Algorithms for Robust Linear Models against Strong Adversarial Corruptions

Yimeng Wang

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2022-97
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2022/EECS-2022-97.html

May 13, 2022

Algorithms for Robust Linear Models against Strong Adversarial Corruptions

by

Yimeng Wang

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Professor Prasad Raghavendra
Research Advisor

5/12/2022
(Date)

* * * * * * *

Jacob Steinhardt

Professor Jacob Steinhardt
Second Reader

05/13/2022
(Date)

Algorithms for Robust Linear Models against Strong Adversarial Corruptions

Yimeng Wang

May 2022

Algorithms for Learning Robust Linear Models under the Strong Contamination model

A Survey of Filtering methods and Sum-of-squares techniques

Yimeng Wang

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in chage:

Professor Prasad Raghavendra, Chair
Professor Jacob Steinhardt

Spring 2022

Abstract

Algorithms for Robust Linear Models against Strong Adversarial Corruptions

A survey of algorithms in Robust Linear Models under the Strong-Contamination Model

by

Yimeng Wang

Master of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Prasad Raghavendra, Chair

Professor Jacob Steinhardt

In real-world data science and machine learning, data are inevitably imperfect. Data contamination comes in many sources. It may come from human errors which can be avoided with more caution. However, it may also come from sources such as systematic measurement errors and adversarial data poisoning that are hard to avoid and even detect. Consequently, there is a need for methods that can perform certain tasks in statistics despite this difficulty. Formally speaking, we want to design efficient algorithms that can provide provable guarantees for learning problems under certain models of contamination.

In the article, we examine some important techniques in the recent development of efficient algorithms for robust statistics, namely filtering-based methods and sum-of-squares techniques. Specifically, we will focus on the problem of learning linear models (including linear regression, generalized linear models etc.) under the strong contamination model. We will fully present and analyze the conditions and consequences of SEVER [DKK$^+$19] and the sum-of-squares-based algorithm for robust linear regression in [KKM20]. SEVER is meta-algorithm that takes in a well-conditioned base learner and output a outlier-robust version of the base learners. The [KKM20] robust linear regression algorithm is an elegant and simple application of sum-of-squares techniques for robust regressions in general including l_1, l_2 and polynomial regression. Both algorithms have $O(\sqrt{\epsilon})$-dependence in error on the fraction of outlier ϵ. We will present and prove the theoretical guarantees of these algorithms which shed lights on future directions in which the error dependence and the required assumptions can be improved.

Acknowledgements

Having spent five years at Berkeley as both an undergraduate student and Masters student, there are many people I am grateful for. I want to express my sincere gratitude to Professor Prasad Raghavendra for advising me over the past year. I genuinely appreciate his insights on both the technical side of research and the general philosophy of how to do great research. I would like to tribute my sincere thanks to my undergraduate advisor Professor Alistair Sinclair for introducing me to doing research in Theoretical Computer Science. I would like to thank Professor Jacob Steinhardt for the intellectual conversations when I was working with him as a teaching assistant. I would like express my thanks to Professor Satish Rao for the wonderful CS 70, CS 170 and CS 270 classes through which I have found my passion.

I would like to thank my parents Jinhua Wang and Jian Li for supporting my decision to pursue an academic career. I am grateful for my partner Christina Jin for supporting me through this journey. Last but not least, I want to express my gratitude to all my friends at Berkeley. This could not have been done without them.

Contents

List of Figures

1 Introduction

1.1 Motivation

Real-world data modeling is difficult, as any data scientist and machine learning practitioner would tell you. Various of factors contribute to this hardness. The modeling part is hard. In real-word applications, we only have limited knowledge of the underlying data distribution. There are numerous models one can choose to employ. How does one choose a model? Ideally, with years of expertise in machine learning and domain-specific knowledge, one would expect to choose the "best model." However, it is ambiguous what the "best" model means as there are various definitions of "goodness" and this definition may change depending on the question one intends to solve. Machine learning practitioners face trade-offs in the sense that optimizing one aspect of the model typically means sacrificing some other aspects. For instance, the time-accuracy trade-off in deep-learning and the cost-performance trade-off for individuals or small groups training large models without much computing power and the famous bias-variance trade-off in statistics.

Getting good data is hard. It is time-consuming and expensive to collect data. Natural scientists require expensive pieces of equipment to observe and collect data. Social science researchers have to carefully design experiments in order to obtain useful data. However, even these costly procedures, we are not guaranteed to have good data to work with.

One example of undesirable data to work with is biological data such as DNA sequencing and expression data. Mislabeling and measurement errors occur frequently which can create systematic outliers [NRF02, JLM08] and it requires painstaking manual effort to remove the outliers. See Figure 1.1 [Li19] for such an example.

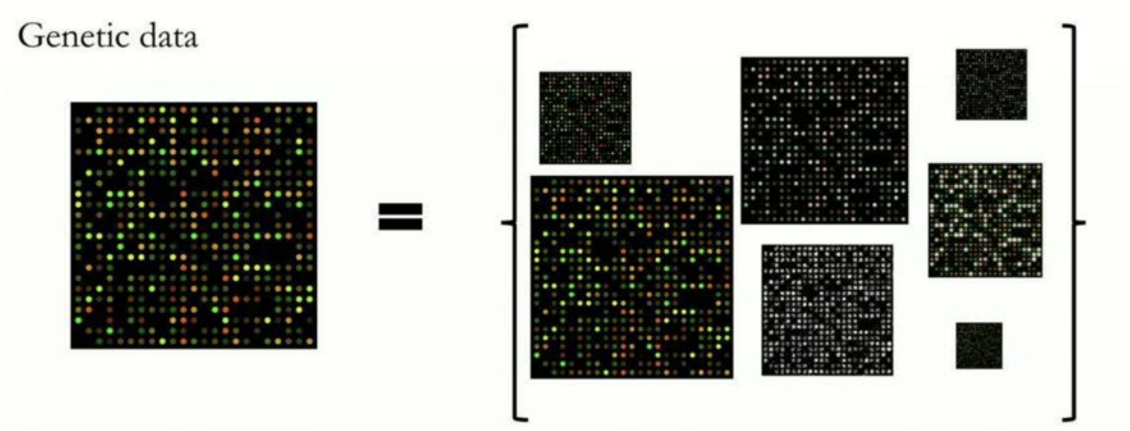

Figure 1: The observed gene expression data (on the left) is a mixture of various heterogeneous gene expression data (on the right). Independently and Identically distributed samples are not realistic to obtain because of cost and technology constraints.

Another example commonly known as *data-poisoning*, comes from computer security. An attacker may hack into the server and maliciously erase or alter certain fractions of the data. In cyber-security, hackers may create fake accounts to insert fake data into the dataset. Other common examples include communication through noisy and adversarial channels, heterogeneous financial data, etc. These examples exemplify cases in which data corruptions cannot be avoided. We need methods to account for these corruptions.

One important thing to notice is that the outliers in the two examples above are not random. Instead, in both of the examples, the outliers are systematic errors that can be purely adversarial, difficult to anticipate and hard to model. This motivates the following important question: *Can we design algorithms that can recover important information (mean, moments, etc.) about the true underlying data distribution despite the presence of certain fraction of arbitrary (and potentially adversarial) outliers?*

In literature, researchers refer to this line of research as *robust statistics*. Let us consider an example of a robust vs non-robust quantity in one-dimension. In an introduction-level statistics class, one is usually presented with to concepts: *mean* and *median*. While mean and median are measures of central tendency of a given distribution, they behave rather differently when the dataset is susceptible to potential corruptions. Mean is not robust to corruptions in the sense that even the presence of a single outlier can arbitrarily shift the mean. On the other hand, median is more robust to corruptions because it is a *location* measure which is stable under corruptions that do not add / delete points at the tails of the distribution. We say that median is a robust estimate of mean in one-dimension. We will define what exactly it means to be a "robust estimate" later. See the nice introduction in [Ste21] for more details of this.

Given that median is a robust estimate of mean in one-dimension, a natural question to ask is: *Can we extend median to higher dimensions as a robust estimate of mean for high-dimensional data?* Sadly, the answer is no. Inference and estimation in high-dimensions are difficult in general [Wai19, Ver18] as most naive extension of low-dimension estimators do not perform well in high dimensions. The coordinate-wise median approach which is a naive extension of median to high dimensions but its error grows at a rate of $\sqrt{d}$ where d is the dimension of the data.

One approach that works well in one-dimension is to remove points that are far from the sample median, i.e. remove the lower and upper quantiles of the observed data. However, in high dimensions, this does not perform well. To see this, let's consider the following example in [Ste21]. Suppose we observe i.i.d samples $x_i, ..., x_n \in \mathbb{R}^d$ from the true underlying distribution $p^* = \mathcal{N}(\mu, I)$. Our goal is to estimate the unknown parameter μ. In this case, the distance $\|x_i - \mu\|_2$ is concentrated around $\sqrt{d}$ with high probability. Hence, if the corrupted points lie at roughly $\sqrt{d}$ from μ, they are indistinguishable from the in-distribution points based on this removal procedure using l_2 distance. See Figure 1.1 [Ste21]. This suggests that these outliers can shift shift the mean by $\Theta(\epsilon\sqrt{d})$ where ϵ is the fraction of corrupted points. Consequently, this approach suffers an error on the order of $\sqrt{d}$ which is meaningless in high dimensions. With more sophisticated approaches, we can obtain much better guarantees in high dimensions. See [CDG18, CDGS20] for some recent advances in robust mean estimation.

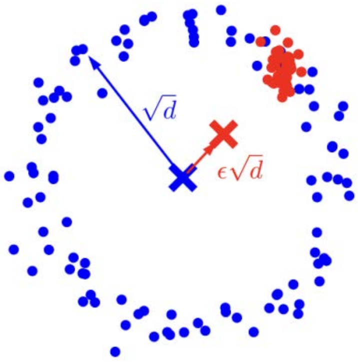

Figure 2: The outliers can lie at a distance of $\sqrt{d}$ without being detected. See [Ste21].

Another challenge robust statistics faces in general is the need for distributional assumptions. For example, one is not able to conclude that points are outliers if they are far from the sample mean if the samples are drawn from a heavy-tail distribution. In other words, our characterization of outliers should depend on the properties of the true underlying distribution.

To summarize, robust statistics is inherently a difficult task because of challenges including but not limited to the following:

1. *Distributional assumptions.* One needs appropriate distributional assumptions for problems in robust statistics. An excess of assumptions may yield results that are not applicable to other circumstances. On the other hand, a lack of assumptions may make the problem too difficult, i.e, information theoretically impossible.

2. *Powerful adversaries.* In this field of study, we are generally interested in worst-case scenarios rather than average-case scenarios. In average-case robustness, one requires the errors to satisfy certain distributions for robust guarantees to be meaningful [Ste21], while our goal in robust statistics is to provide algorithms and provable guarantees to handle unanticipated attacks from malicious and possibly omnipotent adversaries.

3. *High dimensional data.* As explained above, some algorithms that work well in low-dimensions do not naively extend to provide good performance guarantees in high dimensions. See [Wai19] for a detailed exposition of high-dimensional statistics in general. To cope with this, we need

algorithms that has error bounds that do not rely on the dimension d so that it scales well in high dimension.

Our goal is to design *computationally efficient* algorithms that can address these challenges.

1.2 Robust Statistics

In this section, we revisit some earlier attempts to deal with the challenges above in robust statistics literature. Learning in the presence of outliers has been studied in the community since the pioneering work of Tukey in the 1960s [Tuk60]. Classical works in robust statistics focus more on optimality in terms of the minimax risk of robust estimation in some basic settings. See [Ham86, HR11] for technical details of some traditional approaches in this area of study. Also, see [Mor07] for a summary of early discoveries and contributions.

Some popular methods in the area include RANSAC [FB87], minimum covariance determinant [RD99], removal procedure based on k-nearest neighbors [BKNS00] and Hubernizing the loss function [Owe07]. Despite the popularity, these methods either break down in high dimensions or require strong distributional assumptions on the data so that the outliers are easily detectable from the in-distribution points.

Another issue of traditional methods in robust statistic is the lack of attention on the computational aspect of learning. Computational efficiency of estimators was not the main focus in early days of robust statistics. Little attention was paid to the computational aspect of the estimators and some basic computational questions were not well-understood until recently. As an example, let's consider the Tukey median estimator [Tuk75]. Earlier studies have shown good theoretical guarantees of the estimator. See for instance [ZJS20]. In particular, it is known that the Tukey median is a sample-efficient robust mean estimator for spherical Gaussian distributions [DK19]. However, it is NP-Hard to compute in general [JP78] and the approximation accuracy of the proposed heuristics degrades in high dimensions. Hardt and Moitra showed that estimators for robust subspace recovery are inefficient[HM13] while Bernholt proved that robust estimators including LMS, LQS, LTS are hard to compute [Ber06].

Motivated by these unsolved mysteries, recent work in theoretical computer science has developed computationally efficient robust estimators for classical problems including regression [BJK15, SBS17, BJKK18], linear classification [KLS09, PAL17] and mean and covariance estimation [DKK$^+$16, LRV16]. One influential recent work is by Diakonikolas, Kamath, Kane et al. [DKK$^+$19] in which they introduce a new meta-algorithm that take in a base learner and harden the learner to be resistant to outliers. Another highlight in recent years that is of particular interest to us is the work by Klivans, Kothari and Meka [KKM20]. They gave the first polynomial-time algorithm for performing linear or polynomial regression resilient to adversarial corruptions in both examples and labels using a simple and delicate sum-of-squares proof.

The reason we specifically mentioned these two works [DKK$^+$19, KKM20] is that they represent two different yet important methods in recent development in robust statistics. In [DKK$^+$19], Diakonikolas et al. presented a meta-algorithm whose main component includes a *filtering* subroutine. The idea of a *filtering* is relatively intuitive. Since our dataset has been corrupted, we will just *remove bad points from the dataset until all points are reasonably good.* As simple as it might sound, this approach needs to be carefully carried out. How do we determine which points are "bad?" We need a score to quantify the "badness" of each point and remove those that have high scores. There are various ways of defining this. See the survey by Diakonikolas and Kane [DK19]. Some examples include but are not limited to l_2 distance from the sample mean, projection along the principal components of the sample covariance matrix and projection along the top singular vector of the sample gradient matrix of the loss functions [DKK$^+$19]. The performance of filtering-based algorithms depends heavily on choice of the score function. We will investigate into the details and subtleties of filtering-based approaches in the following sections.

In [KKM20], Klivans et al. gave an efficient algorithm for robust linear regression using sum-of-squares methods. Since the pioneering work of [BKS14], *sum-of-squares* methods have been widely studied for designing efficient algorithms for learning problems. The high-level idea is to give a low-degree *sum-of-squares proof* (will define this later) of certain statements (e.g. low-error guarantees for linear regression, unique identifiability of parameters for parameter recovery problems). Then by the "duality" of sum-of-squares proofs and pseudo-distributions, output a pseudo-distribution that can be used to output a solution. This sum-of-squares based approach for learning problems relies heavily on the convexity of the empirical loss function and the crucial fact that the l_2 loss function is a polynomial. How to apply sum-of-squares methods to learning problems with non-convex and non-polynomial loss functions (e.g. logistic regression) is still an open problem.

1.3 Outline of the paper

The main focus of this paper is on linear models. This includes linear regression and generalized linear models including logistic regression. When introducing the Filtering method, we will focus on robust mean estimation for simplicity and clarity. In section 2, we will go over some preliminaries needed for our exposition. We will give formal definitions of common noise models and the statistical problems we are interested in. In section 3, we will go over the fundamentals of the Filtering method which is widely used in the robust statistics community. In section 4, we introduce SEVER algorithm [DKK$^+$19] as an important example of the filtering method in recent years. In section 5, we cover the fundamentals of sum-of-squares method and in section 6, we see how sum-of-squares methods can be applied to give an efficient algorithm for robust linear regression. We conclude in section 7 with summaries of results and some open problems in this area.

2 Preliminaries

In this section, we state the notations we are going to use in the rest of the paper. We will then provide definitions of various contamination models that are common in robust statistic literature.

2.1 Notation

Throughout this paper, we will use capital letters to denote distributions, e.g. P. We will use either lower case or upper case letters to denote random variables depending on the context. We use the notation $x \sim X$ to denote that x is a sample draw from distribution X. Let $x_1, ..., x_n \sim_{i.i.d} X$ to denote n independent and identically distributed samples from distribution X.

We use $P[E]$ to denote the probability of the event E. The expected value and variance / covariance of a random variable X are denoted as $\mathbf{E}[X]$ and $\mathbf{Var}(X)$/ $\mathbf{Cov}(X)$ respectively. Let S be a finite set. Then we use μ_S and Σ_S to denote the sample mean and sample covariance of the set S respectively.

For a vector $v \in \mathbb{R}^d$, we let $\|v\|_2$ to denote its l_2 norm. We use $\|v\|_\infty$ and $\|v\|_1$ to denote the l_∞ and l_1 norm respectively. Let X be a random variable. Then we use the notation $\|X\|_k$ to denote the k-norm $\|X\|_k = \mathbf{E}[X^k]^{1/k}$. Given a matrix A, let $\text{tr}(A)$ be its trace. Let $\|A\|_2$ and $\|A\|_{\text{op}}$ to denote the operator norm of A. We will use these two notations interchangably throughout this paper. We use $\|A\|_{\text{F}}$ to denote the Frobenius norm of A.

We will be using the standard asymptotic notations $O(\cdot), \Omega(\cdot), \Theta(\cdot)$ to denote sample complexities and running time of our algorithms. We will also be using $\tilde{O}(\cdot)$ notation hides logarithmic factors in its argument.

We will use ϵ-corrupted and η-corrupted interchangably when denoting corrupted samples under the strong contamination model. Specifically, we will be using ϵ-corrupted in the first half of this paper when we will be talking about filtering-based methods. In the second half, we will generally using the phrase η-corrupted. The reason for this is that we want to follow the notations in the two main subjects of our discussion, namely [DKK$^+$19] and [KKM20].

2.2 Contamination model

In Huber's contamination model, proposed by Huber in [Hub64], the adversary is oblivious to inliners and can only add outliers. Formally, this contamination model is defined as:

Definition 2.1. *(Huber's contamination model) Suppose $X \sim \mathcal{P}^*$ where $\mathcal{P}^*$ is the true underlying distribution of the random variable X. Let $\eta \in (0, 1/2)$ be a constant. Then under Huber's contamination model (or the η-contamination model), samples $x_1, ..., x_n$ are drawn from the distribution*

$$\mathcal{P} = (1 - \eta)\mathcal{P}^* + \eta\mathcal{N}$$

where $\mathcal{N}$ is an adversarily chosen noise distribution.

In this paper [Hub64], he proposed a robust location estimator that achieve its minimax optimality under Huber's contamination model [CGR17]. Moreover, his work suggests that an optimial estimator under Huber's contamination model must achieve statistical efficiency and resistant to outliers at the same time. Another interesting model of contamination is the *TV-distance corruption model.*

Definition 2.2. *(TV-distance corruption model). Suppose $X \sim \mathcal{P}^*$ where $\mathcal{P}^*$ is the true underlying distribution of the random variable X. Let $\mathcal{P}'$ be another distribution that is at most η away from $\mathcal{P}^*$ in TV distance, i.e, $d_{TV}(\mathcal{P}', \mathcal{P}^*) \leq \eta$. Then under TV-distance corruption model, samples $x_1, ..., x_n$ are drawn from $\mathcal{P}^*$.*

where given two probability distributions p, q defined on the same σ-algebra $\mathcal{F}$ on subsets of the sample space Ω, the total-variation (TV) distance is given by $d_{\text{TV}}(p, q) = \sup_{A \in \mathcal{F}} |p(A) - q(A)|$. This noise model is more functional in nature and has been widely studied in the statistics community. See lecture notes of Steinhardt for some recent results in robust statistics under this noise model [Ste21]. Notice that TV-distance corruption model is strictly strong than Huber's contamination model. The model we are interested in is stronger than both of the models above and is referred to as the *strong contamination model.*

Definition 2.3. *(Strong contamination model). Let $\eta \in (0, 1/2)$ be a constant parameter and let $\mathcal{D}$ be a distribution family over $\mathbb{R}^d$. Given samples $X_1, ..., X_n \sim X$ for some unknown distribution $X \in \mathcal{D}$, the adversary is allow to inspect all samples, remove up to ηn of then and replace the* *removed samples with arbitrary points. This modified set of n points is then given as input to a machine learner.*

Without any specification, we refer to a set of samples as η-corrupted if they are generated through the strong contamination model. Note that in some sense the Strong contamination model is the strongest possible adversary provided one can only remove up to ϵ-fraction of the points. This is true since the adversary can make arbitrary modifications *after inspecting all samples.* Intuitively, this adversary can create the worst case scenarios for our estimation and learning tasks.

3 Filtering

In this section, we will focus on the filtering method that is commonly used in robust statistics. The intuition of the method is quite clear: given some estimation task, we want to remove the *bad* points that can adversarially affect our task and ensure that remaining points are reasonably *good.* We will provide some basic notions needed for the filtering method. Then, we will go over several filtering methods following the exposition in the survey by Diakonikolas and Kane [DK19].

