Distributed Robust Learning

Jiashi Feng  
EECS Department & ICSI  
UC Berkeley  
jshfeng@berkeley.edu

Huan Xu  
ME Department  
National University of Singapore  
mpexuh@nus.edu.sg

Shie Mannor  
EE Department  
Technion  
shie@ee.technion.ac.il

Abstract

We propose a generic distributed learning framework for robust statistical learning on big contaminated data. The Distributed Robust Learning (DRL) framework can reduce the computational cost of traditional robust learning methods by several orders. We provide a sharp analysis on the robustness of DRL, showing that DRL not only preserves the robustness of base robust learning methods, but also tolerates breakdowns of a constant fraction of computing nodes. Moreover, DRL can enhance the breakdown point of existing robust learning methods to be even larger than 50%, under favorable conditions. This enhanced robustness is in sharp contrast with the naive divide and fusion method where the breakdown point may be reduced by several orders. We specialize the DRL framework for two concrete cases: distributed robust PCA and distributed robust regression. We demonstrate the efficiency and the robustness advantages of DRL through comprehensive simulations.

1 Introduction

In the modern era of big data, traditional statistical learning methods are facing many great challenges. The two most significant challenges are: (1) how to scale the current machine learning methods to the large-scale data, and (2) how to obtain the accurate inference results when the data are noisy and may even contain malicious outliers. These two important problems make the scalable robust learning methods become heavily desired.

In robust learning, a statistician receives \(n + n_1\) samples of the form \(\{x_i\}_{i=1}^{n+n_1}\) for unsupervised learning or \(\{(x_i, y_i)\}_{i=1}^{n+n_1}\) for supervised learning. Here \(x_i \in \mathbb{R}^p\) is an observation and \(y_i \in \mathbb{R}\), if exists, is a real-valued response of \(x_i\). Among the \(n + n_1\) data points, \(n_1\) of them may be corrupted by gross noise or are possibly malicious outliers. The goal of robust learning is then to estimate the parameter \(\theta \in \Theta\) of interest with guaranteed closeness to the ground truth \(\theta^*\), even though a constant fraction of outliers may exist in the data.

An example par excellence of robust learning is robust principal component analysis (RPCA) [11, 8, 9]. RPCA aims to estimate the low-dimensional subspace fitting the inliers \(\{x_i\}_{i=1}^{n}\) while being resistant to the negative effect of the outliers \(\{x_i\}_{i=n+1}^{n+n_1}\). Another example is robust linear regression (RLR) [5, 12, 4], where the inlier samples \((x_i, y_i)\) can be described by the linear model \(y_i = \theta^\top x_i\). Also, RLR aims to estimate the regression parameter \(\theta\) without being affected by outliers. Notice that we use the term “inlier” to denote authentic samples generated according to the underlying statistics rule, which does not imply these samples are closer to the origin.
Traditional robust learning methods generally rely on optimizing certain robust statistics [14, 16] or applying some sample trimming strategies [8, 9], whose calculations require loading all the samples into the memory or going through the data multiple times [10]. Thus, the computational cost of those robust learning methods is usually at least linearly dependent on the size of the sample set, \( n_1 + n_2 \). For example, in RPCA [16], the computational cost is \( O((n_1 + n_2)^2 r) \) where \( r \) is the intrinsic dimension of subspace. In robust linear regression [4], the computational cost is super-linear on the sample size: \( O(p(n_1 + n_2) \log(n_1 + n_2)) \). This rapidly increasing computation cost becomes a major obstacle to applying robust learning methods to big data in practice, where the sample size easily reaches the terabyte or even petabyte scale.

Recent years, along with the rapidly increasing of data, distributed learning methods become popular and necessary. Among them, the most popular one is simply MapReduce or divide and fusion. The data are uniformly distributed over several parallel machines and the computation results from the machines are simply fused by taking their average. Such MapReduce framework is able to enhance the computation efficiency by several orders with negligible communication cost. However, naively implementing the robust learning algorithms in such MapReduce framework could destroy the robustness of the algorithm. How to preserve the robustness of the algorithms in the parallel implementation is still an open problem.

In this work, we propose a generic framework for Distributed Robust Learning (DRL) to efficiently process big data yet preserving robustness to outliers. The implementation of DRL follows the strategy similar to map-reduce [7]: DRL first distributes all the samples evenly onto \( k \) machines. Then it implements a base robust learning algorithm to generate an estimation on each of the \( k \) machines. Finally, it merges the \( k \) individual estimates via an efficient and robust aggregation operation. The framework is compatible with any existing robust learning methods and is able to enhance their computational efficiency with a constant factor of at least \( k \) with guaranteed robustness.

The parallel implementation potentially allows significant computational speed-ups and the ability to cope with big data. However, data communication rates between machines is typically slower than their processing speeds. Here we specify the benefits of DRL on reducing the memory and computation cost, and show its communication cost is negligible. (1) Communication cost. In the sample division step, samples can be directly assigned to corresponding machines without communication between different machines. In the aggregation step, the \( i \)th machine only needs to send its estimator \( \hat{\theta}_i \) to a specific machine. Therefore, the total communication cost of DRL is only \( k \cdot s \), where \( s \) is the size of \( \hat{\theta}_i \). (2) Memory cost. Each machine needs to store samples with the size of \( p(n_1 + n_2)/k \). Compared with a single machine case, the memory cost of each machine is reduced by a factor of \( k \). (3) Computation cost. Since the sample size on each machine is reduced by a factor of \( k \) and the computational cost of base robust learning method is (super-)linear with respect to the sample size. The computation cost on each machine may also be reduced by a factor of at least \( k \).

Besides its obvious advantage of enhancing the computation efficiency for handling big data, DRL may also increase the robustness (measured by the breakdown point\(^1\)) of the base robust learning methods. Traditionally, it is deemed that 50% is the maximal breakdown point which can ever be achieved by any robust learning method [11, 16]. However, for distributed learning, the case can be different. At the worst case, DRL also has a breakdown point of 50%. However, under the following favorable cases, DRL may achieve stronger robustness with breakdown point larger than 50%.

