alpha_2$, then

$$\mathbb{P}\left\{\left\|\frac{1}{n}\sum_{i=1}^n h(X_i, \theta) - \mathbb{E}\left[h(X, \theta)\right]\right\| > 2\Delta_1'(n, d, \delta, \sigma_2)\|\theta - \theta^*\|\right\} \leq \frac{\delta}{3}.$$

Remark 5. Similar to $\Delta_1(n, d, \delta, \sigma_2)$, if δ, σ_1, and σ_2 are fixed, and $d = o(n)$, then for all sufficiently large n, it holds that $\Delta_1'(n, d, \delta, \sigma_2) \leq \sigma_2^2/\alpha_2$.

We write $\Delta_1'(n, d, \delta, \sigma_2)$ as Δ_1' or $\Delta_1'(n)$ for short.

Assumption 2 and Assumption 3 can be potentially relaxed at an expense of looser concentration bounds. Note that Assumption 3, roughly speaking, only imposes some smoothness condition w. r. t. the optimal model θ^*. To mimic the Lipschitz continuity of the sample gradients (in stochastic sense), we impose the following assumption, which holds automatically if we strengthen Assumption 3 by replacing θ^* with an arbitrary θ' such that $\theta \neq \theta'$. In general, Assumption 4 is strictly weaker than the strengthened version of Assumption 3.

Assumption 4. For any $\delta \in (0, 1)$, there exists an $M' = M'(n, \delta)$ that is non-increasing in n such that

$$\mathbb{P}\left\{\sup_{\theta, \theta' \in \Theta : \theta \neq \theta'} \frac{\|\frac{1}{n}\sum_{i=1}^n \left(\nabla f(X_i, \theta) - \nabla f(X_i, \theta')\right)\|}{\|\theta - \theta'\|} \leq M'\right\} \geq 1 - \frac{\delta}{3}.$$

With Assumption 2–Assumption 4, we apply the celebrated ϵ-net argument to prove the averaged random gradients *uniformly* converges to $\nabla F(\cdot)$.

For a given real number $r > 0$, define Δ_2 as follows.

$$\Delta_2(n) = \sigma_2\sqrt{\frac{2}{n}}\sqrt{d\log\frac{18M\vee M'}{\sigma_2} + \frac{1}{2}d\log\frac{n}{d} + \log\left(\frac{6\sigma_2^2 r\sqrt{n}}{\alpha_2\sigma_1\delta}\right)}. \tag{26}$$

Proposition 1. *Suppose Assumption 2 – Assumption 4 hold, and $\Theta \subset \{\theta : \|\theta - \theta^*\| \le r\sqrt{d}\}$ for some positive parameter r. For any $\delta \in (0,1)$ and any integer n, recall Δ_1 defined in (22) and define Δ_2 as in (26). If $\Delta_1 \le \sigma_1^2/\alpha_1$ and $\Delta_2 \le \sigma_2^2/\alpha_2$, then*

$$\mathbb{P}\left\{\forall\theta\in\Theta : \left\|\frac{1}{n}\sum_{i=1}^n \nabla f(X_i,\theta) - \nabla F(\theta)\right\| \le 8\Delta_2\|\theta-\theta^*\| + 4\Delta_1\right\} \ge 1-\delta.$$

Proof. The proof is based on the classical ϵ-net argument. Let

$$\tau = \frac{\alpha_2\sigma_1}{2\sigma_2^2}\sqrt{\frac{d}{n}} \quad \text{and} \quad \ell^* = \lceil r\sqrt{d}/\tau\rceil.$$

Henceforth, for ease of exposition, we assume ℓ^* is an integer. For integers $1 \le \ell \le \ell^*$, define

$$\Theta_\ell \triangleq \{\theta : \|\theta-\theta^*\| \le \tau\ell\}.$$

For a given ℓ, let $\theta_1,\ldots,\theta_{N_{\epsilon_\ell}}$ be an ϵ_ℓ-cover of Θ_ℓ, where ϵ_ℓ is given by

$$\epsilon_\ell = \frac{\sigma_2\tau\ell}{M\vee M'}\sqrt{\frac{d}{n}},$$

where $M\vee M' = \max\{M, M'\}$. By [Ver10, Lemma 5.2], $\log N_{\epsilon_\ell} \le d\log(3\tau\ell/\epsilon_\ell)$. Fix any $\theta \in \Theta_\ell$. There exists a $1 \le j_\ell \le N_{\epsilon_\ell}$ such that $\|\theta - \theta_{j_\ell}\|_2 \le \epsilon_\ell$. Recall that we let $\nabla\bar{f}_n(\theta)$ denote $\frac{1}{n}\sum_{i=1}^n \nabla f(X_i,\theta)$. By triangle's inequality,

$$\left\|\nabla\bar{f}_n(\theta) - \nabla F(\theta)\right\| \le \|\nabla F(\theta) - \nabla F(\theta_{j_\ell})\| + \left\|\nabla\bar{f}_n(\theta) - \nabla\bar{f}_n(\theta_{j_\ell})\right\| + \left\|\nabla\bar{f}_n(\theta_{j_\ell}) - \nabla F(\theta_{j_\ell})\right\|. \tag{27}$$

In view of Assumption 1,

$$\|\nabla F(\theta) - \nabla F(\theta_{j_\ell})\| \le M\|\theta - \theta_{j_\ell}\| \le M\epsilon_\ell, \tag{28}$$

where the last inequality holds because by the construction of ϵ-net, and the fact that for a given θ, θ_{j_ℓ} is chosen in such a way that $\|\theta - \theta_{j_\ell}\| \le \epsilon_\ell$.