3.1 Preliminaries

For simplicity, we will be focusing only on the Robust Mean Estimation problem: given an η-corrupted set of samples (under the strong contamination model) from a well-behaved distribution with mean μ, we want to output a vector $\hat{\mu}$ that is close to μ in certain distance metric $\|\cdot\|$. In some sense, robust mean estimation is the simplest task in robust statistics. However, even for this task, it is information-theoretically impossible to devise an algorithm without making any distributional assumptions.

To see this, consider the following example from [DK19]. Let $\mathcal{D} = \{D_x, x \in \mathbb{R}\}$ where D_x is a distribution over $\mathbb{R}$ such that $x \in \mathbb{R}$ is the only point with positive mass, i.e. $P(D_x = x) = \eta > 0$ such that $\mathbf{E}[D_x] = x$. In other words, D_x can be regarded as a mixture of a point-mass and any other arbitrary continuous distributions such that the mean of the mixture is x. For the task of mean estimation, given a η-corrupted sample of n points under the strong contamination model, the adversary can erase all points from the point mass and replace them with arbitrary points on the real line. In this case, all information about the mean is lost and it is information theoretically impossible to recover the mean in this case.

As a result, we need some distributional assumptions in order to make the task even possible. Some common assumptions include parametric families (e.g., $\mathcal{D}$ can be the family of isotropic Gaussian distributions), moment boundedness assumptions (e.g. bounded covariance, bounded higher moments or hypercontractivity) and concentration assumptions (e.g. sub-gaussian or sub-exponential tails).

Another important observation is that in contrast to the uncorrupted setting, in strong contamination model, it is not possible to obtain statistically *consistent* estimators. By "statistically consistent," we mean that the error of our estimator does not go to zero as sample size $n \to \infty$. Typically, there is an information-theoretic lower bound on the minumum attainable error that depends on the level of corruption η and the structural properties of the underlying distribution family [DK19].

As a concrete example of the difficulty of robust mean estimation, let's consider the high-dimensional isotropic Gaussian family.

Fact 3.1. *For any $d \geq 1$, any robust estimator for the mean of $X = \mathcal{N}(\mu, I)$ in $\mathbb{R}^d$ must have an l_2-error of $\Omega(\eta)$, even in Huber's contamination model.*

To see why this is true, let's consider the following example. Let $X = \mathcal{N}(\mu, I)$ and $X' = \mathcal{N}(\mu', I)$ such that $\|\mu_1 - \mu_3\|_2 = \Theta(\eta)$. In Huber's contamination model, which is weaker than the strong contamination model, the adversary can construct two noise distributions N_1 and N_2 over $\mathbb{R}^d$ such that

$$Y = (1-\eta)\mathcal{N}(\mu, I) + \eta N_1 = (1-\eta)\mathcal{N}(\mu', I) + \eta N_2.$$

It is easy to see that such N_1, N_2 exists. Hence, under any of the level-η corruption model we introduced in section 2, the best a robust estimator can do is to learn that the samples are from Y but it cannot tell whether it comes from X or X'. In fact, if the corruption level is at least the TV-distance between X and X', then no robust estimators can distinguish between X and X'. For this TV-distance bound on Gaussian families, see [DMR22].

If the target distribution X is allowed to come from a wider class, then the situation is even worse. Suppose X comes from a class of distribution with sub-gaussian tails with identity covariance, then the information theoretical lower bound of the l_2-error for robust mean estimation is $\Omega(\epsilon\sqrt{\log(1/\epsilon)})$; and for the class of distributions with covariance Σ bounded by $\sigma^2 I$, the l_2-error is $\Omega(\sigma\sqrt{\epsilon})$.

As stated in section 1, another difficulty in robust statistics (and statistics in general), is high-dimensionality. Some natural generalizations of low-degree robust estimators do not work well in high-dimensions in that they suffer a loss that scales with the dimension d. As an example, in one-dimension, median is a robust estimator of mean. Specifically, the median $\hat{\mu}$ of a multiset of size $n = \Omega(\log(1/\delta)/\epsilon^2)$ of η-corrupted samples from a one-dimensional Gaussian $\mathcal{N}(\mu, 1)$ satisfies $|\hat{\mu} - \mu| < \eta$ with probability at least $1 - \delta$. One natural generalization of the one-dimensional median to high dimensions in the coordinate-wise median. As the name suggests, the estimator $\hat{\mu} \in \mathbb{R}^d$ is given by $(\hat{\mu})_i = \text{median}(\{X_j^{(i)}\}_{j=1}^n)$ where $i \in [d]$ and $X_j^{(i)}$ is the projection of X_j along the canonical basis vector e_i. This generalization incurs a l_2-loss of $\Omega(\epsilon\sqrt{d})$. Another natural generalization of the one-dimensional median is via the *geometric median*, i.e. the point x^* that minimizes $\sum_i \|x^{(i)} - x^*\|_2$. Unfortunately, this approach again suffers a loss of $\Omega(\epsilon\sqrt{d})$ the η-fraction of the adversarial points are added all off from the mean in the same direction.

A more sophisticated generalization of one-dimensional median is the *Tukey median* [Tuk75]. It relies on the observation that taking the median of any univariate projection of the input points gives us an approximation to the projected mean. With this observation, we can then output an estimator $\hat{\mu}$ that minimizes the error over the *worst direction.* It can be shown that this estimator $\hat{\mu}$ can obtain a l_2-error of $O(\epsilon)$ with high probability. In other words, via these univariate projections, we can reduce the problem of high-dimensional mean estimation to the problem of one-dimensional mean estimation. However, while this method achieves the optimal error bound, it is computationally infeasible to compute such an estimator as it requires computing and combining univariate projections along infinite many directions. This has been shown to be an NP-Hard problem.

Another possible generalization of is the *median-of-means* framework that is commonly used in heavy-tail statistics literature. In one-dimension, this method can be described as: given n i.i.d samples from distribution X, randomly group them into $[n/k]$ groups, compute the mean within each group and then output the median of the k grouped means. In one-dimension, this estimator is sample efficient and requires minimal assumptions on the underlying distribution X, i.e. only bounded variance. Cherapanamjeri, Hopkins et al. apply this idea with sum-of-squares proofs to obtain a sample-efficient algorithm for high-dimensional estimation problems whose underlying distribution is heavy-tailed [CHK+19]. This high-dimensional generalization uses the following definition of median: for points $X_1, ..., X_k \in \mathbb{R}^d$ and $r > 0$, $x \in \mathbb{R}^d$ is an r-median if for every uni-direction u, we have $| \langle X_i, u \rangle - \langle x, u \rangle | \leq r$ for at least $1/2 + \epsilon$ fraction of $X_1, ..., X_n$ for some small $\epsilon > 0$. Based on our searches, we have found any results involving this estimator in robust statistics. It is unclear to us how this estimator performs for robust mean estimation. This is an potential open problem.

Below, we state a computationally inefficient algorithm that gives the optimal guarantee based on the idea of Tukey median.

Proposition 3.2. *There exists an algorithm that, on input an ϵ-corrupted set of samples from $X = \mathcal{N}(\mu_X, I)$ of size $n = \Omega((d + \log(1/\tau))/\epsilon^2)$ running in poly$(n, 2^d)$ time, and outputs $\hat{\mu} \in \mathbb{R}^d$ such that with probability at least $1 - \epsilon$, it holds that $\|\hat{\mu} - \mu_X\|_2 = O(\epsilon)$.*

Note that this algorithm is not feasible in practice as it runs in poly$(n, 2^d)$. The algorithm to establish this proposition proceeds by using a one-dimensional robust estimator to estimate $\nu \cdot \mu$ for a set of $2^{O(d)}$ unit vectors $\nu \in \mathbb{R}^d$ and then combine these estimates (by solving a large linear program) to obtain an accurate estimate of μ.

We have seen that natural generalizations of one-dimensional robust estimators suffer either from an Euclidean error that scales with $\sqrt{d}$ or computationally inefficiency. Some other methods are needed to devise computationally efficient algorithms for high-dimensional problems.

3.2 Filtering

In this section, we go over the Filtering approach for devising computationally efficient algorithms for robust statistics. For simplicity, we will be mainly focusing on the problem of robust mean estimation of Gaussian families.

3.2.1 Key insight

One challenge in high dimension is that it is difficult to identify corrupted points from the uncorrupted ones. As illustrated in Figure 1.1, for a centered isotropic Gaussian, the uncorrupted points are concentrated around the sphere $S_{d-1}(\sqrt{d}) = \{x \in \mathbb{R}^d | \ \|x\|_2 = \sqrt{d}\}$. Hence, the adversary can points at a distance of $\sqrt{d}$ from the mean without being detected while shifting the mean by $\eta\sqrt{d}$ in random directions. Our task seems hopeless since it is information-theoretically impossible to detect remove these points. However, *do we really need to identify and remove all corrupted points?*

The answer is no. As stated in [DK19], we only need to identify and remove outliers that are "consequential," i.e., *the ones that can significantly impact our estimates of the mean.*

Let's assume with out generality that there are no extreme outliers (as these can be removed via pre-processing). Now, if we have two outliers one at $\sqrt{d} \cdot u$ and one at $-\sqrt{d} \cdot u$ for some unit vector u, then the effect of these two outliers on mean estimation will be canceled. Hence, *the only way* *that the empirical mean can be far from the true mean is if there is a "conspiracy" of many outliers, all producing errors in approximately the same direction.* Consequently, our task of detecting all outliers can be reduced to detecting such *consequential outliers.*

In order to convert the aforementioned insight into a computationally efficient algorithm, we need the following observation. Let T be an η-corrupted set of points drawn from $\mathcal{N}(\mu, I)$. Then based on the observation above, the consequential outliers that significantly move the empirical mean $\hat{\mu}$ must move it in some direction. Formally speaking, there exists some unit vector v such that the projection $v \cdot (\mu - \hat{\mu})$ is large in magnitude. In particular, if an η-fraction of corrupted points in T move the sample average of $v \cdot (U_T - \mu)$ where U_T is the uniform distribution on T by more than δ (δ should be thought of as small, but substantially larger than η), then on average these corrupted points x must have $v \cdot (x - \mu)$ at least δ/η [DK19]. In this case, these corrupted points will have a contribution of at least $\epsilon \cdot (\delta/\epsilon)^2 = \delta^2/\epsilon$ to the variance of $v \cdot U_T$. This observation allows us to devise a computationally efficient algorithm. Specifically, by computing the top eigenvector of the sample covariance matrix, we will be able to know if there exists a unit vector v such that the variance of $v \cdot U_T$ is particularly large.

This allows us to obtain a general framework of an efficient algorithm. Starting with an η-corrupted sample T of distribution X such that $\mathbf{E}[X] = \mu$ and $\mathbf{Cov}(X) = \Sigma$, under the strong contamination model. Let Σ_T be the sample covariance matrix. We then proceed as:

1. Find the eigenvector v^* with the largest eigenvalue λ^*.
2. Compare λ^* with λ the value it should be (in the absence of outliers)
    - If $\lambda^* \approx \lambda$, then there are not any consequential outliers and the empirical mean is close to the true mean. In this case, we return the sample mean as our output.
    - If $\lambda^* >> \lambda$, we have obtained a particular direction v^* along which the projections of the outliers behave significantly differently from the inliers. In this case, we compute a *score* for each points and remove points with high scores and go through the procedure again.

In the last case, we perform some outlier-removal procedure based on some score function along the direction v^*. This removal procedure and the design of the function is rather subtle and relies heavily on our distributional assumptions of the data. Notice that this general framework does not only work for robust mean estimation. For problems such as robust covariance estimation and robust linear regression, a similar procedure can be obtained for the purpose of outlier removal. The difference is that in these problems, we often consider some variants of the sample covariance matrix and might have a more sophisticated outlier removal procedure. See [DKK$^+$16, And08].

In order for the aforementioned framework to work as desired, we also need the inliers to behave reasonably well. Since otherwise, after the removal procedure we will end up with either a small set of samples or a set of samples that behave essentially like the outliers. To this end, we need some conditions on the "good" set of points.

3.3 Stability

Let S be a set of n i.i.d. samples from X. We say that these sample points are "good." Given S, an adversary then inspect all the good points and select up to an η-fraction of points in S and replace them with arbitrary points to obtain an η-corrupted sample T which is given as an input to the algorithm. Note that $|S \cap T|/|S| \geq 1 - \eta$. As suggested above, in order to establish the correctness of the algorithm, we need to show that *with high probability over the choice of the good points S, the algorithm is able to output an accurate estimate of the true mean regardless of how the adversary made those corruptions.*

To this end, we need to impose a *stability* condition on the good set. Specifically, we require the uniform distribution over the good points to be similar to the true distribution in terms of moments and potentially tail bounds. We also want to make sure these similarities carry on with high probability in any subset of S of size $\geq (1-\eta)n$. In particular, we have the following definition.

Definition 3.3. *(Stability) Fix some $0 < \epsilon < 1/2$ and $\delta \geq \epsilon$. We say that a finite set $S \subseteq \mathbb{R}^d$ is $(\epsilon, \delta)-$stable (with respect to a distribution X) if for every unit vector $v \in \mathbb{R}^d$ and every $S' \subseteq S$ with $|S'| \geq (1 - \epsilon)|S|$, the following conditions hold:*

1. $\left| \frac{1}{|S'|} \sum_{x \in S}^{'} v \cdot (x - \mu) \right| \leq \delta$

2. $\left|\frac{1}{|S'|}\sum_{x\in S}^{\prime}(v\cdot(x-\mu))^2-1\right|\leq\delta^2/\epsilon$.

In words, the definition formalizes our intuition that a small fraction of points ($\leq\eta n$) cannot change the mean and variance along any direction v by a lot. This stability condition on the good set is necessary and used in every known robust mean estimation algorithm. The condition depends heavily on the underlying distribution X. In order for the algorithm to be meaningful, we need this stability condition for a large class of distributions. Fortunately, this is true as stated in the following proposition.

Proposition 3.4. *A set of i.i.d. samples from an identity covariance sub-gaussian distribution of size $\Omega(d/\epsilon^2)$ is $(\epsilon, O(\epsilon\sqrt{\log(1/\epsilon)}))$-stable with high probability.*

For a proof sketch of the proposition, see [DK19]. In fact, we do not need the sub-gaussian assumption but instead requires only a boundedness assumption on the covariance matrix in order for stability to hold. Specifically,

Proposition 3.5. *A set of i.i.d. samples from a distribution with covariance $\Sigma\preceq I$ of size $\tilde{\Omega}(d/\epsilon)$ is $(\epsilon, O(\sqrt{\epsilon}))$-stable with high probability.*

One useful fact is that analogous bound can be proved for distributions with identity covariance and bounded higher central moments. For instance, if distribution X has identity covariance and its k-th central moment is boounded from above by a constant, one can show that a set of $\Omega(d/\epsilon^{2-2/k})$ samples is $(\epsilon, O(\epsilon^{1-1/k}))$-stable with high probability.

It remains show that this stability condition suffices for our purpose. Specifically, we want to show that any η-corruption T of a stable set S of samples and bounded sample covariance has the following guarantee: the sample mean of T is a good approximation of the true mean. We have the following lemma.

Lemma 3.6. *(Certificate for empirical mean) Let S be an (ϵ,δ)-stable set with respect to a distribution X, for some $\delta\geq\epsilon>0$. Let T be an ϵ-corrupted version of S. Let μ_T and Σ_T be the empirical mean and covariance of T. If the largest eigenvalue of Σ_T is at most $1+\lambda$, for some $\lambda\geq 0$, then $\|\mu_T-\mu_X\|_2\leq O(\delta+\sqrt{\epsilon\lambda})$.*

Proof. Let $S'=S\cap T$ and $T'=T\backslash S'$, i.e. S' is the set of samples that remain uncorrupted after an adversarial corruption and T' is the set of samples set have been corrupted by the adversary. Without loss of generality, we may assume that $|S'|=(1-\epsilon)|S|$ and $|T'|=\epsilon|S|$ (if this is not satisfied, we can replace S' with a subset of it if necessary). Use $\mu_{S'},\mu_{T'},\Sigma_{S'},\Sigma_{T'}$ to denote the empirical mean and sample covariance matrices of S' and T' respectively. It is easy to show that the following relation holds:

$$\Sigma_T=(1-\epsilon)\Sigma_{S'}+\epsilon\Sigma_{T'}+\epsilon(1-\epsilon)(\mu_{S'}-\mu_{T'})(\mu_{S'}-\mu_{T'})^T.$$

Let λ_T be the maximum eigenvalue of Σ_T and v be the unit vector in the direction of $\mu_{S'}-\mu_{T'}$. By the functional definition of maximum eigenvalue, we have,

$$\begin{aligned}1+\lambda_T\geq\max_{u\in S^{d-1}}u^T\Sigma_T u\geq v^T\Sigma_T v&\\ &=(1-\epsilon)v^T\Sigma_{S'}v+\epsilon v^T\Sigma_{T'}v+\epsilon(1-\epsilon)v^T(\mu_{S'}-\mu_{T'})(\mu_{S'}-\mu_{T'})^Tv\\ &\geq(1-\epsilon)(1-\delta^2/\epsilon)+\epsilon(1-\epsilon)\|\mu_{S'}-\mu_{T'}\|_2^2\\ &\geq 1-O(\delta^2/\epsilon)+(\epsilon/2)\|\mu_{S'}-\mu_{T'}\|_2^2,\end{aligned}$$

were the second last inequality comes from the fact that $\Sigma_{T'}$ is positive semi-definite and the second stability condition of the set S'. By rearranging, we obtain that $\|\mu_{S'}-\mu_{T'}\|_2=O(\delta/\epsilon+\sqrt{\lambda/\epsilon})$. Using the fact that $\mu_T=(1-\epsilon)\mu_{S'}+\epsilon\mu_{T'}$, we have

$$\begin{aligned}\|\mu_T-\mu_X\|_2&=\|(1-\epsilon)\mu_{S'}+\epsilon\mu_{T'}\|_2\\ &=\|\mu_{S'}-\mu_X+\epsilon(\mu_{T'}-\mu_{S'})\|_2\\ &\leq\|\mu_{S'}-\mu_X\|_2+\|\epsilon(\mu_{T'}-\mu_{S'})\|_2\\ &=O(\delta)+\epsilon\cdot O(\delta/\epsilon+\sqrt{\lambda/\epsilon})\\ &=O(\delta+\sqrt{\lambda\epsilon})\end{aligned}$$

where the inequality comes from triangle inequality, the first stability condition and the bound on $\|\mu_{S'}-\mu_{T'}\|_2$ we obtained above. □

This is a nice result as it tells us that under the stability condition , given such a set T with bounded covariance, we can use the sample mean of T as an good approximation of the true mean μ_X. However, we are not always guaranteed the set T has this nice property. To deal with this, we will need a generalization of the above lemma.

Lemma 3.7. *Let S be an (ϵ, δ)-stable set with respect to a distribution X, for some $\delta \geq \epsilon > 0$ with $|S| > 1/\epsilon$. Let W be a probability distribution that differs from U_S, the uniform distribution over S, by at most ϵ in total variation distance. Let μ_W and Σ_W be the mean and covariance of W. If the largest eigenvalue of Σ_W is at most $1 + \lambda$ for some $\lambda \geq 0$, then $\|\mu_W - \mu_X\|_2 \leq O(\delta + \sqrt{\epsilon\lambda})$.*

By letting W be the uniform distribution over T we obtain lemma 3.6. With this generalized lemma, we can now clarify our goal of our outlier removal procedure. Given an ϵ-corrupted sample T, we want to find a distribution W over T such that Σ_W has no large eigenvalues. Moreover, we need to ensure that this distribution W is close in TV-distance to the uniform distribution μ_S. In this case, $W(x)$ quantifies the strength of our belief that $x \in T$ is an inlier. Now our goal becomes relatively clear: we want to efficiently find such a W. To formalize this as an optimization problem, we need the following definition of the space we are optimizing over.

Definition 3.8. *Let S be a $(3\epsilon, \delta)$-stable set with respect to X and let T be an ϵ-corrupted version of S. We define $C_\epsilon(T)$ to be the set of distributions W over T such that $W(x) \leq 1/(|T|(1-\epsilon))$ for all $x \in T$, i.e.,*

$$C_\epsilon(T) = \Big\{W \in \mathcal{P}(T) |\ W(x) \leq \frac{1}{|T|(1-\epsilon)}\ \forall x \in T\Big\}.$$

When the context is clear, we will use that notation C instead of $C_\epsilon(T)$.

Notice that for any distribution W in C, we have $d_{TV}(W, U_S) \leq 3\epsilon$. To see this, note that

$$\begin{aligned} d_{TV}(W, U_S) &= \sum_{x \in T} \max\{W(x) - U_S(x), 0\} \\ &= \sum_{x \in S \cap T} \max\{W(x) - 1/|T|, 0\} + \sum_{x \in T \setminus S} W(x) \\ &\leq \sum_{x \in S \cap T} \frac{\epsilon}{|T|(1-\epsilon)} + \sum_{x \in T \setminus S} \frac{1}{|T|(1-\epsilon)} \\ &\leq |T|\Big(\frac{\epsilon}{|T|(1-\epsilon)}\Big) + \epsilon|T|\Big(\frac{1}{|T|(1-\epsilon)}\Big) \\ &= \frac{2\epsilon}{1-\epsilon} \leq 3\epsilon \end{aligned}$$

where the last inequality holds for $\epsilon \leq 1/3$. This suggests that if we can find a $W \in C_\epsilon(T)$ such that Σ_W has no large eigenvalues, then μ_W is a good approximation of μ_X by the lemma above. On the other hand, we know this constrained optimization problem does indeed have a solution: simply let W be the uniform distribution over $S \cap T$. In this case, by the stability condition, we have that its largest eigenvalue is at most $1 + \delta^2/\epsilon$ and we will then obtain an l_2 error of $O(\delta)$.

