Securing Distributed Machine Learning in High Dimensions

Lili Su
MIT
lilisu@mit.edu

Jiaming Xu
Purdue University
xu972@purdue.edu

April 27, 2018

Abstract

We consider securing a distributed machine learning system wherein the data is kept confidential by its providers who are recruited as workers to help the learner to train a d–dimensional model. In each communication round, up to q out of the m workers suffer Byzantine faults; faulty workers are assumed to have complete knowledge of the system and can collude to behave arbitrarily adversarially against the learner. We assume that each worker keeps a local sample of size n. (Thus, the total number of data points is $N = nm$.) Of particular interest is the high-dimensional regime $d \gg n$.

We propose a secured variant of the classical gradient descent method which can tolerate up to a constant fraction of Byzantine workers. We show that the estimation error of the iterates converges to an estimation error $O(\sqrt{q/N}+\sqrt{d/N})$ in $O(\log N)$ rounds. The core of our method is a robust gradient aggregator based on the iterative filtering algorithm proposed by Steinhardt et al. [SCV18] for robust mean estimation. We establish a uniform concentration of the sample covariance matrix of gradients, and show that the aggregated gradient, as a function of model parameter, converges uniformly to the true gradient function. As a by-product, we develop a new concentration inequality for sample covariance matrices of sub-exponential distributions, which might be of independent interest.

1 Introduction

Many effective and efficient distributed machine learning algorithms [BPC+11, JLY16] and system [MNSJ15, PH96, DG08, LBG+12] have been proposed and implemented. In typical learning frameworks, data is collected from its providers (who might or might not be voluntary) and stored by the learner. Such data collection immediately leads to significant privacy risk and users' privacy concerns. These privacy concerns root in not only pure psychological reasons but also the poor real-world practice of privacy-preserving solutions. In fact, privacy breaches occur frequently, with recent examples including Facebook data leak scandal, iCloud leaks of celebrity photos, and PRISM surveillance program. Putting this privacy risk aside, the data providers often benefit from the learning outputs. For example, in medical applications, although a participant may be embarrassed about his use of drugs, he might benefit from a good learning output that can provide high-accuracy predictions of developing cancer [DJW14].

To resolve this dilemma of data providers, an alternative learning framework has been proposed wherein the training data is kept confidential by its providers from the learner and the providers function as workers [KMR15, fed, DJW14]. We refer to this framework as *learning with external workers*. This framework arises in many practical systems such as Google's *Federated Learning* system [KMR15, fed], wherein Google tries to learn a model with the training data kept confidential on the users' mobile devices. In contrast to the traditional learning framework under which models

are trained within datacenters, in *learning with external workers* the learner faces serious *security* risk. In particular, on the one hand, some external workers may be highly unreliable or even be adversarial. For example, in Google's Federated Learning, the users' mobile devices may not be always reliable and can be easily compromised by hackers. On the other hand, the leaner lacks enough administrative power over those external workers and thus is vulnerable to adversarial attacks. What's worse, each worker/data provider typically has access to a sample of small size comparing to the dimension of the model to train, as is often the case in Federated Learning. Two immediate consequences are: (1) to learn an accurate model, the learner has to interact closely with those external workers, and such close interaction gives the adversary more chances to foil the learning process; (2) identifying the adversarial workers based on abnormality is highly challenging, because it is difficult to distinguish the statistical errors from the adversarial errors due to small sample sizes.

In this paper, we aim to develop strategies to safeguard distributed machine learning against adversarial workers while keeping the following practical constraints in mind: [1]

- Small local samples versus high model dimensions: While the total volume of data over all workers may be large, individual workers may keep only small samples comparing to model dimensions.
- Communication constraints: Transmission between the external workers and the learner typically suffers from high latency and low-throughout. Thus communication between them is a scarce resource. In fact, communication cost is often argued to be the principal constraint in Federated Learning [MMR+16, KMY+16, KMR15].

1.1 System Model

To formally study security issues in distributed maching learning, we consider a setup proposed in [CSX17]. Specifically, let X denote the input data generated from some *unknown* distribution P. Let $\Theta \subset \mathbb{R}^d$ denote the set of all possible model parameters. We consider a risk 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 of the above statistical learning framework is linear regression, where $x = (w, y) \in \mathbb{R}^{d-1} \times \mathbb{R}$ is the feature-response pair and $f(x, \theta) = \frac{1}{2}(\langle w, \theta \rangle - y)^2$. The learner is interested in learning the model parameter choice θ^* which minimizes the *population risk*, i.e.,

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

– assuming that $\mathbb{E}[f(X, \theta)]$ is well-defined. The model choice θ^* is optimal in that it minimizes the expected risk to pay if it is used for prediction in the future over a freshly drawn data.

If P–the underlying distribution of X–were known, then the population risk might be evaluated directly, and θ^* might be computed by solving the minimization problem in (1). Unfortunately, in statistical learning, the distribution P is typically unknown; instead, training data is available for the learner to learn/estimate θ^*. Formally, we assume that there exist N i.i.d. data points $X_i \overset{\text{i.i.d.}}{\sim} P$ in the learning system. In this paper, we consider the distributed/decentralized learning system wherein the training data is kept locally by data providers and cannot be accessed by the learner directly. The learner can only request those providers to compute gradient-like quantities of the locally kept data, as is the case in Federated Learning. We refer to those data providers as workers

[1] Depending on the detailed applications, there might be many other constraints such as unevenly distributed training data, intermittent availability of mobile phones, etc.

as they can be viewed as workers that are "recruited" by the learner. We assume there are m workers, and the N data points are distributed evenly across the m workers. Specifically, the index set $[N]$ is partitioned into m subsets S_j such that $|\mathcal{S}_j| = N/m \triangleq n$, and $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ for $i \neq j$. In the example of Federated Learning, Google is the learner and users' mobile devices, such as smart phones, are the workers. Notably, the local data volume n is often much smaller than the model dimension d, which we refer to as the *high-dimensional regime*.

We further assume the learner communicates with the workers in synchronous communication rounds, while communication among non-faulty workers is not allowed. We leave the asynchronous communication as one future direction. We consider the practical learning scenario wherein the communication rounds are costly, as is the case in Federated Learning.

To capture the unreliability and potential malicious behaviors of the workers, we use the Byzantine fault model [Lyn96]. In particular, we assume that among the m workers, up to q of them might suffer Byzantine faults and thus behave arbitrarily and possibly maliciously. This arbitrarily faulty behavior arises when the workers are reprogrammed and completely controlled by the adversary. We refer to those faulty workers as Byzantine workers. We assume the learner knows the upper bound q – a standard assumption in literature [DKK+17, CSV17, SCV18]. Nevertheless, an effective and efficient learning algorithm that does not call for the knowledge of q as input is highly desired. The set of Byzantine workers is allowed to *change* between communication rounds; the adversary can choose different sets of workers to control across communication rounds. Byzantine workers are assumed to have *complete knowledge* of the system, including the total number of workers m, all N data points over the whole system, the programs that the workers are supposed to run, the program run by the learner, and the realization of the random bits generated by the learner. Moreover, Byzantine workers can collude. We assume that when the adversary gives up the control of a worker, this worker recovers and becomes normal immediately. This implies that Byzantine faults can only cause a worker to lie about the local data to the learner but cannot corrupt the local data. It is easy to see that the Byzantine fault model covers the data-poisoning attacks as a speical case.

As can be seen later, the (mobile) Byzantine faults creates unspecified dependency across communication rounds — a key challenge in our algorithm design and convergence analysis.

1.2 Related Work

There is a recent flurry of work on securing distributed machine learning algorithms against adversarial attacks [CSX17, CSV17, FXM14, SV16, DKK+17, DKK+16a, BMGS17, AAZL18, YCRB18], among which the most related work is [CSX17, CSV17, FXM14, SV16, BMGS17, AAZL18, YCRB18]. Both [SV16, BMGS17] considered a pure optimization framework and characterizations of statistical performance of the learning outputs are left open; whereas [CSX17, AAZL18, YCRB18] studied the same statistical learning framework as we do in this paper.

Among the proposed algorithms, the robust one-shot aggregation algorithm [FXM14] uses only one round communication and hence is communication-efficient. Specifically, in robust one-shot aggregation algorithm, each worker trains a preliminary model based on local sample and reports it to the learner; then the learner robustly aggregates these preliminary models to obtain a final model. However, the correctness of this algorithm relies crucially on the assumption that the local sample size satisfies $n = N/m \gg d$, so that a preliminary model is sufficiently close to the true model, which excludes its applications to the high-dimensional regime.

There have been attempts to robustify stochastic gradient descent (SGD) against adversarial workers [BMGS17, AAZL18]. In every iteration/round of SGD, each worker computes a gradient using only *a single fresh* random data point. Even when the population risk function F is strongly

convex, the mean squared error of SGD is only $O(1/t)$ with t iterations [NJLS09]. Therefore, SGD requires a large number of communication rounds to achieve a small estimation error and at every communication round, a d-dimensional gradient needs to be transmitted from every worker to the learner. This is undesirable for many applications where the model dimensions are high and communication cost is the principal constraint [KMY+16, MMR+16].

In constrast to SGD, under full gradient descent, each worker computes the gradient based on all locally available n data samples. When the population risk function F is strongly convex, full gradient descent converges exponentially fast, and hence requires only a few communication rounds. This makes full gradient descent approach particularly appealing in the high-dimensional regime. The previous work [CSX17] proposed a computationally-efficient algorithm – *Byzantine-resilient gradient descent* (BGD) method, where in each round the learner computes the *geometric median of means* of the gradients reported by the workers. It was shown that BGD converges in logarithmic rounds with an estimation error on the order of $\sqrt{dq/N}$ for $q \geq 1$, which is larger than minimax-optimal error rate $\sqrt{d/N}$ in the failure-free setting by at most a factor of $\sqrt{q}$. In fact, *geometric median* is shown to be fundamentally suboptimal in high dimension [DKK+16b]. In the low dimensional regime where $d = O(1)$, more recent work [YCRB18] obtains an order-optimal error rate based on coordinate-wise median and trimmed mean, but the dependency of error rate on dimension d is still highly suboptimal.

1.3 Contributions

In this work, we propose a robust version of full gradient descent method that tolerates up to a constant fraction of adversarial workers and converges to a statistical estimation error on the order of $O(\sqrt{q/N} + \sqrt{d/N})$ with $O(\log N)$ communication rounds. We adapt the iterative filtering approach originally proposed for robust mean estimation problems [SCV18] to our distributed statistical learning setting. In particular, in each round, we use this iterative filtering approach to robustly aggregate the gradients reported by the external workers.

On the technical front, in contrast to robust mean estimation, in robust gradient descent, we need to deal with the interplay of the randomness of the data and the iterative updates of θ: the iterates and the associated gradients are highly dependent on each other [2]. What's worse, due to the fact that the Byzantine workers can behave arbitrarily, the aforementioned dependency can not even be specified. To handle this, we first establish the concentration of sample covariance matrix of gradients uniformly at all possible model parameters. Then we further prove that our aggregated gradient, as function of θ, converges uniformly to the population gradient function $\nabla F(\cdot)$.

Similar uniform convergence concentration of sample covariance matrix has been derived in [CSV17, Lemma 2.1] under the assumption that the gradients are sub-gaussian. However, it turns out that even in the simplest linear regression example, the gradients are sub-exponential instead of sub-gaussian. To this end, we develop a new concentration inequality for sample covariance matrices of sub-exponential distributions, which might be of independent interest.

2 Our Robust Gradient Descent Approach and Main Results

Recall that our goal is to learn the optimal model choice θ^* defined in (1), i.e., $\theta^* \in \arg\min_{\theta \in \Theta} F(\theta)$, where $F(\theta) \triangleq \mathbb{E}[f(X, \theta)]$ for all $\theta \in \Theta$. A standard approach to estimate θ^* in statistical learning

[2] For online learning with fresh data drawn at each iteration, the gradients computed in a given iteration are independent from the gradients previously computed conditioning on the current iterate. Thus, securing the gradient descent is reduced to the robustly estimating the mean from i.i.d. samples [DKK+17, DKK+16b, CSV17, SCV18].

is via empirical risk minimization. Given N independent copies $X_1, \cdots, X_N$ of X, the empirical risk function is a random function over Θ defined as

$$\frac{1}{N}\sum_{i=1}^{N} f(X_i, \theta), \quad \forall \theta \in \Theta.$$

By the functional law of large numbers, the empirical risk function converges uniformly to the population risk function in probability, i.e., $\sup_{\theta\in\Theta}\left|\frac{1}{N}\sum_{i=1}^{N} f(X_i,\theta) - F(\theta)\right|$ converges to 0 in probability as sample size $N \to \infty$. As a consequence, we expect the minimizer of the empirical risk function (which is random) also converges to the population risk minimizer θ^* in probability. Unfortunately, even a single Byzantine worker can completely skew the empirical risk function and thus foils the whole empirical risk minimization approach [SV16]. While it may be possible to secure the empirical risk minimization using some "robust" versions of empirical risk functions [SV16, CSV17], the characterizations of the estimation error are either unavailable or too loose. Moreover, in our distributed settting, it is costly to transmit the local emprical risk function from every worker to the learner.

In this paper, we take a different approach: Instead of robustifying the empirical risk functions, we aim at robustifying the *learning process*. Specifically, we focus on securing the gradient descent method against the interruption caused by the Byzantine workers during model training. As commented in the introduction, due to the fact that (1) the local sample sizes are typically small, and that (2) the communication is a scarce resource (in terms of rounds), we consider full gradient descent – having each non-faulty worker compute gradient based on the entire local sample.

2.1 Approximate Gradient Descent Method

To motivate our *Approximate Gradient Descent Method*, let's first consider the ideal case where the population risk function F is known. In this case, the learner is able to perform 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, and θ_0 is the given initial guess of θ^*. It is well-known [BV04] that θ_t converges to θ^* exponentially fast under the following standard assumption.

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

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

and

$$\left\|\nabla F(\theta) - \nabla F(\theta')\right\| \leq L\left\|\theta - \theta'\right\|.$$

Note that both M and L may scale in d – the dimension of θ. Under Assumption 1, choosing the stepsize η in (2) to be $M/(2L^2)$, it holds that

$$\|\theta_t - \theta^*\| \leq \left(1 - \left(\frac{M}{2L}\right)^2\right)^{t/2} \|\theta_0 - \theta^*\|. \tag{3}$$

Thus $\|\theta_t - \theta^*\|$ converges to 0 exponentially fast as long as $M/(2L)$ is bounded away from 1. Exponential convergence (or even faster convergence) is highly desired because the communication between the learner and the workers is costly.

