A Huber Loss Minimization Approach to Byzantine Robust Federated Learning

Puning Zhao, Fei Yu, Zhiguo Wan*

Zhejiang Lab
Hangzhou, Zhejiang, China
{pnzhao,yufei,wanzhiguo}@zhejianglab.com

Abstract

Federated learning systems are susceptible to adversarial attacks. To combat this, we introduce a novel aggregator based on Huber loss minimization, and provide a comprehensive theoretical analysis. Under independent and identically distributed (i.i.d) assumption, our approach has several advantages compared to existing methods. Firstly, it has optimal dependence on ϵ, which stands for the ratio of attacked clients. Secondly, our approach does not need precise knowledge of ϵ. Thirdly, it allows different clients to have unequal data sizes. We then broaden our analysis to include non-i.i.d data, such that clients have slightly different distributions.

Introduction

Due to privacy concerns, there are a large number of isolated information islands, resulting in the difficulty of integrating data from various sources. Under such background, a novel machine learning framework called Federated Learning (FL) has arisen in recent years (McMahan et al. 2017). FL consists of numerous clients that store and compute data locally, and a central server that plays the role as a coordinator. In comparison to traditional centralized learning, FL offers distinct advantages in terms of both computational efficiency and privacy protection. As a result, FL is gaining increasing attention and has been widely applied in various domains, including mobile devices, industrial engineering, and healthcare (Yang et al. 2019; Li et al. 2020).

Nevertheless, FL is facing several severe challenges (Kairouz et al. 2021), with one of them being the robustness issue. Due to various factors, including data poisoning, system malfunctions and transmission errors, some clients may send wrong gradient vectors to the server. Consider that these abnormal behaviors are hard to predict and may happen in an unknown manner, it suffices to analyze the most harmful attack, which is typically modeled as Byzantine failure (Lamport, Shostak, and Pease 1982). Under this model, an adversary can modify the gradient values uploaded to the master in arbitrary way. Without proper handling, even a single malicious client can significantly degrade the model performance (Bagdasaryan et al. 2020). Therefore, for the safe deployment of FL, it is important to design effective defense strategies robust to Byzantine attacks.

There have been many existing works on Byzantine robust federated learning problems. In particular, various gradient aggregators have been proposed. Krum (Blanchard et al. 2017) picks the gradient vector uploaded from clients with small nearest neighbor distances. However, the global convergence is not guaranteed. (Chen, Su, and Xu 2017) proposed a geometric median-of-mean method, which ensures that the model weights converge to a point near to the global minimum, as long as $\epsilon < 1/2$, with ϵ being the fraction of Byzantine machines. This method is not perfect since the error has a suboptimal rate of $\tilde{O}(\sqrt{\epsilon d})$. (Yin et al. 2018) analyzed two aggregators. The first one, called coordinate-wise median, is suboptimal if the sample size per client n is smaller than the number of clients m. Unfortunately, this is quite likely in practice. The second one is coordinate-wise trimmed mean, which has optimal dependence with ϵ when ϵ is small. However, as will be discussed later, if ϵ is close to $1/2$, then coordinate-wise trimmed mean is not efficient. Another drawback is that this method needs the precise knowledge of ϵ, which is usually not practical. Moreover, the analysis of these previous methods are based on some simplified assumption, including independent and identically distributed (i.i.d) assumption, and that all clients have nearly equal sample sizes. More theoretical analysis is needed in realistic scenarios with heterogeneous and unbalanced data.

In this paper, we propose a novel approach to Byzantine robust federated learning, which aggregates gradients by minimizing a multi-dimensional Huber loss. As a widely used loss function in robust statistics (Huber 1964, 2004; Hall and Jones 1990; Zhao and Wan 2023), Huber loss combines the advantages of ℓ_1 and ℓ_2 loss, and achieves a trade-off between robustness and consistency. However, the original definition of Huber loss was for scalars. We generalize the original definition, to make it suitable for vectors. In each iteration, given a list of gradient vectors uploaded from clients, the new algorithm obtains the estimated gradient of the underlying global risk function by minimizing the generalized Huber loss, and then use the outcome as the direction of parameter update.

We then provide theoretical analysis of the proposed method. To begin with, it is assumed that training samples are i.i.d, which is common in most of existing works on

*Corresponding author

Byzantine robust FL (Chen, Su, and Xu 2017; Blanchard et al. 2017; Yin et al. 2018). Under i.i.d assumption, two error bounds are derived for balanced and unbalanced data, respectively. Consider that i.i.d assumption may not be realistic, we then make some generalization to allow heterogeneous clients. The result shows that our method is robust to moderate violation of i.i.d assumption. There are several recent works (Li et al. 2019; Pillutla, Kakade, and Harchaoui 2022; Ghosh et al. 2019) that deal with heterogeneous data. These works are mainly designed for the case that the differences between clients are large. In slightly heterogeneous regime, these methods can not achieve comparable statistical rates.

Finally, we discuss how to implement our new algorithm. Our design is motivated by Weiszfeld's algorithm for calculating the geometric median of a set of vectors (Weiszfeld and Plastria 2009). We make some adjustments to Weiszfeld's algorithm, such that it can minimize the multi-dimensional Huber loss.

In general, compared with existing methods, our new approach has several advantages. Firstly, the dependence of statistical risk on the attack ratio ϵ is nearly minimax optimal, up to a logarithm factor. Secondly, our method has desirable performance with unbalanced data. In particular, with an adaptive rule of parameter selection, the statistical rate is the same as the case with balanced data. Thirdly, many existing methods require the precise knowledge of ϵ to set parameters, which is usually not practical. On the contrary, our method works well under the ϵ-agnostic settings.

Contributions

Our contributions are summarized as follows.

- A multi-dimensional Huber loss minimization approach to robust federated learning;
- Theoretical analysis of our method for both balanced and unbalanced data under i.i.d assumption, which also provides a guideline of parameter selection;
- Extension of the analysis above to heterogeneous clients;
- An implementation algorithm of multi-dimensional Huber loss minimization;
- Numerical experiments on both synthesized and real data, which validates the effectiveness of our new method.