By the discussions above, we now know that it suffices to *find any distribution $W \in C$ such that Σ_W has no large eigenvalues.* To efficiently solve this problem, there are two basic algorithmic techniques that can be applied.

1. *The Unknown Convex Programming Method.* This algorithm is based on the observation that C is a convex set and finding a point in C with bounded covariance is *almost* a convex program. It is not exactly a convex program because given any fixed $v \in \mathbb{R}^d$, the variance of $v \cdot W$ is not a convex function of W where $v \cdot W$ denote distribution of W projected onto the direction of v. However, if W has variance significantly larger than $1 + \delta^2/\epsilon$ in some direction, there exists an efficient algorithm that construct a hyperplane separating W from $U_{S \cap T}$, i.e., the uniform distribution over $S \cap T$. This method is a direct application of the discussions we have had so far. But as it relies heavily on the ellipsoid algorithm, it is slow (although polynomial time). We will not go over details of this method in this paper. Interested readers can refer to [DK19] for a detailed discussion of this method.

2. *Filtering.* This method is at the heart of our later discussions. *Filtering* is an iterative outlier removal method that is typically a lot faster as it relies primarily on spectral techniques. Specifically, based on our discussion above, if Σ_W does not have a large eigenvalue, then μ_W

is close to μ_X and we can simply output μ_W as our estimate. On the other hand, if Σ_W has an extremely large eigenvalue, then this suggests that there exists some direction v such that $\mathbf{Var}[v \cdot W]$ is substantially larger than what it should be (when the samples are not corrupted). This can happen if and only if W assigns high probability mass to points x in $T \backslash S$ such that the projection $v \cdot x$ is far from the true mean $v \cdot \mu$. As a result, we can perform a outlier removal procedure by removing points that have extreme values of $v \cdot x$.

One thing to notice is that it is difficult to ensure that only outliers are removed. However, with some careful design, one can ensure that more outliers are removed than inliers which guarantee enough *good* samples that will be used to calculate our final output. On a high-level, the *filtering* algorithm framework is described as: given a W such that Σ_W has large eigenvalues, one filtering step gives a new distribution $W' \in C$ that is closer to $U_{S \cap T}$ than W was. Repeat this process until the current W has no large eigenvalues. We then output μ_W as our final estimate. In the next few sections, we will discuss various concrete realizations of the filtering algorithm. In particular, while a filtering step removes points that are far from the projected mean along some direction v, there are several different ways to quantify this notion of closeness which might have different guarantees depending on the true underlying distribution.

3.4 Filtering

In this section, we introduce some concrete realizations of the filtering framework. Specifically, we will be going over *basic filtering, randomized filtering and universal filtering* which builds on top of each other. Although in the discussion above we let W be any general distributions over T, in most cases, it suffices to consider only the uniform distribution over some set of points. Hence, the common goal of all of the variants of the filtering algorithms below is to remove outliers in the set T so that we are making progress towards $U_{S \cap T}$ which is alternatively denoted as W^*.

3.4.1 Basic Filtering

We first present a filtering algorithm which yields optimal error bound for distributions with identity covariance (or more generally, known covariance) distributions whose univariate projections satisfy appropriate tail bounds. Diakonikolas and Kane refer to this algorithm as *basic filtering.* For simplicity, we will be discussing this algorithm in the context of Gaussian distributions. This algorithm can be easily extended to distributions with weaker concentrations. For example, one can show this algorithm yields the optimal error bound on sub-gaussian, sub-exponential and even inverse polynomial concentrations (most non-heavy-tailed distributions) with some proper adjustments.

As suggested above, in addition to stability condition, we also need a concentration condition defined below.

Definition 3.9. *A set $S \subset \mathbb{R}^d$ is tail-bound-good (with respect to $X = \mathcal{N}(\mu_X, I)$ if for any unit vector v, and any $t > 0$, we have*

$$P_{x \sim \mu_S}[|v \cdot (x - \mu_X)| > 2t + 2] \leq e^{-t^2/2}.$$

where μ_S is the uniform distribution over S.

One can show that this condition hold with high probability if S consists of i.i.d. random samples from X of a sufficiently large polynomial size. See [DKK+16] for a proof of this statement. The reason we need this extra condition on tail probability is that we want to ensure we remove more outliers than inliers after one filtering step. Formally, we have the following lemma.

Lemma 3.10. *Let $\epsilon > 0$ be a sufficiently small constant. Let $S \subset \mathbb{R}^d$ be both $(2\epsilon, \delta)$-stable and tail-bound-good with respect to $X = \mathcal{N}(\mu_X, I)$ with $\delta = c\epsilon\sqrt{\log(1/\epsilon)}$ for $c > 0$ a sufficiently large constant. Let $T \subset \mathbb{R}^d$ be such that $|T \cap S| \geq (1 - \epsilon)\max(|T|, |S|)$ and assume we are given a unit vector $v \in \mathbb{R}^d$ such that $\mathbf{Var}[v \cdot T] > 1 + 2\delta^2/\epsilon$. Then there exists a polynomial time algorithm that returns a set $R \subset T$ satisfying $|R \cap S| < |R|/3$.*

Notice that this lemma suffices for our purpose. Replacing T with $T' = T \backslash R$, we obtain a less noise sample T'. More importantly, the inliners we removed by setting $T' = T \backslash R$ is less than the number of outliers as $|R \cap S| < |R|/3$. Let $A \Delta B$ denote the symmetric difference between A and B. Then it is easy to see that $|S \Delta T'| < |S \Delta T|$. This make sure that the condition $|T \cap S| \geq$

$(1-\epsilon)\max(|T|,|S|)$ is still satisfied by replacing T with T' and thus we can continue to apply this algorithm until $T = T'$ or when we are left with a set of small variance.

The high-level idea of the proof is to create the set R by removing points based on the tail bound. There are some technical but non-educational details in the proof. We will omit the proof here. Interested readers can refer to the proof of Lemma 2.11 in [DK19]. One thing to notice is that the algorithm is purely deterministic as it remove points based on violations of the tail-bound condition satisfied by the inliers. The complete filtering algorithm does the following:

1. Compute $Cov[T]$ and its largest eigenvalue ν.
2. If $\nu \leq 1+\lambda$, output μ_T.
3. If $\nu > 1+\lambda$, remove points based on the *basic filtering* procedure and repeat.

However, this deterministic algorithm fails in certain regimes. As a concrete example, it fails if the only assumption we impose on the true distribution is boundedness of the covariance matrix. To overcome this problem, the power of randomness is needed.

3.4.2 Randomized Filtering

In *randomized filtering*, we remove points based a probability proportional to the non-negative function $f(x)$. In particular, suppose there exists some non-negative function f defined on $S \cup T$ such that $\sum_{x\in T} f(x) \geq 2\sum_{x\in S} f(x)$ where S is the set of uncorrupted points and T is an ϵ-corruption of S. In this case, we can devise a randomized filtering scheme by simply removing points randomly based on the probability that is proportional to $f(x)$ over $S \cup T$. By the property of f, we are guaranteed, in expectation, to remove more outlier than inliers.

Let R_t be the set of points that were removed in the t-th iteration of the randomized filtering algorithm. One key ingredient in the analysis of this algorithm is the quantity $E_t = |R_t \cap S| - |R_t \cap (T\backslash S)|$, i.e. the difference between the number of removed inliners and the number of removed outliers. This algorithm satisfies the desirable property that during the process of this randomized filtering algorithm, the sequence (E_t) is a sub-martingale. Using this, one can show that this algorithm satisfies the desirable properties. Specifically, we have the following theorem.

Theorem 3.11. *Let $S \subset \mathbb{R}^d$ be a $(6\delta,\epsilon)$-stable set (with respect to some distribution X with bounded covariance) and T be an ϵ-corrupted version of S. Suppose that given any $T' \subseteq T$ with $|T' \cap S| \geq (1-6\epsilon)|S|$ for which $\mathbf{Cov}[T']$ has an eigenvalue bigger than $1+\lambda$, for some $\lambda > 0$, there is an efficient algorithm that computes a non-zero function $f : T' \to \mathbb{R}_+$ such that $\sum_{x\in T'} f(x) \geq 2\sum_{x\in T'\cap S} f(x)$. Then there exists a polynomial time randomized algorithm that computes a vector $\hat{\mu}$ that with probability at least $2/3$, it holds that $\|\hat{\mu}-\mu_X\|_2 = O(\delta\sqrt{\epsilon\lambda})$.*

Notice that here we only requires the underlying distribution X to have bounded covariance. The algorithm is given below.

Algorithm 1 Randomized filtering algorithm

```
1. Compute $\mathbf{Cov}[T]$ and its largest eigenvalue $\nu$.
2. If $\nu \leq 1+\lambda$, output $\mu_T$.
3. Else
    • Compute $f$ as guaranteed in the theorem statement.
    • Remove each $x \in T$ with probability $f(x)/\max_{x\in T} f(x)$ and return to Step 1 with the
      new set $T$.
```

We first notice that during each iteration of the algorithm, at least one points is removed as the point $\arg\max_{x\in T} f(x)$ is removed with probability one. Therefore, the runtime of the algorithm is at most $|T|$. For correctness, we will show the following claim.

Claim 3.12. *At each iteration of the algorithm, with probability at least $2/3$, we have that*

$$|S \cap T| \geq (1-6\epsilon)|S|.$$

Assuming this claim, Lemma 3.6 implies our desired final. bound. It remains to prove the above claim.

Proof. (of Claim) Let's consider the sequence of random variables $d(T) = |S\Delta T| = |S\backslash T| + |T\backslash S|$ across iterations of the algorithm. Notice that initially we have $d(T) = 2\epsilon|S|$ and it is lower bounded by 0. At each stage of the algorithm, it is decremented by E_t where E_t is (#inliers removed - #outliers removed). Moreover, we have that

$$\mathbf{E}[E_t] = \sum_{x\in S\backslash T} - \sum_{x\in T\backslash S} f(x) = 2\sum_{x\in S\cap T} - \sum_{x\in T} f(x) \leq 0.$$

This suggests that $(d(t))$ is a sub-martingale until we reach a point where $|S \cap T| \leq (1-6\epsilon)|S|$. However, if we set a stopping time at the first occasion where this condition fails, we note that the expectation of $d(T)$ is at most $2\epsilon|S|$. Since it is also non-negative, we can then apply Markov's inequality and deduce that

$$P(d(T) \geq 2\epsilon|S|) \leq \frac{\mathbf{E}[d(T)]}{2\epsilon|S|} \leq \frac{1}{3}.$$

In other words, with probability at least 2/3, it is never more than $6\epsilon|S|$. Therefore, we know that $|S \cap T| \geq (1-6\epsilon)|S|$ throughout the algorithm. The inequality $|T' \cap S| \geq (1-6\epsilon)|S|$ will continue to hold throughout our algorithm which will eventually yielding such a set with the variance of T' bounded. Then by Lemma 3.6, we can output $\mu_{T'}$ and this will be a good estimate of the true mean μ_X. □

After we have gone through the analysis of the algorithm, there is still one aspect of the algorithm that is uncertain. We only specified that each point $x \in T$ is removed with probability $f(x)/\max_{x\in T} f(x)$. But we have not specified whether each point is removed independently or not. In fact, there are several different methods of this point removal procedure which have different practical implications despite having similar theoretical guarantees. Here we give some natural ways of doing point removal.

- *Randomized Thresholding:* A concrete implementation of the outlier removal procedure does the following: it generates a number y uniformly randomly from the interval $y \in [0, \max_{x\in T} f(x)]$ and then remove all points $x \in T$ such that $f(x) \geq y$. This approach is the practical and the easiest to apply in many settings. After generating y, we can just iterate through all the points and compare with this threshold value. Notice that in this case, the removal of point x_i is not independent of the removal of point x_j for $i \neq j$ as their respectively probabilities of being removed depend on this common threshold y.

- *Independent Removal:* Each point $x \in T$ is removed independently with probability $f(x)/\max_{x\in T} f(x)$. This scheme has the advantage of reducing the variance in $d(t)$. A careful analysis of this removal procedure involving random walk allows one to reduce the failure probability to $\exp(-\Omega(\epsilon|S|))$.

- *Deterministic Reweighting:* In this procedure, each point is assigned a weight in $[0, 1]$ and we will consider the weighted means and covariances. Instead of removing a point, we will decrement a fraction of the weight of a point x proportional to $f(x)$. In some sense, this is similar to the multiplicative weights algorithms in Theoretical Computer Science. This reweighting scheme ensures that the appropriate weighted version of $d(T)$ is non-increasing which implies the correctness of the algorithm.

While these point-removal schemes have similar theoretical guarantees, they behave rather differently in practice. When dealing with real datasets, deterministic reweighting is generally much slower than the other two methods. In summary, we would prefer randomized methods to non-random ones for point-removal in practice. For a in-depth discussion of this topic and some practical results, see [DK19, DKK$^+$19].

3.4.3 Universal Filtering

Lastly, in this subsection, we show how we can use randomized filtering to obtain an universal filters that work requiring only the stability condition. We will work towards a proof of the following theorem.

Theorem 3.13. *Let S be a $(3\epsilon, \delta)$-stable set with respect to a distribution X and let Tbe an ϵ-corrupted version of S. Then there exists a polynomial time algorithm which given T returns $\hat{\mu}$ such that $\|\hat{\mu} - \mu_X\|_2 = O(\delta)$.*

To prove this theorem, we need an efficient algorithm that achieves the desired task. To this end, we will be using the following proposition.

Proposition 3.14. *Let $S \subset \mathbb{R}^d$ be an (ϵ, δ)-stable set for $\epsilon, \delta > 0$ sufficiently small constants and δ at least a sufficiently large multiple of ϵ. Let T be an ϵ-corrupted version of S. Suppose that $\mathbf{Cov}[T]$ has largest eigenvalue $1+\lambda > 1+8\delta^2/\epsilon$. Then there exists a computationally efficient algorithm that on input ϵ, δ, T, computes a non-zero function $f : T \to \mathbb{R}_+$ satisfying $\sum_{x\in T} f(x) \geq 2\sum_{x\in S\cap T} f(x)$.*

If we can obtain such a function f, then we can apply randomized filtering to obtain an sample T'. We can then output T' as our estimator. By Theorem 3.11, we can then conclude our proof of Theorem 3.13.

Proof. (of proposition) We want to construct a function f satisfying the desired properties. To construct such a function, our first step is to compute the sample mean μ_T and the top (unit) eigenvector v of $\mathbf{Cov}[T]$. For all $x \in T$, define the function g as $g(x) = (v \cdot (x - \mu_T))^2$. Let L be the set of $\epsilon \cdot |T|$ elements of T on which $g(x)$ is largest. We define f to be $f(x) = 0$ for $x \notin L$ and $f(x) = g(x)$ for $x \in L$. Notice that

$$\sum_{x\in T} g(x) = |T|\,\mathbf{Var}[v\cdot T] = |T|(1+\lambda).$$

Moreover, for any $S' \subseteq S$ with $|S'| \geq (1-2\epsilon)|S|$, we have that

$$\sum_{x\in S'} g(x) = |S'|(\mathbf{Var}[v\cdot S'] + (v\cdot(\mu_T - \mu_{S'}))^2). \tag{1}$$

By the second stability condition, we have that $|\,\mathbf{Var}[v\cdot S'] - 1| \leq \delta^2/\epsilon$. By the first stability condition and Lemma 3.6, we have

$$\|\mu_T - \mu_{S'}\|_2 \leq \|\mu_T - \mu_X\|_2 + \|\mu_X - \mu_{S'}\|_2 = O(\delta + \sqrt{\epsilon\lambda}).$$

On the other hand, we know that $\lambda \geq 8\delta^2/\epsilon$ by assumption. Together with the bounds above, this suggests that $\sum_{x\in T\setminus S} g(x) \geq (2/3)|S|\lambda$. Moreover, since $|L| \geq |T\setminus S|$ and g takes its largest values on points $x \in L$, we have that

$$\sum_{x\in T} f(x) = \sum_{x\in L} g(x) \geq \sum_{x\in T\setminus S} g(x) \geq (16/3)|S|\delta^2/\epsilon.$$

Applying this to equation 1 with $S' = S$ and $S' = S\setminus L$, we have

$$\begin{aligned}\sum_{x\in S\cap T} f(x) = \sum_{x\in S\cap L} g(x) &= \sum_{x\in S} g(x) - \sum_{x\in S\setminus L} g(x)\\ &= |S|(1 \pm \delta^2/\epsilon + O(\delta^2+\epsilon\lambda)) - |S\setminus L|(1\pm\delta^2/\epsilon + O(\delta^2+\epsilon\lambda))\\ &\leq 2|S|\delta^2/\epsilon + |S|O(\delta^2+\epsilon\lambda).\end{aligned}$$

where the latter quantity is at most $(1/2)\sum_{x\in T} f(x)$ when δ and ϵ/δ are sufficiently small constants. This completes the proof of the proposition. □

3.5 Robust Mean Estimation Algorithm

In this section, we present formal statements of the robust mean estimation algorithm in [DKK+17]. These statements are results of the filtering techniques we covered in this section. We will not be going over the proofs in this paper. See [DKK+17] for details of the proofs and some experimental results. We have the following theorems for robust mean estimation in the sub-gaussian setting and bounded covariance setting respectively.

Theorem 3.15. *Let G be a sub-gaussian distribution on $\mathbb{R}^d$ with parameter $\nu = \Theta(1)$, mean μ_G, covariance I, and $\epsilon > 0$. Let S be an ϵ-corrupted set of samples from G of size $\Omega((d/\epsilon^2)poly\log(d/\epsilon))$. There exists an efficient algorithm that, on input S and $\epsilon > 0$, returns a mean vector $\hat{\mu}$ so that with high probability at least $9/10$ we have $\|\hat{\mu} - \mu_G\|_2 = O(\epsilon\sqrt{\log(1/\epsilon)})$.*

Theorem 3.16. *Let P be a distribution on $\mathbb{R}^d$ with unknown mean vector μ_P and unknown covariance matrix $\Sigma_P \preceq \sigma^2 I$. Let S be an ϵ-corrupted set of samples from P of size $\Theta((d/\epsilon)\log d)$. There exists an efficient algorithm that, on input S and $\epsilon > 0$, with probability $9/10$ outputs $\hat{\mu}$ with $\|\hat{\mu} - \mu_P\|_2 \leq O(\sigma\sqrt{\epsilon})$.*

We can also apply filtering techniques to robustly estimate the covariance of a Gaussian distribution as stated in the following theorem.

Theorem 3.17. *Let $G \sim \mathcal{N}(0, \Sigma)$ be a Gaussian in d dimensions, and let $\epsilon > 0$. Let S be an ϵ-corrupted set of samples from G of size $\Omega((d^2/\epsilon^2)\mathit{poly}\log(d/\epsilon))$. There exists an efficient algorithm that, given S and ϵ, returns the parameters of a Gaussian distribution $G' \sim \mathcal{N}(0, \hat{\Sigma})$ so that with probability at least $9/10$, it holds that $\|I - \Sigma^{-1/2}\hat{\Sigma}\Sigma^{-1/2}\|_F = O(\epsilon \log(1/\epsilon))$.*

4 SEVER algorithm

Having introduced some filtering techniques, in this section, we will go over a recent highlight in the robust statistics community which exploits filtering algorithms. Specifically, we will introduce and analyze the SEVER algorithm proposed in [DKK$^+$19]. In contrast to the robust mean estimation algorithm we introduced in the last section, SEVER is a meta-algorithm that works for a variety of learning tasks. The way it works is that it takes in a *base learner* (e.g. least squares, stochastic gradient descent etc.) and outputs a version of the base learner that is robust to outliers. Moreover, it is an efficient algorithm that scales well with the dimension since it only requires computing the top singular vector of a certain $n \times d$ matrix where n is the number of samples and d is the dimension of the samples beyond running the base learner itself. SEVER algorithm has the following nice properties.

- **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.

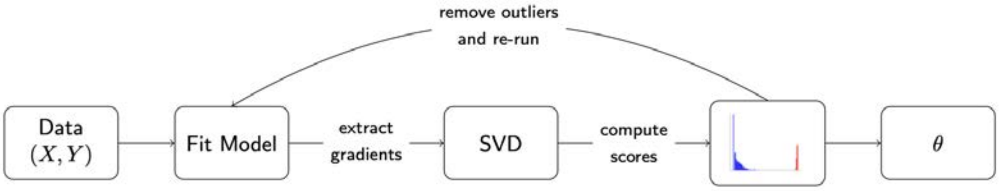

Figure 3: Outline of the SEVER algorithm. Adapted from Figure 1 of [DKK$^+$19].