Now consider the more practical case where the population risk F is unknown. In this case, the population gradient $\nabla F(\theta)$ used in the standard gradient descent update (2) is unavailable. Nevertheless, if the learner, by interacting with the workers (among whom up to q workers may be Byzantine), is able to obtain good estimates of the population gradients $\nabla F(\theta)$, the learner may learn θ^* exponentially fast as well. In particular, if the learner could have access to an approximate gradient function $G(\theta) : \Theta \to \mathbb{R}^d$ such that

$$\|G(\theta) - \nabla F(\theta)\| \leq \xi_1 + \xi_2 \|\theta - \theta^*\|, \quad \forall \theta \in \Theta, \tag{4}$$

where $\xi_1 > 0$ and $\xi_2 > 0$ are two positive precision parameters that are independent of θ, and run the following *approximate* gradient descent:

$$\theta_t = \theta_{t-1} - \eta \times G(\theta_{t-1}), \tag{5}$$

where $\eta = M/(2L^2)$, it follows immediately from (3) and the triangle inequality that

$$\|\theta_t - \theta^*\| \leq \left(\sqrt{1 - M^2/(4L^2)} + M/(2L^2)\xi_2\right) \|\theta_{t-1} - \theta^*\| + M/(2L^2)\xi_1. \tag{6}$$

Hence, as long as $\sqrt{1 - M^2/(4L^2)} + M/(2L^2)\xi_2$ is bounded away from 1, $\|\theta_t - \theta^*\|$ converges to $M/(2L^2)\xi_1$ exponentially fast. See [CSX17] for details. Therefore, to learn θ^*, it remains for the learner to compute an approximate gradient function $G : \Theta \to \mathbb{R}^d$ with small ξ_1 and ξ_2, despite the fact that the training data is kept collectively by the workers among whom up to q of them may be Byzantine.

Algorithm 1 Approximate Gradient Descent Method: Round $t \geq 1$

The learner:

1: *Initialization:* Let θ_0 be an arbitrary point in Θ. Let $\eta = \frac{M}{2L^2}$.
2: Broadcast the current model iterate θ_{t-1} to all workers;
3: Wait to receive all the gradients reported by the m workers; Let $g_j(\theta_{t-1})$ denote the value received from worker j.
If no message from worker j is received, set $g_j(\theta_{t-1})$ to be some arbitrary value;
4: Aggregate gradients: Pass the received gradients to a gradient aggregator $\mathcal{R}$ to obtain an aggregated gradient $G(\theta_{t-1})$, i.e.,

$$G(\theta_{t-1}) \leftarrow \mathcal{R}(g_1(\theta_{t-1}), \cdots, g_j(\theta_{t-1}), \cdots, g_m(\theta_{t-1})). \tag{7}$$

5: Update: $\theta_t \leftarrow \theta_{t-1} - \eta \times G(\theta_{t-1})$;

Worker j:

1: On receipt of θ_{t-1}, compute the gradient at θ_{t-1}, i.e., $\frac{1}{n}\sum_{i \in \mathcal{S}_j} \nabla f(X_i, \theta_{t-1})$;
2: Send $\frac{1}{n}\sum_{i \in \mathcal{S}_j} \nabla f(X_i, \theta_{t-1})$ back to the learner;

Next we describe our approximated (full) gradient descent method given by Algorithm 1. In Algorithm 1, θ_0 is initialized to be an arbitrary point in the parameter space Θ. In each round $t \geq 1$, the learner broadcasts θ_{t-1} to all workers. If worker j is non-faulty at round t, on receipt of θ_{t-1}, it computes its local gradient $\frac{1}{n}\sum_{i\in\mathcal{S}_j} \nabla f(X_i, \theta_{t-1})$ at θ_{t-1} and reports the computed gradient to the learner; if worker j is Byzantine faulty at round t, instead of reporting the true local gradient $\frac{1}{n}\sum_{i\in\mathcal{S}_j} \nabla f(X_i, \theta_{t-1})$, it may report arbitrary value to the learner. Formally,

$$g_j(\theta_{t-1}) = \begin{cases} \frac{1}{n}\sum_{i\in\mathcal{S}_j} \nabla f(X_i, \theta_{t-1}), & \text{if worker } j \text{ is non-faulty at round } t; \\ \star, & \text{otherwise.} \end{cases}$$

where $\star$ denotes arbitrary value. In addition, the Byzantine workers are allowed to use all the information of the system to determine the value to report. In line 4 of Algorithm 1, the learner aggregates the received gradients via a gradient aggregator $\mathcal{R}$ (an algorithmic function) to obtain an approximate gradient $G(\theta_{t-1})$ at θ_{t-1}. In line 5, the learner updates θ_t using a gradient descent step. Since the system is synchronous, the learner can proceed from line 3 to lines 4 and 5.

We will specify our choice of gradient aggregator $\mathcal{R}$ in the next subsection. Before that, we provide a remark on the vulnerability of the standard "averaging" for gradient aggregation.

Remark 1. In the traditional failure-free setting, the learner can simply take the average of the received gradients:

$$G(\theta_{t-1}) = \frac{1}{m}\sum_{j=1}^{m} \frac{1}{n}\sum_{i\in\mathcal{S}_j} \nabla f(X_i, \theta_{t-1}) = \frac{1}{N}\sum_{i=1}^{N} \nabla f(X_i, \theta_{t-1}).$$

By the functional law of large numbers, as $N \to \infty$,

$$\sup_{\theta} \left| \frac{1}{N}\sum_{i=1}^{N} \nabla f(X_i, \theta_{t-1}) - \nabla F(\theta) \right| \to 0 \quad \text{in probability.}$$

Hence, by the property (6) of approximate gradient descent, θ_t converges to θ^* exponentially fast in probability. Unfortunately, averaging is not resilient to Byzantine fault. In fact, even a single Byzantine worker can completely skew the average of the local gradients and thus foils the whole gradient descent algorithm. Neither the average over a randomly selected subset of received gradients is resilient to a single Byzantine fault. This is because Byzantine workers have complete information of the system, including the gradients reported by other workers and the realization of the random bits generated by the learner.

To secure the distributed gradient descent against Byzantine faults, the previous work [CSX17] chooses the gradient aggregator $\mathcal{R}$ to be the *geometric median of means*, which is known to be computationally efficient. The estimation error of the obtained learning algorithm is on the order of $\sqrt{dq/N}$ (where $q \geq 1$), which is larger than the minimax-optimal error rate $\sqrt{d/N}$ in the failure-free setting by at most a factor of $\sqrt{q}$. In the high dimensional regime, the multiplicative $\sqrt{q}$ factor is highly suboptimal. To illustrate this, suppose each worker has only one data point, i.e., $n = 1$. To guarantee a small estimation error, the fraction of Byzantine machines q/m needs to be much smaller than $1/d$.

Next, we present a gradient aggregator which is originally proposed for the problem of robust mean estimation [SCV18]. The existing performance guarantees and analysis are far from being applicable to our statistical learning problems directly. The reason is two-fold: (1) the literature on robust mean estimation [SCV18] often focuses on $q/m = \Theta(1)$, whereas we are mainly interested in $q/m = O(1)$; (2) for robust mean estimation [SCV18], only a concentration of good random data points is needed, whereas in our setting, we need to establish a concentration of random functions.

2.2 Robust Gradient Aggregator

In this subsection, we present the robust gradient aggregator $\mathcal{R}$ that will be used in Algorithm 1. Robust gradient aggregation is closely related to robust mean estimation, formally stated next.

Definition 1 (Robust mean estimation)**.** Let $\mathcal{S} = \{y_1, \cdots, y_m\}$ be a sample of size m, wherein each of the data point y_i is generated independently from an unknown distribution. Among those m data points, up to q of them may be adversarially corrupted. Let $\{\widehat{y}_1, \cdots, \widehat{y}_m\}$ be the observed sample. The goal is to estimate the true mean of the unknown distribution when only corrupted sample $\{\widehat{y}_1, \cdots, \widehat{y}_m\}$ is accessible.

A very interesting recent line of work on robust mean estimation [DKK+16a, DKK+17, SCV18] discovers computationally efficient iterative filtering approaches that can achieve a bounded estimation error as long as at most a *constant* fraction ($q/m = \Theta(1)$) of data is corrupted, irrespective of dimension d. In this paper, we consider an iterative filtering algorithm proposed in [SCV18], formally presented in Algorithm 2. At a high level, Algorithm 2 iteratively finds a direction along which all data points are spread out the most, and filters away data points which have large residual errors projected along this direction, by solving (8) and (9) for a saddle point (W, U). See Appendix D for detailed discussions.

Algorithm 2 Iterative Filtering for Robust Mean Estimation [SCV18]

Input: Corrupted sample $\{\widehat{y}_1, \cdots, \widehat{y}_m\} \subseteq \mathbb{R}^d$, $1 - \alpha \triangleq \epsilon \in [0, \frac{1}{4})$, and $\sigma > 0$.
Initialization: $\mathcal{A} \leftarrow \{1, \cdots, m\}$, $c_i \leftarrow 1$ and $\tau_i \leftarrow 0$ for all $i \in \mathcal{A}$.

while true **do**

Let $W \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{A}|}$ be a minimizer to the following convex program:

$$\min_{\substack{0 \le W_{ji} \le \frac{4-\alpha}{\alpha(2+\alpha)m} \\ \sum_{j \in \mathcal{A}} W_{ji} = 1}} \quad \max_{\substack{U \succeq 0 \\ \mathsf{Tr}(U) \le 1}} \quad \sum_{i \in \mathcal{A}} c_i \left(\widehat{y}_i - \sum_{j \in \mathcal{A}} \widehat{y}_j W_{ji} \right)^\top U \left(\widehat{y}_i - \sum_{j \in \mathcal{A}} \widehat{y}_j W_{ji} \right), \tag{8}$$

and $U \in \mathbb{R}^{d \times d}$ be a maximizer to the following convex program:

$$\max_{\substack{U \succeq 0 \\ \mathsf{Tr}(U) \le 1}} \quad \min_{\substack{0 \le W_{ji} \le \frac{4-\alpha}{\alpha(2+\alpha)m} \\ \sum_{j \in \mathcal{A}} W_{ji} = 1}} \quad \sum_{i \in \mathcal{A}} c_i \left(\widehat{y}_i - \sum_{j \in \mathcal{A}} \widehat{y}_j W_{ji} \right)^\top U \left(\widehat{y}_i - \sum_{j \in \mathcal{A}} \widehat{y}_j W_{ji} \right). \tag{9}$$

For $i \in \mathcal{A}$, $\tau_i \leftarrow \left(\widehat{y}_i - \sum_{j \in \mathcal{A}} \widehat{y}_j W_{ji} \right)^\top U \left(\widehat{y}_i - \sum_{j \in \mathcal{A}} \widehat{y}_j W_{ji} \right)$

if $\sum_{i \in \mathcal{A}} c_i \tau_i > 8m\sigma^2$ **then**

For $i \in \mathcal{A}$, $c_i \leftarrow \left(1 - \frac{\tau_i}{\tau_{\max}}\right) c_i$, where $\tau_{\max} = \max_{i \in \mathcal{A}} \tau_i$.

$\mathcal{A} \leftarrow \mathcal{A} / \left\{ i : \ c_i \le \frac{1}{2} \right\}$.

else

Break **while**–loop.

end if

end while

return $\widehat{\mu} = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \widehat{y}_i$.

In each iteration in the **while**-loop, the column/row indices of $W \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{A}|}$ correspond to the data indices remained in $\mathcal{A}$, and at least one point is removed from set $\mathcal{A}$. Thus, this loop will be executed at most m times. Given the corrupted sample $\{\widehat{y}_1, \cdots, \widehat{y}_m\}$, ϵ, and σ, Algorithm 2 *deterministically* outputs an estimate $\widehat{\mu}$ differing from the true *sample mean* by at most a bounded distance, formally stated next.

Lemma 1. *[SCV18] Let $\mathcal{S} = \{y_1, \cdots, y_m\}$ be the true sample. Define $\mu_{\mathcal{S}} = \frac{1}{m}\sum_{i=1}^{m} y_i$ as the sample mean on $\mathcal{S}$. Let $\{\widehat{y}_1, \cdots, \widehat{y}_m\} \subseteq \mathbb{R}^d$ be the observed sample, which is obtained from $\mathcal{S}$ by adversarially corrupting up to q data points. Suppose that*

$$\left\| \frac{1}{m} \sum_{i\in\mathcal{S}} (y_i - \mu_{\mathcal{S}}) (y_i - \mu_{\mathcal{S}})^\top \right\| \leq \sigma^2. \tag{10}$$

Then for $q/m = \epsilon \leq \frac{1}{4}$, Algorithm 2 outputs a parameter $\widehat{\mu}$ such that

$$\|\widehat{\mu} - \mu_{\mathcal{S}}\| = O(\sigma\sqrt{\epsilon}). \tag{11}$$

Remark 2. Let μ be the true mean of the unknown underlying distribution. By triangle inequality,

$$\|\widehat{\mu} - \mu\| \leq \|\widehat{\mu} - \mu_{\mathcal{S}}\| + \|\mu_{\mathcal{S}} - \mu\| = O\left(\sigma\sqrt{\epsilon}\right) + \|\mu_{\mathcal{S}} - \mu\|.$$

Thus, to characterize $\|\widehat{\mu} - \mu\|$ – the estimation error of the output of $\widehat{\mu}$, it is enough to control the spectral norm $\left\|\frac{1}{m}\sum_{i\in\mathcal{S}} (y_i - \mu_{\mathcal{S}}) (y_i - \mu_{\mathcal{S}})^\top\right\|$ of covariance matrix of the *uncorrupted sample* and the deviation of the empirical average $\|\mu_{\mathcal{S}} - \mu\|$ – the latter of which is standard.

Remark 3. Note that

$$\begin{aligned}
\left\| \frac{1}{m} \sum_{i\in\mathcal{S}} (y_i - \mu_{\mathcal{S}}) (y_i - \mu_{\mathcal{S}})^\top \right\| &= \frac{1}{m} \left\| \left([y_1, \cdots, y_m] - \mu_{\mathcal{S}} \mathbf{1}_m^\top\right) \left([y_1, \cdots, y_m] - \mu_{\mathcal{S}} \mathbf{1}_m^T\right)^\top \right\| \\
&= \frac{1}{m} \left\| [y_1, \cdots, y_m] - \mu_{\mathcal{S}} \mathbf{1}_m^\top \right\|^2 \\
&\leq \frac{1}{m} \left(\left\| [y_1, \cdots, y_m] - \mu \mathbf{1}_m^\top \right\| + \sqrt{m}\, \|\mu - \mu_{\mathcal{S}}\| \right)^2,
\end{aligned}$$

where $\mathbf{1}_m \in \mathbb{R}^m$ is an all-ones vector. Therefore, to guarantee (10), it is enough to bound $\|\mu - \mu_{\mathcal{S}}\|$ and $\frac{1}{\sqrt{m}} \left\| [y_1, \cdots, y_m] - \mu \mathbf{1}_m^\top \right\|$.

Remark 4. The termination condition of Algorithm 2 needs to know σ. In the case where we do not know σ a priori, the termination condition for Algorithm 2 can be replaced by checking the cardinality of set $|\mathcal{A}|$ instead. In particular, we can replace the **if** condition in line 4 as

$$\textbf{if } \left| \mathcal{A} \setminus \left\{ i : \left(1 - \frac{\tau_i}{\tau_{\max}}\right) c_i \leq \frac{1}{2} \right\} \right| \geq \frac{\alpha(2+\alpha)m}{4-\alpha} \quad \textbf{then}$$

The correctness of this modification can be found in Appendix D.3.