The Proposed Method

In this section, we make a precise statement of the framework of the federated learning problem, and then introduce our proposed aggregator based on minimization of multi-dimensional Huber loss.

The framework is shown in Algorithm 1. Suppose there is a server S_0 and m clients $S_1, \ldots, S_m$. Denote $\mathcal{B}$ as the set of Byzantine clients. There are N training samples in total, with each client S_i storing n_i of them. Denote $\mathbf{Z}_{ij}$ as the j-th sample in the i-th client, $j \in \{1, \ldots, n_i\}$. Let $f(\mathbf{w}, \mathbf{z})$ be the loss function of model parameter $\mathbf{w} \in \mathcal{W}$ with respect to sample $\mathbf{z}$, in which $\mathcal{W} \subset \mathbb{R}^d$ is the parameter space. Moreover, define the global risk function as

$$F(\mathbf{w}) = \mathbb{E}[f(\mathbf{w}, \mathbf{Z})], \tag{1}$$

in which $\mathbf{Z}$ follows the global distribution of training samples. In particular, under i.i.d assumption, $\mathbf{Z}$ just follows the distribution of arbitrary $\mathbf{Z}_{ij}$. Otherwise, with heterogeneous clients, $\mathbf{Z}$ follows the average distribution of clients weighted by n_i. The goal is to learn the global minimizer

$$\mathbf{w}^* = \arg\min_{\mathbf{w} \in \mathcal{W}} F(\mathbf{w}). \tag{2}$$

The algorithm starts with an initial parameter $\mathbf{w}_0 \in \mathcal{W}$, At each iteration $t = 0, 1, \ldots$, we find an update $\mathbf{w}_{t+1}$. Ideally, it would be better if we know $\nabla F(\mathbf{w}_t)$, so that we can use a simple gradient descent to update $\mathbf{w}$, i.e. $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla F(\mathbf{w}_t)$, in which η is the learning rate. However, $\nabla F(\mathbf{w}_t)$ is unknown in practice. We need to estimate it using the gradient vectors uploaded from clients. Therefore, at each iteration t, the master broadcasts parameter $\mathbf{w}_t$ to all clients, and then wait for the responses from them. Benign clients send the estimated gradient vectors back to the master, with respect to parameter $\mathbf{w}_t$. On the contrary, Byzantine clients send arbitrary vectors determined by the adversary. To be more precise, denote $\mathbf{X}_{it}$ as the vector received from client i at the t-th iteration, then

$$\mathbf{X}_{it} = \begin{cases} \frac{1}{n_i} \sum_{j=1}^{n_i} \nabla f(\mathbf{w}_t, \mathbf{Z}_{ij}) & \text{if} \quad i \notin \mathcal{B} \\ \star & \text{if} \quad i \in \mathcal{B}, \end{cases} \tag{3}$$

in which $\star$ means arbitrary vector determined by the adversary. After received $\mathbf{X}_{it}$ for all $i = 1, \ldots, m$, the master then updates the parameter $\mathbf{w}$ according to the following rule:

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}_t - \eta g(\mathbf{w}_t)), \tag{4}$$

in which $\Pi_{\mathcal{W}}(\cdot)$ is an Euclidean projection operator, which ensures that the model parameter stays in $\mathcal{W}$. This operator is also used in (Yin et al. 2018; Zhu et al. 2023). η is the learning rate. $g(\mathbf{w}_t)$ is the aggregator function, which estimates $\nabla F(\mathbf{w}_t)$ using $\mathbf{X}_{it}$, $i = 1, \ldots, m$.

Now it remains to design the aggregator function $g(\mathbf{w}_t)$. Our idea is to minimize the Huber loss weighted by the sample sizes in each clients:

$$g(\mathbf{w}_t) = \arg\min_{s} \sum_{i=1}^{m} n_i \phi_i(\|\mathbf{s} - \mathbf{X}_{it}\|), \tag{5}$$

in which $\|\cdot\|$ is the ℓ_2 norm, and

$$\phi_i(u) = \begin{cases} \frac{1}{2}u^2 & \text{if} \quad |u| \le T_i \\ T_i u - \frac{1}{2}T_i^2 & \text{if} \quad |u| > T_i \end{cases} \tag{6}$$

is the Huber loss function. If clients have equal or nearly equal sample size n_i, then we can let T_i to be the same for all i. Otherwise, we may use different T_i. With larger n_i, we let T_i to be smaller. We refer detailed discussion on parameter selection to the next two sections.

Now we explain the intuition of such design. We have two requirements for a good aggregator: consistency without attack, and robustness under attack. Minimizing ℓ_2 loss corresponds to a simple averaging $g_{avg}(\mathbf{w}_t) = (1/m) \sum_{i=1}^{n_i} \mathbf{X}_{it}$,

Algorithm 1: Byzantine Robust Federated Learning

Input: Master machine S_0, working machines $S_1, \ldots, S_m$
Parameter: Initial weight parameter $\mathbf{w}_0 \in \mathcal{W}$, learning rate η, total number of steps T
Output: Estimated weight $\hat{\mathbf{w}}$

for $t = 0, 1, \ldots, T-1$ **do**
  Server: broadcast current parameter $\mathbf{w}_t$ to all clients;
  for $i = 1, \ldots, m$ **in parallel do**
    Client i: compute local gradient $\mathbf{G}_i = (1/n_i)\sum_{j=1}^{n_i} \nabla f(\mathbf{w}_t, \mathbf{Z}_{ij})$;
    if i is benign **then**
      $\mathbf{X}_{it} = \mathbf{G}_i(\mathbf{w}_t)$;
    else
      $\mathbf{X}_{it}$ is an arbitrary d dimensional vector determined by the adversary;
    end if
    send $\mathbf{X}_{it}$ to the server;
  end for
  Server: Receive $\mathbf{X}_{it}$, $i = 1, \ldots, m$ from each client;
  Calculate aggregated gradient $g(\mathbf{w}_t)$ using $\mathbf{X}_{it}$, $i = 1, \ldots, m$;
  Update parameter $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}_t - \eta g(\mathbf{w}_t))$;
end for
return $\hat{\mathbf{w}} = \mathbf{w}_T$