Define event

$$\mathcal{E}_1 = \left\{\sup_{\theta,\theta'\in\Theta:\theta\neq\theta'} \frac{\|\nabla\bar{f}_n(\theta) - \nabla\bar{f}_n(\theta')\|}{\|\theta-\theta'\|} \le M'\right\}.$$

By Assumption 4, we have $\mathbb{P}\{\mathcal{E}_1\} \ge 1 - \delta/3$. On event $\mathcal{E}_1$, we have

$$\sup_{\theta\in\Theta}\left\|\nabla\bar{f}_n(\theta) - \nabla\bar{f}_n(\theta_{j_\ell})\right\| \le M'\epsilon_\ell. \tag{29}$$

By triangle's inequality again,

$$\begin{aligned}\left\|\nabla \bar{f}_n(\theta_{j_\ell}) - \nabla F(\theta_{j_\ell})\right\| &\le \left\|\nabla \bar{f}_n(\theta^*) - \nabla F(\theta^*)\right\| + \left\|\nabla \bar{f}_n(\theta_{j_\ell}) - \nabla \bar{f}_n(\theta^*) - (\nabla F(\theta_{j_\ell}) - \nabla F(\theta^*))\right\| \\ &\le \left\|\nabla \bar{f}_n(\theta^*) - \nabla F(\theta^*)\right\| + \left\|\frac{1}{n}\sum_{i=1}^n h(X_i, \theta_{j_\ell}) - \mathbb{E}\left[h(X, \theta_{j_\ell})\right]\right\|, \end{aligned} \tag{30}$$

where function $h(x, \cdot)$ is defined in (23). Define event

$$\mathcal{E}_2 = \left\{\left\|\nabla \bar{f}_n(\theta^*) - \nabla F(\theta^*)\right\| \le 2\Delta_1\right\}$$

and event

$$\mathcal{F}_\ell = \left\{\sup_{1\le j\le N_\epsilon}\left\|\frac{1}{n}\sum_{i=1}^n h(X_i, \theta_j) - \mathbb{E}\left[h(X, \theta_j)\right]\right\| \le 2\tau\ell\Delta_2\right\},$$

where Δ_2 is defined in (26) and satisfies

$$\Delta_2 = \sqrt{2}\sigma_2\sqrt{\frac{d\log 6 + d\log(3\tau\ell/\epsilon_\ell) + \log(3\ell^*/\delta)}{n}}. \tag{31}$$

In (26), note that Δ_2 is independent of ℓ, due to the choice of ϵ_ℓ made earlier. It is easy to check that (26) and (31) are equivalent.

Since $\Delta_1 \le \sigma_1^2/\alpha_1$, it follows from Lemma 6 that $\mathbb{P}\{\mathcal{E}_2\} \ge 1 - \delta/3$. Similarly, since $\Delta_2 \le \sigma_2^2/\alpha_2$, by Lemma 7, $\mathbb{P}\{\mathcal{F}_\ell\} \ge 1 - \delta/(3\ell^*)$. In particular,

$$\begin{aligned}\mathbb{P}\{\mathcal{F}_\ell^c\} &= \mathbb{P}\left\{\sup_{1\le j\le N_{\epsilon_\ell}}\left\|\frac{1}{n}\sum_{i=1}^n h(X_i, \theta_j) - \mathbb{E}\left[h(X, \theta_j)\right]\right\| > 2\tau\ell\Delta_2\right\} \\ &\le \sum_{j=1}^{N_{\epsilon_\ell}}\mathbb{P}\left\{\left\|\frac{1}{n}\sum_{i=1}^n h(X_i, \theta_j) - \mathbb{E}\left[h(X, \theta_j)\right]\right\| > 2\tau\ell\Delta_2\right\} \\ &\le \frac{\delta}{3\ell^*}\frac{1}{\left(\frac{3\tau\ell}{\epsilon_\ell}\right)^d}\left(\frac{3\tau\ell}{\epsilon_\ell}\right)^d = \frac{\delta}{3\ell^*}. \end{aligned} \tag{32}$$

Therefore, we have $\mathbb{P}\{\mathcal{F}_\ell\} \ge 1 - \delta/(3\ell^*)$.

In conclusion, by combining (27), (28), (29) and (30), it follows that on event $\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{F}_\ell$,

$$\begin{aligned}\sup_{\theta\in\Theta_\ell}\left\|\nabla \bar{f}_n(\theta) - \nabla F(\theta)\right\| &\le (M + M')\epsilon_\ell + 2\Delta_1 + 2\Delta_2\tau\ell \\ &\le 4\Delta_2\tau\ell + 2\Delta_1, \end{aligned}$$

where the last inequality holds due to $(M \vee M')\epsilon_\ell \le \Delta_2\tau\ell$. Let

$$\mathcal{E} = \mathcal{E}_1 \cap \mathcal{E}_2 \cap \left(\cap_{\ell=1}^{\ell^*}\mathcal{F}_\ell\right).$$

It follows from the union bound, $\mathbb{P}\{\mathcal{E}\} \ge 1 - \delta$. Moreover, suppose event $\mathcal{E}$ holds. Then for all $\theta \in \Theta_{\ell^*}$, there exists an $1 \le \ell \le \ell^*$ such that $(\ell - 1)\tau < \|\theta - \theta^*\| \le \ell\tau$. If $\ell \ge 2$, then $\ell \le 2(\ell - 1)$ and thus

$$\left\|\nabla \bar{f}_n(\theta) - \nabla F(\theta)\right\| \le 4\Delta_2\tau\ell + 2\Delta_1 \le 8\Delta_2\|\theta - \theta^*\| + 2\Delta_1.$$

If $\ell = 1$, then

$$\left\|\nabla \bar{f}_n(\theta) - \nabla F(\theta)\right\| \le 4\Delta_2\tau + 2\Delta_1 \le 4\Delta_1,$$

where the last inequality follows from our choice of τ and the assumption that $\Delta_2 \le \sigma_2^2/\alpha_2$ and $\Delta_1 \ge \sigma_1\sqrt{d/n}$. In conclusion, on event $\mathcal{E}$,

$$\sup_{\theta\in\Theta_{\ell^*}} \left\|\nabla \bar{f}_n(\theta) - \nabla F(\theta)\right\| \le 4\Delta_1 + 8\Delta_2\|\theta-\theta^*\|.$$

The proposition follows by the assumption that $\Theta \subset \Theta_{\ell^*}$.