In many practical cases, the state of the samples – being inlier or outlier – may have dependence on their neighbors. Examples include time series data, Markov chain data [1], and images/videos data where spatial correlation is common [3]. Here, the outliers need not be uniformly distributed within the samples, and the fractions of outliers on different machines in DRL are hence no longer identical. Though the machines having too many outliers may individually break down, DRL is able to tolerate their bad performances as aforementioned and take advantage of the machines having less outliers to get better estimation. Under these cases, DRL is able to achieve larger breakdown point. We will make this claim precise in the following sections and provide two concrete examples: distributed robust PCA and distributed robust regression.

\(^1\)Breakdown point is a robustness measure defined as the percentage of corrupted points that can make the output of an algorithm arbitrarily bad.
1.1 Related Works

Distributed learning methods can be roughly divided into two categories: the decentralized gossip-type algorithms [2] and the algorithms constructed on the map-reduce framework [7]. Gossip algorithms do not require a center node to aggregate the results, while they suffer from high communication cost.

The most relevant methods to the proposed DRL are the divide-and-conquer methods [18, 17, 13], which are built on the map-reduce framework. In those works, similar to us, the samples are evenly distributed on the machines and processed in parallel. However, those methods take the simple average of the estimates as the final output, which is not robust to corruption [11]. If one machine breaks down, the final estimation can be arbitrarily bad.

Recently, several online robust learning algorithms were proposed to process the data in a sequential manner [10]. Online learning methods partially mitigate the scalability issue of robust learning by reducing the memory cost for machines. However, the time complexity of those methods depends linearly on the sample size, hardly affordable in practice when dealing with ultra-large datasets.

2 Distributed Robust Learning Framework

2.1 Problem Setup

Given a set of \( n + n_1 \) observations \( X = [x_1, \ldots, x_{n+n_1}] \in \mathbb{R}^{p \times (n+n_1)} \), it contains a mixture of \( n \) authentic samples and \( n_1 \) outliers. Here \( p \) is the dimension of the samples. The authentic samples are generated according to the underlying model (i.e., ground truth) parameterized by \( \theta \in \Theta \). For example, for the problem of Principal Component Analysis (PCA), the authentic samples are generated from a low rank matrix \( \theta \in \mathbb{R}^{p \times d} \) such that \( x = \theta z + e \). Here \( z \in \mathbb{R}^d \) is the underlying signal and \( e \in \mathbb{R}^p \) denotes the additive noise. For the linear regression problem, the observations are pairs of covariate-response \((x, y) \in \mathbb{R}^p \times \mathbb{R}\). The authentic samples follow the linear model parameterized by \( \theta \in \mathbb{R}^p \): \( y = \langle \theta, x \rangle + e \), where \( e \) again denotes the noise. The outliers are the samples arbitrarily corrupted or even maliciously chosen, which do not follow the model that generates the authentic samples. Throughout this paper, we use \( \lambda = \frac{n_1}{n+n_1} \) to denote the outlier fraction.

2.2 Distributed Robust Learning

The implementation of the proposed distributed robust learning (DRL) framework is quite simple. Suppose there are \( k \) machines ready to use. DRL evenly divides the sample set \( \{x_1, \ldots, x_{n+n_1}\} \) into \( k \) subsets. Each subset, denoted as \( X_j \), is assigned to one corresponding machine. On each machine, an appropriate base robust learning algorithm (such as robust PCA or robust regression) is applied to estimate the interested parameter \( \hat{\theta} \) individually and output its estimation \( \hat{\theta}_j \). Then the \( k \) estimators \( \hat{\theta}_1, \ldots, \hat{\theta}_k \) are communicated to and aggregated on a single machine. A merge operation follows to combine the \( k \) estimations to yield the final estimation \( \tilde{\theta} \).

Previous divide-and-conquer distributed learning works [18, 17, 13] commonly merge the \( k \) separate estimations by taking their average, i.e.,

\[
\tilde{\theta} = \frac{1}{k} \sum_{j=1}^{k} \hat{\theta}_j.
\]

However, the above average is fragile to outliers [11] with zero breakdown point. Suppose one of the estimator \( \hat{\theta}_j \) is corrupted (this may be caused by too many outliers in \( X_j \), machine breakdown, or errors when communicating the individual output), the final estimation can be arbitrarily bad. Thus, those methods are not robust and cannot handle sample outliers or machine breakdown. More concretely, suppose the breakdown point of the base learning method is \( \lambda^* \), then in the worst case, the breakdown point of the learning algorithm will be reduced to \( \lambda^*/k \) due to such non-robust fusion operation.

Instead of taking the average of the estimations, DRL calculates the geometric median of the estimations for its robustness [11]. Geometric median is formally defined as follows [15].
**Definition 1** (Geometric Median). Given a finite collection of points \( \theta_1, \ldots, \theta_k \in \Theta \), the geometric median is the point which minimizes the total distance to all the given points, i.e.,

\[
\tilde{\theta} = \text{median}(\theta_1, \ldots, \theta_k) := \arg \min_{\gamma \in \Theta} \sum_{j=1}^{k} \| y - \theta_j \|.
\]

Calculating the geometric median is a convex optimization problem, where any off-the-shelf convex problem solvers can be adopted. Due to the space limitation, we omit the optimization details.

Suppose we have a base robust learning algorithm, denoted as RL\((X, \eta)\) with input \(X\) and algorithmic parameter \(\eta\). DRL provides a distributed implementation of RL\((X, \eta)\), as shown in Algorithm 1.