which is consistent without attack, but not robust. On the contrary, minimizing ℓ_1 loss yields the geometric median of $\mathbf{X}_{it}$, $i = 1, \ldots, m$, which is robust but not consistent even without attack. Therefore, we minimize Huber loss, which combines the advantages of these two methods. At the limit $T_i \to \infty$, ϕ_i becomes ℓ_2 loss, then $g(\mathbf{w}_t)$ is just the sample mean $g_{avg}(\mathbf{w}_t)$. The opposite limit is $T_i \to 0$, under which $g(\mathbf{w}_t)$ is actually the weighted geometric median of $\mathbf{X}_{it}$, $i = 1, \ldots, m$. Between these two extremes, we can set appropriate T_i to achieve a good tradeoff between consistency and Byzantine robustness.

In the following sections, we provide a theoretical analysis of the performance of our method. Following previous works (Yin et al. 2018; Zhu et al. 2023), we discuss three cases separately, which are stated in Assumption 1:

Assumption 1. *Consider the following three types of global loss function F:*

(a) (Strong convex) F is μ-strong convex and L-smooth, and $\mathcal{W}$ is convex;

(b) (General convex) F is convex and L-smooth, with $\mathcal{W} = \{\mathbf{w} | \|\mathbf{w} - \mathbf{w}^*\| \le 2\|\mathbf{w}_0 - \mathbf{w}^*\|\}$;

(c) (Non-convex) F is L-smooth, and $\|\nabla F(\mathbf{w})\| \le M$ for all $\mathbf{w} \in \mathcal{W}$.

For all these three cases, we make the following common assumption on the covering number, which ensures the uniform convergence of the aggregator function:

Assumption 2. *Assume that there exists constants C_W, r_D, such that for all $r < r_D$, the r-covering of $\mathcal{W}$ is bounded by $N_c(r) \le C_W / r^d$.*

Before diving into the detailed analysis, we clarify the notations used in the remainder of this paper first: $a \lesssim b$ if there exists a constant C such that $a \le Cb$. C may depend on μ, σ, L and C_W in the assumptions. Conversely, $a \gtrsim b$ means $a \ge Cb$. Moreover, $a \sim b$ means there exists two constants C_1 and C_2 such that $C_1 b \le a \le C_2 b$. Furthermore, $a = \tilde{O}(b)$ means $a \le Cb \ln^k(N/\delta)$ for some constants C and k. Denote $[m] = \{1, \ldots, m\}$ as the set of numbers from 1 to m. ϵ, q are the ratio and the number of Byzantine clients, respectively, with $q = \epsilon m$. Finally, $\|\cdot\|$ denotes ℓ_2 norm.

Theoretical Analysis for I.I.D Case

In this section, similar to most of previous works, we assume that all samples are i.i.d. We discuss two cases, depending on whether sample sizes are balanced in different clients.

Assumption 3. *$\mathbf{Z}_{ij}$ are i.i.d for all $i = 1, \ldots, m$ and $j = 1, \ldots, n_i$. For any i and j, $\nabla f(\mathbf{w}, \mathbf{Z}_{ij})$ is sub-exponential with parameter σ, i.e. For all λ such that $|\lambda| \le 1/\sigma$,*

$$\sup_{\|\mathbf{v}\|=1} \mathbb{E}\left[e^{\lambda \mathbf{v}^T(\nabla f(\mathbf{w}, \mathbf{Z}_{ij})) - \nabla F(\mathbf{w})}\right] \le e^{\frac{1}{2}\sigma^2\lambda^2}, \tag{7}$$

for any vector $\mathbf{v}$ with $\|\mathbf{v}\| = 1$;

Assumption 3 ensures that with high probability, using the true sample gradients, we are able to identify $\mathbf{w}^*$ defined in (2). Similar assumption was also made in (Chen, Su, and Xu 2017; Cao and Lai 2019).

Balanced Data

In this case, we assume that $n_i = N/m$ for all i, in which N is the total number of training samples, and m is the number of clients. Here we just denote n as the number of samples per client, and the subscript i is omitted. The analysis here can be simply generalized to the case in which n_i are different but are in the same order, i.e. there exists two constants c_1 and c_2, such that $c_1 n \le n_i \le c_2 n$. Since data sizes are balanced, we set equal thresholds in Huber losses, i.e. $T_i = T$ for all i.

We aim to obtain a bound of $\|\mathbf{w}_t - \mathbf{w}^*\|$, the error of the estimation of global minimizer, that holds with probability at least $1 - \delta$. In particular, the following theorem holds. For the sake of simplicity, we state the asymptotic version here, while the finite sample bounds and proofs are shown in the supplementary material.

Theorem 1. *There exists two constants C_1 and C_2, if*

$$C_1\sqrt{\frac{d}{n}\ln\frac{N}{\delta}} \le T \le C_2\sqrt{\frac{d}{n}\ln\frac{N}{\delta}}, \tag{8}$$

then under Assumption 2 and 3, with $|\mathcal{B}| = \epsilon m$ Byzantine clients, the following equations hold with probability at least $1 - \delta$.

