SEVER: A Robust Meta-Algorithm for Stochastic Optimization[1]

Ilias Diakonikolas [*1] **Gautam Kamath** [*2] **Daniel M. Kane** [*3] **Jerry Li** [*4] **Jacob Steinhardt** [*5] **Alistair Stewart** [*6]

Abstract

In high dimensions, most machine learning methods are brittle to even a small fraction of structured outliers. To address this, we introduce a new meta-algorithm that can take in a *base learner* such as least squares or stochastic gradient descent, and harden the learner to be resistant to outliers. Our method, SEVER, possesses strong theoretical guarantees yet is also highly scalable—beyond running the base learner itself, it only requires computing the top singular vector of a certain $n \times d$ matrix. We apply SEVER on a drug design dataset and a spam classification dataset, and find that in both cases it has substantially greater robustness than several baselines.

1. Introduction

Learning in the presence of outliers is a ubiquitous challenge in machine learning; nevertheless, most machine learning methods are very sensitive to outliers in high dimensions. The focus of this work is on designing algorithms that are outlier robust while remaining competitive in terms of accuracy and running time.

We highlight two motivating applications. The first is biological data (e.g., gene expression data), where mislabeling or measurement errors can create systematic outliers (Rosenberg et al., 2002; Li et al., 2008) requiring painstaking manual effort to remove (Paschou et al., 2010). Detecting outliers in such settings is often important either because the outlier observations are of interest themselves or because they might contaminate the downstream statistical analysis. The second motivation is machine learning security, where outliers can be introduced through *data poisoning* attacks (Barreno et al., 2010) where an adversary inserts fake data into the training set (e.g., by creating a fake user account). Recent work shows that for high-dimensional datasets, even a small fraction of outliers can substantially degrade the learned model (Biggio et al., 2012; Newell et al., 2014; Koh & Liang, 2017; Steinhardt et al., 2017; Koh et al., 2018).

Crucially, in both the biological and security settings above, the outliers are not "random" but are instead highly correlated, and could have a complex internal structure that is difficult to model. This leads us to the following conceptual question underlying the present work: *Can we design training algorithms that are robust to the presence of an ε-fraction of arbitrary (and potentially adversarial) outliers?*

Estimation in the presence of outliers is a prototypical goal in robust statistics and has been systematically studied since the pioneering work of Tukey Tukey (1960). Popular methods include RANSAC (Fischler & Bolles, 1981), minimum covariance determinant (Rousseeuw & Driessen, 1999), removal based on k-nearest neighbors (Breunig et al., 2000), and Huberizing the loss (Owen, 2007) (see Hodge & Austin (2004) for a comprehensive survey). However, these classical methods either break down in high dimensions, or only handle "benign" outliers that are obviously different from the rest of the data (see Section 1.1 for futher discussion).

Motivated by this, recent work in theoretical computer science has developed efficient robust estimators for classical problems such as linear classification (Klivans et al., 2009; Awasthi et al., 2014), mean and covariance estimation (Diakonikolas et al., 2016a; Lai et al., 2016), clustering (Charikar et al., 2017), and regression (Bhatia et al., 2015; 2017; Balakrishnan et al., 2017). Nevertheless, the promise of practical high-dimensional robust estimation is yet to be realized; indeed, the aforementioned results generally suffer from one of two shortcomings–either they use sophisticated convex optimization algorithms that do not scale to large datasets, or they are tailored to specific problems of interest

[*]Equal contribution [1]Department of Computer Science, University of Southern California, Los Angeles, California, USA [2]Simons Institute for the Theory of Computing, Berkeley, California, USA [3]Departments of Mathematics and Computer Science and Engineering, University of California, San Diego, California, USA [4]Microsoft Research AI, Redmond, Washington, USA [5]Department of Statistics, University of California, Berkeley, California, USA [6]Web3 Foundation, Zug, Switzerland. Correspondence to: Ilias Diakonikolas <diakonik@usc.edu>, Gautam Kamath <g@csail.mit.edu>, Daniel M. Kane <dakane@cs.ucsd.edu>, Jerry Li <jerrl@microsoft.com>, Jacob Steinhardt <jsteinhardt@berkeley.edu>, Alistair Stewart <stewart.al@gmail.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

[1] Code: https://github.com/hoonose/sever

or specific distributional assumptions on the data, and hence do not have good accuracy on real data.

In this work, we address these shortcomings. We propose an algorithm, SEVER, that is:

- **Robust:** it can handle arbitrary outliers with only a small increase in error, even in high dimensions.
- **General:** it can be applied to most common learning problems including regression and classification, and handles non-convex models such as neural networks.
- **Practical:** the algorithm can be implemented with standard machine learning libraries.

At a high level, our algorithm (depicted in Figure 1 and described in detail in Section 2.1) is a simple "plug-in" outlier detector–first, run whatever learning procedure would be run normally (e.g., least squares in the case of linear regression). Then, consider the matrix of gradients at the optimal parameters, and compute the top singular vector of this matrix. Finally, remove any points whose projection onto this singular vector is too large (and re-train if necessary).

Despite its simplicity, our algorithm possesses strong theoretical guarantees: As long as the real (non-outlying) data is not too heavy-tailed, SEVER is provably robust to outliers–see Section 2 for detailed statements of the theory. At the same time, we show that our algorithm works very well in practice and outperforms a number of natural baseline outlier detectors. In line with our original motivating biological and security applications, we implement our method on two tasks–a linear regression task for predicting protein activity levels (Olier et al., 2018), and a spam classification task based on emails from the Enron corporation (Metsis et al., 2006). Even with a small fraction of outliers, baseline methods perform poorly on these datasets; for instance, on the Enron spam dataset with a 1% fraction of outliers, baseline errors range from 13.4% to 20.5%, while SEVER incurs only 7.3% error (in comparison, the error is 3% in the absence of outliers). Similarly, on the drug design dataset, with 10% corruptions, SEVER achieved 1.42 mean-squared error test error, compared to 1.51-2.33 for the baselines, and 1.23 error on the uncorrupted dataset.