**Algorithm 1** DRL framework

| Input: Sample matrix \(X = [x_1, \ldots, x_{n+n_1}] \in \mathbb{R}^{n \times (n+n_1)}\), number of available machines \(k\). Base robust learning algorithm RL\((., \eta)\). Column-wisely divide \(X\) evenly into \(k\) submatrices: \(X = [X_1, \ldots, X_k]\). |
| Do in parallel |
| \(\tilde{\theta}^{(1)} = \text{RL}(X_1, \eta)\) |
| \(\vdots\) |
| \(\tilde{\theta}^{(k)} = \text{RL}(X_k, \eta)\) |
| Output: \(\tilde{\theta} = \text{median}\left(\tilde{\theta}^{(1)}, \ldots, \tilde{\theta}^{(k)}\right)\) |

### 2.3 Robustness of DLR

The following lemma, straightforwardly derived from Lemma 2.1 in [15], characterizes the robustness of the geometric median.

**Lemma 1** (Robustness of the Geometric Median). Let \(\tilde{\theta}\) be the geometric median of the points \(\theta_1, \ldots, \theta_k \in \Theta\). Fix \(\gamma \in (0, \frac{1}{2})\) and \(C_{\gamma} = (1 - \gamma)\sqrt{\frac{1}{1 - 2\gamma}}\). Suppose there exists a subset \(J \subseteq \{1, \ldots, k\}\) of cardinality \(|J| > (1 - \gamma)k\) such that for all \(j \in J\) and any point \(\theta^* \in \Theta\), \(\|\theta_j - \theta^*\| \leq r\). Then we have \(\|\tilde{\theta} - \theta^*\| \leq C_{\gamma}r\).

In words, given a set of points, their geometric median will be close to a point of interest \(\theta^*\) as long as at least half of them are close to \(\theta^*\). In particular, the geometric median will not be skewed severely even if some of the points are unbounded far away from \(\theta^*\).

Based on the robustness of geometric median, we obtain the following result that characterizes the robustness of DLR to the corrupted estimations on a fraction of machines. Before presenting the details, we define the following necessary quantities. For \(0 < \beta < \alpha < \frac{1}{2}\), define their divergence as \(\psi(\alpha; \beta) = (1 - \alpha)\log \frac{1 - \beta}{1 - \alpha} + \alpha \log \frac{\alpha}{\beta}\) and let \(\beta^* = \beta^*(\alpha) := \max\{\beta \in (0, \alpha) : \psi(\alpha, \beta) \geq 1\}\).

**Theorem 1** (Robustness of DLR). Fix \(\alpha \in (0, \frac{1}{2})\). Assume that \(\theta \in \Theta\) is the parameter of interest. Let \(\hat{\theta}_1, \ldots, \hat{\theta}_k\) be a collection of independent estimations of \(\theta\) from \(k\) machines. Assume that \((1 - \gamma)k\) machines, where \(0 \leq \gamma < \frac{\alpha - \beta^*}{1 - \beta^*}\), provide good estimations satisfying \(\mathbb{P}\left(\|\hat{\theta}_j - \theta\| > \varepsilon\right) \leq \beta^* < \alpha\) and the estimations from the other \(\gamma k\) machines are corrupted arbitrarily. Let \(\tilde{\theta}\) be the output of DLR. Then

\[
\mathbb{P}\left(\|\tilde{\theta} - \theta\| < C_{\alpha}\varepsilon\right) \geq 1 - \exp\left(-k(1 - \gamma)\psi\left(\frac{\alpha - \gamma}{1 - \gamma}, \beta^*\right)\right),
\]

where \(C_{\alpha} = (1 - \alpha)\sqrt{\frac{1}{1 - 2\alpha}}\).

The proof of above theorem is straightforward from the Lemma 1 and Theorem 3.1 in [15]. Basically, Theorem 1 states that even \(\gamma k\) machines break down (because their outliers outnumber the
breakdown point of the base robust algorithm for example), DRL still guarantees that the final estimation has bounded error $C_\alpha \varepsilon$ with high probability.

Note that the above function of $\psi(\beta, \alpha)$ is monotonically decreasing with $\beta$ and monotonically increasing with $\alpha$. The function $C_\alpha$ is monotonically increasing in $\alpha$, which characterizes the bound relaxation after taking the geometric median.

**Remark 1 (Values of $\beta$ and $\alpha$).** To more explicitly appreciate the results given in Theorem 2 we provide some concrete results under specific values of $\beta^*$ and $\alpha$ in Table 1. From the results, we observe that $C_\alpha$ decreases rapidly with decreasing $\beta^*$. When the error bound holds with failure probability less than $\beta = 1 \times 10^{-3}$ for $(1 - \gamma)k$ machines, the value of $C_\alpha$ is very close to 1, which indicates that DRL indeed retains an almost same error bound as the batch method.

**Table 1: Exemplar values of $\beta^*$, $\alpha$ and $C_\alpha$ in Theorem 1**

| $\beta^*$       | $1 \times 10^{-2}$ | $5 \times 10^{-3}$ | $1 \times 10^{-3}$ | $1 \times 10^{-4}$ | $1 \times 10^{-5}$ |
|------------------|--------------------|--------------------|-------------------|-------------------|-------------------|
| $\alpha$         | 0.358              | 0.32               | 0.22              | 0.156             | 0.119             |
| $C_\alpha$       | 1.205              | 1.13               | 1.04              | 1.018             | 1.009             |

**Remark 2 (Trade-off between efficiency and accuracy).** In Theorem 2 there is implicit dependence between the result accuracy and the sample size in each machine through $\beta^*$. For example, in previous robust PCA [10], $\beta^* \sim e^{-cn}$ with $n$ being the sample size. Smaller sample size may incur larger failure probability $\beta^*$. Therefore, we need to trade-off between the efficiency (which favors increasing the number of parallel machines, $k$) and accuracy (which favors decreasing $k$ to increase the sample size on each machine) in practice.