(1) (Strong convex) Under Assumption 1(a), if $\eta \le 1/L$,

$$\|\mathbf{w}_t - \mathbf{w}^*\| \le (1-\rho)^t \|\mathbf{w}_0 - \mathbf{w}^*\| + \frac{2\Delta_A}{\mu}, \tag{9}$$

in which $\rho = \eta\mu/2$;

(2) (General convex) Under Assumption 1(b), with $\eta = 1/L$, after $t_m = (L/\Delta_A)\|\mathbf{w}_0 - \mathbf{w}^*\|_2$ steps,

$$F(\mathbf{w}_{t_m}) - F(\mathbf{w}^*) \le 16\|\mathbf{w}_0 - \mathbf{w}^*\|\Delta_A; \tag{10}$$

(3) (Non-convex) Under Assumption 1(c), with $\eta = 1/L$, after $t_m = (2L/\Delta_A^2)(F(\mathbf{w}_0) - F(\mathbf{w}^*))$ steps, we have

$$\min_{t=0,1,\ldots,t_m} \|\nabla F(\mathbf{w}_t)\| \leq \sqrt{2}\Delta_A, \tag{11}$$

in which

$$\Delta_A \lesssim \left(\frac{1}{\sqrt{1-2\epsilon}}\frac{\epsilon}{\sqrt{n}} + \frac{1}{\sqrt{N}}\right)\sqrt{d\ln\frac{N}{\delta}}. \tag{12}$$

From (8), the selection of parameter T does not rely on the knowledge of ϵ. Moreover, with fixed d, if ϵ is small, our error bound $\tilde{O}(\epsilon/\sqrt{n} + 1/\sqrt{N})$ is nearly optimal, since it matches the information-theoretic minimax lower bound up to a logarithm factor (Yin et al. 2018).

Unbalanced Data

Now we discuss a more realistic setting, such that n_i are different among clients. In this case, we design an adaptive selection rule of T_i. For a benign client S_i with n_i samples, from central limit theorem, the distance between the gradient vector send to the server decays roughly with the squared root of n_i, i.e. $\|\mathbf{X}_i(\mathbf{w}) - \nabla F(\mathbf{w})\| \sim \sqrt{d/n_i}$ with high probability. Therefore, if n_i is large and S_i is benign, then $\mathbf{X}_i(\mathbf{w})$ should be close to most of other gradient vectors. If $\mathbf{X}_i(\mathbf{w})$ is far away from others, then client S_i is highly likely to be Byzantine. However, if n_i is small, even if $\mathbf{X}_i(\mathbf{w})$ is far away from others, we can not infer that S_i is Byzantine, since the variance of $\mathbf{X}_i(\mathbf{w})$ is large even if S_i is benign. With such intuition, we set T_i to be smaller with large n_i, and vice versa.

To ensure the convergence, both the ratio of attacked clients and the ratio of the samples in attacked clients need to be small. Therefore, we slightly change the definition of ϵ as the maximum of the fraction of Byzantine clients, and the fraction of samples stored in Byzantine clients, i.e.

$$\epsilon = \max\left\{\frac{|\mathcal{B}|}{m}, \frac{\sum_{i\in\mathcal{B}} n_i}{N}\right\}. \tag{13}$$

The theoretical results for unbalanced data is shown in Theorem 2.

Theorem 2. *Let*

$$T_i = T_0 + \frac{M}{\sqrt{n_i}}, \tag{14}$$

with $M \sim \sigma\sqrt{d\ln(N/\delta)}$, and

$$\epsilon\sigma\sqrt{\frac{md}{N}\ln\frac{N}{\delta}} \lesssim T_0 \lesssim \sigma\sqrt{\frac{md}{N}\ln\frac{N}{\delta}}. \tag{15}$$

Then under Assumption 2 and 3, the following equations hold for small ϵ with probability $1-\delta$.

(1) (Strong convex) Under Assumption 1(a), if $\eta \leq 1/L$,

$$\|\mathbf{w}_t - \mathbf{w}^*\| \leq (1-\rho)^t \|\mathbf{w}_0 - \mathbf{w}^*\| + \frac{2\Delta_A}{\mu}, \tag{16}$$

in which $\rho = \eta\mu/2$;

(2) (General convex) Under Assumption 1(b), with $\eta = 1/L$, after $t_m = (L/\Delta_B)\|\mathbf{w}_0 - \mathbf{w}^*\|_2$ steps,

$$F(\mathbf{w}_{t_m}) - F(\mathbf{w}^*) \leq 16\|\mathbf{w}_0 - \mathbf{w}^*\|\Delta_B; \tag{17}$$

(3) (Non-convex) Under Assumption 1(c), with $\eta = 1/L$, after $t_m = (2L/\Delta_B^2)(F(\mathbf{w}_0) - F(\mathbf{w}^*))$ steps, we have

$$\min_{t=0,1,\ldots,t_m} \|\nabla F(\mathbf{w}_t)\| \leq \sqrt{2}\Delta_B, \tag{18}$$

in which

$$\Delta_B \lesssim \left(\frac{\epsilon}{1-2\epsilon}\sqrt{\frac{m}{N}} + \frac{1}{\sqrt{N}}\right)\sqrt{d\ln\frac{N}{\delta}}. \tag{19}$$

Now we compare (19) with (12). If ϵ is close to $1/2$, then the error is larger with unbalanced data sizes than the balanced case. In particular, the factor $1/(1-2\epsilon)$ in (19) is larger than the factor $1/\sqrt{1-2\epsilon}$ in (12). However, if ϵ is small, we can neglect the factor $1/(1-2\epsilon)$, thus the only difference between (19) and (12) is that the first term $\epsilon/\sqrt{n}$ in the bracket in (12) is now replaced by $\epsilon\sqrt{m/N}$. Recall that if samples are evenly distributed, then $n = N/m$, thus these two bounds are actually of the same order. Therefore, we have shown a somewhat surprising result that the statistical error rate is not affected by the unbalanced allocation of training samples in clients.

Theoretical Analysis for Non-I.I.D Case

In this section, we assume that clients are heterogeneous. In particular, suppose that for any $\mathbf{w}$, $\mu_i(\mathbf{w})$, $i = 1, \ldots, m$ are m i.i.d random variables with $\mathbb{E}[\mu_i(\mathbf{w})] = F(\mathbf{w})$. Furthermore, assume that $\mathbf{Z}_{ij}$ for $j = 1, \ldots, n_i$ are conditional independent given $\mu_i(\mathbf{w})$, and $\mathbb{E}[\nabla f(\mathbf{w}, \mathbf{Z}_{ij})|\mu_i(\mathbf{w})] = \mu_i(\mathbf{w})$. Now we replace Assumption 3 with the following new assumption.