Our statement of Lemma 1 is slightly different from [SCV18, Proposition 16]. The focus therein is on adversarial *data insertion*, where ϵm arbitrary data points are inserted to a sample of size $(1-\epsilon)m$ in $\mathbb{R}^d$ and the goal is to obtain a robust mean estimator with a constant error independent of the dimension d. In this case, it is enough to have *a* significantly large proportion of good data points which enjoys nice spectral concentration in the sense of (10). In contrast, here we are

dealing with more challenging *data corruption*, where an adversary can look up all m good data points in the original sample, and arbitrarily corrupt any ϵ fraction of them. More importantly, the concentration for the robust mean estimation is point-wise, and ϵ therein is implicitly assumed to be bounded away from 0, i.e., $\epsilon = \Theta(1)$. However, in general statistical learning problem, uniform concentration is typically involved and diminishing ϵ is of primary interest. Thus, in this paper, we require the whole set of good data points to have nice spectral concentration according to (10).

We present the proof of Lemma 1 in Appendix D for completeness.

2.3 Main Results

In this work, we use Algorithm 2 as our robust gradient aggregator $\mathcal{R}$, with inputs

$$\widehat{y}_1(\theta) = g_1(\theta), \ \cdots, \ \widehat{y}_m(\theta) = g_m(\theta),$$

where $g_1(\theta), \cdots, g_m(\theta)$ are the gradients reported by the m workers, among which up to q reported gradients may not be the true local gradients. The true m local gradients are

$$y_1(\theta) = \frac{1}{n}\sum_{i\in\mathcal{S}_1} \nabla f(X_i, \theta), \ \cdots, \ y_m(\theta) = \frac{1}{n}\sum_{i\in\mathcal{S}_m} \nabla f(X_i, \theta),$$

– recalling that $|\mathcal{S}_j| = n = \frac{N}{m}$ for $j \in [m]$. The sample mean $\mu_{\mathcal{S}}(\theta)$ is

$$\frac{1}{m}\sum_{j=1}^{m}\left(\frac{1}{n}\sum_{i\in\mathcal{S}_j}\nabla f(X_i,\theta)\right) = \frac{1}{N}\sum_{i=1}^{N}\nabla f(X_i,\theta),$$

and the population mean $\mathbb{E}[Y(\theta)]$ is $\nabla F(\theta)$. In view of Lemma 1, to guarantee that the aggregated gradient is close to the true gradient, we need to characterize the upper bound in (10) for all $\theta \in \Theta$. From Remarks 2 and 3, we know that to bound the spectral norm of the sample covariance matrix uniformly over $\theta \in \Theta$, it suffices to bound

$$\frac{1}{\sqrt{m}}\left\|\left[\frac{1}{n}\sum_{i\in\mathcal{S}_1}\nabla f(X_i,\theta) - \nabla F(\theta), \ \cdots, \frac{1}{n}\sum_{i\in\mathcal{S}_m}\nabla f(X_i,\theta) - \nabla F(\theta)\right]\right\| \tag{12}$$

and

$$\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f(X_i,\theta) - \nabla F(\theta)\right\| \tag{13}$$

uniformly over all $\theta \in \Theta$.

To this end, we need a set of standard technical assumptions, which are also made in [CSX17]. It can be checked that the standard linear regression satisfies these assumptions [CSX17]. Let $S^{d-1} = \{v \in \mathbb{R}^d : \|v\| = 1\}$ denote the unit Euclidean sphere.

Assumption 2. The sample gradient at the optimal model parameter θ^*, i.e., $\nabla f(X, \theta^*)$, is sub-exponential with constants (σ_1, α_1), i.e., for every unit vector $v \in S^{d-1}$,

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

We further assume the Lipschitz continuity of the sample gradient functions.

Assumption 3. There exists an L' such that

$$\left\|\nabla f(X,\theta) - \nabla f(X,\theta')\right\| \;\leq\; L' \left\|\theta - \theta'\right\| \quad \forall\, \theta, \theta' \in \Theta.$$

For applications where Assumption 3 does not hold deterministically, it suffices to have Assumption 3 hold with high probability for all training data. Notably, L' may be much larger than L and scale polynomially in N and d. However, L' affect our results only by logarithmic factors $\log L'$.

Next define the gradient difference function

$$h(X,\theta) \;=\; \nabla f(X,\theta) - \nabla f(X,\theta^*) - (\nabla F(\theta) - \nabla F(\theta^*))\,. \tag{14}$$

Note that $h(X,\theta)/\left\|\theta - \theta^*\right\|$ characterizes the change of $f(X,\theta) - \nabla F(\theta)$ from $f(X,\theta^*) - \nabla F(\theta^*)$ in terms of distance of θ from θ^*; hence it can viewed as a a local Lipschitz parameter with respect to θ^*.

Assumption 4. The local Lipschitz parameter $h(X,\theta)/\left\|\theta - \theta^*\right\|$ is sub-exponential with constants (σ_2, α_2), i.e.., for fixed θ and v,

$$\mathbb{E}\left[\exp\left(\frac{\lambda\langle h(X,\theta), v\rangle}{\|\theta - \theta^*\|}\right)\right] \leq e^{\sigma_2^2\lambda^2/2}, \quad \forall|\lambda| \leq \frac{1}{\alpha_2}.$$

Notably, Assumption 4 assumes a concentration of the *local* Lipschitz parameter with respect to θ^*, instead of a *global* Lipschitz parameter.

2.3.1 Standard Concentration Inequalities

For any fixed θ, the matrix in (12) is of independent columns. Standard routine to bound the spectral norm of (12) is available, see [Ver10, Theorem 5.44] and [ALPTJ10, Corollary 3.8] for example. To get a uniform concentration result, we can use ϵ–net argument to extend the concentration of a fixed θ to uniform over all $\theta \in \Theta$. Nevertheless, using the standard matrix concentration results, the concentration bound we get is far from being optimal.

The following theorem is a standard concentration inequality for matrices with sub-exponential columns [ALPTJ10, Corollary 3.8].

Theorem 1. *Let A be a $d \times m$ matrix whose columns A_j are i.i.d., zero-mean, sub-exponential random vectors in $\mathbb{R}^d$ with the scaling parameters σ and α. Assume that $\sigma, \alpha = O(1)$ and $m \leq e^{\sqrt{d}}$. There are absolute positive constants C and c such that for every $K \geq 1$, with probability at least $1 - e^{-cK\sqrt{d}}$,*

$$\|A\| \leq CK\left(\sqrt{m} + \sqrt{d}\right).$$

Note that assuming $m \leq e^{\sqrt{d}}$ only loses minimal generality in the high-dimensional regime. The above theorem is tight up to constant factors when the tail probability is on the order of $e^{-\sqrt{d}}$, i.e., when $K = \Theta(1)$ [ALPTJ10][Remark 3.7]. However, in our problem, to guarantee a uniform bound of the spectral norm of (12), we need a tail probability on the order of e^{-d}, i.e., $K \approx \sqrt{d}$. In this case, Theorem 1 yields an upper bound on the order of $\sqrt{md} + d$. Using [Ver10, Theorem 5.44], we can obtain an alternative upper bound $O(\sqrt{m} + d^{3/2})$. Both of these two upper bounds are not tight.

Next, we develop a new matrix concentration inequality, proving a nearly tight upper bound on the order of $\sqrt{m} + d$ up to logarithmic factors.

2.3.2 New Concentration Inequalities

A key step in deriving a concentration inequality for matrices with sub-exponential random vectors is to obtain a large deviation inequality for the sum of independent random variables whose tails decay *slower* than sub-exponential random variables. Note that in this case, the moment generating function may not exist and thus we cannot follow the standard approach to obtain a large deviation inequality by invoking the Chernoff bound. To circumvent this, we partition the support of a real-valued random variable Y into countably many finite segments, and write Y as a summation of component random variables, each of which is supported on its corresponding segment. Due to the fact that each segment is of finite length, we can apply the Bennett's inequality for bounded random variables (cf. Lemma 5). Then we take a union bound to arrive at a concentration result of the original Y. Some additional care is needed in choosing the partition. Our proof is inspired by Proposition 2.1.9 and Excercise 2.1.7 in [Tao12].

Lemma 2. *Let Y be a random variable whose tail probability satisfies*

$$\mathbb{P}\left\{|Y| \geq t\right\} \leq \exp\left(-E(t)\right),$$

where $E(t) : \mathbb{R}_+ \to [0, \infty]$ is a non-decreasing function. Suppose that there exists $t_0 \geq e^2$ such that for all $t \geq t_0$

$$E(e^{k-1}) \geq 2\left(2k + 4\log(k+1) + \log 2 - \log t\right), \quad \forall k \text{ with } 4(k+1)^2 e^k \geq t, \ \forall t \geq t_0, \tag{15}$$

and

$$E(t)/t \text{ is monotone in } t. \tag{16}$$

Let $Y_1, \cdots, Y_m$ be m independent copies of Y. Then if $E(t)/t$ non-decreasing

$$\mathbb{P}\left\{\left|\sum_{j=1}^m Y_j - m\mathbb{E}\left[Y\right]\right| \geq mt\right\} \leq 2\log(mt)\exp\left(-\frac{m}{4(\log(mt)+1)^2}E\left(\frac{t}{4e\log^2 t}\right)\right) + \exp\left(-\frac{1}{2}E\left(\frac{mt}{e}\right)\right); \tag{17}$$

if $E(t)/t$ is non-increasing

$$\mathbb{P}\left\{\left|\sum_{j=1}^m Y_j - m\mathbb{E}\left[Y\right]\right| \geq mt\right\} \leq 2\log(mt)\exp\left(-\frac{1}{4e(\log(mt)+1)^2}E\left(\frac{mt}{e}\right)\right) + \exp\left(-\frac{1}{2}E\left(\frac{mt}{e}\right)\right). \tag{18}$$

Remark 5. Let us consider the following special cases:

- Suppose Y is sub-Gaussian. In this case, $E(t) = ct^2$ for a universal constant $c > 0$. Thus, there exists a universal constant $t_0 \geq e^2$ such that both (15) and (16) hold. It follows from Lemma 2 that for all $t \geq t_0$,

  $$\mathbb{P}\left\{\left|\sum_{j=1}^m Y_j - m\mathbb{E}\left[Y\right]\right| \geq mt\right\} \leq 2\log(mt)\exp\left(-\frac{cmt^2}{64e^2\log^2(emt)\log^4 t}\right) + \exp\left(-\frac{cm^2t^2}{2e^2}\right),$$

  which gives the desired sub-Gaussian tail bound up to logarithmic factors.

- Suppose Y is sub-exponential. In this case, $E(t) = ct$ for a universal constant c. Thus, there exists $t_0 \geq e^2$ that only depends on c such that both (15) and (16) hold. It follows from Lemma 2 that for all $t \geq t_0$ (large deviation region),

$$\mathbb{P}\left\{\left|\sum_{j=1}^{m} Y_j - m\mathbb{E}\left[Y\right]\right| \geq mt\right\} \leq 2\log(mt)\exp\left(-\frac{cmt}{16e\log^2(emt)\log^2 t}\right) + \exp\left(-\frac{cmt}{2e}\right),$$

  which gives the desired sub-exponential tail bound up to logarithmic factors.

- Suppose $Y = Z^2$, where Z is sub-exponential. In this case, $E(t) = c\sqrt{t}$ for a universal constant $c > 0$. Thus, there exists $t_0 \geq e^2$ that only depends on c such that both (15) and (16) hold. It follows from Lemma 2 that for all $t \geq t_0$ (large deviation region),

$$\mathbb{P}\left\{\left|\sum_{j=1}^{m} Y_j - m\mathbb{E}\left[Y\right]\right| \geq mt\right\} \leq 2\log(mt)\exp\left(-\frac{c\sqrt{mt}}{4e\sqrt{e}\log^2(emt)}\right) + \exp\left(-\frac{c\sqrt{mt}}{2\sqrt{e}}\right).$$

Despite the fact that Lemma 2 is loose up to logarithmic factors comparing to the standard sub-gaussian and sub-exponential random variables, Lemma 2 applies to much larger family than the sub-gaussian distributions, and requires much less structures on the distributions. In particular, Lemma 2 does not even require the existence of moment generating function.

The proof of Lemma 2 can be found in Appendix B. Lemma 2 is our key machinery to obtain a concentration inequality for matrices with i.i.d. sub-exponential random vectors.

Theorem 2. *Let A be a $d \times m$ matrix whose columns A_j are independent and identically distributed sub-exponential, zero-mean random vectors in $\mathbb{R}^d$ with parameters (σ, α). Assume that*

$$\sigma/\alpha = \Omega(1). \tag{19}$$

Then with probability at least $1 - \delta$,

$$\|A\| \leq c\left(\sigma\sqrt{m} + \sigma\phi\left(d + \log\frac{1}{\delta}\right) + \alpha\phi^2\left(d + \log\frac{1}{\delta}\right)\right),$$

where c is a universal positive constant and $\phi(x) : \mathbb{R} \to \mathbb{R}$ is a function given by $\phi(x) = \sqrt{x}\log^{3/2}(x)$.

Remark 6. We discuss two consequences of Theorem 2.

- Suppose $\alpha = 0$. In this case, A has sub-Gaussian columns, and Theorem 2 implies that

$$\|A\| \lesssim \sigma\left(\sqrt{m} + \sqrt{d + \log\frac{1}{\delta}}\log^{3/2}\left(d + \log\frac{1}{\delta}\right)\right),$$

  which matches the sub-Gaussian matrix concentration inequality [Ver18][Theorem 5.39] up to logarithmic factors.

- Suppose $\sigma, \alpha = \Theta(1)$, and $\log(1/\delta) = d$. In this case, we get that with probability at least $1 - e^{-d}$,

$$\|A\| \lesssim \sqrt{m} + d\log^3 d \quad \text{implied by Theorem 2}, \tag{20}$$

whereas the analogous bound implied by Theorem 1 is on the order of $\sqrt{md}+d$. The upper bound (20) is tight up to logarithmic factors. To see this, consider an example, where A_j's are i.i.d. isotropic Laplace distribution with the density function given by $f(x) = \prod_{i=1}^{d}(1/\sqrt{2})\exp\left(-\sqrt{2}x_i\right)$ for $x \in \mathbb{R}^d$. In this case, note that

$$\left\{\|A\| \geq \max\{\sqrt{m/2}, d\}\right\} \supseteq \left\{|A_{11}| \geq d \text{ and } \sum_{j=1}^{m} A_{2j}^2 \geq m/2\right\}.$$

Since

$$\mathbb{P}\left\{|A_{11}| \geq d\right\} = \int_{|t|\geq d} \frac{1}{\sqrt{2}} \exp\left(-\sqrt{2}t\right) \mathrm{d}t = \exp\left(-\sqrt{2}d\right),$$

and by Chebyshev's inequality,

$$\mathbb{P}\left\{\sum_{j=1}^{m} A_{2j}^2 \geq m/2\right\} \geq 1 - O(1/m) \geq \frac{1}{2}$$

for m suffciently large, and A_{11} is independent of $\sum_{j=1}^{m} A_{2j}^2$, it follows that

$$\mathbb{P}\left\{\|A\| \geq \max\{\sqrt{m/2}, d\}\right\} \geq \mathbb{P}\left\{|A_{11}| \geq d \text{ and } \sum_{j=1}^{m} A_{2j}^2 \geq m/2\right\} \geq \frac{1}{2}\exp\left(-\sqrt{2}d\right).$$

2.3.3 Convergence of Approximate Gradient Descent Method

Recall from Subsection 2.1 that if the learner could have access to an approximate gradient function $G(\theta) : \Theta \to \mathbb{R}^d$ such that

$$\|G(\theta) - \nabla F(\theta)\| \leq \xi_1 + \xi_2 \|\theta - \theta^*\|, \quad \forall \theta \in \Theta,$$

for small enough $\xi_1 > 0$ and $\xi_2 > 0$ that are independent of θ, and run *approximate* gradient descent in (5), then $\|\theta_t - \theta^*\|$ converges to $M/(2L^2)\xi_1$ exponentially fast.