### 2.4 Overcoming The Barrier of The Breakdown Point

In many real-world datasets such as time series data, images and videos, the outliers may not be uniformly mixed with the inliers. We use following quantities to describe the non-uniformness of the outlier distribution on the $k$ machines. Suppose the total outlier fraction is $\lambda$. Let $\gamma < 1$ denote the fraction of machines on which the outlier fraction is lower bounded by $\rho \lambda$ with $\rho \geq 1$. For simplicity, we assume $\gamma k$ is an integer. Let the vector $\mu_0 = \left[ \frac{1 - \rho \lambda}{1 - \gamma}, \ldots, \frac{1 - \rho \lambda}{1 - \gamma} \right] \in \mathbb{R}^{(1-\gamma)k}$ represent the outlier fractions, when the rest outliers uniformly distribute, on the other $(1 - \gamma)k$ machines. Let $\mu = \mathbb{E}(1-\gamma)^k$ denote the actual outlier fractions on the $(1 - \gamma)k$ machines. We use $d_\mu = \|\mu_0 - \mu\|_\infty$ to measure how far the actual outlier distribution away from uniform distribution.

We here demonstrate that the proposed DRL framework (Algorithm 1) actually benefits from the non-uniform distribution of outliers: the breakdown point of the algorithm may improve from that of the base robust learning methods, under favorable conditions.

**Theorem 2.** Suppose the breakdown point of the base robust learning algorithm is $\lambda^*$. Outliers are not uniformly distributed on the $k$ machines. In particular, on $k' = \gamma k$ with $0 \leq \gamma \leq \frac{\alpha - \beta^*}{1 - \beta^*}$ machines, the fraction of outliers is lower bounded by $\rho \lambda^*$ with $1 \leq \rho < \frac{1}{2}$. If $d_\mu = \|\mu - \mu_0\|_\infty \leq \frac{\gamma(\mu - 1)\lambda^*}{1 - \gamma}$, we have the breakdown point of DRL is $\lambda_{DRL} = \frac{1}{1 - \rho^2} (\lambda^* - d_\mu) \geq \lambda^*$.

In the above theorem, $\rho$, $\gamma$ and $d_\mu$ together describe how non-uniform the outliers are distributed over the machines. When $\rho \rightarrow \frac{1}{2}$, the outliers concentrate on the $\gamma k$ machines, and DRL achieves nearly “infinite” robustness as no outliers are on the other $(1 - \gamma)k$ machines. When $d_\mu = 0$, the outliers uniformly distribute on the non-corrupted $(1 - \gamma)k$ machines, and DRL provides maximal breakdown point enhancement: $\lambda_{DRL} = \frac{1}{1 - \rho^2} \lambda^*$. When $\rho = 1$, the outlier fractions on all the machines are equal to $\lambda^*$. DRL provides no robustness gain as $\lambda_{DRL} = \lambda^*$. We defer the proof of Theorem 2 to the supplementary material. More concretely, we provide the robustness of DRL under three specific and common cases in the following corollary.

**Corollary 1.** Suppose the breakdown point of the base robust learning algorithm is $\lambda^*$. In the average case where the outliers are uniformly distributed on the $k$ machines, then the breakdown point of DRL is also $\lambda^*$. In the most favorable case, where all of the outliers exactly distributed on $(1/2 - \epsilon)k$ machines with $\epsilon$ being an very small positive scale, then the breakdown point of...
DRL is asymptotically increased to $3\lambda^*/2$ when $\epsilon \to 0$. In contrast, in the adversarial case where $(1/2+\epsilon)k$ machines have more outliers than inliers and $(1/2-\epsilon)k$ machines only have inliers, then the breakdown point of DRL is asymptotically reduced to $\lambda^*/2$ when $\epsilon \to 0$.

In the following two sections, we give two practical examples to instantiate the proposed DRL framework and demonstrate how it enhances the robustness (measured using the breakdown point).

3 Example I: Distributed Robust PCA

PCA is known to be fragile to outliers and many robustified PCA methods have been proposed (See [16] and the references therein). Here we develop a novel robust PCA method, which is presented by us in this work, as an illustrating example, which robustifies PCA via a robust sample covariance matrix estimation.

Given a sample matrix $X = [x_1, x_2, \ldots, x_n]$, the standard covariance matrix is computed as $C = XX^\top$, i.e., $C_{ij} = \langle x_i, x_j \rangle, \forall i, j = 1, \ldots, p$. Here $X_i$ denotes the $i$th row vector of matrix $X$. To obtain a robust estimator of the covariance matrix, we replace the vector inner product by a trimmed inner product, $\hat{C}_{ij} = \langle x_i, x_j \rangle_{\Omega}$, as detailed in Algorithm 2. Intuitively, the trimmed inner product removes the outliers having large magnitude and the remaining outliers are bounded by inliers. Thus, the obtained covariance matrix is close to the inlier sample covariance matrix.

Algorithm 2 Trimmed inner product $\langle x, x' \rangle_{\Omega}$

**Input:** Two vectors $x \in \mathbb{R}^N$ and $x' \in \mathbb{R}^N$, trimmed parameter $n_1$.

Compute $q_i = x_i x_i', i = 1, \ldots, N$.

Sort $\{q_i\}$ in ascending order and select the smallest $(N - n_1)$ ones.

Let $\Omega$ be the set of selected indices.

**Output:** $\hat{h} = \sum_{i \in \Omega} q_i$.

After obtaining the robust estimation of covariance matrix, we perform a standard eigenvector decomposition on the covariance matrix to produce the PC estimations. The details of the proposed base RPCA algorithm is given in Algorithm 3.

Algorithm 3 Base RPCA

**Input:** Sample matrix $X = [x_1, \ldots, x_m] \in \mathbb{R}^{p \times m}$, outlier fraction $\lambda$, subspace dimension $r$.

Compute the trimmed covariance matrix $\hat{C}$ as follows (see Algorithm 2):

$\hat{C}_{ij} = \langle X_i^\top, X_j^\top \rangle_{\Omega}$, $\forall i, j = 1, \ldots, p$.

Perform eigen decomposition on $\hat{C}$ and take the eigenvectors corresponding to the largest $r$ eigenvalues $\hat{W}_r = [\hat{w}_1, \ldots, \hat{w}_r]$.