Assumption 4. *(a) $\mu_i(\mathbf{w})$ is sub-exponential with respect to $\nabla F(\mathbf{w})$ with parameter σ_μ, i.e. For all λ such that $|\lambda| \leq 1/\sigma_\mu$,*

$$\sup_{\|\mathbf{v}\|=1} \mathbb{E}\left[e^{\lambda\mathbf{v}^T(\mu_i(\mathbf{w}) - \nabla F(\mathbf{w}))}\right] \leq e^{\frac{1}{2}\sigma_\mu^2\lambda^2}; \tag{20}$$

(b) $\mathbf{Z}_{ij}$ is sub-exponential with respect to $\mu_i(\mathbf{w})$ with parameter σ, i.e. For all λ such that $|\lambda| \leq 1/\sigma$,

$$\sup_{\|\mathbf{v}\|=1} \mathbb{E}\left[e^{\lambda\mathbf{v}^T(\nabla f(\mathbf{w}, \mathbf{Z}_{ij}) - \mu_i(\mathbf{w}))}\right] \leq e^{\frac{1}{2}\sigma^2\lambda^2}, \tag{21}$$

in which $\mathbf{Z}_{ij}$, $j = 1, \ldots, n_i$ are conditional i.i.d given $\mu_i(\mathbf{w})$.

Assumption 4(a) allows heterogeneous data. However, $\mu_i(\mathbf{w})$ follows a sub-exponential distribution with parameter σ_μ, thus the distance can not be too large. (b) requires that within each client, $\mathbf{Z}_{ij}$ is sub-exponential with respect to $\mu_i(\mathbf{w})$ for client i. At the limit of $\sigma_\mu \to 0$, Assumption 4 reduces to Assumption 3. Some existing works assume that the clients are completely different, such as (Ghosh et al. 2019), which groups clients into some clusters with inherently different properties. However, we assume that training samples are collected from sources that are only moderately different. The theoretical result is shown as following.

Theorem 3. *Let* $T_i = T_{0a} + T_{0b} + M/\sqrt{n_i}$*, with* $M \sim \sigma\sqrt{d\ln(N/\delta)}$*, and*

$$\epsilon\sigma\sqrt{\frac{md}{N}\ln\frac{N}{\delta}} \lesssim T_{0a} \lesssim \sigma\sqrt{\frac{md}{N}\ln\frac{N}{\delta}}, \tag{22}$$

and

$$T_{0b} \sim \sigma\sqrt{(d/N)\ln(N/\delta)}, \tag{23}$$

then under Assumption 2 and 4, the following equations holds with probability at least $1-2\delta$*.*

(1) (Strong convex) Under Assumption 1(a), with $\eta \le 1/L$*,*

$$\|\mathbf{w}_t - \mathbf{w}^*\| \le (1-\rho)^t \|\mathbf{w}_0 - \mathbf{w}^*\| + \frac{2\Delta_C}{\mu}, \tag{24}$$

in which $\rho = \eta\mu/2$*;*

(2) (General convex) Under Assumption 1(b), with $\eta = 1/L$*, after* $t_m = (L/\Delta_C)\|\mathbf{w}_0 - \mathbf{w}^*\|_2$ *steps,*

$$F(\mathbf{w}_{t_m}) - F(\mathbf{w}^*) \le 16\|\mathbf{w}_0 - \mathbf{w}^*\|\Delta_C; \tag{25}$$

(3) (Non-convex) Under Assumption 1(c), with $\eta = 1/L$*, after* $t_m = (2L/\Delta_C^2)(F(\mathbf{w}_0) - F(\mathbf{w}^*))$ *steps, we have*

$$\min_{t=0,1,\ldots,t_m} \|\nabla F(\mathbf{w}_t)\| \le \sqrt{2}\Delta_C, \tag{26}$$

in which

$$\Delta_C \lesssim \left(\frac{\epsilon}{1-2\epsilon}\sqrt{\frac{m}{N}} + \frac{1}{\sqrt{N}} + \frac{\sigma_\mu\sqrt{\sum_{i=1}^m n_i^2}}{N}\right)\sqrt{d\ln\frac{N}{\delta}} + \epsilon\sigma_\mu d\ln\frac{N}{\delta}. \tag{27}$$

Implementation

Our design follows the Weiszfeld's algorithm for geometric median (Weiszfeld and Plastria 2009). Suppose there are m vectors, $\mathbf{X}_1, \ldots, \mathbf{X}_m$. Define

$$\mathbf{c} = \arg\min_s \sum_{i=1}^m \phi_i(\|\mathbf{s} - \mathbf{X}_i\|) \tag{28}$$

as the center that minimizes the multi-dimensional Huber loss, in which ϕ_i is defined in (6). Suppose that the algorithm starts from $\mathbf{c}_0$. Then the update rule is

$$\mathbf{c}_{k+1} = \frac{\sum_{i=1}^m \min\left\{1, \frac{T_i}{\|\mathbf{c}_k - \mathbf{X}_i\|}\right\}\mathbf{X}_i}{\sum_{i=1}^m \min\left\{1, \frac{T_i}{\|\mathbf{c}_k - \mathbf{X}_i\|}\right\}}. \tag{29}$$

Our algorithm repeats (29) until convergence. Similar to Weiszfeld's algorithm, despite the fast convergence in almost all practical cases, it is not theoretically guaranteed in general. In section 8 in (Beck and Sabach 2015), it is shown that under some assumptions, an estimate of geometric median with Weiszfeld's algorithm with error tolerance τ needs $O(1/\tau)$ steps. With minor modification, the analysis in (Beck and Sabach 2015) also holds for our new algorithm (29). Moreover, from (29), each step requires $O(md)$ time, thus the overall time complexity is $O(md/\tau)$.

In the future, it is possible to extend some recent works on geometric median, such as (Feldman and Langberg 2011; Cohen et al. 2016) to improve the update rule (29) for multi-dimensional Huber loss minimization.