□

Theorem 3. *Suppose Assumption 2 – Assumption 4 hold, and $\Theta \subset \{\theta : \|\theta-\theta^*\| \le r\sqrt{d}\}$ for some positive parameter r. For any $\delta \in (0,1)$ and any integer n, define $\Delta_1(n)$ and $\Delta_2(n)$ as in (22) and (26), respectively. If $\Delta_1(N/k) \le \sigma_1^2/\alpha_1$ and $\Delta_2(N/k) \le \sigma_2^2/\alpha_2$, then for every $1 \le \ell \le k$,*

$$\mathbb{P}\left\{\forall\theta\in\Theta : C_\alpha\|Z_\ell(\theta)\| \le \xi_2\|\theta-\theta^*\| + \xi_1\right\} \ge 1-\delta,$$

where $\xi_1 = 4C_\alpha \times \Delta_1(N/k)$ and $\xi_2 = 8C_\alpha \times \Delta_2(N/k)$.

Proof. Recall that Z_ℓ is defined in (10). Note that for each $1 \le \ell \le k$, Z_ℓ has the same distribution as the average of N/k i.i.d. random gradients $f(X_i,\theta)$ subtracted by $\nabla F(\theta)$. Hence, Theorem 3 readily follows from Proposition 1. □

Remark 6. Suppose σ_1, α_1, σ_2, α_2 are all of $\Theta(1)$, $\log(M \vee M') = O(\log d)$, $\log(1/\delta) = O(d)$ and $\log r = O(d\log(N/k))$. In this case, Theorem 3 implies that if $N/k = \Omega(C_\alpha^2 d\log(N/k))$, then

$$\xi_1 = O\left(C_\alpha\sqrt{kd/N}\right) \quad \text{and} \quad \xi_2 = O\left(C_\alpha\sqrt{kd\log(N/k)/N}\right).$$

In particular, those assumptions are indeed satisfied under the linear regression model as shown in Lemma 8.

3.3 Main Theorem

By combining Theorem 2, Lemma 5, and Theorem 3, we prove the main theorem.

Theorem 4. *Suppose Assumption 1 – Assumption 4 hold, and $\Theta \subset \{\theta : \|\theta-\theta^*\| \le r\sqrt{d}\}$ for some positive parameter r. Assume $2(1+\epsilon)q \le k \le m$. Fix any constant $\alpha \in (q/k, 1/2)$ and any $\delta > 0$ such that $\delta \le \alpha - q/k$. If $\Delta_1(N/k) \le \sigma_1^2/\alpha_1$, $\Delta_2(N/k) \le \sigma_2^2/\alpha_2$, and*

$$\rho = 1 - \sqrt{1 - L^2/(4M^2)} - \xi_2 L/(2M^2) > 0$$

for $\xi_2 = 8C_\alpha \times \Delta_2(N/k)$, then with probability at least

$$1 - \exp(-kD(\alpha - q/k\|\delta)),$$

the iterates $\{\theta_t\}$ given by Algorithm 2 with $\eta = L/(2M^2)$ satisfy

$$\|\theta_t - \theta^*\| \le (1-\rho)^t\|\theta_0-\theta^*\| + \eta\xi_1/\rho, \quad \forall t \ge 1,$$

where $\xi_1 = 4C_\alpha \times \Delta_1(N/k)$.

Under certain conditions, we are able to further bound ξ_1 and ξ_2. Next we present a formal statement of Theorem 1; it readily follows from Theorem 4.

Theorem 5. *Suppose that Assumption 1 – Assumption 4 hold such that L, M, σ_1, α_1, σ_2, α_2 are all of $\Theta(1)$, and $\log M' = O(\log d)$. Assume that $\Theta \subset \{\theta : \|\theta - \theta^*\| \le r\sqrt{d}\}$ for some positive parameter r such that $\log(r) = O(d\log(N/k))$, and $2(1+\epsilon)q \le k \le m$. Fix any $\alpha \in (q/k, 1/2)$ and any $\delta > 0$ such that $\delta \le \alpha - q/k$ and $\log(1/\delta) = O(d)$. There exist universal positive constants c_1, c_2 such that if*

$$\frac{N}{k} \ge c_1 C_\alpha^2 d \log(N/k),$$

then with probability at least

$$1 - \exp(-kD(\alpha - q/k \| \delta)),$$

the iterates $\{\theta_t\}$ given by Algorithm 2 with $\eta = L/(2M^2)$ satisfy

$$\|\theta_t - \theta^*\| \le \left(\frac{1}{2} + \frac{1}{2}\sqrt{1 - \frac{L^2}{4M^2}}\right)^t \|\theta_0 - \theta^*\| + c_2\sqrt{\frac{dk}{N}}, \quad \forall t \ge 1.$$

Proof. Recall from (22) that

$$\Delta_1(N/k, d, \delta, \sigma_1) = \sqrt{2}\sigma_1\sqrt{\frac{d\log 6 + \log(3/\delta)}{N/k}}.$$

When $\sigma_1 = \Theta(1)$ and $\log(1/\delta) = O(d)$, it holds that $\Delta_1(N/k) = \Theta\left(\sqrt{kd/N}\right)$. Similarly, we have $\Delta_2(N/k) = \Theta\left(\sqrt{\frac{kd\log(N/k)}{N}}\right)$. Hence, there exists an universal positive constant c_1 such that for all $N/k \ge c_1 C_\alpha^2 d\log(N/k)$, it holds that $\Delta_1(N/k) \le \sigma_1^2/\alpha_1$, $\Delta_2(N/k) \le \sigma_2^2/\alpha_2$, and

$$\Delta_2(N/k) \le \frac{M^2}{8C_\alpha L}\left(1 - \sqrt{1 - L^2/(4M^2)}\right). \tag{33}$$

Thus we have

$$\xi_2 = 8C_\alpha \times \Delta_2(N/k) \le \frac{M^2}{L}\left(1 - \sqrt{1 - L^2/(4M^2)}\right),$$

and as a consequence,

$$\rho = 1 - \sqrt{1 - L^2/(4M^2)} - \frac{\xi_2 L}{2M^2} \ge \frac{1}{2} - \frac{1}{2}\sqrt{1 - L^2/(4M^2)} > 0.$$