On a high level, the algorithm is illustrated in the figure above (see Figure 4). Given data $(X, Y) \in \mathbb{R}^{n\times d} \times \mathbb{R}^n$, we first use our base learner to fit a model to the data. The fitted model has some fitted parameter which we denote as w. We then compute the gradients of the loss function on the fitted parameter w for all points $(x, y) \in (X, Y)$ and obtain an $n \times d$ centered gradient matrix. The next is to compute the top singular vector v and project the n centered gradients we obtained onto this direction v. We use the magnitude of the projections as our scores and then perform some point removal procedure based on this score and re-run the algorithm until some end condition has been reached.

Despite the simplicity of the algorithm, some details of the algorithm are quite subtle and we will treat them carefully in the following sections. More importantly, as long as the true underlying distribution is not too heavy-tailed, SEVER is provably robust to outliers and the algorithm works well on real datasets. In the following sections, we will go over some preliminaries needed for the algorithm. Then, we will present the algorithm and carefully analyze it. Lastly, we will analyze its performance for *generalized linear models* and specifically, logistic regression.

4.1 Preliminaries

As described in the high-level outline of the algorithm, the main objects we will be working with are loss functions and their gradients. For the sake of clarity, we will restate the statistical model and the noise model in the context of *data functions* instead of *data points*. The characterizations are completely equivalent and we will use the *data functions* representation simply because it is easier to work with in our setting.

We consider the following formalization of a learning problem. Suppose there is some true underlying distribution p^* over functions $f : \mathcal{H} \to \mathbb{R}$ where $\mathcal{H}$ is the space of possible parameters. Our goal is to find a parameter $w^* \in \mathcal{H}$ that minimizes the risk $\bar{f}(w) := \mathbf{E}_{f\sim p^*}[f(w)]$ given a training set of *data functions* $f_{1:n} = \{f_1, ..., f_n\}$. To better understand this formalization, let's consider a few examples.

For l_2 linear regression with no regularizations, the traditional *data points* formalization specifies that given training data points $(x_1, y_1), ..., (x_n, y_n)$, our goal is to output a linear function $w^* \in \mathcal{H}$

such that $\mathbf{E}_{(x,y)\sim D}[(1/2)(y - w \cdot x)^2]$ is minimized at w^* among all points in $\mathcal{H}$ where D is the true underlying distribution of (x, y). An equivalent formalization in the language of *data functions* is the following: given data functions $\{f_1, ..., f_n\}$ where $f_i(w) = \frac{1}{2}(w \cdot x_i - y_i)^2$, output a parameter $w^* \in \mathcal{H}$ such that $\mathbf{E}_{f\sim p^*}[f]$ is minimized at w^* among all points in $\mathcal{H}$. It is easy to see this two formalizations are exactly equivalent.

For different learning problems, we have different data functions $\{f_1, ..., f_n\}$. In the case of support vector machines (SVM) with hinge loss, we have $f(w) = \max\{0, 1 - y(w \cdot x)\}$. We can see how this formalization is easily to adaptable to various statistical problems as long as we know the loss function f. We will be using the standard l_2 linear regression model as a running example for the theoretical part of this section.

We also need a formal definition of the strong contamination model in the data function formalization.

Definition 4.1. *(ϵ-contamination model) Given $\epsilon > 0$ and a distribution p^* over functions $f : \mathcal{H} \to \mathbb{R}$, data is generated as follows: first we generate n clean (uncorrupted) samples $f_1, ..., f_n$ drawn from p^*. Then an adversary is allowed to inspect the samples and replace up to ϵn of the points with arbitrary samples. The resulting set of samples is then fed into the algorithm. We will call such a set of samples ϵ-corrupted (with respect to p^*).*

As suggested in Figure 4, we need a base learner which we denote as $\mathcal{L}$ for our algorithm. It takes in functions $f_1, ..., f_n$ and outputs a parameter $w \in \mathcal{H}$. In order for the algorithm to work as desired, we want w to be an approximate minimizer of the empirical loss function $(1/n)\sum_{i=1}^{n} f_i(w)$. To this end, we need to define some terms to capture this approximation notion formally.

Definition 4.2. *(γ-approximate critical point) Given a function $f : \mathcal{H} \to \mathbb{R}$, we say $w \in \mathcal{H}$ is a γ-approximate critical point of f such that for all unit vectors v where $w + \delta v \in \mathcal{H}$ for arbitrarily small positive δ, we have that $v \cdot \nabla f(w) \geq -\gamma$.*

This condition mandates that the value of f cannot be decreased much by changing the input w locally, while staying in the domain. When $\mathcal{H} = \mathbb{R}^d$, this notion of approximate critical points reduces to the notion of approximate stationary point (i.e, a point where the gradient is small in magnitude). With this notion in mind, we can now define the desired properties we want from our base learner.

Definition 4.3. *(γ-approximate learner) A learning algorithm $\mathcal{L}$ is called γ-approximate if, for any functions $f_1, ..., f_n : \mathcal{H} \to \mathbb{R}$ 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 $f(x) = (1/n)\sum_{i=1}^{n} f_i(x)$,*

In other words, $\mathcal{L}$ always finds an approximate critical point of the empirical learning objective. To see how this definition is appropriate, we notice that many common algorithms satisfy this property. For instance, for standard l_2 linear regression, gradient descent satisfies this property. Another possibly more straightforward approach is to set the gradient of the empirical objective to 0 and solve for w. While this is not a computationally efficient approach for many problems, the output of this method certainly satisfies the desired property.

4.2 Algorithm

Given a black-box base learner $\mathcal{L}$, the main component of the algorithm is the outlier removal procedure based on gradients at the output parameters. Similar to the high-level idea behind the filtering methods covered in the last section, we want to remove *consequential* outliers that has a large effect on the learned parameters. For this to be true, we want to identify outliers whose gradients are:

1. Systematically pointing in a specific direction.

2. Large in magnitude.

We can detect such points via Singular Value Decomposition (SVD). Specifically, if both 1 and 2 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. We now state our algorithm.

There are a couple of different methods when carrying out the final procedure in practice. We discussed some methods in the randomized filtering section. For in detail discussions of the performances of these methods in practice, see [DKK+19].

For concreteness, let's see how SEVER algorithm works for the problem of standard l_2 linear

Algorithm 2 SEVER algorithm

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

Algorithm 3 Filter(S, τ, σ)

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

regression. Suppose the base learner is a linear solver that solves the equation $x_i(y_i - w \cdot x_i)$, i.e. the equation we obtain by setting the gradient equal to 0, and outputs a solution $\hat{w}$. With this, we then compute the average gradient $\hat{\nabla} = (1/n) \sum_{i=1}^{n} x_i(\hat{w} \cdot x_i - y_i)$ with which we can compute the centered gradient matrix G whose jth row is given by $G_j = x_j(\hat{w} \cdot x_j - y_j) - \hat{\nabla}$. We then compute the top right singular vector of G which we denote as v and project the centered gradients onto this direction. Having done so, we will then obtain a score $\tau_j = (G_j \cdot v)^2$ for each data point. We can then feed those points into Algorithm 3 to randomly remove a fraction of points based on the scores and a pre-determined threshold *sigma*. We repeat this procedure until no points are being removed in the fiterling step.

With appropriate conditions on the underlying distribution over the function class, we obtain the following theoretical guarantee.

Theorem 4.4. *Suppose that functions $f_1, ..., f_n, \bar{f} : \mathcal{H} \to \mathbb{R}$ are bounded below on a closed domain $\mathcal{H}$. Suppose also that they satisfy the following deterministic regularity conditions: there exists a set $I_{good} \subseteq [n]$ with $|I_{good}| \geq (1 - \epsilon)n$ and $\sigma > 0$ such that*

1. $\mathbf{Cov}_{I_{good}}[\nabla f_i(w)] \preceq \sigma^2 I, \ \forall w \in \mathcal{H}$.
2. $\|\nabla \hat{f}(w) - \nabla \bar{f}(w)\|_2 \leq \sigma\sqrt{\epsilon}, \ \forall w \in \mathcal{H}$

where $\hat{f} := (1/|I_{good}|) \sum_{i \in I_{good}} f_i$ and $\bar{f}(w) = \mathbf{E}_{f \sim p^*}[f(w)]$. Then our algorithm SEVER applied to $f_1, ..., f_n$, σ returns a point $w \in \mathcal{H}$ such that, with probability at least $9/10$, is a $(\gamma + O(\sigma\sqrt{\epsilon}))$-approximate critical point of $\bar{f}$.

In words, the first condition states that there exists some good set of samples such that the covariance matrix of the gradients over this set is bounded above. The second condition states that the average gradient over this good set is not far from gradient under the true distribution. One important observation that the error does not depend on the dimension d. This ensures that we have our desired robustness even in high dimensions.

The nest step is to show that these desirable regularity conditions hold with high probability for some natural distributions.

Proposition 4.5. *Let $\mathcal{H} \subset \mathbb{R}^d$ be a closed bounded set with diameter at most r. Let p^* be a distribution over functions $f : \mathcal{H} \to \mathbb{R}$ and $\bar{f} = \mathbf{E}_{f\sim p^*}[f]$. Suppose that for each $w \in \mathcal{H}$ and unit vector v we have $\mathbf{E}_{f\sim p^*}[(v \cdot (\nabla f(w) - \nabla \bar{f}(w)))^2] \leq \sigma^2$. Under appropriate Lipschitz and smoothness assumptions, for $n = \Omega(d \log(r/(\sigma^2\epsilon))/(\sigma^2\epsilon))$, an ϵ-corrupted set of functions drawn i.i.d from $p^*, f_1, ..., f_n$ with high probability satisfy the regularity conditions.*

One nice thing about the algorithm is that it does not require any regularity conditions on the functions $f_1, ..., f_n$. As stated in Theorem 4.4, we are guaranteed to find a approximate critical point. Can we hope to better for convex functions? The answer is yes, we can find a approximate global minimum.

Corollary 4.6. *Suppose $f_1, ..., f_n : \mathcal{H} \to \mathbb{R}$ satisfy the regularity conditions stated in Theorem 4.4 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:*

1. *If $\bar{f}$ is convex, the algorithm finds a $w \in \mathcal{H}$ such that $\bar{f}(w) - \bar{f}(w^*) = O((\sigma\sqrt{\epsilon} + \gamma)r)$.*
2. *If $\bar{f}$ is ξ-strongly convex, the algorithm finds a $w \in \mathcal{H}$ such that $\bar{f}(w) - \bar{f}(w^*) = O((\epsilon\sigma^2 + \gamma^2)/\xi)$.*

4.3 Main idea

Before jumping directly to the proof of Theorem 4.4, we give an overview of SEVER and compare it with a projective gradient method which achieves the same theoretical guarantees. At the heart, SEVER is a gradient based algorithm with better runtime at the expense of a stronger assumption. We start with a high-level description of the problem and examine possible general strategies.

For simplicity, let's assume the true distribution p^* is supported over a convex set of convex functions. Specifically, $f_1, ..., f_n : \mathcal{H} \to \mathbb{R}$ and $\bar{f} = \mathbf{E}_{f\sim p^*}[f]$ are convex. Let $\hat{f} = (1/n)\sum_{i=1}^n f_i$ (also convex). Suppose all the data points are faithful and uncorrupted, we can then apply our favorite convex optimization algorithm to the objective function $\hat{f}$. We can thus obtain a approximate minimizer of $\hat{f}$, which by large sample theory, is also an approximate minimizer of $\hat{f}$.

However, this strategy does not exactly apply in our case as our data is ϵ-corrupted. The main reason is that a *single* adversarially corrupted sample can substantially changes the location of the minimum for $\hat{f}$. Consequently, a good approximation for the minimum of $\hat{f}$ is no longer a good approximation of $\bar{f}$. To overcome this issue, we would like to have an algorithm that can approximate a minimizer of $\bar{f}$ without necessarily giving a minimizer of $\hat{f}$ in the first place. One possible solution to resolve this is the (projective) gradient descent method.

4.3.1 Projective Gradient Descent method

Suppose we are given a set of uncorrupted sample $f_1, ..., f_n : \mathcal{H} \to \mathbb{R}$. Let $\bar{f} = \mathbf{E}_{f\sim p^*}[f]$ and $\hat{f} = (1/n)\sum_{i=1}^n f_i$. Then in order to recover the optimal $w^* \in \mathcal{H}$, one popular algorithm we can employ is projective gradient descent. In each iteration, we simply calculate the gradient $(1/n)\sum_{i=1}^n \nabla f_i(w)$ and update w using the projective gradient update:

$$w \leftarrow \mathcal{P}_{\mathcal{H}}\left(w - \eta\left(\frac{1}{n}\sum_{i=1}^n \nabla f_i(w)\right)\right)$$

where $\eta > 0$ is the step-size and $\mathcal{P}_{\mathcal{H}} : \mathbb{R}^d \to \mathcal{H}$ is the projection onto $\mathcal{H}$ under some distance metric. We then repeat this process until we are not making any progress. Then by some standard argument of projective gradient descent, we obtain an output that is a good approximation of a critical point of $\hat{f}$.

We can use this approach to devise an algorithm for ϵ-corrupted samples. If we can ensure in each iteration, the *gradient* we use in the update procedure is close to the gradient of $\bar{f}$, then the output of the projective gradient descent method is a good approximation of the minimizer of $\bar{f}$. The question now becomes: how can we find such good approximations of $\nabla \bar{f}(w)$?

To this end, the key observation is that approximating the gradient of $\bar{f}$ at a given point, given access to an ϵ-corrupted set of samples, can be reduced to a robust mean estimation problem. Then, we can use any robust mean estimation algorithm as a black-box, output a good approximation of

$\nabla \bar{f}(w)$ under some mild assumptions. Assuming that the covariance matrix of $\nabla f(w)$ for $f \sim p^*$ is bounded, we can then perform gradient descent and compute an approximate minimum for $\bar{f}$.

We will state without proof some key results of this gradient descent algorithm. Compared with SEVER, this algorithm is slower but it also requires a weaker assumption. Specifically, this algorithm only requires that for each $w \in \mathcal{H}$ there exists a set of good functions rather than the existence of a single good set that works simultaneously for all w.

Assumption 4.7. *Fix $0 < \epsilon < 1/2$ and parameter $\sigma \in \mathbb{R}_+$. For each $w \in \mathcal{H}$, there exists an unknown set $I_{good} \subseteq [n]$ with $|I_{good}| \geq (1-\epsilon)n$ "good" functions $\{f_i\}_{i \in I_{good}}$ such that*

$$\left\| \underset{I_{good}}{\mathbf{E}}[(\nabla f_i(w) - \nabla \bar{f}(w))(\nabla f_i(w) - \nabla \bar{f}(w))^T] \right\|_2 \leq \sigma^2$$

and

$$\|\nabla \hat{f}(w) - \nabla \bar{f}(w)\|_2 \leq \sigma\sqrt{\epsilon}$$

where $\hat{f} = (1/|I_{good}|) \sum_{i \in I_{good}} f_i$.

We will be using the result on robust mean estimation in [DKK+17, SCV17]. Specifically, we have the following theorem.

Theorem 4.8. *Let $\mu \in \mathbb{R}^d$ and a collection of points $x_i \in \mathbb{R}^d, i \in [n]$ and $\sigma > 0$. Suppose that there exists $I_{good} \subseteq [n]$ with $|I_{good}| \geq (1-\epsilon)n$ satisfying the following:*

1. $\frac{1}{|I_{good}|} \sum\limits_{i \in I_{good}} (x_i - \mu)(x_i - \mu)^T \preceq \sigma^2 I.$

2. $\left\| \frac{1}{|I_{good}|} \sum\limits_{i \in I_{good}} (x_i - \mu) \right\|_2 \leq \sigma\sqrt{\epsilon}.$

Then if $\epsilon < \epsilon_0$ for some universal constant ϵ_0, there is an efficient algorithm $\mathcal{A}$ which outputs an estimate $\hat{\mu} \in \mathbb{R}^d$ such that $\|\hat{\mu} - \mu\|_2 = O(\sigma\sqrt{\epsilon})$.

With this algorithm $\mathcal{A}$ as a black-box, we have the following theorem.

Theorem 4.9. *For functions $f_1, ..., f_n : \mathcal{H} \to \mathbb{R}$, bounded below on a closed domain $\mathcal{H}$, suppose Assumption 4.7 is satisfied with parameters $\epsilon, \sigma > 0$. Then there exists an efficient algorithm that finds an $O(\sigma\sqrt{\epsilon})$-approximate critical point of $\bar{f}$.*

Proof. (of Theorem 4.9) Applying Algorithm $\mathcal{A}$ to $\{\nabla f_i(w)\}$, we can find an approximation to $\nabla \bar{f}(w)$ with error $O(\sigma\sqrt{\epsilon})$. By standard results in optimization theory, we know that projective gradient algorithm can run efficiently even with approximate gradients. This can then be used to find our estimate with the desired error bound. □

In summary, the projective gradient descent approach first robustly estimates the projected gradient $\nabla \bar{f}(w)$ using an robust mean estimation algorithm as a black-box in each iteration and then perform the gradient update.

4.3.2 SEVER

Now we describe the main idea behind SEVER. SEVER does not use robust mean estimation as a black-box. In contrast, we take advantage of the performance guarantees of our filtering algorithm. As we shall see in the analysis later, the main idea of the analysis is as follows: when we apply our filtering algorithm, we want the following conditions to hold:

1. When FILTER is still removing points, it removes more bad points than good points.

2. When FILTER is not removing points any more, it has reached a point such that the average gradient over the good samples is close to the gradient of $\bar{f}$

3. When the algorithm terminates, we have sufficiently many samples.

If these conditions hold, since the base learner is assumed to output a γ-approximate critical point of the empirical average of the remaining functions, by condition 2 above, we would obtain a γ-approximate critical point of $\bar{f}$. By large sample theory, since we have sufficiently many samples, this approximation is sound.

Notice the assumption that the base learner is an γ-approximate learner is important. It makes sure that before we run the FILTER procedure, we always have reached a approximate critical point of $\hat{f}$. Thus, in contrast to the projective gradient descent approach, SEVER only calls the robust mean estimation routine each time the algorithm reaches an approximate critical point of $\hat{f}$. One of the main reason we prefer this approach is that an iteration of the filter subroutine (Algorithm 3) is more expensive than an iteration of gradient descent. Consequently, it is advantageous to run many steps of gradient descent in between of consecutive runs of the filter subroutine. The speed of SEVER can be further improved if use stochastic gradient descent instead of regular gradient descent.

4.3.3 SEVER and Projective gradient method in practice

One major conceptual difference between SEVER and the projective gradient method is that SEVER works with a *black-box non-robust* learner and requires the filter algorithm used in robust mean estimation. Unlike SEVER, the projective gradient method works with a *black-box robust mean estimation algorithm* and then plugs into the (approximate) stochastic gradient descent method. These two algorithms have similar theoretical runtime guarantees.

However, in practical implementations, SEVER is preferred for several reasons [DKK$^+$19]. First, in practice we find that in practice, Sever often only requires a constant number of runs of the base black-box learner, and so incurs only a constant factor overhead. In contrast, the algorithm presented in this section requires at least linear time per iteration of SGD, since it needs to run a robust mean estimation algorithm on the entire dataset (and the total number of iterations needed is comparable). In contrast, SGD typically runs in constant time per iteration, so this presents a major bottleneck for scalability.

Second, SEVER gives us more freedom when it comes to the choice of base learners. We can then use problem-specific libraries to experiment with various kinds of base learners. On the other hand, projective gradient method does not give us this freedom as there are not many choices for black-box robust mean estimation methods.

4.4 Analysis of SEVER

4.4.1 Proof of main theorem

In this section, we give a full analysis of SEVER algorithm. Specifically, we will give a proof of Theorem 4.4. To this end, let's restate the theorem and the necessary assumptions below.

Assumption 4.10. *(Deterministic Regularity Conditions) Fix $0 < \epsilon < 1/2$. There exists an unknown set $I_{good} \subseteq [n]$ with $|I_{good}| \geq (1-\epsilon)n$ of "good" functions $\{f_i\}_{i \in I_{good}}$ and parameters $\sigma_0, \sigma_1 \in \mathbb{R}_+$ such that:*

$$\left\| \mathop{\mathbf{E}}_{I_{good}} [(\nabla f_i(w) - \nabla \bar{f}(w))(\nabla f_i(w) - \nabla \bar{f}(w))^T] \right\|_2 \leq (\sigma_0 + \sigma_1 \|w^* - w\|_2)^2, \ \forall w \in \mathcal{H}.$$

and

$$\|\nabla \hat{f}(w) - \nabla \bar{f}(w)\|_2 \leq (\sigma_0 + \sigma_1 \|w^* - w\|_2)\sqrt{\epsilon}, \ \forall w \in \mathcal{H}$$

where $\hat{f} = (1/|I_{good}|) \sum_{i \in I_{good}} f_i$.

Comparing the assumption above with Assumption 4.7, one would notice that despite the same assumptions, Assumption 4.10 requires them to hold for a good set uniformly for all points $w \in \mathcal{H}$. In contrast, 4.7 only requires such a set $I_{\text{good}}(w)$ exists for all points $w \in \mathcal{H}$ and $I_{\text{good}}(w)$ need not be the same as $I_{\text{good}}(w')$ for $w \neq w'$.