With Algorithm 2 as our robust gradient aggregator $\mathcal{R}$ in Algorithm 1, we show that the learner can indeed access those desired good approximate gradient.

Theorem 3. *Suppose Assumptions 1, 2, 3, and 4 hold. Assume that* $\log(L + L') = O(\log(Nd))$ *and* $\Theta \subset \{\theta : \|\theta - \theta^*\| \leq r\}$ *for some positive parameter* r *such that* $\log r = O(\log(Nd))$. *Suppose* $N \geq cd^2 \log^8(Nd))$ *for a sufficiently large constant* c *and* $m \leq e^{\sqrt{d}}$. *Let* $G(\theta)$ *(for each* $\theta \in \Theta$*) be the aggregated gradient returned by Algorithm 2. Then with probability at least* $1 - 2e^{-\sqrt{d}}$,

$$\|G(\theta) - \nabla F(\theta)\| \lesssim \left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}}\log^2(Nd)\right)\|\theta - \theta^*\| + \left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}}\right)$$

holds for all $\theta \in \Theta$.

Theorem 3 allows both the Lipschitz constant of the population gradient L and that of the sample gradient L' to scale in d and N. Theorem 3 also allows the diameter of Θ to scale in d and N. We postpone the proof of Theorem 3 to Section 3.

Remark 7. Theorem 3 requires the total sample size $N \gtrsim d^2$ (ignoring the logarithmic factors), which is due to our sub-exponential assumption 3 of local Lipscthiz parameter $h(X,\theta)/\|\theta-\theta^*\|$. This sample size requirement $N \gtrsim d^2$ is inevitable as can be seen from the linear regression example (cf. Remark 8). If instead $h(X,\theta)/\|\theta-\theta^*\|$ is assumed to be sub-Gaussian, then $N \gtrsim d$ suffices.

Theorem 4. *Consider the same setup as Theorem 3. Further assume that $N \geq c'q$ for a sufficiently large constant c', and $M \geq 1$. Then with probability at least $1-2e^{-\sqrt{d}}$,the iterates $\{\theta_t\}$ given by Algorithm 1 with $\eta = L/(2M^2)$ satisfy*

$$\|\theta_t - \theta^*\| \lesssim \left(1 - \frac{M^2}{16L^2}\right)^t \|\theta_0 - \theta^*\| + \left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}}\right).$$

Proof. From Theorem 3, we know that there a constant c_0 such that with probability at least $1-2e^{-\sqrt{d}}$, for all $\theta \in \Theta$,

$$\|G(\theta) - \nabla F(\theta)\| \leq c_0 \left(\left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}}\log^2(Nd)\right)\|\theta-\theta^*\| + \left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}}\right)\right).$$

Let

$$\rho \triangleq \sqrt{1 - \frac{M^2}{4L^2}} + c_0\frac{M}{2L^2}\left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}}\log^2(Nd)\right).$$

Recall that, by definition, $M \leq L$. For sufficiently large constants c' and c such that $N \geq c'q$, $N \geq cd^2\log^8(Nd)$, it holds that

$$c_0\frac{M}{2L^2}\left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}}\log^2(Nd)\right) \leq \frac{M}{16L^2}.$$

Consequently,

$$\rho \leq \sqrt{1 - \frac{M^2}{4L^2}} + \frac{M}{16L^2} \leq 1 - \frac{M^2}{8L^2} + \frac{M}{16L^2} \leq 1 - \frac{M^2}{16L^2},$$

where the last inequality follows from the assumption that $M \geq 1$. From (6), we have

$$\begin{aligned}\|\theta_t - \theta^*\| &\leq \rho^t \|\theta_0 - \theta^*\| + c_0\frac{M}{2L^2}\frac{1}{1-\rho}\left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}}\right)\\ &\leq \left(1 - \frac{M^2}{16L^2}\right)^t\|\theta_0 - \theta^*\| + 8c_0\left(\sqrt{\frac{q}{N}} + \sqrt{\frac{d}{N}}\right),\end{aligned}$$

proving Theorem 4. □

3 Proof of Theorem 3

Recall that we need to bound (12) and (13) uniformly for all $\theta \in \Theta$. Bounding (13) uniformly is relatively easy and has been done in previous work [CSX17, Proposition 3.8].

Proposition 1. *[CSX17, Proposition 3.8] Consider the same setup as Theorem 3. Assume that $N = \Omega(d\log(Nd))$. Then with probability at least $1 - e^{-d}$,*

$$\left\| \frac{1}{N}\sum_{i=1}^{N} \nabla f(X_i,\theta) - \nabla F(\theta) \right\| \lesssim \Delta_2 \|\theta - \theta^*\| + \Delta_1, \quad \forall\ \theta \in \Theta,$$

where

$$\Delta_1 \triangleq \sqrt{\frac{d}{N}}, \ \text{and} \quad \Delta_2 \triangleq \sqrt{\frac{d\log(Nd)}{N}}.$$

It remains to bound (12) uniformly over all $\theta \in \Theta$. For notational convenience, let

$$G(X_{\mathcal{S}},\theta) \triangleq \frac{1}{\sqrt{m}}\left[\tfrac{1}{n}\textstyle\sum_{i\in\mathcal{S}_1} \nabla f(X_i,\theta) - \nabla F(\theta), \ \cdots, \ \tfrac{1}{n}\sum_{i\in\mathcal{S}_m} \nabla f(X_i,\theta) - \nabla F(\theta)\right]. \tag{21}$$

Proposition 2. *Consider the same setup as Theorem 3. With probability at least $1 - 2e^{-\sqrt{d}}$,*

$$\|G(X_{\mathcal{S}},\theta^*)\| \lesssim \Delta_3 \quad \text{and} \quad \|G(X_{\mathcal{S}},\theta) - G(X_{\mathcal{S}},\theta^*)\| \lesssim \Delta_4 \|\theta - \theta^*\|, \ \forall \theta \in \Theta, \tag{22}$$

where

$$\begin{aligned}
\Delta_3 &\triangleq \frac{1}{\sqrt{n}} + \sqrt{\frac{d}{N}}, \\
\Delta_4 &\triangleq \frac{1}{\sqrt{n}} + \frac{1}{\sqrt{N}}\phi\left(d\log\left(r\sqrt{n}(L+L')\right)\right) + \frac{1}{\sqrt{Nn}}\phi^2\left(d\log\left(r\sqrt{n}(L+L')\right)\right).
\end{aligned}$$

It follows from triangle inequality that

$$\|G(X_{\mathcal{S}},\theta)\| \lesssim \Delta_4\|\theta - \theta^*\| + \Delta_3, \ \forall \theta \in \Theta.$$

Remark 8. The uniform upper bound Δ_4 in (22) depends linearly in d (ignoring logarithmic factors). Such linear dependency is inevitable as can be seen from the linear regression example. Suppose $y_i = \langle w_i, \theta^*\rangle + \zeta_i$, 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. The risk function $f(X_i,\theta) = \frac{1}{2}\left(\langle w_i,\theta\rangle - y_i\right)^2$. In this case $\nabla f(X_i,\theta) = w_i w_i^\top(\theta - \theta^*) - w_i\zeta_i$ and $\nabla F(\theta) = \theta - \theta^*$. For simplicity, assume $n = 1$ and $m = N$. Then

$$\begin{aligned}
\sup_{\theta\in S^{d-1}} \|G(X_{\mathcal{S}},\theta) - G(X_{\mathcal{S}},\theta^*)\| &\geq \sup_{\theta\in S^{d-1}} \frac{1}{\sqrt{N}}\left\|(w_1 w_1^\top - \mathbf{I})(\theta - \theta^*)\right\| \\
&= \frac{1}{\sqrt{N}}\left(\|w_1\|^2 - 1\right)\|\theta - \theta^*\| \\
&= O_P\left(\frac{d}{\sqrt{N}}\right)\|\theta - \theta^*\|,
\end{aligned}$$

where the first equality holds by choosing $\theta - \theta^*$ parallel to w_1.

Proof. We prove the two bounds in (22) individually.

Bounding $\|G(X_{\mathcal{S}}, \theta^*)\|$: It follows from Assumption 2 that the columns of $G(X_{\mathcal{S}}, \theta^*)$ are i.i.d. sub-exponential random vectors in $\mathbb{R}^d$ with mean 0 and scaling parameters $\sigma_1/\sqrt{nm}$ and $\alpha_1/(n\sqrt{m})$. The sub-exponential parameters for the scaled matrix $\sqrt{N}\, G(X_{\mathcal{S}}, \theta^*)$ are σ_1 and $\alpha_1/\sqrt{n}$ – recalling that $N = nm$. Applying Theorem 1 to $A = \sqrt{N}\, G(X_{\mathcal{S}}, \theta^*)$, it holds that with probability at least $1 - e^{-\sqrt{d}}$,

$$\|G(X_{\mathcal{S}}, \theta^*)\| = \frac{1}{\sqrt{N}} \|A\| \;\lesssim\; \frac{1}{\sqrt{N}} \left(\sqrt{m} + \sqrt{d}\right) = \frac{1}{\sqrt{n}} + \sqrt{\frac{d}{N}}. \tag{23}$$

Bounding $\|G(X_{\mathcal{S}}, \theta) - G(X_{\mathcal{S}}, \theta^*)\|$ for a fixed $\theta \in \Theta$: For notational convenience, define

$$H(X_{\mathcal{S}}, \theta) \;\triangleq\; G(X_{\mathcal{S}}, \theta) - G(X_{\mathcal{S}}, \theta^*) = \frac{1}{\sqrt{m}} \left[\frac{1}{n} \sum_{i \in \mathcal{S}_1} h(X_i, \theta), \quad \cdots, \quad \frac{1}{n} \sum_{i \in \mathcal{S}_m} h(X_i, \theta)\right], \tag{24}$$

where recall from (14) that the gradient difference function $h(X, \cdot)$ is defined as

$$h(X, \theta) \;=\; \nabla f(X, \theta) - \nabla f(X, \theta^*) - (\nabla F(\theta) - \nabla F(\theta^*)).$$

It follows from Assumption 4 that the columns of $H(X_{\mathcal{S}}, \theta)/\|\theta - \theta^*\|$ are i.i.d. sub-exponential random vectors in $\mathbb{R}^d$ with mean 0 and scaling parameters $\sigma_2/\sqrt{nm}$ and $\alpha_2/(n\sqrt{m})$. Recall that $N = nm$. Applying Theorem 2 to $H(X_{\mathcal{S}}, \theta)/\|\theta - \theta^*\|$, we know that for any fixed θ, with probability at least $1 - \delta$,

$$\begin{aligned}
\|H(X_{\mathcal{S}}, \theta)\| &\lesssim \left(\frac{\sigma_2}{\sqrt{n}} + \frac{\sigma_2}{\sqrt{N}} \phi\left(d + \log\frac{1}{\delta}\right) + \frac{\alpha_2}{\sqrt{Nn}} \phi^2\left(d + \log\frac{1}{\delta}\right)\right) \|\theta - \theta^*\| \\
&\lesssim \left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{N}} \phi\left(d + \log\frac{1}{\delta}\right) + \frac{1}{\sqrt{Nn}} \phi^2\left(d + \log\frac{1}{\delta}\right)\right) \|\theta - \theta^*\|,
\end{aligned} \tag{25}$$

where $\phi(x) = \sqrt{x} \log^{3/2}(x)$, and $\sigma_2 = O(1), \alpha_2 = O(1)$.

ϵ-net argument: We apply ϵ-net argument to extend the point convergence in (25) to the uniform convergence over Θ. In particular, let $\mathcal{N}_{\epsilon_0}$ be an ϵ_0-cover of $\Theta = \{\theta : \|\theta - \theta^*\| \le r\}$ with

$$\epsilon_0 = \frac{\sigma_1}{\sqrt{n}(L + L')}.$$

By [Ver10, Lemma 5.2], we have

$$\log |\mathcal{N}_{\epsilon_0}| \le d \log (r/\epsilon_0) = d \log \frac{r(L + L')\sqrt{n}}{\sigma_1}.$$

By (25) and the union bound, we get that with probability at least $1 - \delta$, for all $\theta \in \mathcal{N}_{\epsilon_0}$

$$\|H(X_{\mathcal{S}}, \theta)\| \lesssim \left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{N}} \phi\left(d + \log\frac{|\mathcal{N}_{\epsilon_0}|}{\delta}\right) + \frac{1}{\sqrt{Nn}} \phi^2\left(d + \log\frac{|\mathcal{N}_{\epsilon_0}|}{\delta}\right)\right) \|\theta - \theta^*\|. \tag{26}$$

So far, we have shown the uniform convergence over net $\mathcal{N}_{\epsilon_0}$. Next, we extend this uniform convergence to the entire set Θ.

For any $\theta \in \Theta$, there exists a $\theta_k \in \mathcal{N}_{\epsilon_0}$ such that $\|\theta - \theta_k\| \le \epsilon_0$. By triangle inequality,

$$\|H(X_{\mathcal{S}}, \theta)\| \le \|H(X_{\mathcal{S}}, \theta_k)\| + \|H(X_{\mathcal{S}}, \theta) - H(X_{\mathcal{S}}, \theta_k)\|.$$

Note that

$$\|H(X_{\mathcal{S}},\theta)-H(X_{\mathcal{S}},\theta_k)\| \le \|H(X_{\mathcal{S}},\theta)-H(X_{\mathcal{S}},\theta_k)\|_{\mathrm{F}} \le \max_{1\le j\le m}\left\|\frac{1}{n}\sum_{i\in\mathcal{S}_j}h(X_i,\theta)-\frac{1}{n}\sum_{i\in\mathcal{S}_j}h(X_i,\theta_k)\right\|$$
$$\overset{(a)}{\le}(L+L')\|\theta-\theta_k\| \le (L+L')\epsilon_0 = \frac{\sigma_1}{\sqrt{n}}, \tag{27}$$

where (a) holds because

$$\left\|\frac{1}{n}\sum_{i\in\mathcal{S}_j}h(X_i,\theta)-\frac{1}{n}\sum_{i\in\mathcal{S}_j}h(X_i,\theta_k)\right\| \le \frac{1}{n}\sum_{i\in\mathcal{S}_j}\|h(X_i,\theta)-h(X_i,\theta_k)\| \le (L+L')\|\theta-\theta_k\|,$$

in view of Assumption 1 and Assumption 3.