Hence, we can apply Theorem 4. Finally, to finish the proof, recall that $\eta = L/(2M^2)$ and $\xi_1 = 4C_\alpha \times \Delta_1(N/k)$; thus the term $\eta\xi_1/\rho$ can be bounded as follows:

$$\frac{\eta\xi_1}{\rho} = \frac{L}{2M^2} \times \frac{4C_\alpha\Delta_1(N/k)}{\rho} \le \frac{L}{2M^2} \times \frac{8C_\alpha\Delta_1(N/k)}{1 - \sqrt{1 - L^2/(4M^2)}} \le c_2\sqrt{\frac{dk}{N}},$$

where c_2 is some universal constant. □

4 Application to Linear Regression

We illustrate our general results by applying them to the classical linear regression problem. Let $X_i = (w_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$ denote the input data and define the risk function $f(X_i, \theta) = \frac{1}{2}\left(\langle w_i, \theta\rangle - y_i\right)^2$. For simplicity, we assume that y_i is indeed generated from a linear model:

$$y_i = \langle w_i, \theta^*\rangle + \zeta_i,$$

where θ^* is an unknown true model parameter, $w_i \sim N(0, \mathbf{I})$ is the covariate vector whose covariance matrix is assumed to be identity, and $\zeta_i \sim N(0,1)$ is i.i.d. additive Gaussian noise independent of w_i's. Intuitively, the inner product $\langle w_i, \theta^* \rangle$ can be viewed as some "measurement" of θ^* – the signal; and ζ_i is the additive noise.

The population risk minimization problem (1) is simply

$$\min_{\theta} \quad \frac{1}{2}\|\theta - \theta^*\|_2^2 + \frac{1}{2},$$

where

$$\begin{aligned} F(\theta) \triangleq \mathbb{E}\left[f(X,\theta)\right] &= \mathbb{E}\left[\frac{1}{2}\left(\langle w, \theta\rangle - y\right)^2\right] \\ &= \mathbb{E}\left[\frac{1}{2}\left(\langle w, \theta\rangle - \langle w, \theta^*\rangle - \zeta\right)^2\right] = \frac{1}{2}\|\theta - \theta^*\|_2^2 + \frac{1}{2}, \end{aligned}$$

for which θ^* is indeed the unique minimum. If the function $F(\cdot)$ can be computed exactly, then θ^* can be read from its expression directly. The standard gradient descent method for minimizing $F(\cdot)$ is also straightforward. The population gradient is $\nabla_\theta F(\theta) = \theta - \theta^*$. It is easy to see that the population risk F is M-Lipschitz continuous with $M = 1$, and L-strongly convex with $L = 1$. Hence, Assumption 1 is satisfied with $M = L = 1$; and the stepsize $\eta = L/(2M^2) = 1/2$.

In practice, unfortunately, since we do not know exactly the distribution of the random input X, we can neither read θ^* from the expression $F(\cdot)$ nor compute the population gradient $\nabla F(\theta)$ exactly. We are only able to approximate the population risk $F(\cdot)$ or the population gradient $\nabla F(\theta)$. Our focus is the gradient approximation. In particular, for a given random sample, the associated random gradient is given by $\nabla f(X,\theta) = w\langle w, \theta - \theta^*\rangle - w\zeta$, where $w \sim \mathcal{N}(0, \mathbf{I})$ and $\zeta \sim \mathcal{N}(0,1)$ that is independent of w.

The following lemma verifies that Assumption 2–Assumption 4 are satisfied with appropriate parameters.

Lemma 8. *Under the linear regression model, the sample gradient function $\nabla f(X,\cdot)$ satisfies*
(1) Assumption 2 with $\sigma_1 = \sqrt{2}$ and $\alpha_1 = \sqrt{2}$,
(2) Assumption 3 with $\sigma_2 = \sqrt{8}$ and $\alpha_2 = 8$,
(3) and Assumption 4 with $M'(\delta) = d + 2\sqrt{d\log(4/\delta)} + 2\log(4/\delta)$.

Proof. We first check Assumption 2. Recall that $\nabla f(X,\theta) = w\langle w, \theta - \theta^*\rangle - w\zeta$, where $w \sim \mathcal{N}(0,\mathbf{I})$ and $\zeta \sim \mathcal{N}(0,1)$ is independent of w. Hence, $\nabla f(X,\theta^*) = -w\zeta$. It follows that for any v in unit sphere B,

$$\langle \nabla f(X,\theta^*), v\rangle = -\zeta \langle w, v\rangle .$$

Because $w \sim \mathcal{N}(0,\mathbf{I})$ and are independent of ζ, it holds that $\langle w, v\rangle \sim \mathcal{N}(0,1)$ and is independent of ζ. Thus, to compute $\mathbb{E}\left[\exp\left(-\lambda\zeta\langle w, v\rangle\right)\right]$, we can use the standard conditioning argument. In particular, for $\lambda^2 < 1$,

$$\begin{aligned} \mathbb{E}\left[\exp\left(\lambda\langle \nabla f(X,\theta^*), v\rangle\right)\right] &= \mathbb{E}\left[\exp\left(-\lambda\zeta\langle w, v\rangle\right)\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[\exp\left(-\lambda y\langle w, v\rangle\right)|\zeta = y\right]\right], \end{aligned} \tag{34}$$

where the expectation of $\mathbb{E}\left[\exp\left(-\lambda y\langle w, v\rangle\right)|\zeta = y\right]$ is taken over the conditional distribution of $\langle w, v\rangle$ conditioning on ζ being y. Since $\langle w, v\rangle$ and ζ are independent of each other, the conditional distribution of $\langle w, v\rangle$ w. r. t. ζ is the same as the unconditional distribution of $\langle w, v\rangle$, which is a

Gaussian distribution. Thus, we can apply the moment generating function of Gaussian distribution to get

$$\mathbb{E}\left[\exp\left(-\lambda y \left\langle w, v\right\rangle\right) | \zeta = y\right] = \exp\left(\lambda^2 y^2/2\right).$$