Theorem 4.11. *Suppose that the functions $f_1, ..., f_n, \bar{f} : \mathcal{H} \to \mathbb{R}$ are bounded below, and that Assumption 4.10 is satisfied. Let $\sigma := \sigma_0 + \sigma_1 \|w^* - w\|_2$. Then SEVER applied to $f_1, ..., f_n, \sigma$ returns a point $w \in \mathcal{H}$ that, with high probability at least $9/10$, is a $(\gamma + O(\sigma\sqrt{\epsilon}))$-approximate critical point of $\bar{f}$.*

In order to give a formal proof of Theorem 4.11, we require the following lemmas. We mentioned these in the last section. First, we want to make sure that each time we run the filter, if it outputs a S' such that $|S'| < |S|$, then it must remove more bad samples than good samples. Second, if FILTER outputs a S' such that $|S| = |S'|$, then it must suggest that we have sufficiently many points and that the average gradient over the good set has to be close to the gradient of $\bar{f}$ at the current output w. To formalize these requirements, we have the following lemmas.

Lemma 4.12. *If the samples satisfy the first condition in Assumption 4.10, and if* $|S| \geq 2n/3$, *then let* S' *be the output of FILTER*(S, τ, σ), *we have that*

$$\mathbf{E}[|I_{good} \cap (S \backslash S')|] \leq \mathbf{E}[|([n] \backslash I_{good}) \cap (S \backslash S')|].$$

Here, $|I_{\text{good}} \cap (S \backslash S')|$ is the number of good points being removed and $|([n] \backslash I_{\text{good}}) \cap (S \backslash S')|$ is the number of bad points being removed.

Lemma 4.13. *If the samples satisfy Assumption 4.10, FILTER*$(S, \tau, \sigma) = S$, *and* $n - |S| \leq 11\epsilon n$, *then*

$$\Big\| \nabla \bar{f}(w) - \frac{1}{|I_{good}|} \sum_{i \in S} \nabla f_i(w) \Big\|_2 \leq O(\sigma\sqrt{\epsilon}).$$

Assuming these two lemmas, we now give a proof of Theorem 4.11.

Proof. (of Theorem 4.11) We first notice that the algorithm must terminate in n steps as during each iteration of the algorithm, we either stop or remove at least one point.

It remains to prove correctness. To see this, note that Lemma 4.12 states that in expectation, FILTER removes more points from $S \backslash I_{\text{good}}$ than from I_{good}. In particular, this suggests that $|([n] \backslash I_{\text{good}}) \cap S| + |I_{\text{good}} \backslash S|$ is a supermartingale. Since its initial size is at most ϵn, with probability at least $9/10$, it never exceeds $10\epsilon n$ and therefore at the end of the algorithm, we have that

$$n - |S| \leq \epsilon n + |I_{\text{good}} \backslash S| \leq 11\epsilon n.$$

This allows us to use Lemma 4.13 to finish the proof, using the fact that w is a γ-approximate critical point of $(1/|I_{\text{good}}|) \sum_{i \in S} \nabla f_i(w)$. □

4.4.2 Proof of lemmas

We now finish the proof of the main theorem by proving Lemma 4.12 and Lemma 4.13.

Proof. (of Lemma 4.12) Let $S_{\text{good}} = S \cap I_{\text{good}}$ and $S_{\text{bad}} = S \backslash I_{\text{good}}$. We want to show that in expectations, more points are removed from S_{bad} than in S_{good}. This is trivially true if FILTER(S, τ, σ). Thus, we can assume without loss of generality that $\mathbf{E}_{i \in S}[\tau_i] \geq 12\sigma$.

Notice that the expected number of points thrown out of S_{good} and S_{bad} are proportional to $\sum_{i \in S_{\text{good}}} \tau_i$ and $\sum_{i \in S_{\text{bad}}} \tau_i$ respectively. Therefore, it suffices to show that $\sum_{i \in S_{\text{bad}}} \tau_i \geq \sum_{i \in S_{\text{good}}} \tau_i$.

To this end, notice that since $\mathbf{Cov}_{i \in I_{\text{good}}}[\nabla f_i(w)] \preceq \sigma^2 I$, we have

$$\begin{aligned}\underset{i \in S_{\text{good}}}{\mathbf{Cov}}[v \cdot \nabla f_i(w)] &\leq \frac{3}{2} \underset{i \in I_{\text{good}}}{\mathbf{Cov}}[v \cdot \nabla f_i(w)] \\ &= \frac{3}{2} \cdot v^T \underset{i \in I_{\text{good}}}{\mathbf{Cov}}[\nabla f_i(w)] v \\ &\leq 2\sigma^2\end{aligned}$$

where the first inequality follows from the assumption that $|S| \geq 2n/3$. Let $\mu_{\text{good}} = \mathbf{E}_{i \in \text{good}}[v \cdot \nabla f_i(w)]$ and $\mu = \mathbf{E}_{i \in S}[v \cdot \nabla f_i(w)]$. Note that

$$\underset{i \in S_{\text{good}}}{\mathbf{E}}[\tau_i] = \underset{i \in S_{\text{good}}}{\mathbf{Cov}}[v \cdot \nabla f_i(w)] + (\mu - \mu_{\text{good}})^2 \leq 2\sigma^2 + (\mu - \mu_{\text{good}})^2.$$

We now split into two cases. Firstly, if $(\mu - \mu_{\text{good}})^2 \leq 4\sigma^2$, then $\mathbf{E}_{i \in S_{\text{good}}}[\tau_i] \leq 6\sigma^2 \leq \mathbf{E}_{i \in S}[\tau_i]/2$. Thus, we have $\sum_{i \in S_{\text{bad}}} \tau_i \geq \sum_{i \in S_{\text{good}}} \tau_i$.

On the other hand, if $(\mu - \mu_{\text{good}})^2 \geq 4\sigma^2$, we let $\mu_{\text{bad}} = \mathbf{E}_{i \in S_{\text{bad}}}[v \cdot \nabla f_i(w)]$. Note that $|\mu - \mu_{\text{bad}}| \cdot |S_{\text{bad}}| = |\mu - \mu_{\text{good}}| \cdot |S_{\text{good}}|$. We then have that

$$\begin{aligned}\underset{i \in S_{\text{bad}}}{\mathbf{E}}[\tau_i] &\geq (\mu - \mu_{\text{bad}})^2 \\ &\geq (\mu - \mu_{\text{bad}})^2 \Big(\frac{|S_{\text{good}}|}{|S_{\text{bad}}|}\Big)^2 \\ &\geq 2\Big(\frac{|S_{\text{good}}|}{|S_{\text{bad}}|}\Big)(\mu - \mu_{\text{bad}})^2 \\ &\geq \Big(\frac{|S_{\text{good}}|}{|S_{\text{bad}}|}\Big) \underset{i \in S_{\text{good}}}{\mathbf{E}}[\tau_i].\end{aligned}$$

Therefore, we have that $\sum_{i \in S_{\text{bad}}} \tau_i \geq \sum_{i \in S_{\text{good}}} \tau_i$. This conclude the proof. □

Lastly, we give a proof of Lemma 4.13.

Proof. (of Lemma 4.13) To prove the lemma, it suffices to prove that

$$\delta := \Big\| \sum_{i \in S} (\nabla f_i(w) - \nabla \bar{f}(w)) \Big\|_2 = O(n\sigma\sqrt{\epsilon}).$$

By the triangle inequality, we have

$$\begin{aligned}\Big\| \sum_{i \in S} (\nabla f_i(w) - \nabla \bar{f}(w)) \Big\|_2 &\le \Big\| \sum_{i \in I_{\text{good}}} (\nabla f_i(w) - \nabla \bar{f}(w)) \Big\|_2 + \Big\| \sum_{i \in (I_{\text{good}} \backslash S)} (\nabla f_i(w) - \nabla \bar{f}(w)) \Big\|_2 \\ &+ \Big\| \sum_{i \in (S \backslash I_{\text{good}})} (\nabla f_i(w) - \nabla \bar{f}(w)) \Big\|_2 \\ &= \Big\| \sum_{i \in (I_{\text{good}} \backslash S)} (\nabla f_i(w) - \nabla \bar{f}(w)) \Big\|_2 + \Big\| \sum_{i \in (S \backslash I_{\text{good}})} (\nabla f_i(w) - \nabla \bar{f}(w)) \Big\|_2 \\ &+ O(n\sqrt{\sigma^2 \epsilon})\end{aligned}$$

where the last line comes from the regularity assumption on the good set I_{good}. We first analyze

$$\Big\| \sum_{i \in (I_{\text{good}} \backslash S)} (\nabla f_i(w) - \nabla \bar{f}(w)) \Big\|_2.$$

By the functional representation of $| \cdot |_2$, we have that

$$\Big\| \sum_{i \in (I_{\text{good}} \backslash S)} (\nabla f_i(w) - \nabla \bar{f}(w)) \Big\|_2 = \sup_{v \in S^{d-1}} \sum_{i \in (I_{\text{good}} \backslash S)} v \cdot (\nabla f_i(w) - \nabla \bar{f}(w)).$$

Note that by assumption, we have

$$\sum_{i \in (I_{\text{good}} \backslash S)} v \cdot (\nabla f_i(w) - \nabla \bar{f}(w))^2 = O(n\sigma^2).$$

Since $|I_{\text{good}} \backslash S| = O(n\epsilon)$, by Cauchy-Schwarz inequality, we have

$$\sum_{i \in (I_{\text{good}} \backslash S)} v \cdot (\nabla f_i(w) - \nabla \bar{f}(w)) = O(\sqrt{(n\sigma^2)(n\epsilon)}) = O(n\sigma\sqrt{\epsilon}).$$

as desired. On the other hand, we have

$$\sum_{i \in S} v \cdot (\nabla f_i(w) - \nabla \bar{f}(w))^2 = \sum_{i \in S} v \cdot (\nabla f_i(w) - \nabla \hat{f}(w))^2 + \delta^2 = O(n\sigma^2) + \delta^2$$

(or otherwise our filter would have removed elements). Since $|S \backslash I_{\text{good}}| = O(n\epsilon)$, we have similarly that

$$\Big\| \sum_{i \in (S \backslash I_{\text{good}})} (\nabla f_i(w) - \nabla \bar{f}(w)) \Big\|_2 = O(n\sigma\sqrt{\epsilon} + \delta\sqrt{n\epsilon}).$$

Combining these together, we can then conclude that

$$\delta = O(n\sigma\sqrt{\epsilon})$$

as desired. □

4.5 Application in Logistic Regression

In this section, we will see how we can apply SEVER to some learning problems. For an application of SEVER to linear regression, see appendix E.1 of [DKK$^+$19]. Our main focus here is generalized linear models and specifically, logistic regression. As we shall see in the later part of this paper, the non-polynomial nature of the loss function in generalized linear models raises some seemingly insurmountable barriers for sum-of-squares methods. We will examine what kind of theoretical guarantees SEVER can give us on these problems.

4.6 SEVER on Generalized Liner Models

Let's formalize our definition of a generalized linear model through the lenses of data functions.

Definition 4.14. *Let $\mathcal{H} \subseteq \mathbb{R}^d$ and $\mathcal{Y}$ be an arbitrary set. Let D_{xy} be a distribution over $\mathcal{H} \times \mathcal{Y}$. For each $Y \in \mathcal{Y}$, let $\sigma_Y : \mathbb{R} \to \mathbb{R}$ be a convex function. The generalized linear model (GLM) over $\mathcal{H} \times \mathcal{Y}$ with distribution D_{xy} and link function σ_Y is the function: $\bar{f} : \mathbb{R}^d \to \mathbb{R}$ defined by $\bar{f}(w) = \mathbf{E}_{X,Y}[f_{X,Y}(w)]$, where*

$$f_{X,Y}(w) := \sigma_Y(w \cdot X).$$

A sample from this GLM is given by $f_{X,Y}(w)$ where $(X,Y) \sim D_{xy}$.

For support vector machines (SVM), the loss function is given by $f_i((w,(x,y))) = L(w,(x_i,y_i)) = \max\{0, 1 - y(w \cdot x)\}$. In the case of logistic regression, we have $f_i((w,(x,y))) = L(w,(x_i,y_i))$ where $L(w,(x_i,y_i))$ is given as

$$L(w,(x_i,y_i)) = \frac{1+y}{2} \ln\Big(\frac{1}{\phi(w \cdot x)}\Big) + \frac{1-y}{2} \ln\Big(\frac{1}{\phi(-w \cdot x)}\Big)$$

where $\phi(t) = 1/(1+e^{-t})$. We can formalize this loss function $f_{(x,y)}(w)$ using maximum likelihood argument as in [Din21].

Our goal in GLMS is to approximately minimize $\bar{f}$ given ϵ-corrupted samples from D_{xy}. Throughout this section, we assume that $\mathcal{H}$ is contained in the ball of radius r around 0, i.e. $\mathcal{H} \subseteq B(0,r)$. Let $w^* = \arg\min_{w \in \mathcal{H}} \bar{f}(w)$ be a minimizer of $\bar{f}$ in $\mathcal{H}$.

One challenge in GLMs is that it is unclear how to demonstrate that Assumption 4.10 holds after taking polynomially many samples from GLM. To rectify this, we show give a different deterministic regularity condition under which we can show SEVER succeeds. We will show that this condition holds after taking polynomially many samples from a GLM. This alternative deterministic regularity condition is stated below.

Assumption 4.15. *Fix $0 < \epsilon < 1/2$. There exists an unknown set $I_{good} \subseteq [n]$ with $|I_{good}| \geq (1-\epsilon)n$ of "good" functions $\{f_i\}_{i \in I_{good}}$ and parameters $\sigma_0, \sigma_2 \in \mathbb{R}_+$ such that the following conditions hold simultaneously:*

- $\left\| \mathbf{E}_{I_{good}}[(\nabla f_i(w) - \nabla \bar{f}(w))(\nabla f_i(w) - \nabla \bar{f}(w))^T] \right\|_2 \leq \sigma_0^2, \ \forall w \in \mathcal{H}.$
- $\|\nabla \hat{f}(w^*) - \nabla \bar{f}(w^*)\|_2 \leq \sigma_0 \sqrt{\epsilon}.$
- $|\hat{f}(w) - \bar{f}(w)| \leq \sigma_2 \sqrt{\epsilon}$ *for all* $w \in \mathcal{H}$.

where $\hat{f} = (1/|I_{good})| \sum_{i \in I_{good}} f_i$.

Comparing this with Assumption 4.10, we notice that the first two conditions here are looser than the conditions in Assumption 4.10. To see this, notice that if we assume the conditions in in Assumption 4.10, we can obtain the first two conditions here by setting $\sigma_1 = 0$ and σ_0 as it is. More importantly, both of the assumptions in Assumption 4.10 are required to hold for all $w \in \mathcal{H}$ while here we only require the conditions to hold for $w = w^*$. The last condition here states that the sample average is close to the true function f^*. This is not a strong restriction as it easily follows from standard concentration results when n is large. One intuitive reason of why need these somewhat relaxed conditions, is that although convex, the loss functions in a generalized linear model typically take in complicated forms which make it hard to certify uniform closeness of gradients and convariances with polynomially many samples.

With the modified regularity conditions for GLMs, we now have the following theorem.

Theorem 4.16. *For functions $f_1, ..., f_n : \mathcal{H} \to \mathbb{R}$, suppose that Assumption 4.15 holds and that $\mathcal{H}$ is convex. Then, for some universal constants ϵ_0, there is an algorithm which, with probability at least $9/10$, finds a $w \in \mathcal{H}$ such that*

$$\bar{f}(w) - \bar{f}(w^*) = r(\gamma + O(\sigma_0\sqrt{\epsilon})) + O(\sigma^2\sqrt{\epsilon}).$$

If the link functions are ξ-strongly convex, the algorithm finds a $w \in \mathcal{H}$ such that

$$\bar{f}(w) - \bar{f}(w^*) = 2\frac{(\gamma + O(\sigma_0\sqrt{\epsilon}))^2}{\xi} + O(\sigma_2\sqrt{\epsilon}).$$

Towards a proof of the theorem, we need the following lemma.

Lemma 4.17. *Let $f_1, ..., f_n$ satisfy Assumption 4.15. Then with probability at least* $9/10$*, SEVER applied to $f_1, ..., f_n$, σ_0 returns a point $w \in \mathcal{H}$ which is a $(\gamma + O(\sigma_0\sqrt{\epsilon}))$-approximate critical point of $\hat{f}$.*

We will omit the proof of the lemma here. The proof is a simple condition checking and apply Theorem 4.11. Given Lemma 4.17, we can now give a proof of Theorem 4.16.

Proof. (of Theorem 4.16) Let $w \in \mathcal{H}$ be the output of SEVER. By Assumption 4.15, we know that $\hat{f}(w^*) \geq \bar{f}(w^*) - \sigma_2\sqrt{\epsilon}$ and moreover, it is a $\gamma + \sigma_0\sqrt{\epsilon}$-approximate critical point of $\hat{f}$ by Lemma 4.17.

By the convexity of the link functions, we know that their empirical average $\hat{f}$ is also convex. Hence, by Corollary 4.6, we know that $\hat{f}(w) - \hat{f}(w^*) \leq r(\gamma + O(\sigma_0\sqrt{\epsilon}))$. By the last condition of Assumption 4.15, we can then conclude that

$$\bar{f}(w) - \bar{f}(w^*) \leq r(\gamma + O(\sigma_0\sqrt{\epsilon})) + O(\sigma_2\sqrt{\epsilon})$$

as desired. The bound for strongly convex functions follow from the exact same argument using the strongly-convex part of Corollary 4.6. □

Lastly, we will state without proof the following proposition which states that Assumption 4.15 holds with high probability in GLMs with mild assumptions.

Proposition 4.18. *Let $\mathcal{H} \subseteq \mathbb{R}^d$ and let $\mathcal{Y}$ be an arbitrary set. Let $f_1, ..., f_n$ be obtained by picking f_i i.i.d. at random from a GLM $\bar{f}$ over $\mathcal{H} \times \mathcal{Y}$ with distribution D_{xy} and link functions σ_Y, where*

$$n = \Omega\Big(\frac{d\log(dr/\epsilon)}{\epsilon}\Big).$$

Suppose more over that the following conditions all hold:

1. $\mathbf{E}_{X\sim D_{xy}}[XX^T] \preceq I$.
2. $|\sigma_Y'(t)| \leq 1$ *for all* $Y \in \mathcal{Y}$ *and* $t \in \mathbb{R}$.
3. $|\sigma_Y(0)| \leq 1$ *for all* $Y \in \mathcal{Y}$.

Then with probability at least 9/10 over the original set of samples, there is a set of $(1-\epsilon)n$ of the f_i that satisfy Assumption 4.15 on $\mathcal{H}$ with $\sigma_0 = 2, \sigma_1 = 0$ and $\sigma_2 = 1 + r$.

Interested readers can find a formal proof of the statement in Section C.2 of [DKK$^+$19].

4.7 SEVER for Logistic Regression

In this section, we demonstrate how we can apply SEVER to the problem of logistic regression as a special case of GLMS. In logistic regression, we are given $(X_i, Y_i) \in \mathbb{R}^d \times \{\pm 1\}$ for $i \in [n]$ which are drawn from some distribution D_{xy}. Our goal is to model the probability of a point belonging to either the class $+1$ or the class -1. To this end, logistic regressions model the $y = 1$ with probability $\phi(w \cdot x)$ and $y = -1$ with probability $\phi(-w \cdot x)$ where $\phi(t) = 1/(1 + e^{-t})$. We define the loss function via the log-likelihood. Let $f_i(w, (x_i, y_i)) = L(w, (x_i, y_i))$. We have

$$L(w, (x_i, y_i)) = \frac{1+y}{2}\ln\Big(\frac{1}{\phi(w\cdot x)}\Big) + \frac{1-y}{2}\ln\Big(\frac{1}{\phi(-w\cdot x)}\Big) = \frac{1}{2}(-\ln(\phi(w\cdot x)\phi(-w\cdot x)) - y(w\cdot x)).$$

The gradient of this function is given by

$$\nabla L(w, (x, y)) = \frac{1}{2}(\phi(w\cdot x) - \phi(-w\cdot x) - y)x.$$

Our goal is to find a $\hat{w}$ that approximately minimizes the objective function

$$\bar{f}(w) = \mathop{\mathbf{E}}_{(X,Y)\sim D_{xy}}[L(w, (X, Y))].$$

For our results to work for logistic regression, we need the following assumptions.

Assumption 4.19. *Given the model for logistic regression as described above, we assume the following conditions hold:*

- $\mathbf{E}_{X\sim D_x}[XX^T] \preceq I$.
- D_x *is* $\epsilon^{1/4}\sqrt{\log(1/\epsilon)}$*-anticoncentrated.*

With this assumption, we then have the following theorem.