Combining (26) and (27), we have that with probability at least $1-\delta$, for any $\theta\in\Theta$,

$$\begin{aligned}\|H(X_{\mathcal{S}},\theta)\| &\le \|H(X_{\mathcal{S}},\theta_k)\|+\frac{\sigma_1}{\sqrt{n}}\\ &\lesssim \left(\frac{1}{\sqrt{n}}+\frac{1}{\sqrt{N}}\phi\left(d+\log\frac{|\mathcal{N}_{\epsilon_0}|}{\delta}\right)+\frac{1}{\sqrt{Nn}}\phi^2\left(d+\log\frac{|\mathcal{N}_{\epsilon_0}|}{\delta}\right)\right)\|\theta-\theta^*\|+\frac{1}{\sqrt{n}}.\end{aligned}$$

Choose $\delta=e^{-d}$, we get with probability at least $1-e^{-d}$,

$$\begin{aligned}\|H(X_{\mathcal{S}},\theta)\| \lesssim &\left(\frac{1}{\sqrt{n}}+\frac{1}{\sqrt{N}}\phi\left(d\log\left(r\sqrt{n}(L+L')\right)\right)+\frac{1}{\sqrt{Nn}}\phi^2\left(d\log\left(r\sqrt{n}(L+L')\right)\right)\right)\|\theta-\theta^*\|\\ &+\frac{1}{\sqrt{n}}.\end{aligned} \tag{28}$$

holds for all $\theta\in\Theta$.

Putting all pieces together Combing (23) and (28), we conclude Proposition 2.

□

Finish the proof of Theorem 3:

Recall that $\log(L+L')=O(\log(Nd))$, $\Theta\subset\{\theta:\|\theta-\theta^*\|\le r\}$ for some positive parameter r such that $\log r=O(\log(Nd))$, and that $N=\Omega(d^2\log^8(Nd))$. Then,

$$\begin{aligned}\Delta_4 &\triangleq \frac{1}{\sqrt{n}}+\frac{1}{\sqrt{N}}\phi\left(d\log\left(r\sqrt{n}(L+L')\right)\right)+\frac{1}{\sqrt{Nn}}\phi^2\left(d\log\left(r\sqrt{n}(L+L')\right)\right)\\ &\lesssim \frac{1}{\sqrt{n}}+\sqrt{\frac{d}{N}}\log^2(Nd)+\frac{1}{\sqrt{n}} \lesssim \frac{1}{\sqrt{n}}+\sqrt{\frac{d}{N}}\log^2(Nd).\end{aligned}$$

Let $\mathcal{E}_1$ and $\mathcal{E}_2$ denote the two events on which the conclusions in Proposition 1 and Proposition 2 hold, respectively. On event $\mathcal{E}_1\cap\mathcal{E}_2$, for each $\theta\in\Theta$, condition (10) in Lemma 1 is satisfied with

$$c_2\left(\frac{1}{\sqrt{n}}+\sqrt{\frac{d}{N}}\log^2(Nd)\right)\|\theta-\theta^*\|+\left(\sqrt{\frac{1}{n}}+\sqrt{\frac{d}{N}}\right)\log(Nd). \tag{29}$$

for sufficently large constant c_2. In view of Lemma 1, Remarks 2 and 3, we have

$$\|G(\theta)-\nabla F(\theta)\| \lesssim \left(\sqrt{\frac{q}{m}}\Delta_4+\Delta_2\right)\|\theta-\theta^*\|+\sqrt{\frac{q}{m}}\Delta_3+\Delta_1.$$

The proof is complete by invoking Proposition 1 and Proposition 2.

Appendices

A Preliminaries

The following lemma presents the standard and well-known concentration inequality for sum of independent sub-exponential random variables.

Lemma 3. *[Ver18] Let $Y_1, \ldots, Y_m$ denote a sequence of independent random variables, where Y_j's are sub-exponential with scaling parameters (σ_j, α_j) and mean 0. Then $\sum_{j=1}^m Y_j$ is sub-exponential with scaling parameters (σ_*, α_*), where $\sigma_*^2 = \sum_{j=1}^m \sigma_j^2$ and $\alpha_* = \max_{1\le j\le m} \alpha_j$. Moreover,*

$$\mathbb{P}\left\{\sum_{j=1}^m Y_j \ge t\right\} \le \begin{cases} \exp\left(-\frac{t^2}{2\sigma_*^2}\right) & \text{if } 0 \le t \le \sigma_*^2/\alpha_*; \\ \exp\left(-\frac{t}{2\alpha_*}\right) & o.w. \end{cases}$$

The following lemma gives an upper bound to the spectral norm of the covariance matrix of a sub-exponential random vector.

Lemma 4. *Let $Y \in \mathbb{R}^d$ denote a zero-mean, sub-exponential random vector with scaling parameters (σ, α), and Σ denote its covariance matrix $\Sigma = \mathbb{E}\left[YY^\top\right]$. Then*

$$\|\Sigma\| \le 4\sigma^2 + 16\alpha^2.$$

Proof. First recall that

$$\|\Sigma\| \;=\; \sup_{v\in S^{d-1}} v^\top \Sigma v = \sup_{v\in S^{d-1}} v^\top \mathbb{E}\left[YY^\top\right] v = \sup_{v\in S^{d-1}} \mathbb{E}\left[\langle Y, v\rangle^2\right].$$

For each unit vector v, from [Ver18, Exercise 1.2.3], we have

$$\begin{aligned} \mathbb{E}\left[\langle Y, v\rangle^2\right] &= \int_0^\infty 2t\ \mathbb{P}\left\{|\langle Y, v\rangle| \ge t\right\} dt \\ &\le \int_0^\infty 4t\ \exp\left(-\frac{1}{2}\min\left\{\frac{t^2}{\sigma^2}, \frac{t}{\alpha}\right\}\right) dt \\ &\le 4\sigma^2 + 16\alpha^2. \end{aligned} \tag{30}$$

Note that the above upper bound is independent of v. The lemma follows by combining the last two displayed equations. □

B Proof of Lemma 2

We first quote a classical concentration inequality for sum of independent, bounded random variables.

Lemma 5 (Bennett's inequality)**.** *Let $Y_1, \cdots, Y_m$ be independent random variables. Assume that $|Y_j - \mathbb{E}[Y_j]| \le B$ almost surely for every j. Then, for any $t > 0$, we have*

$$\mathbb{P}\left\{\sum_{j=1}^m (Y_j - \mathbb{E}[Y_j]) \ge t\right\} \le \exp\left(-\frac{\sigma^2}{B^2} \cdot h\left(\frac{Bt}{\sigma^2}\right)\right),$$

where $\sigma^2 = \sum_{j=1}^m \mathsf{var}(Y_j)$ is the variance of the sum, and

$$h(u) = (1+u)\log(1+u) - u.$$

Proof of Lemma 2. We use the idea of truncation. In this proof, we adopt the convention that $\frac{1}{0} = +\infty$.

For each copy $j = 1, \cdots, m$, we partition Y_j into countably many pieces as follows: Let

$$\begin{aligned} Y_{j,0} &= Y_j \mathbf{1}_{\{|Y_j| \le 1\}} \\ Y_{j,k} &= Y_j \mathbf{1}_{\left\{e^{k-1} \le |Y_j| \le e^k\right\}}, \text{ for } k = 1, 2, \ldots \end{aligned}$$

It is easy to see that

$$Y_j = \sum_{k=0}^{\infty} Y_{j,k}, \quad \text{for } j = 1, \cdots, m.$$

Let $S = \sum_{j=1}^m Y_j$. We have

$$S = \sum_{j=1}^m Y_j = \sum_{j=1}^m \left(\sum_{k=0}^{\infty} Y_{j,k} \right) = \sum_{k=0}^{\infty} \sum_{j=1}^m Y_{j,k} = \sum_{k=0}^{\infty} S_k,$$

where $S_k \triangleq \sum_{j=1}^m Y_{j,k}$, for $k = 0, 1, \cdots$. Thus,

$$\mathbb{P}\left\{ \left| \sum_{j=1}^m Y_j - m\mathbb{E}[Y] \right| > mt \right\} = \mathbb{P}\{|S - \mathbb{E}[S]| > mt\} = \mathbb{P}\left\{ \left| \sum_{k=0}^{\infty} (S_k - \mathbb{E}[S_k]) \right| > mt \right\}$$

To bound $\mathbb{P}\left\{ \left| \sum_{j=1}^m Y_j - m\mathbb{E}[Y] \right| > mt \right\}$ for a given t, our plan is to find a sequence of t_k (which depends on t) such that

$$\{|S - \mathbb{E}[S]| > mt\} \quad \subseteq \quad \cup_{k=0}^{\infty} \{|S_k - \mathbb{E}[S_k]| > mt_k\}, \tag{31}$$

and

$$\mathbb{P}\{|S_k - \mathbb{E}[S_k]| > mt_k\}$$

is small enough to apply the union bound over all k.

In this proof, we choose $t_k = \frac{t}{2(k+1)^2}$ for $k = 0, 1, \cdots$. It is easy to see that (31) holds.

Next, we bound $\mathbb{P}\{|S_k - \mathbb{E}[S_k]| > mt_k\}$ for each k. For given $t \ge t_0$, define

$$k_0 \triangleq \inf\left\{ k \in \mathbb{Z} : 4e^k (k+1)^2 \ge t \right\}. \tag{32}$$

We are particularly interested in the setting when $t \ge t_0 \ge e^2$, which implies that

$$1 \le k_0 \le \log t - 1, \tag{33}$$

noting that $4e^{\log t - 1}(\log t - 1 + 1)^2 \ge t$.

Case 1: $0 \le k \le k_0 - 1$. It is easy to see that when $t \ge t_0 \ge e^2$, $k_0 \ge 1$. Thus, case 1 is well posed. As per the definition of (32), for all $0 \le k \le k_0 - 1$, it holds that $4e^k(k+1)^2 < t$. That is,

$$2e^k < \frac{t}{2(k+1)^2} = t_k. \tag{34}$$

On the other hand, by construction of $Y_{j,k}$ we have almost surely

$$|Y_{j,k} - \mathbb{E}[Y_{j,k}]| \le 2e^k, \quad \text{for all } k. \tag{35}$$

Thus, almost surely

$$|S_k - \mathbb{E}[S_k]| = \left|\sum_{j=1}^m Y_{j,k} - \mathbb{E}\left[\sum_{j=1}^m Y_{j,k}\right]\right| \le \sum_{j=1}^m |Y_{j,k} - \mathbb{E}[Y_{j,k}]| \le 2me^k \quad \text{for all } k,$$

i.e.,

$$\mathbb{P}\left\{|S_k - \mathbb{E}[S_k]| > 2me^k\right\} = 0 \quad \text{for all } k.$$

By (34), we have when $0 \le k \le k_0 - 1$,

$$\mathbb{P}\left\{|S_k - \mathbb{E}[S_k]| > mt_k\right\} \le \mathbb{P}\left\{|S_k - \mathbb{E}[S_k]| > 2me^k\right\} = 0, \tag{36}$$

Case 2: $k_0 \le k \le \log(mt)$. For each k in this range, we will apply Bennett's inequality.

From (35), we know that for any fixed k, the random variable $|Y_{j,k} - \mathbb{E}[Y_{j,k}]| \le 2e^k$. The variance of $Y_{j,k}$ can be bounded as follows: for $k \ge 1$

$$\mathsf{var}(Y_{j,k}) = \mathbb{E}\left[(Y_{j,k} - \mathbb{E}[Y_{j,k}])^2\right] \le \mathbb{E}\left[Y_{j,k}^2\right] \le e^{2k}\mathbb{P}\left\{|Y_i| \ge e^{k-1}\right\} \le e^{2k}\exp\left(-E\left(e^{k-1}\right)\right). \tag{37}$$

For notational convenience, define

$$\sigma_k^2 \triangleq e^{2k}\exp\left(-E(e^{k-1})\right). \tag{38}$$

To see that σ_k^2 is well-defined, recall that we adopt the convention that $\frac{1}{0} = \infty$ and $\exp(-\infty) = 0$.

For each k in this case, i.e., $k_0 \le k \le \log(mt)$, by Lemma 5, we get

$$\begin{aligned}\mathbb{P}\left\{|S_k - \mathbb{E}[S_k]| \ge mt_k\right\} &= \mathbb{P}\left\{\left|\sum_{j=1}^m (Y_{j,k} - \mathbb{E}[Y_{j,k}])\right| \ge mt_k\right\} \\ &\le 2\exp\left(-\frac{\sum_{j=1}^m \mathsf{var}(Y_{j,k})}{e^{2(k+1)}} \cdot h\left(\frac{e^{(k+1)}mt_k}{\sum_{j=1}^m \mathsf{var}(Y_{j,k})}\right)\right),\end{aligned}$$

Note that when $u > 0$, it holds that $h(u) \ge u\log(u/e)$. So we have

$$\begin{aligned}\mathbb{P}\left\{|S_k - \mathbb{E}[S_k]| \ge mt_k\right\} &\le 2\exp\left(-\frac{\sum_{j=1}^m \mathsf{var}(Y_{j,k})}{e^{2(k+1)}} \cdot \frac{e^{(k+1)}mt_k}{\sum_{j=1}^m \mathsf{var}(Y_{j,k})}\log\left(\frac{e^{(k+1)}mt_k}{e\sum_{j=1}^m \mathsf{var}(Y_{j,k})}\right)\right) \\ &= 2\exp\left(-\frac{mt_k}{e^{(k+1)}}\log\left(\frac{e^k mt_k}{\sum_{j=1}^m \mathsf{var}(Y_{j,k})}\right)\right) \\ &\le 2\exp\left(-\frac{mt_k}{e^{(k+1)}}\log\left(\frac{e^k t_k}{\sigma_k^2}\right)\right),\end{aligned} \tag{39}$$

where the last inequality follows from the fact that $\sum_{j=1}^m \mathsf{var}(Y_{j,k}) \le m\sigma_k^2$. We proceed to bound

$\log\left(\frac{e^k t_k}{\sigma_k^2}\right)$ using the assumption (15).

$$\begin{aligned}
\log\left(\frac{e^k t_k}{\sigma_k^2}\right) &= \log\left(\frac{e^k t}{2(k+1)^2 e^{2k}\exp\left(-E\left(e^{k-1}\right)\right)}\right) \\
&= \log\left(\frac{t}{2(k+1)^2 e^{k}\exp\left(-E\left(e^{k-1}\right)\right)}\right) \\
&= \log t - \left(\log 2 + 2\log(k+1) + k - E\left(e^{k-1}\right)\right) \\
&= E(e^{k-1}) - (\log 2 + 2\log(k+1) + k - \log t) \\
&\overset{(a)}{\geq} \frac{1}{2}E(e^{k-1}) + (2k + 4\log(k+1) + \log 2 - \log t) - (\log 2 + 2\log(k+1) + k - \log t) \\
&\geq \frac{1}{2}E(e^{k-1}) + 2\log(k+1) + k, \qquad (40)
\end{aligned}$$

where inequality (a) holds due to the assumption (15). Combining the last displayed equation with (39) yields

$$\begin{aligned}
\mathbb{P}\left\{|S_k - \mathbb{E}\left[S_k\right]| \geq mt_k\right\} &\leq 2\exp\left(-\frac{mt_k}{2e^{(k+1)}}E(e^{k-1})\right) \\
&= 2\exp\left(-\frac{mt}{4(k+1)^2 e^{(k+1)}}E(e^{k-1})\right) \\
&\leq 2\exp\left(-\frac{mt}{4(\log(mt)+1)^2 e^{(k+1)}}E(e^{k-1})\right), \qquad (41)
\end{aligned}$$

where the last inequality holds because in the case under consideration, $k_0 \leq k \leq \log(mt)$. To proceed, we use the monotonicity assumption of $E(t)/t$. If $E(t)/t$ is non-decreasing (increasing), we can bound (41) as

$$\begin{aligned}
\mathbb{P}\left\{|S_k - \mathbb{E}\left[S_k\right]| \geq mt_k\right\} &\overset{(a)}{\leq} 2\exp\left(-\frac{mt}{4(\log(mt)+1)^2 e^{(k_0+1)}}E(e^{k_0-1})\right) \\
&\overset{(b)}{\leq} 2\exp\left(-\frac{mt}{4(\log(mt)+1)^2 t}E\left(\frac{t}{4e(k_0+1)^2}\right)\right) \\
&\overset{(c)}{\leq} 2\exp\left(-\frac{m}{4(\log(mt)+1)^2}E\left(\frac{t}{4e\log^2 t}\right)\right), \qquad (42)
\end{aligned}$$

where (a) holds because $k_0 \leq k \leq \log(mt)$; (b) holds because $k_0 \leq \log t - 1$, $4e^{k_0}(k_0+1)^2 \geq t$, and that $E(\cdot)$ is non-decreasing; (c) follows from $k_0 \leq \log t - 1$, and that $E(\cdot)$ is non-decreasing.