Comparison with Related Work

Now we compare our results with several existing popular approaches, including Krum (Blanchard et al. 2017), geometric median-of-means (Chen, Su, and Xu 2017) (GMM), coordinate-wise median (Yin et al. 2018) (CWM), coordinate-wise trimmed mean (Yin et al. 2018) (CWTM), and recent methods based on high dimensional robust statistics (Shejwalkar and Houmansadr 2021; Zhu et al. 2023) (HDRS). To begin with, we consider the i.i.d case. In particular, we compare the following five aspects listed as following:

- Whether the method relies on precise knowledge of ϵ;
- Under sub-exponential and strong convex assumption, with sufficiently large number of iterations t, whether the statistical error rate of $\|\hat{\mathbf{w}} - \mathbf{w}^*\|$ is optimal (or nearly optimal up to a logarithm factor) in ϵ. Here the optimal rate is $\epsilon/\sqrt{n} + \sqrt{d/N}$ (Hopkins and Li 2019; Yin et al. 2018);
- Whether the performance is good if ϵ is close to $1/2$. In particular, whether error blows up by a factor of no more than $1/\sqrt{1-2\epsilon}$, as is shown in (12);
- Whether the time complexity of aggregator is linear or nearly linear, i.e. $O(m)$ or $O(m\log m)$;
- Whether the error rate is optimal in dimensionality d;

The results are shown in Table 1, in which the five aspects mentioned above correspond to its five columns.

Method	ϵ-agnostic	ϵ-opt.	near $1/2$	linear	d-opt.
Krum	No	No	No	No	No
GMM	No	No	Yes	Yes	No
CWM	Yes	No	Yes	Yes	No
CWTM	No	Yes	No	Yes	No
HDRS	No	No	No	No	Yes
Ours	**Yes**	**Yes**	**Yes**	**Yes**	**No**

Table 1: Comparison of our method with existing aggregators under i.i.d assumption in five aspects. "ϵ-agnostic" means that the method does not rely on precise knowledge of ϵ. "ϵ-opt." refers to optimal dependence of error rate on ϵ. Besides, "near $1/2$" means good performance with ϵ close to $1/2$. Moreover, "linear" stands for linear time complexity in m. Finally, "d-opt." means optimality of error rate in d.

For the first aspect, only coordinate-wise median and our method do not rely on precise knowledge of ϵ. For other methods, ϵ significantly affects the parameter selection. For example, coordinate-wise trimmed mean need to set the trim threshold to be ϵ in both sides. However, it is quite unlikely to have exact knowledge of ϵ in practice. If an upper bound α is known, such that $\epsilon < \alpha$, then α can be used to set the parameter, but the accuracy will be sacrificed. Secondly, for optimal dependence in ϵ, Krum is not guaranteed to converge globally. Coordinate-wise median is not optimal if $n \lesssim m$. Geometric median-of-mean only has a $\tilde{O}(\sqrt{\epsilon d/n})$ dependence. Thirdly, for the performance with ϵ close to $1/2$, coordinate-wise trimmed mean is not optimal. In particular, the error rate Δ_{CWTM}, which plays the same role as

Δ_A in Theorem 1, is

$$\Delta_{CWTM} = \tilde{O}\left(\frac{d}{1-2\epsilon}\left(\frac{\epsilon}{\sqrt{n}} + \frac{1}{\sqrt{N}}\right)\right), \tag{30}$$

which blows up by a factor of $1/(1-2\epsilon)$, higher than $1/\sqrt{1-2\epsilon}$. Intuitively, with ϵ close to $1/2$, coordinate-wise trimmed mean is not efficient because it removes most of samples, thus somewhat wastes the data. For the fourth column in Table 1, the time complexity has been discussed in previous section.

The only drawback of our method is that the dependence on d is not optimal. In recent years, there are some new proposed methods for high dimensional robust mean estimation (Diakonikolas et al. 2016, 2017, 2021). These methods can be used as the aggregator function in federated learning (Su and Xu 2018; Shejwalkar and Houmansadr 2021; Zhu et al. 2023), so that the dependence on d becomes optimal. However, the ϵ dependence is $\sqrt{\epsilon}$, which is optimal only if we merely require the covariance matrix to have bounded operator norms. Under a more restrictive sub-exponential assumption, the convergence can not be further improved. Moreover, the time complexity is much higher.

One may wonder if it is possible to design an aggregator that satisfy all of the five properties listed in Table 1. However, (Hopkins and Li 2019) has shown that for robust mean estimation problem, as long as $P \neq NP$, under sub-exponential assumption, polynomial time complexity, optimal dependence in ϵ, and constant factor in d are three goals that can not be achieved together. Since FL relies on robust mean estimation of gradient vectors as the aggregator function, we conjecture that it is impossible to satisfy all five properties in Table 1.

Right now we have compared our results with existing methods under i.i.d assumptions. There are also some related work focusing on non-i.i.d cases, but the assumptions are crucially different. For example, (Li et al. 2019) proposed RSA. Instead of designing a gradient aggregator, (Li et al. 2019) conducts model aggregation, which tries to reach a consensus between different models. This method is suitable for the cases in which the distributions in clients are significantly different. However, under the i.i.d limit, (Li et al. 2019) does not have bounds comparable to existing methods under i.i.d assumption.

Numerical Results

This section shows numerical experiments. Despite the fact that our method has the advantage of not relying on knowledge of ϵ, in this section, we just assume that all baseline methods know ϵ exactly, including Krum, GMM, CWM and CWTM mentioned in the previous section. The detailed algorithms and parameter selection rules are shown in Section 1 in the supplementary material. If ϵ is unknown, then the advantage of our method should be larger than what is described in this section.