1.1. Comparison to Prior Work

As mentioned above, the myriad classical approaches to robust estimation perform poorly in high dimensions or against worst-case outliers. For instance, RANSAC (Fischler & Bolles, 1981) randomly subsamples points such that no outliers remain with decent probability; since we need at least d points to fit a d-dimensional model, this requires at most $O(1/d)$ outliers. k-nearest neighbors (Breunig et al., 2000) similarly suffers from the curse of dimensionality when d is large. The minimum covariance determinant (Rousseeuw & Driessen, 1999) only applies when the number of data points n exceeds $2d$, which does not hold for the datasets we consider (it also has other issues such as computational intractability). A final natural method is to limit the effect of points with large loss (via e.g. Huberization (Owen, 2007)), but as Koh et al. (2018) show (and we experimentally confirm), correlated outliers often have *lower* loss than the real data under the learned model.

These issues have motivated work on high-dimensional robust statistics going back to Tukey (Tukey, 1975). However, it was not until much later that efficient algorithms with favorable properties were first proposed. (Klivans et al., 2009) gave the first efficient algorithms for robustly classification under the assumption that the distribution of the good data is isotropic and log-concave. Subsequently, (Awasthi et al., 2014) obtained an improved and nearly optimal robust algorithm for this problem. Two concurrent works (Diakonikolas et al., 2016a; Lai et al., 2016) gave the first efficient robust estimators for several other tasks including mean and covariance estimation. There has since been considerable study of algorithmic robust estimation in high dimensions, including learning graphical models (Diakonikolas et al., 2016b), understanding computation-robustness tradeoffs (Diakonikolas et al., 2017d; 2018), establishing connections to PAC learning (Diakonikolas et al., 2017c), tolerating more noise by outputting a list of hypotheses (Charikar et al., 2017; Meister & Valiant, 2018; Diakonikolas et al., 2017b), robust estimation of discrete structures (Steinhardt, 2017; Qiao & Valiant, 2018; Steinhardt et al., 2018), and robust estimation via sum-of-squares (Kothari & Steurer, 2017; Hopkins & Li, 2017; Kothari & Steinhardt, 2017).

Despite this progress, these recent theoretical papers typically focus on designing specialized algorithms for specific settings (such as mean estimation or linear classification for specific families of distributions) rather than on designing general algorithms. The only exception is (Charikar et al., 2017), which provides a robust meta-algorithm for stochastic convex optimization in a similar setting to ours. However, that algorithm (i) requires solving a large semidefinite program and (ii) incurs a significant loss in performance relative to standard training *even in the absence of outliers*. On the other hand, (Diakonikolas et al., 2017a) provide a practical implementation of the robust mean and covariance estimation algorithms of (Diakonikolas et al., 2016a), but do not consider more general learning tasks.

A number of papers (Nasrabadi et al., 2011; Nguyen & Tran, 2013; Bhatia et al., 2015; 2017) have proposed efficient algorithms for a type of robust linear regression. However, these works consider a restrictive corruption model that only allows adversarial corruptions to the responses (but not the covariates). On the other hand, (Balakrishnan et al., 2017) studies (sparse) linear regression and, more broadly, generalized linear models (GLMs) under a robustness model

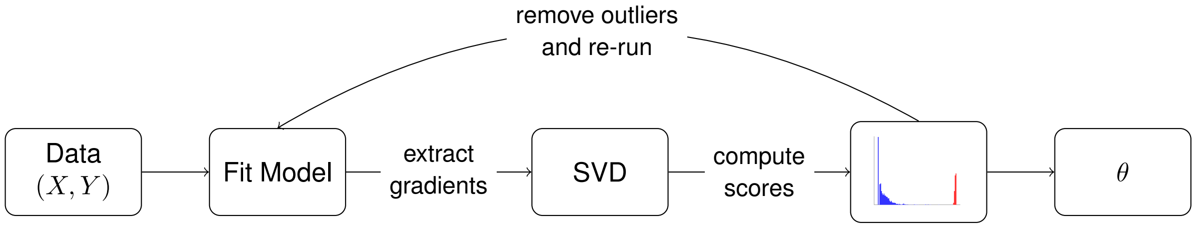

Figure 1. Illustration of the Sever pipeline. We first use any machine learning algorithm to fit a model to the data. Then, we extract gradients for each data point at the learned parameters, and take the singular value decomposition of the gradients. We use this to compute an outlier score for each data point. If we detect outliers, we remove them and re-run the learning algorithm; otherwise, we output the learned parameters.

similar to ours. The main issues with this algorithm are that (i) it requires running the ellipsoid method (hence does not scale) and (ii) it crucially assumes Gaussianity of the covariates, which is unlikely to hold in practice.

In a related direction, Steinhardt et al. (2017) provide a method for analyzing outlier detectors in the context of linear classification, either certifying robustness or generating an attack if the learner is not robust. The outlier detector they analyze is brittle in high dimensions, motivating the need for the robust algorithms presented in the current work. Later work by the same authors showed how to bypass a number of common outlier detection methods (Koh et al., 2018). We use these recent strong attacks as part of our evaluation and show that our algorithm is more robust.

Concurrent Works. (Prasad et al., 2018) independently obtained a robust algorithm for stochastic convex optimization by combining gradient descent with robust mean estimation. For the case of linear regression, (Diakonikolas et al., 2019) provide efficient robust algorithms with near-optimal error guarantees under various distributional assumptions and establish matching computational-robustness tradeoffs.

2. Framework and Algorithm

We will consider stochastic optimization tasks, where there is some true distribution p^* over functions $f : \mathcal{H} \to \mathbb{R}$, and our goal is to find a parameter vector $w^* \in \mathcal{H}$ minimizing $\overline{f}(w) \stackrel{\text{def}}{=} \mathbb{E}_{f \sim p^*}[f(w)]$. Here we assume $\mathcal{H} \subseteq \mathbb{R}^d$ is a space of possible parameters. As an example, we consider linear regression, where $f(w) = \frac{1}{2}(w \cdot x - y)^2$ for (x, y) drawn from the data distribution; or support vector machines, where $f(w) = \max\{0, 1 - y(w \cdot x)\}$.