If $E(t)/t$ is non-increasing, we can bound (41) as

$$\begin{aligned}
\mathbb{P}\left\{|S_k - \mathbb{E}\left[S_k\right]| \geq mt_k\right\} &\leq 2\exp\left(-\frac{mt}{4(\log(mt)+1)^2 e^{\log(mt)+1}}E(e^{\log(mt)-1})\right) \\
&= 2\exp\left(-\frac{1}{4e(\log(mt)+1)^2}E\left(\frac{mt}{e}\right)\right). \qquad (43)
\end{aligned}$$

Case 3: $k \geq \log(mt)$. In this case, we use the Chebyshev's inequality:

$$\begin{aligned}\mathbb{P}\left\{|S_k - \mathbb{E}\left[S_k\right]| \geq mt_k\right\} &\leq \frac{\sigma_k^2}{t_k} = \exp\left(-\log\frac{t_k}{\sigma_k^2}\right) \\ &\overset{(a)}{\leq} \frac{1}{(k+1)^2}\exp\left(-\frac{1}{2}E(e^{k-1})\right) \\ &\leq \frac{1}{(k+1)^2}\exp\left(-\frac{1}{2}E\left(\frac{mt}{e}\right)\right), \end{aligned} \tag{44}$$

where (a) follows from (40); the last inequality follows from the fact that $E(u)$ is increasing (non-decreasing) in u.

For a fix t, summing over all $k \in \mathbb{N}$, we have

$$\begin{aligned}\mathbb{P}\left\{\left|\sum_{j=1}^m Y_j - m\mathbb{E}\left[Y\right]\right| \geq mt\right\} &\leq \sum_{k=0}^{\infty}\mathbb{P}\left\{|S_k - \mathbb{E}\left[S_k\right]| \geq mt_k\right\} \\ &= \sum_{k=0}^{k_0-1}\mathbb{P}\left\{|S_k - \mathbb{E}\left[S_k\right]| \geq mt_k\right\} + \sum_{k=k_0}^{\log(mt)}\mathbb{P}\left\{|S_k - \mathbb{E}\left[S_k\right]| \geq mt_k\right\} \\ &\quad + \sum_{\log(mt)+1}^{\infty}\mathbb{P}\left\{|S_k - \mathbb{E}\left[S_k\right]| \geq mt_k\right\} \\ &\leq 0 + \exp\left(-\frac{1}{2}E\left(\frac{mt}{e}\right)\right) + \sum_{k=k_0}^{\log(mt)}\mathbb{P}\left\{|S_k - \mathbb{E}\left[S_k\right]| \geq mt_k\right\}. \end{aligned}$$

Therefore, we have if $E(t)/t$ non-decreasing

$$\begin{aligned}\mathbb{P}\left\{\left|\sum_{j=1}^m Y_j - m\mathbb{E}\left[Y\right]\right| \geq mt\right\} &\leq 2\log(mt)\exp\left(-\frac{m}{4(\log(mt)+1)^2}E\left(\frac{t}{4e\log^2 t}\right)\right) \\ &\quad + \exp\left(-\frac{1}{2}E\left(\frac{mt}{e}\right)\right); \end{aligned}$$

if $E(t)/t$ is non-increasing

$$\begin{aligned}\mathbb{P}\left\{\left|\sum_{j=1}^m Y_j - m\mathbb{E}\left[Y\right]\right| \geq mt\right\} &\leq 2\log(mt)\exp\left(-\frac{1}{4e(\log(mt)+1)^2}E\left(\frac{mt}{e}\right)\right) \\ &\quad + \exp\left(-\frac{1}{2}E\left(\frac{mt}{e}\right)\right). \end{aligned}$$

□

C Proof of Theorem 2

Proof of Theorem 2. Recall $\Sigma = \mathbb{E}\left[A_1 A_1^\top\right]$. Then

$$\|A\|^2 = \left\|AA^\top\right\| \leq \left\|AA^\top - m\Sigma\right\| + m\left\|\Sigma\right\|.$$

In view of Lemma 4, we have $\|\Sigma\| \le 4\sigma^2 + 16\alpha^2$. It remains to bound $\left\|AA^\top - m\Sigma\right\|$. Note that

$$\left\|AA^\top - m\Sigma\right\| = \sup_{v\in S^{d-1}} \left|v^\top \left(AA^\top - m\Sigma\right) v\right| = \sup_{v\in S^{d-1}} \left|\sum_{j=1}^m \left(\langle A_j, v\rangle^2 - \mathbb{E}\left[\langle A_j, v\rangle^2\right]\right)\right|.$$

Fix a $v \in S^{d-1}$. Note that $\langle A_j, v\rangle$ is zero-mean sub-exponential random variable with parameter (σ, α). For $j = 1, \cdots, m$, define

$$Y_j = \langle A_j, v\rangle^2 / \sigma^2. \tag{45}$$

It follows from Lemma 3 that

$$\mathbb{P}\left\{|Y_j| \ge t\right\} = \mathbb{P}\left\{|\langle A_j, v\rangle| \ge \sigma\sqrt{t}\right\} \le 2\exp\left(-\min\left\{\frac{t}{2}, \frac{\sigma\sqrt{t}}{2\alpha}\right\}\right),$$

We apply Lemma 2 to $Y_1, \cdots, Y_m$ with

$$E(t) = \min\left\{\frac{t}{2}, \frac{\sigma\sqrt{t}}{2\alpha}\right\} - \log 2,$$

which is non-decreasing in t. By assumption $\sigma/\alpha = \Omega(1)$, it follows that $E(t)$ scales as $\sqrt{t}$ in t. Thus there exits $t_0 \ge e^2$ such that (15) holds. In addition, $E(t)/t$ is non-increasing. Therefore, (18) in Lemma 2 applies, i.e., for all $t \ge t_0$,

$$\begin{aligned}\mathbb{P}\left\{\left|\sum_{j=1}^m (Y_j - \mathbb{E}[Y_j])\right| \ge mt\right\} &\le 2\log(mt)\exp\left(-\frac{1}{4e\log^2(emt)}E\left(\frac{mt}{e}\right)\right) + \exp\left(-\frac{1}{2}E\left(\frac{mt}{e}\right)\right) \\ &\le 4\log(mt)\exp\left(-\frac{1}{4e\log^2(emt)}E\left(\frac{mt}{e}\right)\right). \end{aligned} \tag{46}$$

Next, we apply ϵ-net argument. Let $\mathcal{N}_{\frac{1}{4}}$ be the $\frac{1}{4}$–net of the unit sphere S^{d-1}. From [Ver10, Lemma 5.2], we know that $\left|\mathcal{N}_{\frac{1}{4}}\right| \le 9^d$. In addition, it follows from [Ver10, Lemma 5.4] that

$$\left\|AA^\top - \Sigma\right\| \le 2\sup_{v\in\mathcal{N}_{\frac{1}{4}}} \left|\sum_{j=1}^m \left(\langle A_j, v\rangle^2 - \mathbb{E}\left[\langle A_j, v\rangle^2\right]\right)\right|.$$

Hence,

$$\begin{aligned}\mathbb{P}\left\{\left\|AA^\top - \Sigma\right\| \ge 2\sigma^2 mt\right\} &\le \mathbb{P}\left\{\sup_{v\in\mathcal{N}_{\frac{1}{4}}} \left|\sum_{j=1}^m \left(\langle A_j, v\rangle^2 - \mathbb{E}\left[\langle A_j, v\rangle^2\right]\right)\right| \ge \sigma^2 mt\right\} \\ &\le \left|\mathcal{N}_{\frac{1}{4}}\right| \mathbb{P}\left\{\left|\sum_{j=1}^m \left(\langle A_j, v\rangle^2 - \mathbb{E}\left[\langle A_j, v\rangle^2\right]\right)\right| \ge \sigma^2 mt\right\} \\ &\le 9^d \mathbb{P}\left\{\left|\sum_{j=1}^m (Y_j - \mathbb{E}[Y_j])\right| \ge mt\right\} \quad \text{by definition of } Y_j \text{ in (45)} \\ &\le \exp\left(-\frac{1}{4e\log^2(emt)}E\left(\frac{mt}{e}\right) + \log 4 + \log\log(mt) + d\log 9\right) \text{ by (46)}.\end{aligned}$$

To complete the proof, we need to choose mt so that the right hand side of the last inequality is smaller than δ. In other words, we need to find $x \geq mt_0$ such that

$$\frac{1}{4e\log^2(ex)}E(x/e) - \log\log x \geq \log\frac{4}{\delta} + d\log 9 \triangleq a.$$

One such x is given by

$$x = c\left(a\log^3 a + \frac{\alpha^2}{\sigma^2}a^2\log^6 a + m\right),$$

where c is a sufficiently large constant. Therefore, we choose

$$mt = c\left(\left(d+\log\frac{1}{\delta}\right)\log^3\left(d+\log\frac{1}{\delta}\right) + \frac{\alpha^2}{\sigma^2}\left(d+\log\frac{1}{\delta}\right)^2\log^6\left(d+\log\frac{1}{\delta}\right) + m\right)$$

The lemma follows by taking the square root of mt. □

D Robust Mean Estimation

We present the proof of Lemma 1 for completeness. For ease of exposition, in the sequel, we let

$$\alpha \triangleq 1-\epsilon \quad \text{and} \quad \widetilde{\sigma}^2 = 2\sigma^2.$$

We first need a minimax identity between the min-max problem (8) and max-min problem (9). For $W \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{A}|}$ and $U \in \mathbb{R}^{d\times d}$, define a function $\psi : (W,U) \to \mathbb{R}$ as:

$$\psi(W,U) = \sum_{i\in\mathcal{A}} c_i \left(\widehat{y}_i - \sum_{j\in\mathcal{A}}\widehat{y}_j W_{ji}\right)^\top U \left(\widehat{y}_i - \sum_{j\in\mathcal{A}}\widehat{y}_j W_{ji}\right).$$

Also, let $\mathcal{W}$ denote the set of all column stochastic matrices $W \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{A}|}$ such that $0 \leq W_{ji} \leq \frac{4-\alpha}{\alpha(2+\alpha)m}$, and $\mathcal{U}$ denote the set of all positive semi-definite matrices $U \in \mathbb{R}^{d\times d}$ such that $\mathsf{Tr}(U) \leq 1$. Then the min-max program (8) can be rewritten as

$$W^* \in \arg\min_{W\in\mathcal{W}} \max_{U\in\mathcal{U}} \psi(W,U) = \arg\min_{W\in\mathcal{W}} \left\| \sum_{i\in\mathcal{A}} c_i \left(\widehat{y}_i - \sum_{j\in\mathcal{A}}\widehat{y}_j W_{ji}\right)\left(\widehat{y}_i - \sum_{j\in\mathcal{A}}\widehat{y}_j W_{ji}\right)^\top \right\|. \quad (47)$$

and the max-min program (9) can be rewritten as

$$U^* \in \max_{U\in\mathcal{U}} \min_{W\in\mathcal{W}} \psi(W,U).$$

Note that $\psi(W,U)$ is convex in W for a fixed U and concave (in fact linear) in U for a fixed W. By von Neumann's minimax theorem, we have

$$\min_{W\in\mathcal{W}} \max_{U\in\mathcal{U}} \psi(W,U) = \max_{U\in\mathcal{U}} \min_{W\in\mathcal{W}} \psi(W,U) = \psi(W^*,U^*).$$

Moreover, (W^*,U^*) is a saddle point, i.e.,

$$W^* \in \arg\min_{W\in\mathcal{W}} \psi(W,U^*), \quad (48)$$

$$U^* \in \arg\max_{U\in\mathcal{U}} \psi(W^*,U). \quad (49)$$

The saddle point properties (48) and (49) are crucial to prove Lemma 1.

Moreover, by condition (10), the underlying true sample $\mathcal{S}$ (of size m) satisfies the following condition:

$$\left\| \frac{1}{m} \sum_{i=1}^{m} (y_i - \mu_{\mathcal{S}})(y_i - \mu_{\mathcal{S}})^\top \right\| \leq \sigma^2,$$

where $\mu_{\mathcal{S}} = \frac{1}{m} \sum_{i=1}^{m} y_i$. Recall that up to q points in $\mathcal{S}$ are corrupted. Let $\mathcal{S}_0 \subseteq \mathcal{S}$ be a subset of uncorrupted subset of $\mathcal{S}$ of size $m - q = (1 - \epsilon)m = \alpha m$. Notably, since q is only an upper bound on the number of corrupted data points, the choice of subset $\mathcal{S}_0$ may not be unique. Nevertheless, for any choice of subset $\mathcal{S}_0$, the following holds:

$$\begin{aligned}
\left\| \frac{1}{|\mathcal{S}_0|} \sum_{i \in \mathcal{S}_0} (y_i - \mu_{\mathcal{S}})(y_i - \mu_{\mathcal{S}})^\top \right\| &= \frac{1}{|\mathcal{S}_0|} \left\| \sum_{i \in \mathcal{S}_0} (y_i - \mu_{\mathcal{S}})(y_i - \mu_{\mathcal{S}})^\top \right\| \\
&\leq \frac{1}{|\mathcal{S}_0|} \left\| \sum_{i=1}^{m} (y_i - \mu_{\mathcal{S}})(y_i - \mu_{\mathcal{S}})^\top \right\| \\
&\leq \frac{1}{\alpha} \sigma^2 \\
&\leq 2\sigma^2, \qquad (50)
\end{aligned}$$

where the last inequality follows because by assumption, $\alpha = 1 - \epsilon \geq \frac{3}{4} \geq \frac{1}{2}$.