Theorem 4.20. *Let $\epsilon > 0$, and let D_{xy} be a distribution over pairs (X, Y), where the marginal distribution D_x satisfies Assumption 4.19. Then there exists an algorithm that with probability $9/10$, given $O(d\log(d/\epsilon)/\epsilon)$ many ϵ-noisy samples from D_{xy}, returns a $\hat{w}$ such that for any optimal solution w^*, we have*

$$\mathop{\mathbf{E}}_{(X,Y)\sim D_{xy}}[L(\hat{w}, (X, Y))] \le \mathop{\mathbf{E}}_{(X,Y)\sim D_{xy}}[L(w^*, (X, Y))] + O(\epsilon^{1/4}\sqrt{\log(1/\epsilon)}).$$

The main idea is to make some restrictions so that we can apply our results for generalized linear models. To this end, we will restrict our search over w to $\mathcal{H}$, a ball of radius $r = \epsilon^{-1/4}\sqrt{\log(1/\epsilon)}$. As we will see, this restriction comes at a cost of at most $O(\epsilon^{1/4}\sqrt{\log(1/\epsilon)})$ in our algorithm's loss. In this restricted search space, we can then argue that the problem satisfies the conditions of Proposition 4.18. This allows to say that with polynomially-many samples, we can obtain a set of functions $\{f_i\}_{i=1}^n$ satisfying the conditions of Assumption 4.15, enabling us to invoke Theorem 4.16 and finish the proof.

To start, we will show that there is a $w' \in \mathcal{H}$ with loss close to w^* due to the anti-concentration assumption.

Lemma 4.21. *Let w' be a rescaling of w^* such that $\|w'\|_2 \le \epsilon^{-1/4}\sqrt{\ln(1/\epsilon)}$. Then we have*

$$\mathop{\mathbf{E}}_{(X,Y)\sim D_{xy}}[L(w', (X, Y))] \le \mathop{\mathbf{E}}_{(X,Y)\sim D_{xy}}[L(w^*, (X, Y))] + O(\epsilon^{1/4}\sqrt{\ln(1/\epsilon)}).$$

Proof. (of lemma) We will be using the following fact about $\phi(t) = 1/(1 + e^{-t})$:

$$|t| \le -\ln(\phi(t)\phi(-t)) \le |t| + 3\exp(-t).$$

One can see a proof of this fact in Claim E.13 of [DKK+19]. Notice that we always have $|w' \cdot x| - y(w' \cdot x) \le |w^* \cdot x| - y(w^* \cdot x)$. To see this, we do a case analysis on y. Notice we always have $\operatorname{sign}(w' \cdot x) = \operatorname{sign}(w^* \cdot x)$ since w' is a scaled version of w^*. When $y = \operatorname{sign}(w' \cdot x) = \operatorname{sign}(w^* \cdot x)$, both sides of the inequalities are 0. When $y = -\operatorname{sign}(w' \cdot x) = -\operatorname{sign}(w^* \cdot x)$, then the inequality becomes $2|w' \cdot x| \le 2|w^* \cdot x|$, which holds since $\|w'\|_2 \le \|w^*\|_2$. Combining this with the fact above, we have

$$\begin{aligned}
-\ln(\phi(w' \cdot x)\phi(-w' \cdot x)) - y(w' \cdot x) - 3\exp(-3|w' \cdot x|) &\le |w' \cdot x| - y(w' \cdot x) \\
&\le |w^* \cdot x| - y(w^* \cdot x) \\
&\le -\ln(\phi(w^* \cdot x)\phi(-w^* \cdot x)) - y(w^* \cdot x)
\end{aligned}$$

where the first and last inequality comes from the fact above and the second inequality is a result of our case analysis. Equivalently, we have that for any $y \in \{\pm 1\}$, the following result:

$$L(w', (X, Y)) \le L(w^*, (X, Y)) + 3\exp(-3|w' \cdot x|).$$

Therefore, if $|w' \cdot x| \le (1/3)\ln(1/\epsilon)$, then $L(w', (X, Y)) \le L(w^*, (X, Y)) + 3/2$. If $|w' \cdot x| \ge (1/3)\ln(1/\epsilon)$, then $L(w', (X, Y)) \le L(w^*, (X, Y)) + (3/2)\epsilon$. On the other hand, since $\|w'\|_2 \le \epsilon^{-1/4}\sqrt{\ln(1/\epsilon)}$ and D_x is $\epsilon^{1/4}\sqrt{\ln(1/\epsilon)}$-anticoncentrated, we have that

$$P_{D_x}\left[|w' \cdot x| \le \frac{1}{3}\ln(1/\epsilon)\right] \le O(\epsilon^{1/4}\sqrt{\ln(1/\epsilon)}).$$

With this, we can then conclude that

$$\mathop{\mathbf{E}}_{(X,Y)\sim D_{xy}}[L(w', (X, Y))] \le \mathop{\mathbf{E}}_{(X,Y)\sim D_{xy}}[L(w^*, (X, Y))] + O(\epsilon^{1/4}\sqrt{\ln(1/\epsilon)})$$

as desired. □

Now we are ready to a proof of our main theorem (Theorem 4.20).

Proof. (of Theorem 4.20) We show that we can apply the general framework for Generalized Linear Models for our problem. In particular, we want to show the conditions of Proposition 4.18 are satisfied by our logistic model. The link function is given by $\sigma_y(t) = (1/2)(-\ln(\phi(t)\phi(-t) - yt)$, giving us the loss function $L(w, (x, y)) = \sigma_y(w \cdot x)$. Let $\mathcal{H} = B_2(r) = \{x \in \mathbb{R}^d | \ \|x\|_2 \leq r\}$ where $r = \epsilon^{-1/4}\sqrt{\ln(1/\epsilon)}$.

Condition 1 is satisfied by our assumption in Assumption 4.19. For $y \in \{-1, +1\}$, we have $\sigma'_y(t) = (1/2)(\phi(t) - \phi(-t) - y)$, giving us $|\sigma'_y(t)| \leq 1$ for all t and y, satisfying Condition 2. Lastly, we have $\sigma_y(0) = \ln 2 < 1$ for all y, satisfying Condition 3. Therefore, by Proposition 4.18, if we have $O(d \log(dr/\epsilon)/\epsilon)$ ϵ-corrupted samples, then the conditions in Assumption 4.15 are satisfied with $\sigma_0 = 2, \sigma_1 = 0$ and $\sigma_2 = 1 + \epsilon^{-1/4}\sqrt{\ln(1/\epsilon)}$ with probability at least 9/10.

Now by Theorem 4.16, since the loss function is convex, we can obtain a vector $\hat{w}$ such that

$$\bar{f}(\hat{w}) - \bar{f}(w^{*'}) = O((\sigma_0 r + \sigma_1 r^2 + \sigma_2)\sqrt{\epsilon}) = O(\epsilon^{1/4}\sqrt{\ln(1/\epsilon)}).$$

where $w^{*'}$ is the minimizer of $\bar{f}$ on $\mathcal{H}$. As a result, we conclude that

$$\begin{aligned}\bar{f}(\hat{w}) &\leq \bar{f}(w^{*'}) + O(\epsilon^{1/4}\sqrt{\ln(1/\epsilon)}) \\ &\leq \bar{f}(w^{'}) + O(\epsilon^{1/4}\sqrt{\ln(1/\epsilon)}) \\ &\leq \bar{f}(w^{*}) + O(\epsilon^{1/4}\sqrt{\ln(1/\epsilon)})\end{aligned}$$

where the second inequality follows since $w^{*'}$ is the minimizer of $\bar{f}$ on $\mathcal{H}$ and the third inequality comes from Lemma 4.21. □

5 Sum of Squares proofs and optimization

So far we have been talking about filtering-based method in robust statistics. We now switch gears to another set of tools that have been popular in the robust statistics community in recent years. Specifically, the methods we are going to cover make crucial use of sum-of-squares techniques in which the key element is that *the existence of a low-degree sum-of-squares proof of certain statements implies the existence of an efficient algorithm.* We will define what a *low-degree sum-of-squares* means and how that implies an efficient algorithm.

Compared with filtering-based approaches, sum-of-squares based methods provide a general recipe for parameter recovery and learning problems. In some sense, sum-of-squares methods are like programming in that you are manipulating the polynomials at hand to give a low-degree proof of the statement you want to prove. In many concrete examples, sum-of-squares based methods are nearly optimal. However, sum-of-squares-based methods have issues as well. First, for learning problems, it crucially relies on the convexity and the polynomial structure of the loss function. Hence, it can be applied to problems like l_1 and l_2 regressions but it is not known how it can be adapted to problems with non-polynomial loss functions, for instance, logistic regression. So in this sense, sum-of-squares methods do not give enough "coverage" for learning problems.

Another issue is its practicality. Unlike SEVER and the projective gradient descent algorithm which can be easily implemented using standard libraries in machine learning, sum-of-squares-based algorithms require solving a large semi-definite program and as far as our searches went, the best known existing package can only handle low moment tensors in a small number of variates, which deem it infeasible for practical purposes yet.

Nevertheless, the method itself is elegant and has near-optimal performances for many learning problems. It is also extremely exciting and interesting to see how a "proof" can be converted into an efficient algorithm. We will introduce the basics in sum-of-squares methods, use the one-dimensional Gaussian mixture model as a simple example and in the next section, we will see how we can use sum-of-squares to obtain an efficient algorithm for robust linear regression.

In this section, we define pseudo-distributions and sum-of-squares proofs. We will present a few important conclusions that will be used throughout this article. We will follow the exposition in [KKM20]. For a more systematic study of sum-of-squares proofs, see [BS16]. The proofs of the propositions below can be found in the appendix in [MSS16].

Let $x = (x_1, ..., x_n)$ be a tuple of n indeterminates. Let $\mathbb{R}[x]$ be the ring of polynomials with coefficients in $\mathbb{R}$ and indeterminates in $x_1, ..., x_n$. We say that a polynomial $p \in \mathbb{R}[x]$ is a *sum-of-squares (sos)* if there are polynomials $q_1, ..., q_r \in \mathbb{R}[x]$ such that $p = q_1^2 + ... + q_r^2$.

5.1 Pseudo-distributions

As the name suggests, pseudo-distributions are generalizations of probability distributions. Consider a probability distribution p with finite support, i.e., $|\text{supp}(p)| < \infty$. Then we must have that $\sum_{p(x)} = 1$ and $p(x) \geq 0$ for all $x \in \text{supp}(x)$. For pseudo-distributions with finite support, we can similarly describe it using its mass function. However, we relax the condition that the mass function has to be non-negative over the support. Instead, we only require that a pseudo-distribution passes certain low-degree non-negativity tests.

Specifically, we say a finitely-supported function $D : \mathbb{R}^n \to R$ is a *level-l pseudo-distribution* if $\sum_{x \in \text{supp}(D)} D(x) = 1$ and $\sum_{x \in \text{supp}(D)} D(x) f(x)^2 \geq 0$ for every polynomial f of degree at most $l/2$. It is easy to see that every level-∞ pseudo-distribution satisfies $D(x) \geq 0$ for all $x \in \text{supp}(x)$ and is thus a probability distribution. We define the *pseudo-expectation* of a function f on $\mathbb{R}^d$ with respect to a pseudo-distribution D, denoted as $\tilde{E}_{D(x)} f(x)$, as

$$\tilde{\mathbf{E}}_{D(x)} f(x) = \sum_{x \in \text{supp}(x)} D(x) f(x).$$

Consider monomials of $(x_1, ..., x_n)$ which can be expressed as $x_1^{e_1} x_{m_2}^{e_2} ... x_n^{e_n}$ where $e_i \geq 0 \ \forall i$. with pseudo-expectation $\tilde{\mathbf{E}}_{D(x)} f(x)[x_{m_1} x_{m_2} ... x_{m_l}]$. The *degree-$l$ moment tensor* of a pseudo-distribution D is given by the tensor $\mathbf{E}_{D(x)}(1, x_1, ..., x_n)^{\otimes l}$ whose entries are exactly the pseudo-expectations of all the monomials in $(1, x_1, ..., x_n)$ with degree at most l. The set of all degree-l moment tensors of probability distributions is a convex set. Similarly, the set of all degree-l moment tensors of degree d pseudo-distributions is also convex.

One important desirable property of pseudo-distributions is the existence of an efficient separation oracle for this convex set of degree-l moment tensors while for probability distributions, such a separation oracle does not exist. In particular, there is a separation oracle running in time $n^{O(l)}$ for the convex set of the degree-l moment tensors of all level-l pseudo-distributions.

Fact 5.1. *[Las01, Nes00, Par00, Sho87] For any $n, l \in \mathbb{N}$, there exists a $n^{O(l)}$-time weak separation oracle for the following set:*

$$\{\tilde{E}_{D(x)}(1, x_1, ..., x_n)^{\otimes l} \mid \textit{degree-}l \textit{ pseudo-distribution } D \textit{ over } \mathbb{R}^n\}.$$

The oracle is an application of the ellipsoid method. Interested readers can find the exact oracle in [GLS81]. The equivalence of weak separation and optimization (see [GLS81] and this proposition allows us to efficiently optimize over pseudo-distributions (approximately i.e., some rounding procedures needed to convert the output to a solution). In literature, this algorithm is often referred to as the sum-of-squares algorithm. One important feature of this algorithm is that it not only works for general pseudo-distributions, it can also be adapted to work for *constrained pseudo-distributions.*

Definition 5.2. *(Constrained pseudo-distributions). Let D be a level-l pseudo-distribution over $\mathbb{R}^n$. Let $\mathcal{A} = \{f_1 \geq 0, ..., f_m \geq 0\}$ be a system of m polynomial inequality constraints. We say that D satisfies the system of constraints $\mathcal{A}$ at degree r, denoted $D \underset{r}{\models} \mathcal{A}$, if for every $S \subseteq [m]$ and every sum-of-squares polynomial h with deg $h + \sum\limits_{i \in S} \max\{\textit{deg } f_i, r\} \leq l$,*

$$\tilde{\mathbf{E}}_D h \cdot \prod_{i \in S} f_i \geq 0.$$

We write $D \models A$ (without specifying the degree) if $D \underset{0}{\models} \mathcal{A}$ holds. Let p be a polynomial with degree deg $p \leq \infty$. Let $c(p) \in \mathbb{R}^{\deg\, p}$ denote the vector of coefficients of p in the monomial basis. Define the norm of p as $\|p\| = \|c(p)\|_2$, i.e., the Euclidean norm of coefficient vector. We say that $D \underset{r}{\models} \mathcal{A}$ *approximately* if the above inequalities are satisfied up to an error of $\epsilon = 2^{-n^l} \cdot \|h\| \cdot \prod_{i \in S} \|f_i\|$. In other words, $\tilde{\mathbf{E}}_D h \cdot \prod\limits_{i \in S} f_i \geq -\epsilon$. Notice that the choice of Euclidean norm is not important (i.e., can be l_1, l_∞ and other norms) since the error of choosing a different norm can be absorbed in to the 2^{-n^l} factor by equivalence of norms. If D is a discrete probability distribution, then we have $D \models \mathcal{A}$ if and only if D is supported on solutions to the constraints $\mathcal{A}$.

We say that a system $\mathcal{A}$ of polynomial constraints is *explicitly bounded* if it contains a constraint of the form $\{\|x\|^2 \leq M\}$. As a consequence of Proposition 5.1 and [GLS81], we have the following result,

Fact 5.3. *(Efficient Optimization over Pseudo-distributions). There exists an $(n + m)^{O(l)}$-time algorithm that, given any explicitly bounded and satisfiable system $\mathcal{A}$ of m polynomial constraints in n variables, outputs a level-l pseudo-distribution that satisfies $\mathcal{A}$ approximately.*

Here we assume the bit-complexity of the constraints in $\mathcal{A}$ is $(n + m)^{O(1)}$. We will be using the following properties of pseudo-distributions frequently.

Fact 5.4. *(Pseudo-distribution Cauchy-Schwarz inequality) If $\tilde{\mu}$ is a degree r pseudo-distribution and f, g are polynomials of degree at most $r/2$ then*

$$\left(\tilde{\mathbf{E}}_{\tilde{\mu}} fg\right)^2 \leq \left(\tilde{\mathbf{E}}_{\tilde{\mu}} f^2\right)\left(\tilde{\mathbf{E}}_{\tilde{\mu}} g^2\right).$$

Fact 5.5 (Pseudo-distribution Hölder's inequality)**.** *Let f, g be sum-of-squares polynomials. Let p, q be positive integers so that $1/p + 1/q = 1$. Then for any pseudo-distribution $\tilde{\mu}$ of degree $r \geq pq \cdot \textit{deg}(f) \cdot \textit{deg}(g)$, we have:*

$$(\tilde{\mathbf{E}}_{\tilde{\mu}}[f \cdot g]^{pq}) \leq \tilde{\mathbf{E}}[f^p]^q \cdot \tilde{\mathbf{E}}[g^q]^p$$

In particular, for all even integers $k \geq 2$, and polynomial f with $\textit{deg}(f) \cdot k \leq r$,

$$(\tilde{\mathbf{E}}_{\tilde{\mu}}[f])^k \leq \tilde{E}_{\tilde{\mu}}[f^k].$$

5.2 Sum-of-squares proofs

The dual concept of pseudo-distribution is the concept of *sum-of-squares proofs.*

Definition 5.6. *Let $f_1, ..., f_r$ and g be multivariate polynomials in x. A sum-of-squares proof that the constraints $\mathcal{A} = \{f_1 \geq 0, ..., f_r \geq 0\}$ imply the constraint g consists of (sum-of-squares) polynomials $(p_S)_{S \subseteq \in [m]}$ such that*

$$g = \sum_{S \subseteq [m]} p_S \cdot \prod_{i \in S} f_i.$$

It is easy to see that a sum-of-squares proof certifies that $g \geq 0$. We say that this proof has *degree* l if for every set $S \subseteq [m]$, the polynomial $p_S \prod_{i \in S} f_i$ has degree at most l. If there is a degree l sum-of-squares proof that $\mathcal{A} = \{f_i \geq 0 | i \leq r\}$ implies $\{g \geq 0\}$, we write:

$$\mathcal{A} \vdash_{l} \{g \geq 0\}.$$

Sum-of-squares proofs extend naturally to polynomial systems that involve equalities of the form $\{p_i = 0\}$ via the following equivalence: $p_i = 0 \iff p_i \geq 0$ and $-p_i \geq 0$. Let $\mathcal{P}_n$ denote the set of all polynomials on n variables with real coefficients. For all polynomials $f, g : \mathbb{R}^n \to \mathbb{R}$ and for all functions $F, G, H \in \mathcal{F}$ where $\mathcal{F} = \{F : \mathbb{R}^n \to \mathbb{R}^m | (F(x))_i = p_i(x), p_i \in \mathcal{P}_n\}$, i.e., each coordinate of the output of the function is a polynomial of the input. Then we have the following rules:

- Addition:
$$\frac{\mathcal{A} \vdash \{f \geq 0, g \geq 0\}}{\mathcal{A} \vdash \{f + g \geq 0\}}$$

- Multiplication:
$$\frac{\mathcal{A} \vdash_{l} \{f \geq 0\}, \mathcal{A} \vdash_{l'} \{g \geq 0\}}{\mathcal{A} \vdash_{l+l'} \{f \cdot g \geq 0\}}$$

- Transitivity:
$$\frac{\mathcal{A} \vdash_{l} \mathcal{B}, \mathcal{B} \vdash_{l'} \mathcal{C}}{A \vdash_{l+l'} \mathcal{C}}$$

- Substitution:
$$\frac{\{F \geq 0\} \vdash_{l} \{G \geq 0\}}{\{F(H) \geq 0 \vdash_{l \cdot \deg(H)} G(H) \geq 0\}}.$$

where $\frac{A}{B}$ means that "A implies B." We also have the following fact,

Fact 5.7. *For polynomial systems $\mathcal{A}, \mathcal{B}$ and polynomials p, q, if $A \vdash_{l} \{p \geq 0\}$ and $B \vdash_{l'} \{q \geq 0\}$, then we have:*

$$\mathcal{A} \cup \mathcal{B} \vdash_{\max(l,l')} \{p(x) + q(x) \geq 0\}$$

and

$$\mathcal{A} \cup \mathcal{B} \vdash_{l,l'} \{p(x)q(x) \geq 0\}.$$

Intuitively, sum-of-squares proofs can be a proof system for statements involving polynomial inequalities. It is indeed true that low-degree sum-of-squares proofs are sound and complete if we take low-level pseudo-distributions as models. This allows us to deduce properties of pseudo-distributions that satisfy certain constraints via sum-of-squares proofs.

Fact 5.8. *(Soundness) If $D \models_{r} \mathcal{A}$ for a level-l pseudo-distribution D and there exists a sum-of-squares proof $\mathcal{A} \vdash_{r'} \mathcal{B}$, then $D \models_{r'(r+1)} \mathcal{B}$.*

If the pseudo-distribution D satisfies $\mathcal{A}$ approximately, then we still have the soundness of sum-of-squares proof provided we have an upper bound on the bit-complexity of the sum-of-squares $\mathcal{A} \vdash_{r'} \mathcal{B}$. The bit-complexity of all sum-of-squares proofs in our arguments will be $n^{O(l)}$ which allows to talk about approximate satisfiability of a polynomial system $\mathcal{A}$ based on the above fact.

In addition to soundness, we also have completeness.