Then, the right-hand side of (34) becomes

$$\begin{aligned}\mathbb{E}\left[\exp\left(\lambda \left\langle \nabla f(X, \theta^*), v\right\rangle\right)\right] &= \mathbb{E}\left[\mathbb{E}\left[\exp\left(-\lambda y \left\langle w, v\right\rangle\right) | \zeta = y\right]\right] \\ &= \mathbb{E}\left[\exp\left(\lambda^2 \zeta^2/2\right)\right] \\ &\overset{(a)}{=} (1-\lambda^2)^{-1/2}, \end{aligned} \tag{35}$$

where equality (a) follows from the moment generating function of χ^2 distribution, i.e.,

$$\mathbb{E}\left[\exp\left(t\zeta^2\right)\right] = (1-2t)^{-1/2} \quad \text{for} \quad t < 1/2.$$

Using the fact that $1 - \lambda^2 \geq e^{-2\lambda^2}$ for $\lambda^2 \leq 1/2$, it follows that

$$\mathbb{E}\left[\exp\left(\lambda \left\langle \nabla f(X, \theta), v\right\rangle\right)\right] \leq e^{\lambda^2}, \quad \forall |\lambda| \leq \frac{1}{\sqrt{2}}.$$

Thus Assumption 2 holds with $\sigma_1 = \sqrt{2}$ and $\alpha_1 = \sqrt{2}$.

Next, we verify Assumption 4. Note that $\nabla^2 f(X, \theta) = ww^\top$ and hence it suffices to show that

$$\mathbb{P}\left\{\left\|\frac{1}{n}\sum_{i=1}^n \nabla^2 f(X_i, \theta)\right\| \leq M'\right\} = \mathbb{P}\left\{\left\|\frac{1}{n}\sum_{i=1}^n w_i w_i^\top\right\| \leq M'\right\} \geq 1 - \frac{\delta}{3},$$

for some M' depending on n, d, and δ.

Let $W = [w_1, w_2, \ldots, w_n]$ denote the $d \times n$ matrix whose columns are given by w_i's. Then $\sum_{i=1}^n w_i w_i^\top = WW^\top$. Also, the spectral norm of $WW^\top$ equals $\|W\|^2$. Therefore,

$$\mathbb{P}\left\{\left\|\frac{1}{n}\sum_{i=1}^n w_i w_i^\top\right\| \leq M'\right\} = \mathbb{P}\left\{\|W\| \leq \sqrt{nM'}\right\}.$$

Note that W is an $d \times n$ matrix with i.i.d. standard Gaussian entries. Standard Gaussian matrix concentration inequality (see, e.g., [Ver10, Corollary 5.35]) states that for every $t \geq 0$,

$$\mathbb{P}\left\{\|W\| \leq \sqrt{n} + \sqrt{d} + t\right\} \geq 1 - \exp(-t^2/2).$$

Plugging $t = \sqrt{2\log(4/\delta)}$ and setting

$$M' = \frac{1}{n}\left(\sqrt{n} + \sqrt{d} + \sqrt{2\log(4/\delta)}\right)^2$$

complete the proof.

Finally, we verify Assumption 3. Recall that the gradient difference $h(X, \theta)$ is given by $h(X, \theta) = w\langle w, \theta - \theta^*\rangle$, and $\mathbb{E}\left[h(X, \theta)\right] = \theta - \theta^*$. It follows that for any vector v in unit sphere B,

$$\left\langle h(X, \theta) - \mathbb{E}\left[h(X, \theta)\right], v\right\rangle = \langle w, \theta - \theta^*\rangle\langle w, v\rangle - \langle \theta - \theta^*, v\rangle.$$

For a fixed $\theta \in \Theta$ with $\theta \neq \theta^*$ and let $\tau = \|\theta - \theta^*\| > 0$. Then we have the following orthogonal decomposition: $\theta - \theta^* = \sqrt{\gamma} v + \sqrt{\eta} v_\perp$, where $\gamma + \eta = \tau^2$, and $v_\perp$ denote an vector perpendicular to v. It follows that

$$\langle w, \theta - \theta^* \rangle \langle w, v \rangle - \langle \theta - \theta^*, v \rangle = \sqrt{\gamma} \langle w, v \rangle^2 - \sqrt{\gamma} + \sqrt{\eta} \langle w, v_\perp \rangle \langle w, v \rangle.$$

It is easy to see that random variables $\langle w, v_\perp \rangle \sim \mathcal{N}(0,1)$ and $\langle w, v \rangle \sim \mathcal{N}(0,1)$ are jointly Gaussian. In addition, we have

$$\begin{aligned} \mathbb{E}\left[\langle w, v_\perp \rangle \langle w, v \rangle\right] &= \mathbb{E}\left[v_\perp^\top w w^\top v\right] \\ &= v_\perp^\top \mathbb{E}\left[w w^\top\right] v = v_\perp^\top \mathbf{I} v = 0. \end{aligned}$$

Thus, $\langle w, v_\perp \rangle \sim \mathcal{N}(0,1)$ and $\langle w, v \rangle \sim \mathcal{N}(0,1)$ are mutually independent.

For any λ with $\lambda\sqrt{\gamma} < 1/4$ and $\lambda^2 \eta < 1/4$,

$$\begin{aligned} &\mathbb{E}\left[\exp\left(\lambda \langle h(X,\theta) - \mathbb{E}\left[h(X,\theta)\right], v \rangle\right)\right] \\ &= \mathbb{E}\left[\exp\left(\lambda\sqrt{\gamma}\left(\langle w, v\rangle^2 - 1\right) + \lambda\sqrt{\eta}\langle w, v_\perp\rangle\langle w, v\rangle\right)\right] \\ &\leq \sqrt{\mathbb{E}\left[e^{2\lambda\sqrt{\gamma}(\langle w,v\rangle^2 - 1)}\right]\mathbb{E}\left[e^{2\lambda\sqrt{\eta}\langle w,v_\perp\rangle\langle w,v\rangle}\right]} \\ &= e^{-\lambda\sqrt{\gamma}}\sqrt{\mathbb{E}\left[e^{2\lambda\sqrt{\gamma}(\langle w,v\rangle^2)}\right]}\sqrt{\mathbb{E}\left[e^{2\lambda\sqrt{\eta}\langle w,v_\perp\rangle\langle w,v\rangle}\right]} \\ &= e^{-\lambda\sqrt{\gamma}}\left(1 - 4\lambda\sqrt{\gamma}\right)^{-1/4}\left(1 - 4\lambda^2\eta\right)^{-1/4}, \end{aligned}$$

where the first inequality holds due to Cauchy-Schwartz's inequality, and the last equality follows by plugging in the moment generating functions for χ^2 distributions as well as using the conditioning argument that is similar to the derivation of (34).