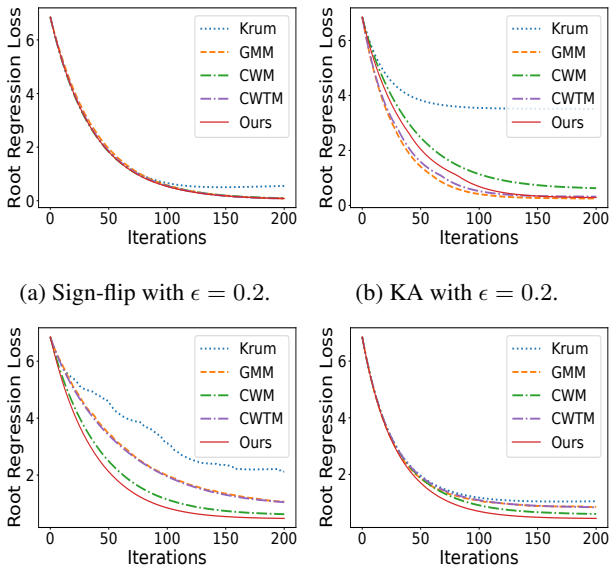
Ideally, robustness against Byzantine failure needs to be tested using optimal attack strategies. However, the optimization problems are hard to solve. In this section, we use four attack strategies. One of them is a simple sign-flip attack, which flips the sign of gradient vectors. Other three are some approximate strategies that are tailored to specific aggregators, including Krum Attack (KA) and Trimmed Mean Attack (TMA) described in (Fang et al. 2020) that are nearly optimal for Krum and CWTM, respectively. Moreover, for a fair comparison, we have designed an attack strategy for our new proposed method, called Huber Loss Minimization Attack (HLMA). The details of all these attacks are shown in the supplementary material.

Synthesized Data

To begin with, we run experiments with distributed linear regression. The model is

$$V_j = \langle \mathbf{U}_j, \mathbf{w}^* \rangle + W_j, \tag{31}$$

in which $\mathbf{U}_j, \mathbf{w}^* \in \mathbb{R}^d$, with $\mathbf{w}^*$ being the true parameter, and W_j is the noise following standard normal distribution. In this experiment, we set $d = 50$. Firstly, we generate all elements of $\mathbf{w}^*$ from distribution $\mathcal{N}(0,1)$. We then obtain $N = 10,000$ samples $(\mathbf{U}_j, V_j)$, $j = 1, \ldots, N$. These samples are evenly divided into $m = 500$ clients. For baseline methods including Krum, GMM, CWTM, the parameters are all set optimally, according to Section 1 in the supplementary material. Our new Huber loss minimization approach uses $T = 1$ for all clients. We run 200 iterations in total, with learning rate $\eta = 0.02$. The results are shown in Figure 1 for all four attack strategies mentioned above, in which we plot the square root of the ℓ_2 regression loss against the number of iterations.



(a) Sign-flip with $\epsilon = 0.2$. (b) KA with $\epsilon = 0.2$.

(c) TMA with $\epsilon = 0.2$. (d) HLMA with $\epsilon = 0.2$.

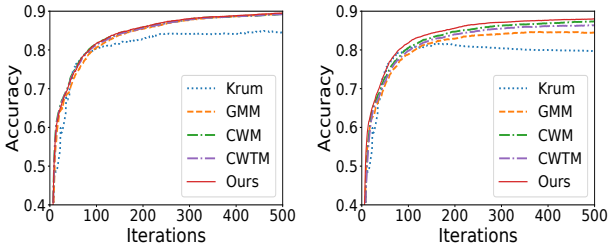
Figure 1: Comparison of our new method and several baselines against Krum Attack and Trimmed Mean Attack for synthesized data with $\epsilon = 0.2$.

From Figure 1, it can be observed that our new method (red solid curve) works well for all four types of attacks, even with HLMA that is specifically designed for our

method. On the contrary, from Figure 1 (b), Krum aggregator (blue dotted curve) fails under KA that is tailored to Krum. Moreover, (c) shows that under TMA which is designed for coordinate-wise trimmed mean, GMM and CWTM (dashed curves with orange and purple colors, respectively) perform significantly worse than our method. CWM appears to be robust, but the performance is not as good as our approach in general.

Real Data

Now we use MNIST dataset (LeCun 1998), which has $60,000$ images of handwritten digits for training, and $12,000$ images for testing. The sizes of these images are 28×28. In this experiment, we use a neural network with one hidden layer between input and output. The size of hidden layer is 32. Training samples are evenly allocated into $m = 500$ clients. We set $T = 0.2$ here. The results with $\epsilon = 0.2$ is shown in Figure 2, while the case with $\epsilon = 0.4$ is shown in the supplementary material, in which we plot the curve of accuracy on the test data against the number of iterations.



(a) SignFlip with $\epsilon = 0.2$. (b) KA with $\epsilon = 0.2$.

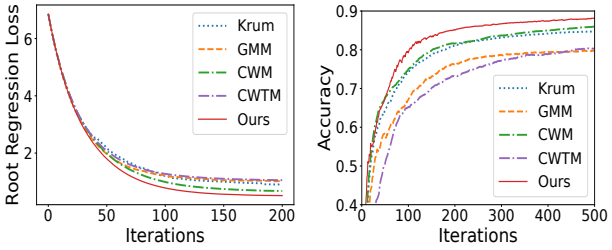
(c) TMA with $\epsilon = 0.2$. (d) HLMA with $\epsilon = 0.2$.

Figure 2: Comparison of our new method and several baselines against sign-flip, KA, TMA and HLMA for MNIST data, with $\epsilon = 0.2$.

Similar to synthesized data, experiments on MNIST show that our new method outperforms other methods. Krum is still highly susceptible to KA. CWM and CWTM appear to be only slightly worse than our method. However, in the supplementary material, we show that with $\epsilon = 0.4$, under TMA, these two methods are much worse than ours, especially CWTM. These results agree with our discussion in the previous section.

Unbalanced Sample Allocation

We then evaluate the adaptive selection rule (14) for unbalanced data. In this experiment, samples are still allocated in different clients randomly, but the sample sizes are no longer equal. Details of the allocation rule are shown in Section 3 in the supplementary material. We use $T_i = 2/\sqrt{n_i}$ as the threshold of each clients. Other settings remain the same as previous experiments. For simplicity, we only show the result with HLMA in Figure 3.