To help us learn the parameter vector w^*, we have access to a *training set* of n functions $f_{1:n} \stackrel{\text{def}}{=} \{f_1, \dots, f_n\}$. (For linear regression, we would have $f_i(w) = \frac{1}{2}(w \cdot x_i - y_i)^2$, where (x_i, y_i) is an observed data point.) However, unlike the classical (uncorrupted) setting where we assume that $f_1, \dots, f_n \sim p^*$, we allow for an ε-fraction of the points to be arbitrary outliers:

Definition 2.1 (ε-contamination model)**.** Given $\varepsilon > 0$ and a distribution p^* over functions $f : \mathcal{H} \to \mathbb{R}$, data is generated as follows: first, n clean samples $f_1, \dots, f_n$ are drawn from p^*. Then, an *adversary* is allowed to inspect the samples and replace any εn of them with arbitrary samples. The resulting set of points is then given to the algorithm. We call such a set of samples *ε-corrupted (with respect to p^*).*

Our theoretical results hold in the ε-contamination model, the adversary is allowed to both add and remove points. Our experimental evaluation uses corrupted instances in which the adversary is only allowed to add corrupted points. Additive corruptions essentially correspond to Huber's contamination model (Huber, 1964) in robust statistics.

Finally, we will often assume access to a black-box learner, which we denote by $\mathcal{L}$, which takes in functions $f_1, \dots, f_n$ and outputs a parameter vector $w \in \mathcal{H}$. We want to stipulate that $\mathcal{L}$ approximately minimizes $\frac{1}{n}\sum_{i=1}^{n} f_i(w)$. For this purpose, we introduce the following definition:

Definition 2.2 (γ-approximate critical point)**.** Given a function $f : \mathcal{H} \to \mathbb{R}$, a γ-approximate critical point of f, is a point $w \in \mathcal{H}$ so that for all unit vectors v where $w + \delta v \in \mathcal{H}$ for arbitrarily small positive δ, we have that $v \cdot \nabla f(w) \geq -\gamma$.

Essentially, the above definition means that the value of f cannot be decreased much by changing the input w locally, while staying within the domain. The condition enforces that moving in any direction v either causes us to leave $\mathcal{H}$ or causes f to decrease at a rate at most γ. It should be noted that when $\mathcal{H} = \mathbb{R}^d$, our above notion of approximate critical point reduces to the standard notion of approximate stationary point (i.e., a point where the magnitude of the gradient is small). We now define a *γ-approximate* learner:

Definition 2.3 (γ-approximate learner)**.** A learning algorithm $\mathcal{L}$ is called *γ-approximate* if, for any functions $f_1, \dots, f_n : \mathcal{H} \to \mathbb{R}$ each bounded below on a closed domain $\mathcal{H}$, the output $w = \mathcal{L}(f_{1:n})$ of $\mathcal{L}$ is a γ-approximate critical point of $f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)$.

In other words, $\mathcal{L}$ always finds an approximate critical point of the empirical learning objective. We note that most common learning algorithms (such as stochastic gradient descent) satisfy the γ-approximate learner property.

2.1. Algorithm and Theory

As outlined in Figure 1, our algorithm works by post-processing the gradients of a black-box learning algorithm. The basic intuition is as follows: we want to ensure that the outliers do not have a large effect on the learned parameters. Intuitively, for the outliers to have such an effect, their corresponding gradients should be (i) large in magnitude and (ii) systematically pointing in a specific direction. We can detect this via singular value decomposition–if both (i) and (ii) hold then the outliers should be responsible for a large singular value in the matrix of gradients, which allows us to detect and remove them. This is shown more formally via the pseudocode in Algorithm 1.

Algorithm 1 SEVER$(f_{1:n}, \mathcal{L}, \sigma)$

```
1:  Input: Sample functions $f_1, \ldots, f_n : \mathcal{H} \to \mathbb{R}$,
    bounded below on a closed domain $\mathcal{H}$, $\gamma$-approximate
    learner $\mathcal{L}$, and parameter $\sigma \in \mathbb{R}_+$.
2:  Initialize $S \leftarrow \{1, \ldots, n\}$.
3:  repeat
4:      $w \leftarrow \mathcal{L}(\{f_i\}_{i \in S})$.
5:      Let $\widehat{\nabla} = \frac{1}{|S|} \sum_{i \in S} \nabla f_i(w)$.
6:      Let $G = [\nabla f_i(w) - \widehat{\nabla}]_{i \in S}$ be the $|S| \times d$ matrix of
        centered gradients.
7:      Let $v$ be the top right singular vector of $G$.
8:      Compute the vector $\tau$ of outlier scores defined via
        $\tau_i = \left( (\nabla f_i(w) - \widehat{\nabla}) \cdot v \right)^2$.
9:      $S' \leftarrow S$
10:     $S \leftarrow$ FILTER$(S', \tau, \sigma)$ ▷ Remove some $i$'s with the
        largest scores $\tau_i$ from $S$; see Algorithm 2.
11: until $S = S'$.
12: Return $w$.
```

Algorithm 2 FILTER(S, τ, σ)

```
1: Input: Set $S \subseteq [n]$, vector $\tau$ of outlier scores, and
   parameter $\sigma \in \mathbb{R}_+$.
2: If $\sum_i \tau_i \leq c \cdot \sigma$, for some constant $c > 1$, return $S$ ▷
   We only filter out points if the variance is larger than an
   appropriately chosen threshold.
3: Draw $T$ from the uniform distribution on $[0, \max_i \tau_i]$.
4: Return $\{i \in S : \tau_i < T\}$.
```

Theoretical Guarantees. Our first theoretical result says that as long as the data is not too heavy-tailed, SEVER will find an approximate critical point of the true function $\overline{f}$, even in the presence of outliers.

Theorem 2.1. *Suppose that functions $f_1, \ldots, f_n, \overline{f} : \mathcal{H} \to \mathbb{R}$ are bounded below on a closed domain $\mathcal{H}$, and suppose that they satisfy the following deterministic regularity conditions: There exists a set $I_{\text{good}} \subseteq [n]$ with $|I_{\text{good}}| \geq (1-\varepsilon)n$ and $\sigma > 0$ such that $\text{Cov}_{I_{\text{good}}}[\nabla f_i(w)] \preceq \sigma^2 I$, $w \in \mathcal{H}$, and $\|\nabla \hat{f}(w) - \nabla \overline{f}(w)\|_2 \leq \sigma\sqrt{\varepsilon}$, $w \in \mathcal{H}$, where $\hat{f} \overset{def}{=} (1/|I_{\text{good}}|) \sum_{i \in I_{\text{good}}} f_i$. Then our algorithm SEVER applied to $f_1, \ldots, f_n, \sigma$ returns a point $w \in \mathcal{H}$ that, with probability at least $9/10$, is a $(\gamma + O(\sigma\sqrt{\varepsilon}))$-approximate critical point of $\overline{f}$.*

The key take-away from Theorem 2.1 is that the error guarantee has no dependence on the underlying dimension d. In contrast, most natural algorithms incur an error that grows with d, and hence have poor robustness in high dimensions.