Using the fact that $e^{-t}/\sqrt{1-2t} \leq e^{2t^2}$ for $|t| \leq 1/4$ and $1 - t \geq e^{-4t}$ for $0 \leq t \leq 1/2$, it follows that for $\lambda^2 \leq 1/(64\tau^2)$,

$$\begin{aligned} \mathbb{E}\left[\exp\left(\lambda \langle h(X,\theta) - \mathbb{E}\left[h(X,\theta)\right], v \rangle\right)\right] &\leq \exp\left(4\lambda^2(\gamma + \eta)\right) \\ &\leq \exp\left(4\lambda^2\tau^2\right). \end{aligned}$$

Hence, Assumption 3 holds with $\sigma_2 = \sqrt{8}$ and $\alpha_2 = 8$.

□

Thus, according to Theorem 1, our Byzantine Gradient Descent method can robustly solve the linear regression problem exponentially fast with high probability – formally stated the following corollary.

Corollary 1 (Linear regression)**.** *Under the aforementioned least-squares model for linear regression, assume* $\Theta \subset \{\theta : \|\theta - \theta^*\| \leq r\sqrt{d}\}$ *for* $r > 0$ *such that* $\log r = O(d \log(N/k))$*. Suppose that* $2(1+\epsilon)q \leq k \leq m$*. Fix any* $\alpha \in (q/k, 1/2)$ *and any* $\delta > 0$ *such that* $\delta \leq \alpha - q/k$ *and* $\log(1/\delta) = O(d)$*, there exist universal constants* $c_1, c_2 > 0$ *such that if* $N/k \geq c_1 C_\alpha^2 d \log(N/k)$*. Then with probability at least* $1 - \exp(-kD((\alpha - q/k)\|\delta))$*, the iterates* $\{\theta_t\}$ *given by Algorithm 2 with* $\eta = 1/2$ *satisfy*

$$\|\theta_t - \theta^*\| \leq \left(\frac{1}{2} + \frac{\sqrt{3}}{4}\right)^t \|\theta_0 - \theta^*\| + c_2 C_\alpha \sqrt{\frac{dk}{N}}, \quad \forall t \geq 1.$$

Note that in Corollary 1, we assume the "searching space" Θ belongs to some range, which may grow with d and N/k. This assumption is rather mild since in practice; we typically do have some prior knowledge about the range of θ^*.

5 Related Work

The present paper intersects with two main areas of research: statistical machine learning and distributed computing. Most related to our work is [BMGS17] that we became aware of when preparing this paper. It also studies distributed optimization in adversarial settings, but the setup is different from ours. In particular, their focus is solving an optimization problem, where all m working machines have access to a common dataset $\{x_i\}_{i=1}^N$ and the goal is to collectively compute the minimizer $\widehat{\theta}$ of the average cost $Q(\theta) = (1/N)\sum_{i=1}^N f(x_i, \theta)$. Importantly, the dataset $\{x_i\}_{i=1}^N$ are assumed to be deterministic. In contrast, we adopt the standard statistical learning framework, where each working machine only has access to its own data samples, which are assumed to be generated from some unknown distribution μ, and the goal is to estimate the optimal model parameter θ^* that minimizes the true prediction error $\mathbb{E}_{X\sim\mu}[f(X, \theta)]$ — as mentioned, characterizing the statistical estimation accuracy is a main focus of ours. Our algorithmic approaches and main results are also significantly different. The almost sure convergence is proved in [BMGS17] without an explicit characterization of convergence speed nor the estimation errors.

Our work is also closely related to the literature on robust parameter estimation using geometric median. It is shown in [LR91] that geometric median has a breakdown point of 0.5, that is, given a collection of n vectors in $\mathbb{R}^d$, at least $\lfloor (n+1)/2 \rfloor / n$ number of points needs to be corrupted in order to arbitrarily perturb the geometric median. A more quantitative robustness result is recently derived in [M$^+$15, Lemma 2.1]. The geometric median has been applied to distributed machine learning under the one-shot aggregation framework [FXM14], under the restrictive assumption that the number of data available in each working machine satisfies $N/m \gg d$. While we also apply geometric median-of-mean as a sub-routine, our problem setup, overall algorithms and main results are completely different.

A recent line of work [DKK$^+$16, LRV16] presents polynomial algorithms to consistently estimate the mean and covariance of a distribution from N i.i.d. samples in $\mathbb{R}^d$, in the presence of an ϵ fraction of malicious errors for sufficiently small ϵ, while geometric median is proved to fail when $\epsilon = \Omega(1/\sqrt{d})$. However, it is unclear how to directly apply their results to our gradient descent setting, where our goal is to robustly estimate a d-dimensional gradient function from i.i.d. sample gradient functions.

On the technical front, a crucial step in our convergence proof is to show the geometric median of means of n i.i.d. random gradients converges to the underlying gradient function $\nabla F(\theta)$ uniformly over θ. Our proof builds on several ideas from the empirical process theory, which guarantees uniform convergence of the empirical risk function $(1/n)\sum_{i=1}^n f(X_i, \cdot)$ to the population risk $F(\cdot)$. However, what we need is the uniform convergence of empirical *gradient* function $(1/n)\sum_{i=1}^n \nabla f(X_i, \cdot)$, as well as its *geometric median* version, to the population gradient function $\nabla F(\cdot)$. To this end, we use concentration inequalities to first establish point-wise convergence and then boost it to uniform convergence via the celebrated ϵ-net argument. Similar ideas have been used recently in the work [MBM16], which studies the stationary points of the empirical risk function.