**Output:** $\hat{W}_r$.

We integrate the above robust PCA method into our proposed DRL framework and yield the distributed robust PCA method, as given in Algorithm 4.

Note that in the robust PCA, we cannot directly take median of the output eigenvectors. This is because the eigenvectors may rotate arbitrarily while still span the same subspace. Thus the eigenvector output can be correct but still arbitrarily far away from the ground truth.

Following Theorem 3 provides a robustness guarantee for the distributed RPCA and in particular states that the breakdown point of base robust PCA can be enhanced by DRL with a factor of $1 - \frac{\gamma}{1-\gamma^\star}$.

**Theorem 3.** Suppose the overall outlier fraction is $\lambda$. Samples are divided onto $k$ machines. Suppose on $\gamma k$ machines, the outlier fraction is lower bounded by $\rho \lambda$, with $1 \leq \rho \leq \frac{1}{2}$. Let $d_{\mu}$ denote the deviation of outlier distribution from uniform one on the other $(1 - \gamma)k$ machines. Let $\hat{\text{Proj}}_r$ be the output of Algorithm 4. If $0 \leq \gamma < \frac{\alpha - \beta^\star}{1 - \beta^\star}$, with probability of $1 - \exp \left(-k(1 - \gamma)\psi \left(\frac{\alpha - \beta^\star}{1 - \gamma^\star}, \beta^\star\right)\right)$,
However, when outliers are uniformly distributed on the case when outlier are uniformly distributed over all the machines, i.e., \( \rho \). Due to the space limitation, the proof is provided in the supplementary material. Note that in the Theorem 4 (Performance of RoTR [5])

Here, we refer the readers to [5] for more details about (Algorithm 5) as a base robust regression method, and then integrate this base robust regression method into the framework in Algorithm 1. Here we provide an example of distributed robust regression algorithm, also under the framework of

4 Example II: Distributed Robust Regression

Here, we provide an example of distributed robust regression algorithm, also under the framework of DRL. Similar to the above robust PCA, we adopt the robustified thresholding (RoTR) regression [5] (Algorithm 5) as a base robust regression method, and then integrate this base robust regression method into the framework in Algorithm 1. Here we refer the readers to [5] for more details about Algorithm 5. The estimation error of the RoTR is bounded as in Theorem 4.

Algorithm 4 Distributed RPCA

| Input: Sample matrix \( X = [x_1, \ldots, x_{n+n_1}] \in \mathbb{R}^{p \times (n+n_1)} \), computer node number \( k \), subspace dimension \( r \), outlier fraction \( \lambda \). Column-wisely divide \( X \) evenly into \( k \) submatrices: \( X = [X_1, \ldots, X_k] \).
| Do in parallel 
| \( \hat{W}^{(1)} = \text{RPCA}(X_1, r, \lambda) \)
| \( \vdots \)
| \( \hat{W}^{(k)} = \text{RPCA}(X_k, r, \lambda) \)
| End parallel 
| Communicate \( \hat{W}^{(1)}, \ldots, \hat{W}^{(k)} \) to an aggregating machine.
| Calculate \( \text{Proj}_{\hat{r}} = \hat{W}^{(j)} (\hat{W}^{(j)} \mathbf{T}, i = 1, \ldots, k. \)
| Output: \( \text{Proj}_{\hat{r}} = \text{median} (\text{Proj}_{\hat{r}}^{(1)}, \ldots, \text{Proj}_{\hat{r}}^{(k)}) \)

we have

\[
\| \text{Proj}_{\hat{r}} - \text{Proj}_{r} \|_F \leq C_{\alpha} 2^\frac{(1 - \rho \gamma)\lambda + d_\mu (1 - \gamma)}{\Delta_r (1 - \gamma)(1 - d_\mu) - (1 - \rho \gamma)\lambda} \sigma_x^2 \log p.
\]

Moreover, the breakdown point of the RPCA algorithm is enhanced from \( \frac{\Delta_r}{\Delta_r + 4C_0 \sigma_x^2 \log p} \) to \( \frac{\Delta_r}{\Delta_r + 4C_0 \sigma_x^2 \log p} \frac{1-\gamma}{1-\rho \gamma} \), given \( d_\mu \leq \frac{\Delta_r}{\Delta_r + 4C_0 \sigma_x^2 \log p} - \frac{\Delta_r}{\Delta_r + 4C_0 \sigma_x^2 \log p} \frac{1-\rho \gamma}{1-\gamma} \).

Due to the space limitation, the proof is provided in the supplementary material. Note that in the case when outlier are uniformly distributed over all the machines, i.e., \( \rho = 1 \), we get no robustness gain from DRL. Nevertheless, the computation time is significantly reduced by the DRL framework. However, when outliers are uniformly distributed on the \( (1 - \gamma)k \) machines, i.e., \( d_\mu = 0 \), we get the strongest breakdown point enhancement with a factor of nearly \( \frac{1-\rho \gamma}{1-\gamma} \) (assuming \( C_{\alpha} \to 1 \)).

4 Example II: Distributed Robust Regression

Here, we provide an example of distributed robust regression algorithm, also under the framework of DRL. Similar to the above robust PCA, we adopt the robustified thresholding (RoTR) regression [5] (Algorithm 5) as a base robust regression method, and then integrate this base robust regression method into the framework in Algorithm 1. Here we refer the readers to [5] for more details about Algorithm 5. The estimation error of the RoTR is bounded as in Theorem 4.

Algorithm 5 Base Robust Regression - RoTR

| Input: Covariate matrix \( X = [x_1, \ldots, x_{n+n_1}] \in \mathbb{R}^{p \times (n+n_1)} \) and response \( y \in \mathbb{R}^{n+n_1} \), outlier fraction \( \lambda \). For \( j = 1, \ldots, p \), compute \( \theta(j) = \langle y, X_j \rangle_{\lambda(n+n_1)} \).
| Output: \( \hat{\theta} \).