In the supplementary material (Proposition B.5), we show that under some mild niceness assumptions on p^*, the deterministic regularity conditions are satisfied with high probability with polynomially many samples.

While Theorem 2.1 is very general and holds even for non-convex loss functions, we might in general hope for more than an approximate critical point. In particular, as a corollary of Theorem 2.1, for convex problems we can guarantee that we find an approximate global minimum.

Corollary 2.2. *Suppose that $f_1, \ldots, f_n : \mathcal{H} \to \mathbb{R}$ satisfy the regularity conditions (i) and (ii), and that $\mathcal{H}$ is convex with ℓ_2-radius r. Then, with probability at least $9/10$, the output of SEVER satisfies the following: If $\overline{f}$ is convex, the algorithm finds a $w \in \mathcal{H}$ such that $\overline{f}(w) - \overline{f}(w^*) = O((\sigma\sqrt{\varepsilon} + \gamma)r)$. If $\overline{f}$ is ξ-strongly convex, the algorithm finds a $w \in \mathcal{H}$ such that $\overline{f}(w) - \overline{f}(w^*) = O\left((\varepsilon\sigma^2 + \gamma^2)/\xi\right)$.*

Practical Considerations. For our theory to hold, we need to use the randomized filtering algorithm shown in Algorithm 2 (which is essentially the robust mean estimation algorithm of (Diakonikolas et al., 2017a)), and filter until the stopping condition in line 1 of Algorithm 1 is satisfied. However, in practice we found that the following simpler algorithm worked well: in each iteration simply remove the top p fraction of outliers according to the scores τ_i, and instead of using a specific stopping condition, simply repeat the filter for r iterations in total. This is the version of SEVER that we use in our experiments in Section 3.

Concrete Applications. In the supplementary material (Sections C and E), we provide several concrete applications of our general theorem, particularly involved with optimization problems related to learning generalized linear models. This includes hinge, logistic, and least-squares loss.

2.2. Overview of SEVER and its Analysis

For simplicity of the exposition, we restrict ourselves to the important special case where the functions involved are convex. We have a probability distribution p^* over convex functions on some convex domain $\mathcal{H} \subseteq \mathbb{R}^d$ and we wish to minimize the function $\overline{f} = \mathbb{E}_{f \sim p^*}[f]$. This problem is well-understood in the absence of corruptions: Under mild assumptions, if we take sufficiently many samples from p^*, their average $\hat{f}$ approximates $\overline{f}$ pointwise with high probability. Hence, we can use standard methods from convex optimization to minimize $\hat{f}$, which will in turn minimize $\overline{f}$.

In the robust setting, stochastic optimization becomes quite challenging: Even for the most basic special cases of this problem (e.g., mean estimation, linear regression) a *single* adversarially corrupted sample can substantially change the location of the minimum for $\hat{f}$. Moreover, naive outlier removal methods can only tolerate a negligible fraction ε of corruptions (corresponding to $\varepsilon = O(d^{-1/2})$).

A first idea to get around this obstacle is the following: consider the standard (projected) gradient descent method used to minimize $\hat{f}$. This algorithm would proceed by repeatedly computing the gradient of $\hat{f}$ at appropriate points and using it to update the current location. The issue is that adversarial corruptions can completely compromise this algorithm's behavior, since they can substantially change the gradient of $\hat{f}$ at the chosen points. The key observation is that approximating the gradient of $\overline{f}$ at a given point, given access to an ε-corrupted set of samples, can be viewed as a robust mean estimation problem. We can thus use the robust mean estimation algorithm of (Diakonikolas et al., 2017a), which succeeds under fairly mild assumptions about the good samples. Assuming that the covariance matrix of $\nabla f(w)$, $f \sim p^*$, is bounded, we can thus "simulate" gradient descent and approximately minimize $\overline{f}$.

In summary, the first algorithmic idea is to use a robust mean estimation routine as a black-box in order to robustly estimate the gradient at *each* iteration of (projected) gradient descent. This yields a simple robust method for stochastic optimization with polynomial sample complexity and running time in a very general setting.

We now describe SEVER (Algorithm 1) and the main insight behind it. SEVER only calls our robust mean estimation routine (which is essentially the filtering method of (Diakonikolas et al., 2017a) for outlier removal) each time the algorithm reaches an approximate critical point of $\hat{f}$. There are two main motivations for this approach: First, we empirically observed that if we iteratively filter samples, keeping the subset with the samples removed, then few iterations of the filter remove points. Second, an iteration of the filter subroutine (Algorithm 2) is more expensive than an iteration of gradient descent. Therefore, it is advantageous to run many steps of gradient descent on the current set of corrupted samples between consecutive filtering steps. This idea is further improved by using stochastic gradient descent, rather than computing the average at each step.

An important feature of our analysis is that SEVER does not use a robust mean estimation routine as a black box. In contrast, we take advantage of the performance guarantees of our filtering algorithm. The main idea is as follows: Suppose that we have reached an approximate critical point w of $\hat{f}$ and at this step we apply our filtering algorithm. By the performance guarantees of the latter algorithm we are in one of two cases: either the filtering algorithm removes a set of corrupted functions or it certifies that the gradient of $\hat{f}$ is "close" to the gradient of $\overline{f}$ at w. In the first case, we make progress as we produce a "cleaner" set of functions. In the second case, our certification implies that the point w is also an approximate critical point of $\overline{f}$ and we are done.

3. Experiments

In this section we apply SEVER to regression and classification problems. As our base learners, we used ridge regression and an SVM, respectively. We implemented the latter as a quadratic program, using Gurobi (Gurobi Optimization, Inc., 2016) as a backend solver and YALMIP (Löfberg, 2004) as the modeling language.