6 Discussion

In this paper, we consider the machine learning scenario where the model is trained in an unsecured environment. As a result of this, the model training procedure needs to be robust to adversarial interruptions. Based on the geometric median of means, we propose a communication-efficient and robust method for the parameter server to aggregate the gradients reported by the unreliable

workers. In each iteration, the parameter server first groups the received gradients into non-overlapping batches to increase the "similarity" of the Byzantine-free batches; and then takes the median of the batch gradients to cripple the interruption of Byzantine machines.

There are many other interesting directions. We list a few of them as follows.

- As mentioned in the introduction, Federated Learning is proposed due to the users' concerns about privacy breaches. In Federated Learning, the training data is kept locally on user's devices, which indeed grants users the control of their data. Nevertheless, to have a high-quality model trained, information about their data need to be extracted. In our future work, we would like to provide a precise characterization of the minimal amount of privacy has to be sacrificed in the Federated Learning paradigm.

- In addition to security, low volume of local data and communication constraints, there are many other practical challenges such as intermittent availability of mobile phones, i.e., communication asynchrony. Although our algorithm only needs $\log(N)$ rounds, a single synchronous round may be significantly "delayed" by the slow machines. We would like to adapt our algorithms to the asynchronous setting.

- In Byzantine fault models, we assume the Byzantine adversaries know the realization of the random bits generated by the parameter server. Depending on the applications, this assumption can possibly be relaxed, which may lead to simpler algorithms. A simple idea to defend against the relaxed Byzantine faults is to selects a subset of received gradients at each iteration and then takes the average over the selected gradients. One selection rule is random selection and another one is to select the gradients of the small ℓ_2 norms. It would be interesting to investigate the performance of these two selection rules and compare them with the geometric median.

A Proof of Lemma 2.1

Proof. Let $S = \{i : \|z_i\| \le r\}$. For any $i \in S$, we have

$$\|z_* - z_i\| \ge \|z_*\| - \|z_i\| \ge \|z_*\| - 2r + \|z_i\|.$$

Moreover, by triangle's inequality, for all $i \notin S$, we have

$$\|z_* - z_i\| \ge \|z_i\| - \|z_*\|$$

Combining the last two displayed equations yields that

$$\sum_{i=1}^{n} \|z_* - z_i\| \ge \sum_{i=1}^{n} \|z_i\| + (2|S| - n)\|z_*\| - 2|S|r.$$

Since z_* is a $(1+\gamma)$-approximate solution of $\sum_{i=1}^n \|z - z_i\|$, it follows that

$$\sum_{i=1}^{n} \|z_i\| + (2|S| - n)\|z_*\| - 2|S|r \le (1+\gamma) \min_z \|z - z_i\|.$$

Note that $\sum_{i=1}^n \|z_i\| = \sum_{i=1}^n \|0 - z_i\| \ge \min_z \sum_{i=1}^n \|z - z_i\|$. Hence, it further implies that

$$(2|S| - n)\|z_*\| - 2|S|r \le \gamma \min_z \sum_{i=1}^{n} \|z - z_i\|,$$

and thus

$$\|z_*\| \le \frac{2|S|r}{2|S|-n} + \frac{\gamma \min_z \sum_{i=1}^n \|z - z_i\|}{2|S|-n} \le \frac{2(1-\alpha)r}{1-2\alpha} + \frac{\gamma \min_z \sum_{i=1}^n \|z - z_i\|}{(1-2\alpha)n},$$

where the last inequality holds due to $|S| \ge (1-\alpha)n$ by the assumption. □

B Proof of Lemma 3.2

Proof. By (13) and the fact that $\nabla F(\theta^*) = \mathbf{0}$, we have

$$\begin{aligned}
\|\theta' - \theta^*\|^2 &= \|\theta - \theta^* - \eta \nabla F(\theta)\|^2 \\
&= \|\theta - \theta^* - \eta \left(\nabla F(\theta) - \nabla F(\theta^*)\right)\|^2 \\
&= \|\theta - \theta^*\|^2 + \eta^2 \|\nabla F(\theta) - \nabla F(\theta^*)\|^2 - 2\eta \left\langle \theta - \theta^*, \nabla F(\theta) - \nabla F(\theta^*) \right\rangle.
\end{aligned}$$

By Assumption 1, we have

$$\|\nabla F(\theta) - \nabla F(\theta^*)\| \le M \|\theta - \theta^*\|,$$

$$F(\theta) \ge F(\theta^*) + \langle \nabla F(\theta^*), \theta - \theta^* \rangle + \frac{L}{2}\|\theta - \theta^*\|^2,$$

and

$$F(\theta^*) \ge F(\theta) + \langle \nabla F(\theta), \theta^* - \theta \rangle.$$

Summing up the last two displayed equations yields that

$$0 \ge \langle \nabla F(\theta) - \nabla F(\theta^*), \theta^* - \theta \rangle + \frac{L}{2}\|\theta - \theta^*\|^2.$$

Therefore,

$$\|\theta' - \theta^*\|^2 \le \left(1 + \eta^2 M^2 - \eta L\right) \|\theta - \theta^*\|^2.$$

The conclusion follows by the choosing $\eta = L/2M^2$. □

C Concentration Inequality for Sub-exponential Random Variables

Definition 1 (Sub-exponential)**.** Random variable X with mean μ is sub-exponential if $\exists\ \nu > 0$ and $\alpha > 0$ such that

$$\mathbb{E}\left[\exp\left(\lambda(X - \mu)\right)\right] \le \exp\left(\frac{\nu^2\lambda^2}{2}\right), \quad \forall |\lambda| \le \frac{1}{\alpha}.$$

Theorem 6. *If $X_1, \ldots, X_n$ are independent random variables where X_i's are sub-exponential with scaling parameters (ν_i, α_i) and mean μ_i, then $\sum_{i=1}^n X_i$ is sub-exponential with scaling parameters (ν_*, α_*), where $\nu_*^2 = \sum_{i=1}^n \nu_i^2$ and $\alpha_* = max_{1 \le i \le n} \alpha_i$. Moreover,*

$$\mathbb{P}\left\{\sum_{i=1}^n (X_i - \mu_i) \ge t\right\} \le \begin{cases} \exp\left(-t^2/(2\nu_*^2)\right) & \text{if } 0 \le t \le \nu_*^2/\alpha_* \\ \exp\left(-t/(2\alpha_*)\right) & o.w. \end{cases}$$

Acknowledgement

Y. Chen was partially supported by the National Science Foundation under CRII award 1657420 and grant CCF-1704828, and by the School of Operations Research and Information Engineering at Cornell University. L. Su was partially supported by the National Science Foundation Grant ECCS-1610543 and NSF Science & Technology Center for Science of Information Grant CCF-0939370.