Theorem 4 (Performance of RoTR [5]). Under sub-Gaussian design with \( \Sigma_x = I \), the following holds with probability at least \( 1 - p^{-2} \). The output of RoTR satisfies the \( \ell_2 \) bound:

\[
\| \hat{\theta} - \theta^* \|_2 \leq c \| \theta^* \|_2 \sqrt{1 + \frac{\sigma_x^2}{\| \theta^* \|_2^2} \frac{\lambda}{1-\lambda} \sqrt{p \log p}}.
\]

Here \( c \) is a constant independent of \( p, n, \lambda \).

The above performance guarantee on RoTR is directly adopted from [5]. To get the breakdown point of the RoTR method, w.l.o.g., we assume that \( \| \theta^* \|_2 = 1 \). Then, to ensure the above bound is
non-trivial, i.e., \( \| \hat{\theta} - \theta^* \|_2 \leq 1 \), the outlier fraction \( \lambda = \frac{n_1}{n_1 + n_1} \) needs to satisfy
\[
\lambda \leq \frac{1}{1 + c \sqrt{1 + \sigma^2} \sqrt{p} \log p}.
\]

Similar to the distributed RPCA, we have the following performance guarantee for distributed RoTR.

**Theorem 5** (Performance of distributed RoTR). Suppose the overall outlier fraction is \( \lambda \). Samples are divided onto \( k \) machines. Suppose on \( \gamma k \) machines, the outlier fraction is lower bounded by \( \rho \lambda \), with \( 1 \leq \rho \leq \frac{1}{2} \). Let \( d_{\mu} \) denote the deviation of outlier distribution from uniform one on the other \((1 - \gamma)k\) machines. Let \( \hat{\theta} \) be the output of distributed RoTR. If \( 0 \leq \gamma < \frac{\alpha - \beta}{1 - \beta^2} \), with probability of \( 1 - \exp \left( -k(1 - \gamma) \psi \left( \frac{\alpha - \beta}{1 - \beta^2}, \beta^* \right) \right) \), we have
\[
\| \hat{\theta} - \theta^* \|_2 \leq C'_\alpha \| \theta^* \|_2 \sqrt{1 + \frac{\sigma^2}{\| \theta^* \|_2^2} \left( \frac{(1 - \rho \gamma) \lambda + d_{\mu} (1 - \gamma)}{(1 - \gamma) (1 - d_{\mu})} - \frac{d_{\mu} (1 - \gamma)}{1 - \rho \gamma \left( 1 + \frac{1}{C'_\alpha \sqrt{1 + \sigma^2} \sqrt{p} \log p} \right)} \right)}.
\]

where \( C'_\alpha = C_\alpha c \) with \( c \) being the constant in Theorem 4. Furthermore, assume \( \| \theta^* \|_2 = 1 \), we have the breakdown point of the distributed RoTR is
\[
\lambda = \lambda \leq 1 - \gamma - \frac{1 - d_{\mu}}{1 - \rho \gamma} + C'_\alpha \sqrt{1 + \frac{\sigma^2}{\| \theta^* \|_2^2} \left( \frac{d_{\mu} (1 - \gamma)}{1 - \rho \gamma \left( 1 + \frac{1}{C'_\alpha \sqrt{1 + \sigma^2} \sqrt{p} \log p} \right)} \right)}.
\]

When \( d_{\mu} = 0 \), the breakdown point is also enhanced by a factor of \( \frac{1 - \alpha}{1 - \beta^2} \) when \( C_\alpha \to 1 \). We again defer the proof to the supplementary material.

## 5 Simulations

We devote this section to comparing the distributed robust learning (DRL) algorithms, including distributed RPCA and distributed robust linear regression (RLR), with their centralized counterparts.

In the simulations of RPCA, samples are generated according to \( x_i = \theta^* z_i + v_i \). Here the signal \( z_i \in \mathbb{R}^d \) is sampled from normal distribution: \( z_i \sim \mathcal{N}(0, I_d) \). The noise \( v_i \in \mathbb{R}^p \) is sampled as: \( v_i \sim \mathcal{N}(0, \sigma_z I_p) \). The underlying matrix \( \theta^* \in \mathbb{R}^{p \times d} \) is randomly generated whose columns are then orthogonalized. The entries of outliers \( x_o \in \mathbb{R}^p \) are i.i.d. random variables from uniform distribution \([-\alpha, \alpha] \). We use the distance between two projection matrices to measure the subspace estimation error: \( \| \text{proj} - \text{proj}^* \|_F / \| \text{proj}^* \|_F \). Here \( \text{proj} \) is the output estimates and \( \text{proj}^* = \theta^* \theta^*^\top \).

In the simulations of RLR, samples \((x_i, y_i)\) are generated according to \( y_i = \theta^*^\top x_i + v_i \). Here the model parameter \( \theta^* \) is randomly sampled from \( \mathcal{N}(0, I_p) \), and \( x_i \in \mathbb{R}^p \) is also sampled from normal distribution: \( x_i \in \mathcal{N}(0, I_p) \). The noise \( v_i \in \mathbb{R} \) is sampled as: \( v_i \sim \mathcal{N}(0, \sigma_e) \). The entries of outlier \( x_o \) are also i.i.d. randomly sampled from uniform distribution \([-\sigma_o, \sigma_o] \). The response of outlier is generated by \( y_o = -\theta^*^\top x_o + v \). We use \( \| \theta^* - \theta \|_2 / \| \theta^* \|_2 \) to measure the error. Here \( \hat{\theta} \) is the output estimate.