In both cases, we ran the base learner and then extracted gradients for each data point at the learned parameters. We then centered the gradients and ran MATLAB's `svds` method to compute the top singular vector v, and removed the top p fraction of points i with the largest *outlier score* τ_i, computed as the squared magnitude of the projection onto v (see Algorithm 1). We repeated this for r iterations in total. For classification, we centered the gradients (and removed points) separately for each class, for improved performance.

We compared our method to six baseline methods. All but one of these all have the same high-level form as SEVER (run the base learner then filter top p fraction of points with the largest score), but use a different definition of the score τ_i for deciding which points to filter: **noDefense**: no points are removed, **l2**: remove points where the covariate x has large ℓ_2 distance from the mean, **loss**: remove points with large loss (measured at the parameters output by the base learner), **gradient**: remove points with large gradient (in ℓ_2-norm), **gradientCentered**: remove points whose gradients are far from the mean gradient in ℓ_2-norm, **RANSAC**: repeatedly subsample points uniformly at random, and find the best fit with the subsample. Then, choose the best fit amongst this set of learners. Note that this method is not "filter-based".[2] **gradientCentered** differs from our method in that

[2] In practice, heuristics must often be applied to choose the best fit. In our experiments, we "cheat" slightly by in fact choosing

it removes large gradients in terms of ℓ_2-norm, rather than projection onto the top singular vector.

Both ridge regression and SVM have a single hyperparameter (the regularization coefficient). We optimized this based on the uncorrupted data and then kept it fixed throughout our experiments. In addition, since the data do not already have outliers, we added varying amounts of outliers (ranging from 0.5% to 10% of the clean data); this process is described in more detail below.

3.1. Ridge Regression

For ridge regression, we tested our method on a synthetic Gaussian dataset as well as a drug discovery dataset. The synthetic dataset consists of observations (x_i, y_i) where $x_i \in \mathbb{R}^{500}$ has independent standard Gaussian entries, and $y_i = \langle x_i, w^* \rangle + 0.1 z_i$, where z_i is also Gaussian. We generated 5000 training and 100 test points. The drug discovery dataset was obtained from the ChEMBL database and was originally curated by Olier et al. (2018); it consists of 4084 data points in 410 dimensions; we split this into a training and test set of 3084 and 1000 points, respectively.

Centering the data points decreased error noticeably on the drug discovery dataset; scaling each coordinate to have variance 1 decreased error by a small amount on the synthetic data. To center with outliers, we used the robust mean estimation algorithm from (Diakonikolas et al., 2017a).

Adding outliers. We devised a method of generating outliers that fools all of the baselines while still inducing high test error. At a high level, the outliers cause ridge regression to output $w = 0$, so the model always predicts $y = 0$. If (X, y) are the true data points and labels, this can be achieved by setting each outlier point $(X_{\text{bad}}, y_{\text{bad}})$ as $X_{\text{bad}} = \frac{1}{\alpha \cdot n_{\text{bad}}} y^\top X$ and $y_{\text{bad}} = -\beta$, where n_{bad} is the number of outliers we add, and α and β are hyperparameters. If $\alpha = \beta$, one can check that $w = 0$ is the unique minimizer for ridge regression on the perturbed dataset. By tuning α and β, we can then obtain attacks that fool all the baselines while damaging the model (we tune α and β separately to give an additional degree of freedom to the attack). To increase the error, we also found it useful to perturb each individual X_{bad} by a small amount of Gaussian noise. We found that this method generated successful attacks as long as the fraction of outliers was at least roughly 2% for synthetic data, and roughly 5% for the drug discovery data.

Results. In Figure 2 we compare the test error of our defense against the baselines as we increase the fraction ε of added outliers. To avoid cluttering the figure, we only show the performance of **l2**, **loss**, **gradientCentered**, **RANSAC**, and SEVER; the performance of the remaining baselines is qualitatively similar to the baselines in Figure 2.

For all filter methods, we iterate the defense $r = 4$ times, each time removing the $p = \varepsilon/2$ fraction of points with largest score. For consistency, for each defense and each value of ε we ran the defense 3 times on fresh attack points and display the median of the 3 test errors.

When the attack parameters α and β are tuned to defeat the baselines (Figure 2 left and center), our defense substantially outperforms the baselines as soon as we cross $\varepsilon \approx 1.5\%$ for synthetic data, and $\varepsilon \approx 5.5\%$ for the drug discovery data. In fact, most of the baselines do worse than not removing any outliers at all (this is because they end up mostly removing good data points, which causes the outliers to have a larger effect). Even when α and β are instead tuned to defeat SEVER, its resulting error remains small (Figure 2 right).

To understand why the baselines fail to detect the outliers, in Figure 3 we show a representative sample of the histograms of scores of the uncorrupted points overlaid with the scores of the outliers, for both synthetic data and the drug discovery dataset with $\varepsilon = 0.1$, after one run of the base learner. The scores of the outliers lie well within the distribution of scores of the uncorrupted points. Thus, it would be impossible for the baselines to remove them without also removing a large fraction of uncorrupted points.

Interestingly, for small ε all of the methods improve upon the uncorrupted test error for the drug discovery data; this appears to be due to a small number of natural outliers in the data that all of the methods successfully remove.

3.2. Support Vector Machines

We describe our experimental results for SVMs; we tested our method on a synthetic Gaussian dataset as well as a spam classification task. Similarly to before, the synthetic data consists of observations (x_i, y_i), where $x_i \in \mathbb{R}^{500}$ has independent standard Gaussian entries, and $y_i = \text{sign}(\langle x_i, w^* \rangle + 0.1 z_i)$, where z_i is also Gaussian and w^* is the true parameters (drawn at random from the unit sphere). The spam dataset comes from the Enron corpus Metsis et al. (2006), and consists of 4137 training points and 1035 test points in 5116 dimensions. To generate attacks, we used the data poisoning algorithm presented in Koh et al. (2018).

In contrast to ridge regression, we did not center and rescale these datasets as it had a minimal effect on results.

In all experiments for this section, each method removed the top $p = \frac{n_- + n_+}{\min\{n_+, n_-\}} \cdot \frac{\varepsilon}{r}$ of highest-scoring points for each of $r = 2$ iterations, where n_+ and n_- are the number of positive and negative training points respectively. This expression for p is chosen in order to account for class

the best fit post-hoc by reporting the best error achieved by any learner in this way. Despite strengthening **RANSAC** in this way, we observe that it still has poor performance.