References

[AS00] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. *SIGMOD Rec.*, 29(2):439–450, May 2000. 2

[BMGS17] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Byzantine-tolerant machine learning. *arXiv preprint arXiv:1703.02757*, 2017. 23

[BPC+11] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011. 1, 5

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 5

[CCZ+13] Hervé Cardot, Peggy Cénac, Pierre-André Zitt, et al. Efficient and fast estimation of the geometric median in hilbert spaces with an averaged stochastic gradient algorithm. *Bernoulli*, 19(1):18–43, 2013. 7

[CLM+16] Michael B Cohen, Yin Tat Lee, Gary Miller, Jakub Pachocki, and Aaron Sidford. Geometric median in nearly linear time. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, pages 9–21. ACM, 2016. 7, 10

[DG08] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008. 1

[DKK+16] Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 655–664. IEEE, 2016. 23

[DWJ13] John Duchi, Martin J Wainwright, and Michael I Jordan. Local privacy and minimax bounds: Sharp rates for probability estimation. In *Advances in Neural Information Processing Systems*, pages 1529–1537, 2013. 2

[FXM14] Jiashi Feng, Huan Xu, and Shie Mannor. Distributed robust learning. *arXiv preprint arXiv:1409.5937*, 2014. 23

[JLY16] Michael I Jordan, Jason D Lee, and Yun Yang. Communication-efficient distributed statistical inference. *arXiv preprint arXiv:1605.07689*, 2016. 1

[Kem87] JHB Kemperman. The median of a finite measure on a banach space. *Statistical data analysis based on the L1-norm and related methods (Neuchâtel, 1987)*, pages 217–230, 1987. 7

[KMR15] Jakub Konečnỳ, Brendan McMahan, and Daniel Ramage. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015. 2

[KPS02] Charlie Kaufman, Radia Perlman, and Mike Speciner. *Network security: private communication in a public world*. Prentice Hall Press, 2002. 1

[LBG+12] Yucheng Low, Danny Bickson, Joseph Gonzalez, Carlos Guestrin, Aapo Kyrola, and Joseph M Hellerstein. Distributed graphlab: a framework for machine learning and data mining in the cloud. *Proceedings of the VLDB Endowment*, 5(8):716–727, 2012. 1

[LR91] Hendrik P Lopuhaa and Peter J Rousseeuw. Breakdown points of affine equivariant estimators of multivariate location and covariance matrices. *The Annals of Statistics*, pages 229–248, 1991. 23

[LRV16] Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 665–674. IEEE, 2016. 23

[Lyn96] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996. 3

[M+15] Stanislav Minsker et al. Geometric median and robust estimation in banach spaces. *Bernoulli*, 21(4):2308–2335, 2015. 7, 23

[MBM16] Song Mei, Yu Bai, and Andrea Montanari. The landscape of empirical risk for non-convex losses. *arXiv preprint arXiv:1607.06534*, 2016. 23

[MD+87] P Milasevic, GR Ducharme, et al. Uniqueness of the spatial median. *The Annals of Statistics*, 15(3):1332–1333, 1987. 7

[MNO+10] Jyrki Möttönen, Klaus Nordhausen, Hannu Oja, et al. Asymptotic theory of the spatial median. In *Nonparametrics and Robustness in Modern Statistical Inference and Time Series Analysis: A Festschrift in honor of Professor Jana Jurečková*, pages 182–193. Institute of Mathematical Statistics, 2010. 7

[MNSJ15] Philipp Moritz, Robert Nishihara, Ion Stoica, and Michael I Jordan. Sparknet: Training deep networks in spark. *arXiv preprint arXiv:1511.06051*, 2015. 1

[MR10] Brendan McMahan and Daniel Ramage. Federated learning: Collaborative machine learning without centralized training data. *https://research.googleblog.com/2017/04/federated-learning-collaborative.html*, Accessed: 2017-04-10. 2, 4

[PH96] Foster J Provost and Daniel N Hennessy. Scaling up: Distributed machine learning with cooperation. In *AAAI/IAAI, Vol. 1*, pages 74–79. Citeseer, 1996. 1

[PP02] Charles P. Pfleeger and Shari Lawrence Pfleeger. *Security in Computing*. Prentice Hall Professional Technical Reference, 3rd edition, 2002. 1

[SB98] Hanif D Sherali and Dimitri P Bertsekas. Network optimization: Continuous and discrete models, 1998. 6

[Ver10] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. *Arxiv preprint arxiv:1011.3027*, 2010. 13, 14, 16, 21

[Wu17] Yihong Wu. Lecture Notes on Information-theoretic Methods For High-dimensional Statistics. http://www.stat.yale.edu/ yw562/teaching/it-stats.pdf, April 2017. 5

[WWRL10] Cong Wang, Qian Wang, Kui Ren, and Wenjing Lou. Privacy-preserving public auditing for data storage security in cloud computing. In *Infocom, 2010 proceedings ieee*, pages 1–9. Ieee, 2010. 1

[ZDW13] Yuchen Zhang, John C. Duchi, and Martin J. Wainwright. Communication-efficient algorithms for statistical optimization. *Journal of Machine Learning Research*, 14:3321–3363, 2013. 4

[ZDW15] Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *J. Mach. Learn. Res*, 16:3299–3340, 2015. 4