We conduct simulations with varying outlier fraction \( \lambda \). When \( \lambda \leq 0.5 \), outliers are uniformly distributed on the machines and on each machine the outlier fraction is around \( \lambda \). When \( \lambda > 0.5 \), outliers are not uniformly distributed. More precisely, when \( \lambda = 0.6 \), on \( \gamma = 50\% \) of the machines, the outlier fraction is 0.8, while on the other 50\% machines the outlier fraction is 0.4. Similarly, when \( \lambda = 0.7 \), the outlier fractions are 0.9 (on 50\% of the machines) and 0.5 (on the other 50\% of the machines) respectively. The simulations for both problems are repeated 10 times. The average and variance of the estimation errors are plotted in Fig. 1(a) and Fig. 1(b) respectively.

The simulations are implemented on a PC with 2.83GHz Quad CPU and 8GB RAM. It takes centralized RPCA around 60 seconds to handle \( 1 \times 10^6 \) samples with dimensionality of 100. In contrast, distributed RPCA only costs 0.6 seconds by using \( k = 100 \) parallel procedures. The communication cost here is negligible since only \( 100 \times 5 \) eigenvector matrices are communicated. For RLR simulations, we also observe about \( k \) (here \( k = 100 \)) times improvement on time efficiency.
As for the performance, from Fig. 1(a) and Fig. 1(b) we observe that when $\lambda \leq 0.5$, DRL and centralized PCA and LR achieve similar performances, which are much better than non-robust standard PCA and LR algorithms. Actually, standard PCA and LR begin to break down when $\lambda = 0.1$. This demonstrates that DRL preserves the robustness of centralized algorithms. When $\lambda$ increases to 0.6, centralized algorithms (blue lines) break down sharply, as the outliers outnumber their maximal breakdown point of 0.5. In contrast, distributed RPCA and RLR still present guaranteed robustness and perform much better, which demonstrate that the DRL framework indeed enhances the robustness of the base robust learning methods.

6 Conclusions

We developed a generic DRL framework that processes the data subsets in parallel and aggregates the results from different subsets via taking their geometric median. DRL not only enhances the time and memory efficiency of robust learning significantly but also preserves the robustness of the base learning algorithms. Moreover, when the outliers are not uniformly distributed, the proposed framework is able to further enhance the robustness (measured using the breakdown point) of the base learning algorithms. We then provided two concrete examples, distributed RPCA and distributed robust regression, to demonstrate how DRL works.

References

[1] Vic Barnett and Toby Lewis. Outliers in statistical data, volume 3. Wiley New York, 1994.
[2] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. Information Theory, IEEE Transactions on, 52(6):2508–2530, 2006.
[3] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? Journal of the ACM (JACM), 58(3):11, 2011.
[4] Yudong Chen and Constantine Caramanis. Noisy and missing data regression: Distribution-oblivious support recovery. In ICML, 2013.
[5] Yudong Chen, Constantine Caramanis, and Shie Mannor. Robust sparse regression under adversarial corruption. In ICML, 2013.
[6] Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. iii. SIAM Journal on Numerical Analysis, 7(1):1–46, 1970.
[7] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. Communications of the ACM, 51(1):107–113, 2008.
[8] David L Donoho and Miriam Gasko. Breakdown properties of location estimates based on halfspace depth and projected outlyingness. The Annals of Statistics, pages 1803–1827, 1992.
A Proof of Theorem 2

Proof. Let $n_1$ denote the total number of samples. In the cases described by the theorem, the total number of outliers in these $k'$ machines is lower bounded by $\rho \lambda n_1 k'/k = \rho \lambda \gamma n_2$. The total number of outliers on the other $k - k' = (1 - \gamma)k$ machines is then upper bounded by $\lambda n_2 - \rho \gamma \lambda n_2 = (1 - \rho \gamma)\lambda n_2$. Therefore, for the remained $(1 - \gamma)k$ machines, the maximal outlier fraction among the $(1 - \gamma)k$ machines is $\frac{1 - \rho \gamma}{1 - \rho \gamma} \lambda + \lambda^*$. Letting $\frac{1 - \rho \gamma}{1 - \rho \gamma} \lambda + \lambda^* \leq \lambda^*$ ensures that all the $(1 - \gamma)k$ machines do not breakdown. According to Theorem 1 when $0 \leq \gamma \leq \alpha - \beta^*$, the error of DRL is lower bounded. Thus DRL can tolerate outlier fraction up to $\lambda$ with

$$\lambda = \frac{1 - \gamma}{1 - \rho \gamma} \left( \lambda^* - \lambda \right).$$

Here $0 \leq \gamma \leq \frac{\alpha - \beta^*}{1 - \beta^*}$ and $1 \leq \rho < \frac{1}{\gamma}$ to ensure $1 - \rho \gamma > 0$.

B Proof of Theorem 3

Proof. According to the proof of Theorem 4 in [5], we have, for the covariance matrix constructed in Algorithm 3,

$$\|\hat{C} - C\|_{\infty} \leq \frac{n_1 \log p \sigma_x^2}{n} \leq \lambda \frac{\sigma_x^2 \log p}{1 - \lambda} = \frac{\lambda}{1 - \lambda} \sigma_x^2 \log p$$

with high probability. Let $\Delta_r = \sigma_r - \sigma_{r+1}$, where $\sigma_r$ denotes the $r$th largest eigenvalue of $C$. Then, applying the Davis-Kahan perturbation theorem [6], we have, whenever $\|\hat{C} - C\|_{\infty} \leq \frac{1}{4} \Delta_r$,

$$\|\text{Proj}_r - \text{Proj}_{r+1}\|_{\infty} \leq \frac{2\|\hat{C} - C\|_{\infty}}{\Delta_r} \leq \frac{2 \lambda}{\Delta_r} \frac{\sigma_x^2 \log p}{1 - \lambda}.$$  \hspace{1cm} (1)

Note that the above condition gives an upper bound on the outlier fraction:

$$\frac{\lambda}{1 - \lambda} \sigma_x^2 \log p \leq \frac{1}{4} \Delta_r,$$

which yields,

$$\lambda \leq \frac{\Delta_r}{\Delta_r + 4 \sigma_x^2 \log p}.$$
It says that the breakdown point of the base RPCA algorithm is $\frac{\Delta_r}{\Delta_r + 4\sigma_x^2 \log p}$.