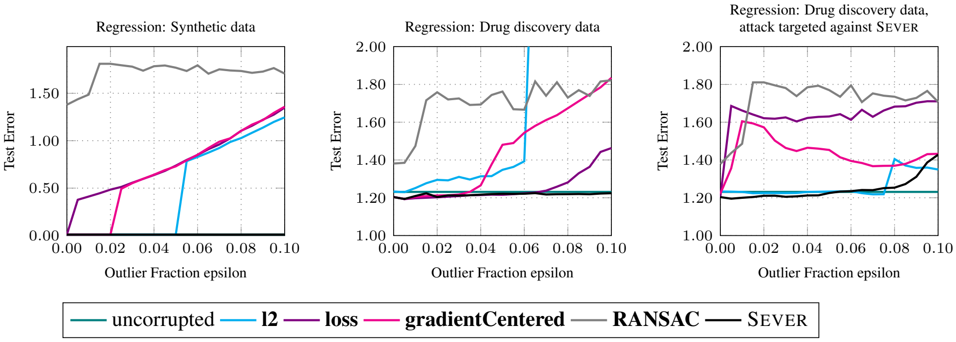

Figure 2. ε vs test error for baselines and SEVER on synthetic data and the drug discovery dataset.

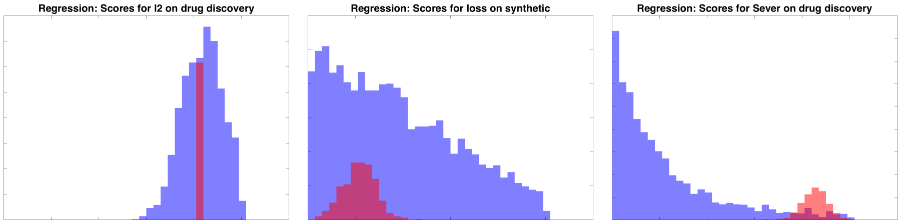

Figure 3. A representative set of histograms of scores for baselines and SEVER on synthetic data and a drug discovery dataset. For the baselines, the scores for the outliers (in red) are inside the bulk of the distribution of the scores of the true dataset (in blue) and thus hard to detect, whereas the scores for the outliers assigned by SEVER are clearly within the tail of the distribution and easily detectable.

imbalance, which is extreme in the case of the Enron dataset – if the attacker plants all the outliers in the smaller class, then a smaller value of p would remove too few points, even with a perfect detection method.

Synthetic results. We considered fractions of outliers ranging from $\varepsilon = 0.005$ to $\varepsilon = 0.03$. By performing a sweep across hyperparameters of the attack, we generated 56 distinct sets of attacks for each value of ε. In Figure 4, we show results for the attack where the **loss** baselines does the worst, as well as for the attack where our method does the worst. When attacks are most effective against **loss**, SEVER substantially outperforms it, nearly matching the test accuracy of 5.8% on the uncorrupted data, while **loss** performs worse than 30% error at just a 1.5% fraction of injected outliers. Even when attacks are most effective against SEVER, it still outperforms **loss**, achieving a test error of at most 9.05%. We note that other baselines behaved qualitatively similarly to **loss**, results are displayed in the supplement.

Spam results. For results on Enron, we used the same values of ε, and considered 96 distinct hyperparameters for the attack. There was not a single attack that simultaneously defeated all of the baselines, so in Figure 4 we show two attacks that do well against different sets of baselines, as well as the attack that performs best against our method.

At $\varepsilon = 0.01$, the worst performance of our method against all attacks was 7.34%, in contrast to $13.43\% - 20.48\%$ for the baselines (note that the accuracy is 3% in the absence of outliers). However, at $\varepsilon = 0.03$, while we still outperform the baselines, our error is relatively large—13.53%.

To investigate this further, we looked at all 48 attacks and found that while on 42 out of 48 attacks our error never exceeded 7%, on 6 of the attacks (including the attack in Figure 4) the error was substantially higher. Figure 5 shows what is happening. The leftmost figure displays the scores assigned by SEVER after the first iteration, where red bars indicate outliers. While some outliers are assigned extremely large scores and thus detected, several outliers are correctly classified and thus have 0 gradient. However, once we remove the first set of outliers, some outliers which were previously correctly classified now have large score, as displayed in the middle figure. Another iteration of this process produces the rightmost figure, where almost all the remaining outliers have large score and will thus be removed by SEVER. This demonstrates that some outliers may be hidden until other outliers are removed, necessitating multiple iterations.

Motivated by this, we re-ran our method against the 6 attacks using $r = 3$ iterations instead of 2 (and decreasing p as per the expression above). After this change, all 6 of the attacks had error at most 7.4%.

4. Discussion

We have presented an algorithm that has both strong theoretical robustness in the presence of outliers, and performs well on real datasets. SEVER is based on the idea that learning can often be cast as the problem of finding an approximate stationary point of the loss, which can in turn be cast as a

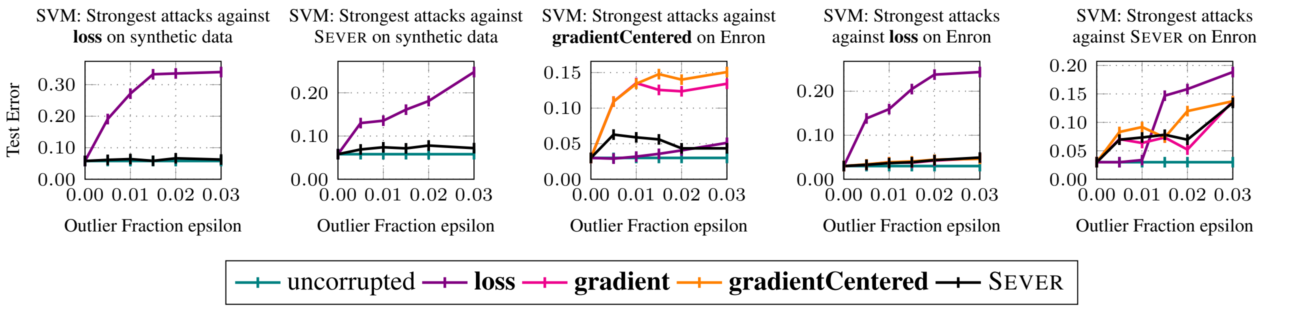

Figure 4. ε vs test error for baselines and SEVER, for SVM on synthetic and Enron data.