Fact 5.9. *(Completeness) Suppose $d \geq r' \geq r$ and $\mathcal{A}$ is a collection of polynomial constraints with degree at most r and $\mathcal{A} \vdash \{\sum_{i=1}^n x_i^2 \leq B\}$ for some finite B. Let $\{g \geq 0\}$ be a polynomial constraint. If every degree-d pseudo-distribution that satisfies $D \models \mathcal{A}$ also satisfies $D \models_{r'} \{g \geq 0\}$, then for every $\epsilon > 0$, there is a sum-of-squares proof $\mathcal{A} \vdash_{d} \{g \geq -\epsilon\}$.*

We will also be using the following sum-of-squares version of Hölder's inequality.

Fact 5.10. *(Sum-of-squares Hölder's inequality) Let $f_1, ..., f_n$ and $g_1, ..., g_n$ be sum-of-squares polynomials over $\mathbb{R}^d$. Let p, q be positive integers such that $1/p + 1/q = 1$. Then,*

$$\vdash^{f_1,...,f_n,g_1,...,g_n}_{pq} \Big\{ \Big(\frac{1}{n}\sum_{i=1}^{n} f_i g_i\Big)^{pq} \leq \Big(\frac{1}{n}\sum_{i=1}^{n} f_i^p\Big)^q \Big(\frac{1}{n}\sum_{i=1}^{n} g_i^p\Big)^q \Big\}.$$

where $\vdash^{d}_{x} \mathcal{A}$ denotes that there exists a degree d sum-of-squares proof of the polynomial system $\mathcal{A}$ in terms of the variable x. We also have the following facts.

Fact 5.11. *(Sum-of-squares triangle inequality) Let x, y be indeterminates (possibly sum-of-squares polynomials). Let t be a power of 2. Then*

$$\vdash_{t} (a+b)^t \leq 2^{t-1}(a^t + b^t).$$

Fact 5.12. *For any sum-of-squares polynomials $f_1, ..., f_n$, we have*

$$\vdash^{f_1,f_2,...,f_n}_{k} \Big\{ \big(\sum_{i=1}^{n} f_i\big)^k \leq n^k\big(\sum_{i=1}^{n} a_i^k\big) \Big\}.$$

5.3 Application: Mixture of Gaussians

In this section, we give an example of sum-of-squares methods in statistics. In particular, we will give a sum-of-squares proof of one key statement which leads to an efficient algorithm for the *mixture of Gaussians* problem.

Let $\mu_1, ..., \mu_k$ be k vectors in $\mathbb{R}^d$ such that $\|\mu_i - \mu_j\|_2 \geq \Delta$ for all $i, j \in [k]$ such that $i \neq j$. Consider the following spherical Gaussian distributions $p_1, ..., p_k$ where $p_k = \mathcal{N}(\mu_k, I)$. Suppose we observe n i.i.d samples $x_1, ..., x_n \in \mathbb{R}^d$ each drawn by selecting $j \sim [k]$ uniformly and then drawing $X_i \sim \mathcal{N}(\mu_j, I)$. Let $S_1, ..., S_k$ be the induced partition of $[n]$ such that $i \in S_j$ if sample x_i was drawn from $\mathcal{N}(\mu_j, I)$. Our goal is to output a partition $\{T_1, ..., T_k\}$ of $[n]$, each of size n/k, such that

$$|S_i \cap T_i| \geq (1 - \delta) \cdot \frac{n}{k}$$

for each $i \in [k]$ and some small parameter $\delta > 0$ up to relabelings of the clusters. One recent result on this problem has been proved in three independent works.

Theorem 5.13. *[HL17, KS17, DKS17] For arbitrarily large $t \in \mathbb{N}$, there is an algorithm requiring $n = d^{O(t)}k^{O(1)}$ samples from the equidistributed mixture of Gaussians model running in time $n^{O(t)}$ that outputs a partition $T_1, ..., T_k$ of $[n]$ into k partitions each of size $N = n/k$ such that the following holds with high probability for some universal constant C,*

$$\frac{|S_i \cap T_i|}{N} \geq 1 - k^{10} \cdot \Big(\frac{C\sqrt{t}}{\Delta}\Big)^t.$$

where *equidistributed* means that we have exactly n/k samples form each Gaussian in the mixture. We will not be going through the exact proof of the theorem. Interested readers can refer to [HL17, KS17, DKS17] or the nice blog posts by Sam Hopkins [Hop18]. In the remainder of this section, we will for simplicity assume that $d = 1$, i.e. $x_i \in \mathbb{R}$ for all $i \in [n]$. Towards the proof of Theorem 5.13, a low-degree sum-of-squares proof of the following lemma is needed.

Lemma 5.14. *Let $S, S' \subseteq \mathbb{R}$ such that $|S| = |S'| = N$. Let X, X' be uniform samplers from S and S' respectively. Let $\mu = \mathbf{E}[X]$ and $\mu' = \mathbf{E}[X']$. Suppose X, X' satisfies the t-th moment bound*

$$\mathbf{E}\,|X - \mu|^t \leq 2 \cdot t^{t/2} \text{ and } \mathbf{E}\,|X' - \mu'|^t \leq 2 \cdot t^{t/2}.$$

Then we have

$$|\mu - \mu'| \leq 4\sqrt{t} \cdot \Big(\frac{|S \cap S'|}{N}\Big)^{-1/t}.$$

As the current statement of the lemma stands, it is hard to give a sum-of-squares of proofs because of the exponents in the conclusion. However, if we are able to give a low-degree sum-of-squares proof of an equivalent statement of the lemma above, then we able to output a pseudo-distribution $\tilde{p}$ by Fact 5.3 and soundness of sum-of-squares proof (Fact 5.8). We can then use it to give an efficient algorithm for the problem and thus prove Theorem 5.13. To this end, let's consider the following *almost* equivalent statement of Lemma 5.14.

Lemma 5.15. *Let $X_1, ..., X_n \in \mathbb{R}$. Let $S \subseteq [n]$ such that $|S| = N$. Denote its mean to be $\mu_S = \mathbf{E}_{i\sim S}[X_i]$. Let t be a power of 2. Suppose S satisfies*

$$\mathop{\mathbf{E}}_{i\sim S} |X_i - \mu_S|^t \le 2 \cdot t^{t/2}. \tag{2}$$

Let $w_1, ..., w_n$ be indeterminates. Let $\mathcal{A}$ be the following set of equations and inequalities.

$$w_i^2 = w_i \textit{ for } i \in [n] \tag{3}$$

$$\sum_{i\in[n]} w_i = N \tag{4}$$

$$\frac{1}{N}\sum_{i\in[n]} w_i \cdot (X_i - \mu)^t \le 2 \cdot t^{t/2}. \tag{5}$$

Then

$$\mathcal{A} \Big|\!\frac{}{O(t)} \left(\frac{|S\cap T|}{N}\right)^t \cdot (\mu - \mu_S)^t \le 2^{O(t)} \cdot t^{t/2} \cdot \left(\frac{|S\cap T|}{N}\right)^{t-1}.$$

where $|S\cap T|(w) = \sum\limits_{i\in S} w_i$. Notice the differences between the two lemmas: (a) The inequality in the conclusion of Lemma 5.15 is raised to the t-th power. (b) The inequality in the conclusion of Lemma 5.15 has an extra factor of $|S \cap T|/N$ on both sides. The first difference is easy to understand since we want all exponents to be positive integers in order to give a sum-of-squares proof. The second difference is more subtle and we will need it for our purpose. Interested readers can refer to [Hop18, HL17].

In the statement of Lemma 5.15, we introduced the indeterminates $\{w_i\}_{i=1}^n$. The way to think about these indeterminates is that they are indicators of membership in the set. In other words, suppose we know exactly which points belong to the set S, then $w_i = 1$ if $X_i \in S$ and $w_i = 0$ if $X_i \notin S$. However, we do not possess this information. Hence, the set of constraints $\mathcal{A}$ can be regarded as a relaxed version of this set of membership constraints.

To see this, (3) is equivalent to saying $w_i = 1$ or $w_i = 0$, i.e, each point either belongs to the set S or not. (4) together with (3) suggests that there are exactly N points in $\{X_i\}_{i=1}^n$ that belong to the set S. Notice that if we knew exactly which points belong to S, (4) would become $\sum_{i\in S} w_i = N$. This is why we say $\mathcal{A}$ is a relaxation. Lastly, (5) is simply (2) with intermediates. We will be using the same idea again in the next section when we talk about sum-of-squares methods for robust linear regression.

It is easy to see that $\mathcal{A}$ is explicitly bounded. In order to apply Fact 5.3, we in addition need that $\mathcal{A}$ is satisfiable. This is indeed true if we let $w_i = 1$ for points that belong to S and $w_i = 0$ otherwise. Now we are ready to give a sum-of-squares proof of Lemma 5.15. We follow the steps outlined in [Hop18].

Proof. (of Lemma 5.15) First, notice that

$$|S\cap T|^t \cdot (\mu - \mu_S)^t = \Big(\sum_{i\in S} w_i\Big)^t \cdot (\mu - \mu_S)^t = \Big(\sum_{i\in S} w_i[(\mu - X_i) - (\mu_S - X_i)]\Big)^t.$$

Then, by sum-of-squares Hölder's inequality, we have

$$\mathcal{A} \Big|\!\frac{}{O(t)} \Big(\sum_{i\in S} w_i[(\mu - X_i) - (\mu_S - X_i)]\Big)^t \le \Big(\sum_{i\in S} w_i\Big)^{t-1} \cdot \sum_{i\in S} w_i[(\mu - X_i) - (\mu_S - X_i)]^t.$$

Substituting $w_i^2 - w_i = 0$ into the inequality above, we have

$$\mathcal{A} \Big|\!\frac{}{O(t)} \Big(\sum_{i\in S} w_i[(\mu - X_i) - (\mu_S - X_i)]\Big)^t \le \Big(\sum_{i\in S} w_i^2\Big)^{t-1} \cdot \sum_{i\in S} w_i^2[(\mu - X_i) - (\mu_S - X_i)]^t.$$

Now notice that the polynomial $\sum_{i\in S} w_i^2\Big)^{t-1}$ is a sum-of-squares polynomial. Applying the sum-of-squares triangle inequality (Fact 5.11) with $a = \mu - X_i$ and $b = -(\mu_S - X_i)$, we obtain

$$\mathcal{A} \Big|\!\frac{}{O(t)} \Big(\sum_{i\in S} w_i[(\mu - X_i) - (\mu_S - X_i)]\Big)^t \le 2^t \Big(\sum_{i\in S} w_i^2\Big)^{t-1} \cdot \sum_{i\in S} w_i^2(\mu - X_i)^t + w_i^2(\mu_S - X_i)^t.$$

Adding the sum-of-squares $2^t(\sum_{i \in S} w_i)^{t-1} \cdot \sum_{i \notin S} w_i^2(\mu - X_i)^t + w_i^2(\mu_S - X_i)^t$ gives us

$$\mathcal{A} \Big|\frac{}{O(t)} \Big(\sum_{i \in S} w_i[(\mu - X_i) - (\mu_S - X_i)]\Big)^t \leq 2^t \Big(\sum_{i \in S} w_i^2\Big)^{t-1} \cdot \sum_{i \in [n]} w_i^2(\mu - X_i)^t + w_i^2(\mu_S - X_i)^t.$$

Using the equation $w_i^2 - w_i = 0$, we can get a degree-2 sum-of-squares proof of the fact $w_i^2 \leq 1$. Substituting these into the inequality, we get

$$\mathcal{A} \Big|\frac{}{O(t)} \Big(\sum_{i \in S} w_i[(\mu - X_i) - (\mu_S - X_i)]\Big)^t \leq 2^t \Big(\sum_{i \in S} w_i^2\Big)^{t-1} \cdot \sum_{i \in [n]} w_i(\mu - X_i)^t + (\mu_S - X_i)^t.$$

Since $\mathbf{E}_{i \in S}(\mu_S - X_i)^t \leq 2 \cdot t^{t/2}$ and $\mathcal{A} \Big|\frac{}{O(t)} \sum_{i \in [n]} w_i(X_i - \mu)^t \leq 2 \cdot t^{t/2} \cdot N$, we have

$$\mathcal{A} \Big|\frac{}{O(t)} \Big(\sum_{i \in S} w_i[(\mu - X_i) - (\mu_S - X_i)]\Big)^t \leq 2^{O(t)} \cdot \Big(\sum_{i \in S} w_i^2\Big)^{t-1} \cdot t^{t/2} \cdot N.$$

Finally, using $w_i^2 - w_i = 0$ and dividing both sides of the inequality by N^t, we conclude that

$$\mathcal{A} \Big|\frac{}{O(t)} \Big(\sum_{i \in S} \frac{w_i}{N}[(\mu - X_i) - (\mu_S - X_i)]\Big)^t \leq 2^{O(t)} \cdot \Big(\frac{\sum_{i \in S} w_i^2}{N}\Big)^{t-1} \cdot t^{t/2}$$

$$\mathcal{A} \Big|\frac{}{O(t)} \Big(\frac{|S \cap T|}{N}\Big)^t \cdot (\mu - \mu_S)^t \leq 2^{O(t)} \cdot t^{t/2} \cdot \Big(\frac{|S \cap T|}{N}\Big)^{t-1}.$$

□

With an sum-of-squares proof of Lemma 5.15, one is in good position to prove Theorem 5.13. See [Hop18, HL17] for details of the reminder of the proof.

6 Robust Linear Regression via Sum-of-squares

In this section, we present the sum-of-squares based algorithm for robust linear regression by Klivans, Kothari and Meka [KKM20]. Concretely, we will describe the robust certifiability condition needed for the method. We will describe the algorithm and analyze its performance.

6.1 Setup

For a real-valued random variable X and integer $k \geq 0$, we let $\|X\|_k = \mathbf{E}[X^k]^{1/k}$. Given a distribution $\mathcal{D}$ over $\mathbb{R}^d \times \mathbb{R}$ and a vector $l \in \mathbb{R}^d$, define $\mathrm{err}_{\mathcal{D}}(l) = \mathbf{E}_{(x,y)\sim\mathcal{D}}[(\langle l, x\rangle - y)^2]$ and let $\mathrm{opt}(\mathcal{D}) = \min_{l\in\mathbb{R}^d} \mathrm{err}_{\mathcal{D}}(l)$. In the classical setting of unregularized linear regression, we are given access to n i.i.d samples (x_i, y_i) from a distribution D over $\mathbb{R}^d \times \mathbb{R}$ and our goal is to find a linear function l that minimizes $\mathrm{err}_D(l)$. By definition, we always have $\mathrm{err}_D(l) \geq \mathrm{opt}(\mathcal{D})$.

In outlier-robust linear regression, our goal is similar except that the samples we observe are not fully faithful in the sense that up to an η-fraction of the samples may be arbitrarily corrupted. The model of corruption we are considering is the strong contamination model 2.3. For the purpose of clarity, we will revisit the definition here and state it in a version specifically for robust linear regression.

Definition 6.1. *(η-Corrupted samples). Let $\mathcal{D}$ be a distribution over $\mathbb{R}^d \times \mathbb{R}$. A set $\mathcal{U} \subset \mathbb{R}^d \times \mathbb{R}$ is said to be an η-corrupted training set drawn from $\mathcal{D}$ if is formed in the following fashion: generate a set X of i.i.d samples from $\mathcal{D}$ and arbitrarily modify any η fraction to produce $\mathcal{U}$.*

The adversary can modify up to an η-fraction of the samples however they want after inspecting all the inliners X as long as $|\mathcal{U} \cap X|/|X| \geq 1 - \eta$. With this model of corruption, our goal in robust linear regression is: given access to an η-corrupted training set $\mathcal{U}$ from $\mathcal{D}$, find a linear function $f = \langle \cdot, l\rangle$ for some $l \in \mathbb{R}^d$ such that the error $\mathrm{err}_{\mathcal{D}}(l)$ under the true distribution $\mathcal{D}$ is as close to $\mathrm{opt}(\mathcal{D})$ as possible.

In the classical unregularized least-squares setting where the labels y_i are bounded. The least-squares estimator is consistent in the realizable-case. Specifically, let $\mathcal{D}$ be a distribution over $\mathbb{R}^d \times [-1, 1]$. Note here the condition $y_i \in [-1, 1]$ is not important as long as the labels are bounded. Let $\{(x_i, y_i)\}_{i=1}^n$ be i.i.d samples from $\mathcal{D}$. Let $\hat{l} = \arg\min_{l\in\mathbb{R}^d}(1/n)\sum_{i=1}^n (y_i - \langle l, x_i\rangle)^2$ be the least-squares estimator. Then we have the following guarantees (see [GKKW02]),

$$\mathrm{err}_{\mathcal{D}}(\hat{l}) \leq \frac{O(d)}{n} + 8 \cdot \arg\min_{l} \mathrm{err}_{\mathcal{D}}(l).$$

In particular, in the realizable case where there is a true linear function l^* such that $y_i = \langle l^*, x_i\rangle$ for all $i \in [n]$, then $err_{\mathcal{D}}(\hat{l})$ decays at a rate of $1/n$ and goes to 0 as $n \to \infty$ without any additional distributional assumption on $\mathcal{D}$. A natural question to ask is: *Can we obtain a consistent estimator for robust linear regression in the realizable case?*

The answer is no. As stated in section 1, one of the difficulties in robust statistics is the need for distributional assumptions even for certain task to be possible. In our case of robust linear regression, a counterexample is given in the proof of Lemma 6.1 in [KKM20] which suggests that without any distributional assumption, even in the realizable case, $err_{\mathcal{D}}(l)$ is lower bounded by some universal constant c.

Hence, some distributional assumptions on $\mathcal{D}$ are needed in order for a consistent estimator in the realizable case to exist. To this end, we impose the hypercontrativity assumption on $\mathcal{D}$.

Definition 6.2. *(Hypercontractivity) For a function $f : \mathbb{R}^d \to \mathbb{R}$, We say a distribution $\mathcal{D}$ on $\mathbb{R}^d$ is (C, k)-hypercontractive if for all $r \leq k/2$, we have*

$$\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\langle x, l\rangle^{2r}] \leq \left(C(r) \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\langle x, l\rangle^2]\right)^r.$$

In other words, the $2r$-th moment of $\langle x, l\rangle$ is controlled by the r-th power of the second moment up to a constant factor. In addition, we say that $\mathcal{D}$ is *certifiably (C, k)-hypercontractive* if there is a degree-k sum-of-squares proof of the above inequality. Notice that the above condition is invariant under affine transformations. It is also not restrictive in a sense that many common, well-studied distributions such as Gaussians, affine transformations of log-concave distributions, uniform distributions over Boolean hypercubes and product distribution of bounded random variables all satisfy this condition [KKM20]. Now we are ready to state the main theorem in this section.

Theorem 6.3. *Let $\mathcal{D}$ be a distribution on $\mathbb{R}^d \times [-M, M]$ and $\mathcal{D}_X$ be its marginal distribution which is certifiably $(C, 4)$-hypercontractive. Let the optimal solution $l^* = \arg\min_l err_{\mathcal{D}}(l)$ have polynomial-bit complexity. Then for all $\epsilon > 0$ and $\eta < c/C^2$ for a universal constant $c > 0$, there exists an algorithm $\mathcal{A}$ with run-time $poly(d, 1/\eta, 1/\epsilon, M)$ that given a polynomial-size η-corrupted training set $\mathcal{U}$, outputs a linear function l such that with probability at least $1 - \epsilon$,*

$$err_{\mathcal{D}}(l) \leq (1 + O(\sqrt{\eta})) \cdot opt(\mathcal{D}) + O(\sqrt{\eta}) \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}[(y - \langle l^*, x\rangle)^4]^{1/2} + \epsilon.$$

Notice that if there exists a true linear functional l^* such that $y_i = \langle l^*, x_i\rangle$ then algorithm $\mathcal{A}$ has guarantee $\text{err}_{\mathcal{D}}(l) \leq \epsilon$. This suggests that the error approaches 0 at a polynomial rate. If we have a higher level of hypercontractivity, then we will have improved bound:

$$\text{err}_{\mathcal{D}}(l) < (1 + O(C)\eta^{1-2/k})\text{opt}_B(\mathcal{D}) + O(C)\eta^{1-2/k}\big(\mathop{\mathbf{E}}_{D}(y - \langle l^*, x\rangle)^k\big)^{2/k} + \epsilon$$

where $\mathcal{D}$ is assumed to be (C, k)-hypercontractive. We will work towards a proof of this improved bound. Theorem 6.3 follows naturally by setting $k = 4$.

6.2 High-level idea

Let $X = \{(x_i, y_i)\}_{i=1}^n$ be a set of uncorrupted samples from the underlying distribution $\mathcal{D}$ and $\mathcal{U} = \{(u_1, v_1)\}_{i=1}^n$ be an η-corruption of X. Let $\hat{D}$ be the empirical distribution of the uncorrputed dataset, i.e, the uniform distribution over X. If given $\mathcal{U}$, we can find a linear function $\hat{l}$ that has low error on $\hat{\mathcal{D}}$, then l will also have a low error on $\mathcal{D}$ with high probability by a standard generalization argument.

Robust Certifiability Lemma. Assuming we have unlimited computation power, then one reasonable thing to do is to do a brute force search over all subsets T of $\mathcal{U}$ such that $|T| \geq (1-\eta)|\mathcal{U}|$, fit a linear function over all such subsets T and output the one that has the smallest error on T. Since the uncorrupted set of points $X\backslash(X \cap U)$ must be a subset of some T, we naturally expect that the output of such brute-force procedures has low error bounds over $\hat{D}$. However, this is not always the case. We need a "robust certifiability lemma" which make it sufficient to find a subset T of size $\geq (1 - \eta)n$ and a linear function l such that the least squares error of l over T is small.