As commented in Subsection 2.2, Algorithm 2 terminates within m iterations. For ease of exposition, we use $t = 1, 2, \cdots$ to denote the iteration number in the loop from line 3 to line 9. We use $c_i(t)$, $\tau_i(t)$, and $\mathcal{A}(t)$ to denote the quantities of interest at iteration t. Note that weights c_i and set $\mathcal{A}$ may be updated throughout an iteration. Therefore, we use $\mathcal{A}'(t)$ and $c_i'(t)$ to denote the updated quantities at the end of iteration t. Note that $c_i'(t-1) = c_i(t)$ and $\mathcal{A}'(t-1) = \mathcal{A}(t)$.

D.1 Two auxiliary lemmas

We first show that when Algorithm 2 terminates, most of data points in $\mathcal{S}_0$ are remained in $\mathcal{A}$.

Lemma 6. *For every iteration $t \geq 1$ in the* **while***–loop of Algorithm 2,*

$$\sum_{i \in \mathcal{S}_0 \cap \mathcal{A}(t)} c_i(t)\tau_i(t) \;\leq\; \alpha m \widetilde{\sigma}^2 \qquad (51)$$

$$\sum_{i \in \mathcal{S}_0} (1 - c_i(t)) \leq \frac{\alpha}{4} \sum_{i=1}^{m} (1 - c_i(t)) \qquad (52)$$

$$|\mathcal{S}_0 \cap \mathcal{A}(t)| \geq \frac{\alpha(2+\alpha)m}{4-\alpha}. \qquad (53)$$

Intuitively, Lemma 6 says that in every iteration: (1) the summation of the projected residual error over the non-corrupted data is small; (2) the weights of non-corrupted data points are reduced by a relatively small amount; (3) and more importantly, most non-corrupted data points are not removed.

Proof of Lemma 6. The proof is by induction on (52) and (53). Recall that we use $t = 1, \cdots$ to denote the iteration number in the **while**–loop.

Base case: $t = 1$**.** Note that $\mathcal{A}(1) = [m]$, and $c_i(1) = 1$ for all $i \in \mathcal{A}(1)$. Therefore, (52) and (53) hold for $t = 1$ trivially.

Induction Step: Suppose (52) and (53) hold for t, and the **while**– has not terminate at iteration t. We aim to show (52) and (53) hold for $t + 1$.

We first prove (51) holds for t. Recall that

$$\tau_i(t) = \left(y_i - \sum_{j \in \mathcal{A}(t)} \widehat{y}_j W_{ji}(t) \right)^\top U(t) \left(y_i - \sum_{j \in \mathcal{A}(t)} \widehat{y}_j W_{ji}(t) \right),$$

where $W(t)$ is a minimizer to (8) and $U(t)$ is a maximizer to (9) at iteration t, respectively. Since $(W(t), U(t))$ is a saddle point, it follows from (48) that $W(t) \in \arg\min_{W \in \mathcal{W}} \psi(W, U(t))$. Moreover, this minimization is decoupled over all data points in $\mathcal{A}(t)$ and hence each column of $W(t)$ is optimized independently. Therefore, by letting $W_{*i}(t)$ denote the column of $W(t)$ corresponding to $i \in \mathcal{A}(t)$, we have

$$\begin{aligned} W_{*i}(t) \in \arg\min_{w} & \left(y_i - \sum_{j \in \mathcal{A}(t)} \widehat{y}_j w_j \right)^\top U(t) \left(y_i - \sum_{j \in \mathcal{A}(t)} \widehat{y}_j w_j \right) \\ \text{s. t.} & \sum_{j \in \mathcal{A}(t)} w_j = 1 \\ & 0 \le w_j \le \frac{4 - \alpha}{\alpha(2 + \alpha)m}. \end{aligned} \tag{54}$$

Let $\widetilde{w} \in \mathbb{R}^{|\mathcal{A}(t)|}$ be the column stochastic vector such that

$$\widetilde{w}_j \triangleq \frac{\mathbf{1}_{\{j \in \mathcal{S}_0 \cap \mathcal{A}(t)\}}}{|\mathcal{S}_0 \cap \mathcal{A}(t)|}, \quad \forall\ j \in \mathcal{A}(t).$$

By the induction hypothesis, $\widetilde{w}$ is feasible to (54). Let $Y_{\mathcal{A}(t)} \in \mathbb{R}^{d \times n}$ be the matrix with $\widehat{y}_i$ with $i \in \mathcal{A}(t)$ as columns. Moreover,

$$Y_{\mathcal{A}(t)} \widetilde{w} = \sum_{j \in \mathcal{A}(t)} \widehat{y}_j \widetilde{w}_j = \frac{1}{|\mathcal{S}_0 \cap \mathcal{A}(t)|} \sum_{j \in \mathcal{S}_0 \cap \mathcal{A}(t)} y_j \ \triangleq\ \mu_{\mathcal{S}_0 \cap \mathcal{A}(t)}.$$

Thus, we have

$$\begin{aligned}
\sum_{i\in\mathcal{S}_0\cap\mathcal{A}(t)} c_i(t)\tau_i(t) &\overset{(a)}{\leq} \sum_{i\in\mathcal{S}_0\cap\mathcal{A}(t)} c_i(t)(y_i-\mu_{\mathcal{S}_0\cap\mathcal{A}(t)})^\top U(t)\left(y_i-\mu_{\mathcal{S}_0\cap\mathcal{A}(t)}\right)\\
&\overset{(b)}{\leq} \sum_{i\in\mathcal{S}_0\cap\mathcal{A}(t)} (y_i-\mu_{\mathcal{S}_0\cap\mathcal{A}(t)})^\top U(t)\left(y_i-\mu_{\mathcal{S}_0\cap\mathcal{A}(t)}\right)\\
&\overset{(c)}{\leq} \sum_{i\in\mathcal{S}_0\cap\mathcal{A}(t)} (y_i-\mu_{\mathcal{S}})^\top U(t)\,(y_i-\mu_{\mathcal{S}})\\
&\leq \sum_{i\in\mathcal{S}_0} (y_i-\mu_{\mathcal{S}})^\top U(t)\,(y_i-\mu_{\mathcal{S}})\\
&\overset{(d)}{\leq} \mathsf{Tr}\left(U(t)\right)\left\|\sum_{i\in\mathcal{S}_0}(y_i-\mu_{\mathcal{S}})(y_i-\mu_{\mathcal{S}})^\top\right\|\\
&\overset{(e)}{\leq} \alpha m\widetilde{\sigma}^2,
\end{aligned}$$

where (a) holds by the optimality of $W_{*i}(t)$ to (54); (b) holds because $c_i(t)\leq 1$ and $U(t)\succeq 0$; (c) holds because $\mu_{\mathcal{S}_0\cap\mathcal{A}(t)}=\frac{1}{|\mathcal{S}_0\cap\mathcal{A}(t)|}\sum_{i\in\mathcal{S}_0\cap\mathcal{A}(t)} y_i$ is a minimizer of the quadratic form $\sum_{i\in\mathcal{S}_0\cap\mathcal{A}(t)}(y_i-u)^\top U(t)\,(y_i-u)$, as a function of u; (d) holds because $|\langle A,B\rangle|\leq\|A\|\,\|B\|_*$, where $\|B\|_*$ is the sum of singlular values of B and $\|B\|_*=\mathsf{Tr}(B)$ when $B\succeq 0$; (e) follows by (50) and the facts that $|\mathcal{S}_0|\leq\alpha m$ and $\mathsf{Tr}(U(t))=1$.

Next we prove (52) and (53). Since by induction hypothesis the **while**– has not terminate at iteration t, it follows that

$$\sum_{i\in\mathcal{A}(t)} c_i(t)\tau_i(t) > 4m\widetilde{\sigma}^2. \tag{55}$$

Note that the weights of the data points that do not lie in $\mathcal{A}(t)$ are not updated in iteration t, i.e., $c_i'(t)=c_i(t)$ for $i\notin\mathcal{A}(t)$. As a consequence, we have

$$\begin{aligned}
\sum_{i\in\mathcal{S}_0}\left(1-c_i'(t)\right) &= \sum_{i\in\mathcal{S}_0}(1-c_i(t)) + \sum_{i\in\mathcal{S}_0\cap\mathcal{A}(t)}\left(c_i(t)-c_i'(t)\right)\\
&\leq \frac{\alpha}{4}\sum_{i=1}^{m}(1-c_i(t)) + \frac{1}{\tau_{\max}(t)}\sum_{i\in\mathcal{S}_0\cap\mathcal{A}(t)}\tau_i(t)c_i(t),
\end{aligned}\tag{56}$$

where the last inequality follows from induction hypothesis. Furthermore, we have

$$\frac{1}{\tau_{\max}(t)}\sum_{i\in\mathcal{S}_0\cap\mathcal{A}(t)}\tau_i(t)c_i(t)\overset{(a)}{\leq}\frac{1}{\tau_{\max}(t)}\alpha m\widetilde{\sigma}^2\overset{(b)}{<}\frac{\alpha}{4\tau_{\max}(t)}\sum_{i\in\mathcal{A}(t)}\tau_i(t)c_i(t),$$

where (a) holds because we have shown that (51) holds for t; (b) holds because we have assumed without loss of generality that $\sum_{i\in\mathcal{A}(t)} c_i(t)\tau_i^*(t)>4m\widetilde{\sigma}^2$.

Thus, (56) can be further bounded as

$$\begin{aligned}
\sum_{i\in\mathcal{S}_0}\left(1-c_i'(t)\right) &\le \frac{\alpha}{4}\sum_{i=1}^{m}(1-c_i(t)) + \frac{\alpha}{4\tau_{\max}(t)}\sum_{i\in\mathcal{A}(t)}\tau_i(t)c_i(t)\\
&= \frac{\alpha}{4}\left(\sum_{i\notin\mathcal{A}(t)}(1-c_i(t)) + \sum_{i\in\mathcal{A}(t)}(1-c_i(t)) + \frac{1}{\tau_{\max}(t)}\sum_{i\in\mathcal{A}(t)}\tau_i^*(t)c_i(t)\right)\\
&= \frac{\alpha}{4}\left(\sum_{i\notin\mathcal{A}(t)}(1-c_i'(t)) + \sum_{i\in\mathcal{A}(t)}\left(1-\left(1-\frac{\tau_i^*(t)}{\tau_{\max}(t)}\right)c_i(t)\right)\right)\\
&= \frac{\alpha}{4}\sum_{i=1}^{m}(1-c_i'(t)),
\end{aligned}$$

proving (52) for $t+1$. We rewrite (52) for $t+1$ as

$$\sum_{i\in\mathcal{S}_0}\left(1-c_i'(t)\right) \le \frac{\alpha}{4-\alpha}\sum_{i\notin\mathcal{S}_0}\left(1-c_i'(t)\right).$$

One the one hand, we have

$$\sum_{i\notin\mathcal{S}_0}\left(1-c_i'(t)\right) \le |\mathcal{S}_0^c| \le (1-\alpha)m.$$

On the other hand,

$$\sum_{i\in\mathcal{S}_0}\left(1-c_i'(t)\right) \ge \sum_{i\in\mathcal{S}_0\setminus\mathcal{A}'(t)}\left(1-c_i'(t)\right) \ge \frac{1}{2}\left|\mathcal{S}_0\setminus\mathcal{A}'(t)\right|,$$

where the last inequality holds from the fact that $c_i'(t)\le 1/2$ for all $i\notin\mathcal{A}'(t)$ – by the data removal criterion in Algorithm 2. Combining the last three displayed equations, we get that

$$\left|\mathcal{S}_0\setminus\mathcal{A}'(t)\right| \le \frac{2\alpha(1-\alpha)}{4-\alpha}m,$$

proving (52) for $t+1$. The proof of Lemma 6 is complete. □

Let W be the minimizer of (8) when the **while**–loop terminates. Let W_1 be the result of zeroing out all singular values of W that are greater than 0.9.

Lemma 7. *The matrix $W_0 = (W-W_1)(I-W_1)^{-1}$ is a column stochastic matrix, and the rank of the weight matrix W_0 is one.*

Remark 9. Let $X_{\mathcal{A}}\subseteq\mathbb{R}^{d\times|\mathcal{A}|}$ be the data matrix with columns being the data points in $\mathcal{A}$. Let $Z = X_{\mathcal{A}}W_0$. Since W_0 is rank one, all the $|\mathcal{A}|$ columns in the matrix Z are identical. Denote

$$Z = [\widetilde{\mu},\cdots,\widetilde{\mu}]. \tag{57}$$

Then $\widetilde{\mu}$ is a weighted average of the points in $\mathcal{A}$.

Proof. We first show that W_0 is a column stochastic matrix:

$$\begin{aligned}\mathbf{1}^\top W_0 &= \mathbf{1}^\top (W - W_1)(I - W_1)^{-1} \overset{(a)}{=} (\mathbf{1}^\top - \mathbf{1}^\top W_1)(I - W_1)^{-1} \\ &= \mathbf{1}^\top (I - W_1)(I - W_1)^{-1} = \mathbf{1}^\top,\end{aligned}$$

where (a) follows because W is column stochastic.

Next we show that rank of W_0 is one. From (8), we know that $\|W\|_{\mathrm{F}}^2 \le \frac{4-\alpha}{\alpha(2+\alpha)}$. To see this,

$$\|W\|_{\mathrm{F}}^2 = \sum_{i,j\in\mathcal{A}} W_{ji}^2 \le \sum_{i,j\in\mathcal{A}} \left(W_{ji} \cdot \max_{i,j\in\mathcal{A}} W_{ji}\right) \le \left(\sum_{i,j\in\mathcal{A}} W_{ji}\right) \frac{4-\alpha}{\alpha(2+\alpha)m} \le \frac{4-\alpha}{\alpha(2+\alpha)}.$$

When $\alpha \ge \frac{3}{4}$,

$$\frac{4-\alpha}{\alpha(2+\alpha)} \le \frac{52}{33} < 2 \times 0.9^2.$$

Hence, at most one singular value of W can be greater than 0.9. Moreover, since W is column stochastic, its largest singular value is at least 1. Thus, $W - W_1$ is of rank one. As a consequence, W_0 is of rank one. □

D.2 Proof of Lemma 1

Recall that our goal is to show

$$\|\mu_{\mathcal{S}} - \widehat{\mu}\| = O(\sigma\sqrt{1-\alpha}),$$

where $\widehat{\mu} = \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}} \widehat{y}_i$ is the algorithm output. Recall $Y_{\mathcal{A}} \subseteq \mathbb{R}^{d\times|\mathcal{A}|}$ is the data matrix with columns being the data points in $\mathcal{A}$. In view of Remark 9, columns of $Z = Y_{\mathcal{A}} W_0$ are identical and denoted by $\widetilde{\mu}$. Our proof is divided into two steps:

- We first show that points in $\mathcal{A}$ are clustered around the center $\widetilde{\mu}$. In addition, by (53) in Lemma 6, the set $\mathcal{A}$ mainly consists of uncorrupted data. As a conseqeuence, we are able to show that

$$\widehat{\mu} = \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}} \widehat{y}_i \approx \frac{1}{|\mathcal{S}_0\cap\mathcal{A}|}\sum_{i\in\mathcal{S}_0\cap\mathcal{A}} \widehat{y}_i = \frac{1}{|\mathcal{S}_0\cap\mathcal{A}|}\sum_{i\in\mathcal{S}_0\cap\mathcal{A}} y_i. \tag{58}$$

- By (50), points in $\mathcal{S}_0$ are clustered around the center $\mu_{\mathcal{S}}$. In addition, by (53) in Lemma 6, most of the points in $\mathcal{S}_0$ have been preserved. Thus we are able to show that

$$\mu_{\mathcal{S}} = \frac{1}{m}\sum_{i=1}^{m} y_i \approx \frac{1}{|\mathcal{S}_0\cap\mathcal{A}|}\sum_{i\in\mathcal{S}_0\cap\mathcal{A}} y_i. \tag{59}$$

Putting these two pieces together, the proof of Lemma 1 is complete.