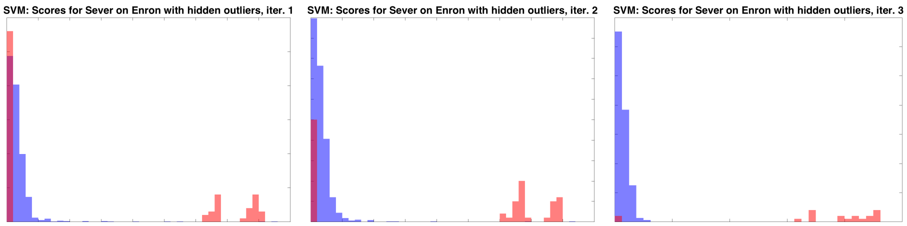

Figure 5. An illustration of why multiple rounds of filtering are necessary: inliers in one round may become outliers in subsequent rounds.

robust mean estimation problem, allowing us to leverage existing techniques for efficient robust mean estimation.

There are a number of directions along which SEVER could be improved: first, it could be extended to handle more general assumptions on the data; second, it could be strengthened to achieve better error bounds in terms of the fraction of outliers; finally, one could imagine *automatically learning* a feature representation in which SEVER performs well. We discuss each of these ideas in detail below.

More general assumptions. The main underlying assumption on which SEVER rests is that the top singular value of the gradients of the data is small. While this appeared to hold true on the datasets we considered, a common occurence in practice is for there to be *a few* large singular values, together with *many* small singular values. It would be desirable to design a version of SEVER that can take advantage of such phenomena. Also, it would be worthwhile to do a more detailed empirical analysis across a wide variety of datasets investigating properties that can enable robust estimation (the notion of *resilience* in (Steinhardt et al., 2018) could provide a template for finding such properties).

Stronger robustness to outliers. In theory, SEVER has a $O(\sqrt{\varepsilon})$ dependence in error on the fraction ε of outliers (see Theorem 2.1). While without stronger assumptions this is likely not possible to improve, in practice we would prefer to have a dependence closer to $O(\varepsilon)$. Therefore, it would also be useful to improve SEVER to have such an $O(\varepsilon)$-dependence under stronger but realistic assumptions. Unfortunately, all existing algorithms for robust mean estimation that achieve error better than $O(\sqrt{\varepsilon})$ either rely on strong distributional assumptions such as Gaussianity (Diakonikolas et al., 2016a; Lai et al., 2016), or else require expensive computation involving e.g. sum-of-squares optimization (Hopkins & Li, 2017; Kothari & Steinhardt, 2017; Kothari & Steurer, 2017). Improving the robustness of SEVER thus requires improvements on the robust mean estimation algorithm that SEVER uses as a primitive.

Learning a favorable representation. We note that SEVER performs best when the features have small covariance and strong predictive power. One situation in particular where this holds is when there are many approximately independent features that are predictive of the true signal.

It would be interesting to try to learn a representation with such a property. This could be done, for instance, by training a neural network with some cost function that encourages independent features (some ideas along these general lines are discussed in Bengio (2017)). An issue is how to learn such a representation robustly; one idea is learn a representation on a dataset that is known to be free of outliers, and hope that the representation is useful on other datasets in the same application domain.

Beyond these specific questions, we view the general investigation of robust methods (both empirically and theoretically) as an important step as machine learning moves forwards. Indeed, as machine learning is applied in increasingly many situations and in increasingly automated ways, it is important to attend to robustness considerations so that machine learning systems behave reliably and avoid costly errors. While the bulk of recent work has highlighted the vulnerabilities of machine learning (e.g. (Szegedy et al., 2014; Li et al., 2016; Steinhardt et al., 2017; Eykholt et al., 2018; Chen et al., 2017)), we are optimistic that practical algorithms backed by principled theory can finally patch these vulnerabilities and lead to truly reliable systems.

Acknowledgements

The authors were supported by NSF Awards CCF-1652862, CCF-1617730, CCF-1650733, CCF-1741137, CCF-1553288, CCF-1453261, CCF-1565235, ONR N00014-12-1-0999, Sloan Research Fellowships, a Google Faculty Research Award, an NSF Graduate Research Fellowship, a Fannie & John Hertz Foundation Fellowship, a Future of Life Institute grant, and a USC startup grant.

References

Awasthi, P., Balcan, M. F., and Long, P. M. The power of localization for efficiently learning linear separators with noise. In *Symposium on Theory of Computing (STOC)*, pp. 449–458, 2014.

Balakrishnan, S., Du, S. S., Li, J., and Singh, A. Computationally efficient robust sparse estimation in high dimensions. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017*, pp. 169–212, 2017.

Barreno, M., Nelson, B., Joseph, A. D., and Tygar, J. D. The security of machine learning. *Machine Learning*, 81 (2):121–148, 2010.

Bengio, Y. The consciousness prior. *arXiv preprint arXiv:1709.08568*, 2017.

Bhatia, K., Jain, P., and Kar, P. Robust regression via hard thresholding. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, pp. 721–729, 2015.

Bhatia, K., Jain, P., Kamalaruban, P., and Kar, P. Consistent robust regression. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pp. 2107–2116, 2017.

Biggio, B., Nelson, B., and Laskov, P. Poisoning attacks against support vector machines. In *International Conference on Machine Learning (ICML)*, pp. 1467–1474, 2012.

Breunig, M. M., Kriegel, H., Ng, R. T., and Sander, J. Lof: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pp. 93–104. ACM, 2000.

Charikar, M., Steinhardt, J., and Valiant, G. Learning from untrusted data. In *Proceedings of STOC 2017*, pp. 47–60, 2017.

Chen, X., Liu, C., Li, B., Lu, K., and Song, D. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., and Stewart, A. Robust estimators in high dimensions without the computational intractability. In *Proceedings of FOCS'16*, pp. 655–664, 2016a.