(a) Synthesized data. (b) MNIST data.

Figure 3: Experiments on unbalanced data for HLMA, with $\epsilon = 0.2$.

The result shows that with unbalanced data and our adaptive choice of T_i, the advantage of our method becomes more obvious, especially for MNIST data. The comparison of Figure 3(b) with Figure 2(d) shows that our method achieves nearly the same performance as the balanced case, while other methods are somewhat negatively affected by the unbalanced data allocation.

Finally, we also conduct experiments for heterogeneous data. In particular, (31) is slightly modified, such that clients have different distributions. In this case, numerical experiments show that the performances of all methods are negatively affected by the heterogeneity, but our method is relatively more stable. Details are shown in Section 4 in the supplementary material.

Conclusion

In this paper, we have proposed a novel approach for Byzantine robust federated learning based on Huber loss minimization. Theoretical analyses have been conducted under the initial i.i.d assumption with balanced data, and subsequently extended to unbalanced and heterogeneous scenarios. Our method offers several advantages over existing approaches, including optimal statistical rate with fixed dimension, convenient selection of parameters without knowledge of Byzantine fraction ϵ, and suitability for clients with unbalanced data. Furthermore, we have presented an algorithm to implement the multi-dimensional Huber loss minimization. The effectiveness of our approach is validated by numerical experiments.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.62132008, 62272425 and

61972229), the Key Research Project of Zhejiang Lab (No.2022PD0AC02, 2022PD0AC01 and K2022PD1BB01), the Natural Science Foundation of Jiangsu Province (BK20220075) and the Fok Ying-Tong Education Foundation for Young Teachers in the Higher Education Institutions of China (No.20193218210004).

References

Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; and Shmatikov, V. 2020. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, 2938–2948.

Beck, A.; and Sabach, S. 2015. Weiszfeld's method: Old and new results. *Journal of Optimization Theory and Applications*, 164: 1–40.

Blanchard, P.; El Mhamdi, E. M.; Guerraoui, R.; and Stainer, J. 2017. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, volume 30.

Cao, X.; and Lai, L. 2019. Distributed gradient descent algorithm robust to an arbitrary number of byzantine attackers. *IEEE Transactions on Signal Processing*, 67(22): 5850–5864.

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

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

Diakonikolas, I.; Kamath, G.; Kane, D. M.; Li, J.; Moitra, A.; and Stewart, A. 2016. Robust Estimators in High Dimensions without the Computational Intractability. In *57th Annual Symposium on Foundations of Computer Science*, 655–664.

Diakonikolas, I.; Kamath, G.; Kane, D. M.; Li, J.; Moitra, A.; and Stewart, A. 2017. Being robust (in high dimensions) can be practical. In *International Conference on Machine Learning*, 999–1008.

Diakonikolas, I.; Kane, D.; Kongsgaard, D.; Li, J.; and Tian, K. 2021. List-decodable mean estimation in nearly-pca time. In *Advances in Neural Information Processing Systems*, volume 34, 10195–10208.

Fang, M.; Cao, X.; Jia, J.; and Gong, N. 2020. Local model poisoning attacks to Byzantine-Robust federated learning. In *29th USENIX security symposium (USENIX Security 20)*, 1605–1622.

Feldman, D.; and Langberg, M. 2011. A unified framework for approximating and clustering data. In *Proceedings of the forty-third annual ACM Symposium on Theory of Computing*, 569–578.

Ghosh, A.; Hong, J.; Yin, D.; and Ramchandran, K. 2019. Robust federated learning in a heterogeneous environment. *arXiv preprint arXiv:1906.06629*.

Hall, P.; and Jones, M. 1990. Adaptive M-estimation in nonparametric regression. *The Annals of Statistics*, 1712–1728.

Hopkins, S. B.; and Li, J. 2019. How Hard is Robust Mean Estimation? In *Conference on Learning Theory*, volume 99, 1649–1682.

Huber, P. J. 1964. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 73–101.

Huber, P. J. 2004. *Robust statistics*, volume 523. John Wiley & Sons.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2): 1–210.

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

LeCun, Y. 1998. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

Li, L.; Fan, Y.; Tse, M.; and Lin, K.-Y. 2020. A review of applications in federated learning. *Computers & Industrial Engineering*, 149: 106854.

Li, L.; Xu, W.; Chen, T.; Giannakis, G. B.; and Ling, Q. 2019. RSA: Byzantine-Robust Stochastic Aggregation Methods for Distributed Learning from Heterogeneous Datasets. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI*, 1544–1551.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282.

Pillutla, K.; Kakade, S. M.; and Harchaoui, Z. 2022. Robust aggregation for federated learning. *IEEE Transactions on Signal Processing*, 70: 1142–1154.

Shejwalkar, V.; and Houmansadr, A. 2021. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *28th Annual Network and Distributed System Security Symposium, 2021*. The Internet Society.

Su, L.; and Xu, J. 2018. Securing distributed machine learning in high dimensions. *arXiv preprint arXiv:1804.10140*, 1536–1233.

Weiszfeld, E.; and Plastria, F. 2009. On the point for which the sum of the distances to *n* given points is minimum. *Annals of Operations Research*, 167(1): 7–41.

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2): 1–19.

Yin, D.; Chen, Y.; Kannan, R.; and Bartlett, P. 2018. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International Conference on Machine Learning*, 5650–5659.

Zhao, P.; and Wan, Z. 2023. Robust Nonparametric Regression under Poisoning Attack. *arXiv preprint arXiv:2305.16771*.

Zhu, B.; Wang, L.; Pang, Q.; Wang, S.; Jiao, J.; Song, D.; and Jordan, M. I. 2023. Byzantine-Robust Federated Learning with Optimal Statistical Rates. In *International Conference on Artificial Intelligence and Statistics*, 3151–3178.