Step 1: We show (58).

When the **while**–loop terminates, in view of (47), we have

$$\left\| Y_{\mathcal{A}}(I-W)\mathsf{diag}\left\{(c_{\mathcal{A}})^{\frac{1}{2}}\right\}\right\| \le 2\sqrt{m}\widetilde{\sigma}., \tag{60}$$

where $\mathsf{diag}\left\{(c_{\mathcal{A}})^{\frac{1}{2}}\right\}$ is the diagonal matrix with diagonal entries given by $\{c_i^{1/2}\}_{i\in\mathcal{A}}$. We will show that $\widehat{y}_i \approx \widetilde{\mu}$ for all $i \in \mathcal{A}$. For this purpose, it is enough to show $\|Y_{\mathcal{A}} - Z\|$ is small:

$$\begin{aligned}
\left\|Y_{\mathcal{A}} - \widetilde{\mu}\mathbf{1}^T\right\| = \|Y_{\mathcal{A}} - Z\| &= \|Y_{\mathcal{A}} - Y_{\mathcal{A}}W_0\| \\
&= \left\|Y_{\mathcal{A}}(I-W_1)(I-W_1)^{-1} - Y_{\mathcal{A}}(W-W_1)(I-W_1)^{-1}\right\| \\
&= \left\|Y_{\mathcal{A}}(I-W)(I-W_1)^{-1}\right\| \\
&\le \|Y_{\mathcal{A}}(I-W)\| \left\|(I-W_1)^{-1}\right\| \\
&\overset{(a)}{\le} \|Y_{\mathcal{A}}(I-W)\| \times 10 \\
&\overset{(b)}{\le} 10\sqrt{2}\left\|Y_{\mathcal{A}}(I-W)\mathsf{diag}\left\{(c_{\mathcal{A}})^{\frac{1}{2}}\right\}\right\| \\
&\overset{(c)}{\le} 20\sqrt{2m}\widetilde{\sigma},
\end{aligned}$$

where (a) holds because the largest singular value of W_1 is at most 0.9; (b) holds because $c_i \ge \frac{1}{2}$ for all $i \in \mathcal{A}$; (c) follows from (60).

Fix any $0 < \epsilon' < 1/2$. Let $\mathcal{T} \subseteq \mathcal{A}$ such that $|\mathcal{T}| \ge (1-\epsilon')|\mathcal{A}|$. We have

$$\begin{aligned}
\left\|\frac{1}{|\mathcal{T}|}\sum_{i\in\mathcal{T}}\widehat{y}_i \;-\; \widehat{\mu}\right\| &= \left\|\frac{1}{|\mathcal{T}|}\sum_{i\in\mathcal{T}}\widehat{y}_i - \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}}\widehat{y}_i\right\| = \left\|\frac{1}{|\mathcal{T}|}\sum_{i\in\mathcal{T}}(\widehat{y}_i-\widetilde{\mu}) - \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}}(\widehat{y}_i-\widetilde{\mu})\right\| \\
&= \left\|\left(\frac{1}{|\mathcal{T}|}-\frac{1}{|\mathcal{A}|}\right)\sum_{i\in\mathcal{T}}(\widehat{y}_i-\widetilde{\mu}) - \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}/\mathcal{T}}(\widehat{y}_i-\widetilde{\mu})\right\| \\
&\overset{(a)}{\le} \frac{|\mathcal{A}|-|\mathcal{T}|}{|\mathcal{T}||\mathcal{A}|}\left\|[Y_{\mathcal{A}}-Z]_{\mathcal{T}}\,\mathbf{1}\right\| + \frac{1}{|\mathcal{A}|}\left\|[Y_{\mathcal{A}}-Z]_{\mathcal{A}/\mathcal{T}}\,\mathbf{1}\right\| \\
&= \left(\frac{|\mathcal{A}|-|\mathcal{T}|}{\sqrt{|\mathcal{T}|}\,|\mathcal{A}|} + \frac{\sqrt{|\mathcal{A}/\mathcal{T}|}}{|\mathcal{A}|}\right)\|Y_{\mathcal{A}}-Z\| \\
&\le 80\sqrt{2}\widetilde{\sigma}\sqrt{\epsilon'}, \qquad (61)
\end{aligned}$$

where $[Y_{\mathcal{A}} - Z]_{\mathcal{T}}$ denotes the submatrix of $Y_{\mathcal{A}} - Z$ – restricting to columns in $\mathcal{T}$, and $\mathbf{1} \in \mathbb{R}^{|\mathcal{T}|}$; the last inequality holds because $\epsilon' < 1/2$ and

$$|\mathcal{A}| \ge |\mathcal{A}\cap\mathcal{S}_0| \ge \frac{\alpha(2+\alpha)}{4-\alpha}m.$$

Note that

$$\frac{\alpha(2+\alpha)}{4-\alpha} \ge 1 - \frac{5}{3}(1-\alpha) \Leftrightarrow (\alpha-1)^2 \ge 0.$$

Thus, $|\mathcal{A} - \mathcal{A}\cap\mathcal{S}_0| \le \frac{5}{3}(1-\alpha)m$. Choosing $\mathcal{T} = \mathcal{A}\cap\mathcal{S}_0$, we obtain

$$\|\mu_{\mathcal{S}_0\cap\mathcal{A}} - \widehat{\mu}\| \le 80\sqrt{2}\widetilde{\sigma}\sqrt{5(1-\alpha)/3} \le 160\widetilde{\sigma}\sqrt{1-\alpha} = O(\widetilde{\sigma}\sqrt{1-\alpha}). \qquad (62)$$

Step 2: We show (59). The proof of (59) is similar to that of (58).

Recall that $\mu_{\mathcal{S}} = \frac{1}{m}\sum_{i=1}^m y_i$, and that $\mu_{\mathcal{S}_0\cap\mathcal{A}} = \frac{1}{|\mathcal{S}_0\cap\mathcal{A}|}\sum_{i\in\mathcal{S}_0\cap\mathcal{A}} y_i$. We have

$$\begin{aligned}
\|\mu_{\mathcal{S}} - \mu_{\mathcal{S}_0\cap\mathcal{A}}\| &= \left\|\mu_{\mathcal{S}} - \frac{1}{|\mathcal{A}\cap\mathcal{S}_0|}\sum_{i\in\mathcal{S}_0\cap\mathcal{A}} y_i\right\| \\
&= \left\|\frac{1}{|\mathcal{A}\cap\mathcal{S}_0|}\sum_{i\in\mathcal{A}\cap\mathcal{S}_0}(y_i - \mu_{\mathcal{S}})\right\| \\
&= \frac{1}{|\mathcal{A}\cap\mathcal{S}_0|}\|[Y_{\mathcal{A}\cap\mathcal{S}_0} - \mu_S]\mathbf{1}\| \\
&\le \frac{\sqrt{|\mathcal{S}_0|}}{\sqrt{|\mathcal{A}\cap\mathcal{S}_0|}}\widetilde{\sigma} \\
&\le \sqrt{\frac{4-\alpha}{\alpha(2+\alpha)}}\sqrt{1-\alpha}\widetilde{\sigma} \le \sqrt{2(1-\alpha)}\widetilde{\sigma}.
\end{aligned}$$

D.3 Modification of Algorithm 2

Recall that Algorithm 2 needs to know the upper bound σ in (10) to terminate. If we do not know σ a priori, as commented in Remark 4, we can modify the termination condition of Algorithm 2 as follows: If

$$\left|\mathcal{A}\setminus\left\{i:\ \left(1-\frac{\tau_i}{\tau_{\max}}\right)c_i \le \frac{1}{2}\right\}\right| \ge \frac{\alpha(2+\alpha)m}{4-\alpha},$$

we update $c_i \leftarrow \left(1-\frac{\tau_i}{\tau_{\max}}\right)c_i$ and remove $\left\{i:\ c_i \le \frac{1}{2}\right\}$ from $\mathcal{A}$; otherwise, we break the **while**–loop.

Similar to the original Algorithm 2, in the modified Algorithm 2, in each iteration of the **while**–loop at least one point will be removed. Thus, the modified Algorithm 2 terminates in at most m iterations. Suppose the modified Algorithm 2 terminates at iteration t^*. By the modified code we know $|\mathcal{A}(t^*)| \ge \frac{\alpha(2+\alpha)m}{4-\alpha}$; otherwise, the algorithm terminates earlier than t^*. By the termination condition, we also know that

$$\left|\mathcal{A}(t^*) - \left\{i:\ \left(1-\frac{\tau_i}{\tau_{\max}}\right)c_i \le \frac{1}{2}\right\}\right| < \frac{\alpha(2+\alpha)m}{4-\alpha}. \tag{63}$$

Claim 1. There exists an iteration $t' \le t^*$ such that $\sum_{i\in\mathcal{A}(t')} c_i(t')\tau_i(t') \le 8m\sigma^2$.

Proof. We prove by contradiction. Suppose

$$\sum_{i\in\mathcal{A}(t)} c_i(t)\tau_i(t) > 8m\sigma^2, \quad \forall t \le t^*. \tag{64}$$

Note that the modified Algorithm 2 and the original Algorithm 2 differ only in their termination conditions. Recall that the original termination condition is only used in the proof of Lemma 6 to conclude that (55) holds when the **while**-loop does not terminate. Thus, under the hypothesis (given in the last displayed equation), Lemma 6 still holds. It follows that

$$\begin{aligned}
\left|\mathcal{A}(t^*) - \left\{i:\ \left(1-\frac{\tau_i}{\tau_{\max}}\right)c_i \le \frac{1}{2}\right\}\right| &\ge \left|\mathcal{S}_0 \cap \left(\mathcal{A}(t^*) - \left\{i:\ \left(1-\frac{\tau_i}{\tau_{\max}}\right)c_i \le \frac{1}{2}\right\}\right)\right| \\
&\ge \frac{\alpha(2+\alpha)m}{4-\alpha},
\end{aligned}$$

which leads to a contradiction. ∎

Since $\mathcal{A}(t)$ is monotone decreasing, it follows that $\mathcal{A}(t^*) \subseteq \mathcal{A}(t')$. Moreover,

$$|\mathcal{A}(t^*)| \geq \frac{\alpha(2+\alpha)m}{4-\alpha} \geq \frac{\alpha(2+\alpha)}{4-\alpha} |\mathcal{A}(t')| \geq \left(1 - \frac{5}{3}(1-\alpha)\right) |\mathcal{A}(t')| .$$

By (61), we know

$$\left\| \frac{1}{|\mathcal{A}(t^*)|} \sum_{i \in \mathcal{A}(t^*)} \widehat{y}_i - \frac{1}{|\mathcal{A}(t')|} \sum_{i \in \mathcal{A}(t')} \widehat{y}_i \right\| \leq 80\sqrt{2}\widetilde{\sigma} \sqrt{\frac{5}{3}(1-\alpha)} = O(\sigma\sqrt{1-\alpha}).$$

From Lemma 1, we know

$$\left\| \frac{1}{|\mathcal{A}(t')|} \sum_{i \in \mathcal{A}(t')} \widehat{y}_i - \mu_{\mathcal{S}} \right\| = O(\sigma\sqrt{1-\alpha}).$$

Combining the last two displayed equations, we have

$$\left\| \frac{1}{|\mathcal{A}(t^*)|} \sum_{i \in \mathcal{A}(t^*)} \widehat{y}_i - \mu_{\mathcal{S}} \right\| = O(\sigma\sqrt{1-\alpha}).$$

Acknowledgment

J. Xu was supported in part by the NSF Grant CCF-1755960.

References

[AAZL18] Dan Alistarh, Zeyuan Allen-Zhu, and Jerry Li. Byzantine stochastic gradient descent. *arXiv preprint arXiv:1803.08917*, 2018. 3

[ALPTJ10] Radosław Adamczak, Alexander Litvak, Alain Pajor, and Nicole Tomczak-Jaegermann. Quantitative estimates of the convergence of the empirical covariance matrix in log-concave ensembles. *Journal of the American Mathematical Society*, 23(2):535–561, 2010. 11

[BMGS17] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Byzantine-tolerant machine learning. *arXiv preprint arXiv:1703.02757*, 2017. 3

[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

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

[CSV17] Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 47–60, New York, NY, USA, 2017. ACM. 3, 4, 5

[CSX17] Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proc. ACM Meas. Anal. Comput. Syst.*, 1(2):44:1–44:25, December 2017. 2, 3, 4, 6, 7, 10, 15, 16

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

[DJW14] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Privacy aware learning. *J. ACM*, 61(6):38:1–38:57, December 2014. 1

[DKK+16a] I. Diakonikolas, G. Kamath, D. M. Kane, J. Li, A. Moitra, and A. Stewart. Robust estimators in high dimensions without the computational intractability. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 655–664, Oct 2016. 3, 8

[DKK+16b] 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. 4

[DKK+17] Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Being robust (in high dimensions) can be practical. *CoRR*, abs/1703.00893, 2017. 3, 4, 8

[fed] Federated learning: Collaborative machine learning without centralized training data. https://research.googleblog.com/2017/04/federated-learning-collaborative.html. Accessed: 2017-04-10. 1

[FXM14] Jiashi Feng, Huan Xu, and Shie Mannor. Distributed robust learning. *arXiv preprint arXiv:1409.5937*, 2014. 3

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

[KMR15] Jakub Konečnỳ, Brendan McMahan, and Daniel Ramage. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575*, 2015. 1, 2

[KMY+16] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016. 2, 4

[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

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

[MMR+16] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016. 2, 4

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

[NJLS09] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009. 3

[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

[SCV18] Jacob Steinhardt, Moses Charikar, and Gregory Valiant. Resilience: A Criterion for Learning in the Presence of Arbitrary Outliers. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*, volume 94 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 45:1–45:21, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. 1, 3, 4, 7, 8, 9

[SV16] Lili Su and Nitin H. Vaidya. Fault-tolerant multi-agent optimization: Optimal iterative distributed algorithms. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, PODC '16, pages 425–434, New York, NY, USA, 2016. ACM. 3, 5

[Tao12] T. Tao. *Topics in random matrix theory.* American Mathematical Society, Providence, RI, USA, 2012. 12

[Ver10] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010. 11, 17, 24

[Ver18] Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science.* Cambridge university press, 2018. 13, 19

[YCRB18] Dong Yin, Yudong Chen, Kannan Ramchandran, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. *arXiv preprint arXiv:1803.01498*, 2018. 3, 4