Diakonikolas, I., Kane, D. M., and Stewart, A. Robust learning of fixed-structure bayesian networks. *CoRR*, abs/1606.07384, 2016b.

Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., and Stewart, A. Being robust (in high dimensions) can be practical. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, pp. 999–1008, 2017a. Full version available at https://arxiv.org/abs/1703.00893.

Diakonikolas, I., Kane, D. M., and Stewart, A. List-decodable robust mean estimation and learning mixtures of spherical gaussians. *CoRR*, abs/1711.07211, 2017b. URL http://arxiv.org/abs/1711.07211.

Diakonikolas, I., Kane, D. M., and Stewart, A. Learning geometric concepts with nasty noise. *CoRR*, abs/1707.01242, 2017c. URL http://arxiv.org/abs/1707.01242.

Diakonikolas, I., Kane, D. M., and Stewart, A. Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017*, pp. 73–84, 2017d. Full version available at http://arxiv.org/abs/1611.03473.

Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., and Stewart, A. Robustly learning a gaussian: Getting optimal error, efficiently. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018*, pp. 2683–2702, 2018. Full version available at https://arxiv.org/abs/1704.03866.

Diakonikolas, I., Kong, W., and Stewart, A. Efficient algorithms and lower bounds for robust linear regression. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019*, pp. 2745–2754, 2019.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., and Song, D. Robust physical-world attacks on deep learning visual classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Fischler, M. A. and Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2016.

Hodge, V. and Austin, J. A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2):85–126, 2004.

Hopkins, S. B. and Li, J. Mixture models, robustness, and sum of squares proofs. *CoRR*, abs/1711.07454, 2017. URL http://arxiv.org/abs/1711.07454.

Huber, P. J. Robust estimation of a location parameter. *Ann. Math. Statist.*, 35(1):73–101, 03 1964.

Klivans, A. R., Long, P. M., and Servedio, R. A. Learning halfspaces with malicious noise. *Journal of Machine Learning Research (JMLR)*, 10:2715–2740, 2009.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning (ICML)*, 2017.

Koh, P. W., Steinhardt, J., and Liang, P. Stronger data poisoning attacks break data sanitization defenses. *arXiv preprint arXiv:1811.00741*, 2018.

Kothari, P. K. and Steinhardt, J. Better agnostic clustering via relaxed tensor norms. *CoRR*, abs/1711.07465, 2017. URL http://arxiv.org/abs/1711.07465.

Kothari, P. K. and Steurer, D. Outlier-robust moment-estimation via sum-of-squares. *CoRR*, abs/1711.11581, 2017. URL http://arxiv.org/abs/1711.11581.

Lai, K. A., Rao, A. B., and Vempala, S. Agnostic estimation of mean and covariance. In *Proceedings of FOCS'16*, 2016.

Li, B., Wang, Y., Singh, A., and Vorobeychik, Y. Data poisoning attacks on factorization-based collaborative filtering. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Li, J., Absher, D., Tang, H., Southwick, A., Casto, A., Ramachandran, S., Cann, H., Barsh, G., Feldman, M., Cavalli-Sforza, L., and Myers, R. Worldwide human relationships inferred from genome-wide patterns of variation. *Science*, 319:1100–1104, 2008.

Löfberg, J. YALMIP: A toolbox for modeling and optimization in MATLAB. In *CACSD*, 2004.

Meister, M. and Valiant, G. A data prism: Semi-verified learning in the small-alpha regime. In *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pp. 1530–1546. PMLR, 06–09 Jul 2018.

Metsis, V., Androutsopoulos, I., and Paliouras, G. Spam filtering with naive Bayes – which naive Bayes? In *CEAS*, volume 17, pp. 28–69, 2006.

Nasrabadi, N. M., Tran, T. D., and Nguyen, N. Robust lasso with missing and grossly corrupted observations. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

Newell, A., Potharaju, R., Xiang, L., and Nita-Rotaru, C. On the practicality of integrity attacks on document-level sentiment analysis. In *Workshop on Artificial Intelligence and Security (AISec)*, pp. 83–93, 2014.

Nguyen, N. H. and Tran, T. D. Exact recoverability from dense corrupted observations via ℓ_1-minimization. *IEEE Transactions on Information Theory*, 59(4):2017–2035, 2013.

Olier, I., Sadawi, N., Bickerton, G. R., Vanschoren, J., Grosan, C., Soldatova, L., and King, R. D. Meta-qsar: a large-scale application of meta-learning to drug design and discovery. *Machine Learning*, 107(1):285–311, Jan 2018. ISSN 1573-0565. doi: 10.1007/s10994-017-5685-x. URL https://doi.org/10.1007/s10994-017-5685-x.

Owen, A. B. A robust hybrid of lasso and ridge regression. *Contemporary Mathematics*, 443(7):59–72, 2007.

Paschou, P., Lewis, J., Javed, A., and Drineas, P. Ancestry informative markers for fine-scale individual assignment to worldwide populations. *Journal of Medical Genetics*, 47:835–847, 2010.

Prasad, A., Suggala, A. S., Balakrishnan, S., and Ravikumar, P. Robust estimation via robust gradient estimation. *CoRR*, abs/1802.06485, 2018. URL http://arxiv.org/abs/1802.06485.

Qiao, M. and Valiant, G. Learning discrete distributions from untrusted batches. In *Innovations in Theoretical Computer Science (ITCS)*, 2018.

Rosenberg, N., Pritchard, J., Weber, J., Cann, H., Kidd, K., Zhivotovsky, L., and Feldman, M. Genetic structure of human populations. *Science*, 298:2381–2385, 2002.

Rousseeuw, P. J. and Driessen, K. V. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

Steinhardt, J. Does robustness imply tractability? A lower bound for planted clique in the semi-random model. *arXiv*, 2017.

Steinhardt, J., Koh, P. W., and Liang, P. Certified defenses for data poisoning attacks. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

Steinhardt, J., Charikar, M., and Valiant, G. Resilience: A criterion for learning in the presence of arbitrary outliers. In *Innovations in Theoretical Computer Science (ITCS)*, 2018.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Tukey, J. Mathematics and picturing of data. In *Proceedings of ICM*, volume 6, pp. 523–531, 1975.

Tukey, J. W. A survey of sampling from contaminated distributions. *Contributions to probability and statistics*, 2:448–485, 1960.