In the distributed RPCA algorithm, the total $n + n_1$ samples are divided onto $k$ machines. On the $\gamma k$ machines, the outlier fractions are lower bounded by $\rho \lambda$ with $\rho \geq 1$, where $\lambda \triangleq \frac{n_1}{n + n_1}$. On the remaining $(1 - \gamma)k$ machines, the maximal outlier fraction is $\lambda' = \frac{1 - \rho \gamma}{1 - \gamma} + d_\mu$. (Refer to Theorem 2 and proof therein for the definition of $d_\mu$.)

Thus, for these $(1 - \gamma)k$ machines, the estimated covariance matrix $\hat{C}(i)$ satisfies

$$\| \hat{C}(i) - C \|_2 \leq \frac{\lambda'}{1 - \lambda'} \sigma_x^2 \log p = \frac{(1 - \rho \gamma) + d_\mu(1 - \gamma)}{(1 - \gamma)(1 - d_\mu) - (1 - \rho \gamma) \lambda} \sigma_x^2 \log p.$$ 

Substituting into Eqn. (1), we obtain that the estimated projection matrix is bounded as,

$$\| \text{Proj} - \hat{C}(i) - C \|_2 \leq \frac{2}{\Delta_r} \| \hat{C}(i) - C \|_2 \leq \frac{2}{\Delta_r} \frac{(1 - \rho \gamma) + d_\mu(1 - \gamma)}{(1 - \gamma)(1 - d_\mu) - (1 - \rho \gamma) \lambda} \sigma_x^2 \log p.$$ 

Let $\hat{\text{Proj}} = \text{median} \left( \hat{\text{Proj}}^{(1)}, \ldots, \hat{\text{Proj}}^{((1-\gamma)k)} \right)$. A direct application of Theorem 3 gives

$$\| \hat{\text{Proj}} - \text{Proj} \|_F \leq C_\alpha \frac{2}{\Delta_r} \frac{(1 - \rho \gamma) + d_\mu(1 - \gamma)}{(1 - \gamma)(1 - d_\mu) - (1 - \rho \gamma) \lambda} \sigma_x^2 \log p.$$ 

Again, the above inequality holds when

$$\lambda' = \frac{1 - \rho \gamma}{1 - \gamma} + d_\mu \leq \frac{\Delta_r}{\Delta_r + 4C_\alpha \sigma_x^2 \log p},$$

which is equivalent to:

$$\lambda \leq \left( \frac{\Delta_r}{\Delta_r + 4C_\alpha \sigma_x^2 \log p} - d_\mu \right) \frac{1 - \gamma}{1 - \rho \gamma}.$$ 

The breakdown point is enhanced from $\frac{1}{\Delta_r + 4C_\alpha \sigma_x^2 \log p}$ to $\left( \frac{1}{\Delta_r + 4C_\alpha \sigma_x^2 \log p} - d_\mu \right) \frac{1 - \gamma}{1 - \rho \gamma}$.

To gain the robustness enhancement, it should be satisfied that

$$\left( \frac{\Delta_r}{\Delta_r + 4C_\alpha \sigma_x^2 \log p} - d_\mu \right) \frac{1 - \gamma}{1 - \rho \gamma} \leq \frac{\Delta_r}{\Delta_r + 4\sigma_x^2 \log p}.$$ 

The above requirement is equivalent to:

$$d_\mu \leq \frac{\Delta_r}{\Delta_r + 4C_\alpha \sigma_x^2 \log p} - \frac{\Delta_r}{\Delta_r + 4\sigma_x^2 \log p} \frac{1 - \rho \gamma}{1 - \gamma}.$$ 

When $C_\alpha \rightarrow 1$, it can be simplified as:

$$d_\mu \leq \frac{\Delta_r}{\Delta_r + 4\sigma_x^2 \log p} \frac{\gamma(\rho - 1)}{1 - \gamma}.$$ 

\[\square\]

\section{Proof of Theorem 5}

According to the proof of Theorem 3 on the $(1 - \gamma)k$ machines, the maximal outlier fraction is $\lambda' = \frac{1 - \rho \gamma}{1 - \gamma} + d_\mu$. Then, based on the results in Theorem 4 and Theorem 1, it is straightforward to get:

$$\| \tilde{\theta} - \theta^* \|_2 \leq C_\alpha \| \theta^* \|_2 \sqrt{1 + \frac{\sigma_x^2}{\| \theta^* \|_2^2} \left( \frac{\lambda'}{1 - \lambda'} \sqrt{p} \log p \right)}$$

$$= C_\alpha \| \theta^* \|_2 \sqrt{1 + \frac{\sigma_x^2}{\| \theta^* \|_2^2} \left( \frac{(1 - \rho \gamma) + d_\mu(1 - \gamma)}{(1 - \gamma)(1 - d_\mu) - (1 - \rho \gamma) \lambda} \sqrt{p} \log p \right)},$$
where $C'_\alpha = C_\alpha c$ with $c$ being the constant in Theorem 4.

Since we assume $\|\theta^*\|_2 = 1$, to make sure the bound is non-trivial, we need to require that,

$$C'_\alpha \sqrt{1 + \sigma^2} \left( \frac{(1 - \rho \gamma) \lambda + d_\mu (1 - \gamma)}{(1 - \gamma)(1 - d_\mu) - (1 - \rho \gamma) \lambda} \sqrt{p} \log p \right) \leq 1.$$ 

Solve the above inequality and we finally obtain

$$\lambda \leq \frac{1 - \gamma}{1 - \rho \gamma} \frac{1 - d_\mu}{1 + C'_\alpha \sqrt{1 + \sigma^2} \sqrt{p} \log p} - \frac{d_\mu (1 - \gamma)}{(1 - \rho \gamma) \left( 1 + \frac{1}{C'_\alpha \sqrt{1 + \sigma^2} \sqrt{p} \log p} \right)}.$$