Relaxation. With the robust certifiability lemma, it suffices to find a set T and linear functional l described above. However, this problem is a non-convex quadratic optimization problem and having an efficient algorithm for this problem is thus difficult. To overcome this issue, we introduce variables $w_1, ..., w_n$ just as we did in the Gaussian mixture model in section 5.3. If we assume $(x, y) \sim \mathcal{D}$ and $\mathcal{D}$ is hypercontractive, then with high probability the empirical distribution $\hat{D}$ is also hypercontractive. Using this fact, we can adopt the following strategy: use $\mathcal{U}$ to find a subset T' of samples of size $\geq (1 - \eta)n$ and a linear function l such that:

1. l has small loss over T'.

2. The empirical distribution D' over T' is close to $\hat{D}$.

To achieve this goal, we consider the following optimization program:

$$\min_{w,l,X'} \frac{1}{n}\sum_{i=1}^{n}(y_i' - \langle l, x_i'\rangle) \text{ subject to}$$

$$\mathcal{A} = \begin{cases} w_i^2 = w_i \ \forall i \in [n] \\ \sum\limits_{i=1}^{n} w_i = (1 - \eta) \cdot n \\ w_i \cdot (u_i - x_i') = 0 \ \forall i \in [n] \\ w_i \cdot (v_i - y_i') = 0 \ \forall i \in [n] \end{cases}$$

Similar as in section 5.3, the optimization problem above is a relaxation of the original problem and it is clearly satifiable since we can simply let $w_i = 1$ if the i-th sample is uncorrupted and $w_i = 0$ otherwise. With our robust certifiability lemma (we will specify this in the next section), any solution to the relaxed optimization problem satisfies

$$\text{err}_{\mathcal{D}'}(l) \leq (1 + O(\sqrt{\eta})) \cdot \text{err}_{\mathcal{D}'}(l) + O(\sqrt{\eta}). \tag{6}$$

Still, this problem is a quadratic optimization problem and is NP-Hard to optimize in general. However, we do not need to exactly solve the system. It suffices to output a distribution or a *pseudodistribution*.

Pseudo-expectation and sum-of-squares proof. Let μ be a distribution over (w, l, X') that satisfies $\mathcal{A}$. Since equation 6 is satisfied for all solutions to the optimization problem, then we have

$$\mathop{\mathbf{E}}_{\mu}[\mathrm{err}_{\hat{D}}(l)] \leq (1 + O(\sqrt{\eta}))\mathrm{opt}_{\mu} + O(\sqrt{\eta}).$$

By convexity of the square loss, we have

$$\mathrm{err}_{\hat{\mathcal{D}}}(\mathop{\mathbf{E}}_{\mu}[l]) \leq \mathop{\mathbf{E}}_{\mu}[\mathrm{err}_{\hat{\mathcal{D}}}(l)] \leq (1 + O(\sqrt{\eta}))\mathrm{opt}_{\mu} + O(\sqrt{\eta}). \tag{7}$$

Consequently, if we can output a distribution-like object $\tilde{\mu}$ that satisfies inequality 7 above, then we can simply output $\mathbf{E}_{\tilde{\mu}}[l]$ as our desired output. The object we are considering here are pseudo-distributions introduced in section 5. Specifically, if we can obtain a low-degree sum-of-squares proof of equation 6, then based on 5.3, we can efficiently compute a pseudo-distribution under which equation 6 is satisfied. Finally, we can then output $\tilde{E}_{\tilde{\mu}}[l]$ as our final output with the desired guarantees. In the following sections we will work towards proofs of the two important intermediate steps:

1. The inequality 6 which is the conclusion of the robust certifiability lemma.
2. A low-degree sum-of-squares proof of inequality 7.

6.3 Robust Certifiability

In this section, we give a formal statement and a proof of the robust certifiability lemma. The lemma states the following.

Lemma 6.4. *(Robust Certifiability for l_2 Regression) Let $\mathcal{D}, \mathcal{D}'$ be distributions on $\mathbb{R}^d \times \mathbb{R}$ such that $\|\mathcal{D} - \mathcal{D}'\|_{TV} \leq \epsilon$ and the marginal distribution $\mathcal{D}_x$ of $\mathcal{D}$ on x is k-certifiably C-hypercontractive for some $C : [k] \to \mathbb{R}_+$ and for some even integer $k \geq 4$. Then for any $l, l^* \in \mathbb{R}^d$ and any η such that $2C(k/2)\eta^{1-2/k} < 0.9$, we have:*

$$err_{\mathcal{D}}(l) \leq (1 + O(C(k/2))\eta^{1-2/k}) \cdot err_{\mathcal{D}'}(l) + O(C(k/2))\eta^{1-2/k} \cdot \big(\mathop{\mathbf{E}}_{\mathcal{D}}[y - \langle l^*, x\rangle]^k\big)^{2/k}.$$

For the purpose of robust linear regression, it is helpful to think of $\mathcal{D}$ to be the uniform distribution over the uncorrupted samples X and think of $\mathcal{D}'$ as the distribution on the samples that serve as "good" samples, i.e. the certificates.

Proof. Fix some vector $l \in \mathbb{R}^d$. Let G be a coupling between $\mathcal{D}, \mathcal{D}'$. In other words, G is a joint distribution on (x, y) such that the marginal on (x', y') is $\mathcal{D}'$ and the marginal on (x, y) is $\mathcal{D}$. In addition, G satisfies the following condition: $\mathbf{P}_G 1\{(x, y) = (x', y')\} = 1 - \eta$.

Let $((x, y), (x', y')) \sim \mathcal{G}$ and write $1 = 1\{(x, y) = (x', y')\} + 1\{(x, y) \neq (x', y')\}$. We obtain the following:

$$\begin{aligned}
\mathop{\mathbf{E}}_{\mathcal{D}}[(y - \langle l, x\rangle)^2] &= \mathop{\mathbf{E}}_{\mathcal{G}}[1\{(x, y) = (x', y')\}(y - \langle l, x\rangle)^2] + \mathop{\mathbf{E}}_{\mathcal{G}}[1\{(x, y) \neq (x', y')\}(y - \langle l, x\rangle)^2] \\
&= \mathop{\mathbf{E}}_{\mathcal{G}}[1\{(x, y) = (x', y')\}(y' - \langle l, x'\rangle)^2] + \mathop{\mathbf{E}}_{\mathcal{G}}[1\{(x, y) \neq (x', y')\}(y - \langle l, x\rangle)^2] \\
&\leq \mathrm{err}_{\mathcal{D}'}(l) + \Big(\mathop{\mathbf{E}}_{\mathcal{G}}[1\{(x, y) \neq (x', y')\}^{k/(k-2)}]\Big)^{1-2/k}\Big(\mathop{\mathbf{E}}_{\mathcal{D}}[(y - \langle l, x\rangle)^k]\Big)^{2/k} \\
&\leq \mathrm{err}_{\mathcal{D}'}(l) + \eta^{1-2/k} \cdot \Big(\mathbf{E}[(y - \langle l, x\rangle)^k]\Big)^{2/k}.
\end{aligned} \tag{8}$$

where the second inequality substitutes in the condition $(x, y) = (x', y')$ in the first term. The first inequality comes from Hölder's inequality and the fact that

$$\mathop{\mathbf{E}}_{\mathcal{G}}[1\{(x, y) = (x', y')\}(y - \langle l, x\rangle)^2] \leq \mathop{\mathbf{E}}_{\mathcal{D}'}(y - \langle l, x\rangle)^2 = \mathrm{err}_{\mathcal{D}'}(l).$$

To bound $\|y - \langle l, x\rangle\|_k$, we apply Minkowski's inequality and get

$$\|y - \langle l, x\rangle\|_k \leq \|y - \langle l^*, x\rangle\|_k + \|y - \langle l - l^*, x\rangle\|_k.$$

On the other hand, by the hypercontractivity of $\mathcal{D}_X$, we get

$$\|\langle l - l^*, x\rangle\|_k \leq \sqrt{C(k/2)} \cdot \|\langle l - l^*, x\rangle\|_2.$$

In addition, by the triangle inequality, we have

$$\begin{aligned}\|\langle l - l^*, x\rangle\|_2 &\leq \|y - \langle l^*, x\rangle\|_2 + \|y - \langle l, x\rangle\|_2 \\ &\leq \|y - \langle l^*, x\rangle\|_k + \|y - \langle l, x\rangle\|_2\end{aligned}$$

Combining the inequalities above, we obtain

$$\|y - \langle l, x\rangle\|_k \leq (1 + \sqrt{C(k/2)})\|y - \langle l, x\rangle\|_k + \sqrt{C(k/2)}\|y - \langle l, x\rangle\|_2.$$

Now we can apply the standard inequality $(a+b)^2 \leq 2a^2 + 2b^2$ and the fact that $2(1+\sqrt{C(k/2)})^2 \leq 8C(k/2)$ and deduce that

$$\|y - \langle l, x\rangle\|_k^2 \leq 8C(k/2)\|y - \langle l^*, x\rangle\|_k^2 + 2C(k/2)\mathrm{err}_{\mathcal{D}}.$$

Substituting this upper bound into inequality 8, we have the following

$$\mathrm{err}_{\mathcal{D}} \leq \mathrm{err}_{\mathcal{D}'} + 8\eta^{1-2/k}C(k/2) \cdot \|y - \langle l^*, x\rangle\|_k^2 + 2\eta^{1-2/k}C(k/2)\mathrm{err}_{\mathcal{D}}.$$

Observe that $1/(1-2\eta^{1-2/k}C(k/2)) \leq 1+O(C(k/2))\eta^{1-2/k}$. Using this observation and rearranging the inequality, we obtain our final conclusion

$$\mathrm{err}_{\mathcal{D}}(l) \leq (1 + O(C(k/2))\eta^{1-2/k}) \cdot \mathrm{err}_{\mathcal{D}'}(l) + O(C(k/2))\eta^{1-2/k} \cdot \left(\mathop{\mathbf{E}}_{\mathcal{D}}[y - \langle l^*, x\rangle]^k\right)^{2/k}.$$

□

If instead of assuming hypercontractivity of linear functions, we assume hypercontractivity of polynomials, then we would obtain a similar generalization result for robust polynomial regressions. See Appendix A of [KKM20] for the exact statements.

6.4 Algorithm

Now we are ready to formally given the algorithm for robust linear regression. Let $\mathcal{D}$ denote the uncorrupted distribution on $\mathbb{R}^d \times \mathbb{R}$ and $\mathcal{D}_x$ denote the marginal distribution on x. Let $X = \{(x_i, y_i)\}_{i=1}^n$ denote the set of i.i.d uncorrupted samples from $\mathcal{D}$. For the purpose of our generalization argument, we will assume that all linear functions wer are considering have big complexity upper bounded by B. Let $\mathrm{opt}(\mathcal{D})$ denote the optimum least squares error of any linear function of bit complexity B on $\mathcal{D}$.

Let $\hat{\mathcal{D}}$ denote the uniform distribution over the uncorrupted samples X. Similar as above, we write $\mathrm{opt}(\hat{\mathcal{D}})$ as the optimum least squares error of any linear function with bit complexity bounded by B with respect to $\hat{\mathcal{D}}$. We will write $\mathcal{U} = \{(u_i, v_i)\}_{i=1}^n$ to denote an η-corruption of X under the strong contamination model. Note that our algorithm does not have access to $X, \mathcal{D}, \hat{\mathcal{D}}$. It only has access to $\mathcal{U}$. Lastly, given $l \in \mathbb{R}^d$, we define the *truncated linear function* as

$$l_M(x) = \begin{cases}\langle l, x\rangle & \text{if } |\langle l, x\rangle| \leq M \\ \mathrm{sign}(\langle l, x\rangle) \cdot M & \text{otherwise.}\end{cases}$$

Note that this truncated linear function is only used for generalization purposes as are often used even for regression without corruptions. As introduced in the earlier sections, in order to apply sum-of-squares techniques, we need to introduce variables $w_1, ..., w_n$ for each sample which serve as variables that we are optimizing over in the relaxed optimization problem. Intuitively, in the ideal case, $w_i = 0$ if the i-th sample if not corrupted and 0 otherwise. In this case, we can exactly identify the uncorrupted samples, fit a least-squares model and output a linear function that has low error over $\mathcal{D}$ by the robust certifiability lemma. As a relaxation of this condition, the following set of constraints in $w_i's, l, x_i'$ denoted as $\mathcal{P}_{\mathcal{U},\eta}$ must be satisfied:

$$\mathcal{P}_{\mathcal{U},\eta} = \begin{cases}\sum_{i=1}^n w_i = (1-\eta)n & \\ w_i^2 = w_i & \forall i \in [n]. \\ w_i \cdot (u_i - x_i') = 0 & \forall i \in [n]. \\ w_i \cdot (v_i - y_i') = 0 & \forall i \in [n].\end{cases}$$

Notice that for the last two constraints, if $w_i = 1$, we have $u_i = x'_i$ and $u_i \neq x'_i$ if $w_i = 0$. This corresponds to our desired behavior as w_i is intended to denote candidacy of the i-th point in the uncorrupted set. Now we are ready to describe the algorithm.

Algorithm 4 Algorithm for Robust l_2 Linear Regression via sum-of-squares

Given:

- η: A bound of the fraction of adversarial corruptions.
- $\mathcal{U}$: An η-corruption of a labeled sample X of size n sampled from a (C,k)-certifiably hypercontractive distribution $\mathcal{D}$.

Operation:

- Find a level-k pseudo-distribution $\tilde{\mu}$ that satisfies $\mathcal{P}_{\mathcal{U},\eta}$ and minimizes
$$\tilde{\mathbf{E}}_{\tilde{\mu}}\Big[\Big(\frac{1}{n}\sum_{i=1}^{n}(y'_i - \langle l, x'_i\rangle)^2\Big)^{k/2}\Big].$$
Denote this optimum value as $\mathrm{opt}_{\mathrm{alg}}$. Let $\widehat{\mathrm{opt}}_{SOS}$ be a positive real number such that $\widehat{\mathrm{opt}}_{SOS} = \mathrm{opt}_{\mathrm{alg}}^{2/k}$.
- Output $\hat{l} = \tilde{\mathbf{E}}_{\tilde{\mu}} l$.

We now state the main theorem in this section.

Theorem 6.5. *Let $\mathcal{D}$ be a distribution on $\mathbb{R}^d \times [-M, M]$ for some positive real number M such that the marginal on $\mathbb{R}^d$ is (C,k)-certifiably hypercontractive distribution. Let $opt_B(\mathcal{D}) = \min_l \mathbf{E}_{\mathcal{D}}[(y - \langle l, x\rangle)^2]$ where the minimum is taken over all $l \in \mathbb{R}^d$ with bounded bit complexity B. Let l^* be such a minimizer.*

Fix any even $k \geq 4$ and any $\epsilon > 0$. Let X be an i.i.d. sample from $\mathcal{D}$ of size $n \geq n_0 = poly(d^k, B, M, 1/\epsilon)$. Then, with probability at least $1 - \epsilon$ over the distribution of X, given any η-corruption $\mathcal{U}$ of X and η as input, there is a polynomial time algorithm (Algorithm 4) that outputs a $l \in \mathbb{R}^d$ such that for $C = C(k/2)$,

$$err_{\mathcal{D}}(l_M) < (1 + O(C)\eta^{1-2/k})opt_B(\mathcal{D}) + O(C)\eta^{1-2/k}\Big(\mathop{\mathbf{E}}_{\mathcal{D}}[(y - \langle l^*, x\rangle)^k]\Big)^{2/k} + \epsilon.$$

As mentioned above, the bounded-label assumption and the bounded-bit-complexity are present mainly to obtain generalization results for linear regression. By setting $k = 4$, we obtain Theorem 6.3.

6.5 Analysis of algorithm

In this section, we give a proof of the main theorem (Theorem 6.5). As explained in section 6.2, the proof is divided into main components. In part (1), we will show that the output of the optimization problem has low-error (*optimization error*) guarantees over the empirical distribution. In part (2), we will show that the model generalizes well in a sense that a low-error bound over the empirical distribution $\hat{\mathcal{D}}$ implies a low-error bound over the true distribution $\mathcal{D}$. Specifically, we will work towards proofs of the following two lemmas. Let $\widehat{\mathrm{opt}}_k = (1/n)\sum_{i=1}^{n}((y_i - \langle l^*, x\rangle)^k)^{2/k}$ and $\mathrm{opt}_k(\mathcal{D}) = \mathbf{E}_{(x,y)\sim\mathcal{D}}[(y - \langle l^*, x\rangle)^k]^{2/k}$. Then, we have

Lemma 6.6. *(Bounding the optimization error) Under the assumptions of Theorem 6.5, with probability at least $1 - \epsilon$, we have*

$$err_{\hat{\mathcal{D}}}(\hat{l}) \leq (1 + C(k/2)\eta^{1-2/k}) \cdot \widehat{opt}_{SOS} + O(C(k/2)) \cdot \eta^{1-2/k} \cdot \widehat{opt}_k.$$

Lemma 6.7. *(Bounding the generalization error) Under the assumptions of Theorem 6.5, with probability at least $1 - \epsilon$, we have*

1. *$\widehat{opt}_{SOS} \leq opt(\mathcal{D}) + \epsilon$.*

2. $err_{\mathcal{D}}(\hat{l}_M) \le err_{\hat{\mathcal{D}}}(\hat{l}) + \epsilon$.

Before we give a proof of these two lemmas, let's see how we can prove Theorem 6.5 assuming they are true. To this end, we need one more lemma.

Lemma 6.8. *For every distribution $\mathcal{D}$ on $\mathbb{R}^d \times \mathbb{R}$ such that $\nu = \mathbf{E}_{\mathcal{D}}(y - \langle l^*, x\rangle)^k < \infty$, there exists a distribution $\mathcal{F}$ such that $\|\mathcal{D} - \mathcal{F}\|_{TV} \le \eta$ and $(y - \langle l^*, x\rangle)^k$ is absolutely bounded in the support of $\mathcal{F}$ by ν/η.*

Proof. (of Lemma 6.8) Set $\mathcal{F} = \mathcal{D}|(y - \langle l^*, x\rangle)^k \le \nu/\eta)$. It is easy to check that $\mathcal{F}$ satisfies the desired properties. □

Notice that an η-corrupted sample of $\mathcal{D}$ can be regarded as an 2η-corrupted sample of $\mathcal{F}$. Thus, we can use Hoefdding bound for concentration to show the convergence of the expectation of $(y - \langle l^*, x\rangle)^k$ since $(y - \langle l^*, x\rangle)^k$ is bounded in $\mathcal{F}$. With this in hand, we are now ready to give a proof of Theorem 6.5.

Proof. (of Theorem 6.5) Recall that X is a set of i.i.d samples from $\mathcal{D}$ of size n and $\hat{D}$ is the empirical distribution over X. Let $\nu = \mathbf{E}_{\mathcal{D}}(y - \langle l^*, x\rangle)^k < \infty$. Then by Lemma 6.8, $(y - \langle l^*, x\rangle)^k < \infty$ is absolutely bounded in $\mathcal{F}$ which allows us to apply Hoeffding's bound. We have that if $n \ge \nu \log(1/\delta)/\eta\epsilon^2$, then with probability at least $1 - \epsilon$, the following holds

$$\widehat{opt}_k = \mathop{\mathbf{E}}_{\hat{\mathcal{D}}}[(y - \langle l^*, x\rangle)^k] \le \mathop{\mathbf{E}}_{\mathcal{D}}(y - \langle l^*, x\rangle)^k + \epsilon = \mathrm{opt}_k + \epsilon.$$

Now, by the inequality above and Lemma 6.7, 6.6, we have that with probability at least $1 - O(\epsilon)$,

$$\mathrm{err}_{\mathcal{D}}(l_M) \le (1 + O(C)\eta^{1-2/k}) \cdot \mathrm{opt}(\mathcal{D}) + O(C)\eta^{1-2/k} \cdot \mathrm{opt}_k + O(C\epsilon).$$

The theorem now follows with a proper choice of ϵ. □

6.5.1 Bounding the optimization error

In this section, we give a proof of Lemma 6.6. The high-level idea is to give a sum-of-squares proof of the error bound implied by the robust certifiability lemma 6.4. Having this, we would obtain a pseudo-distribution $\tilde{\mu}$ over the variables of interest via sum-of-squares optimization. Then by convexity of l_2-loss, we can conclude the lemma with the output $\tilde{\mathbf{E}}_{\tilde{\mu}}[l]$. In particular, let (w, l, X') satisfy the inequalities $\mathcal{P}_{\mathcal{U},\eta}$. Then applying Lemma 6.4 to $\hat{\mathcal{D}}$ and the uniform distribution over X', we have

$$\mathrm{err}_{\hat{\mathcal{D}}}(l) \le (1 + cC\eta^{1-2/k})\Big(\frac{1}{n}\sum_{i=1}^{n}(y_i' - \langle l, x_i'\rangle)^2\Big) + cC\eta^{1-2/k} \cdot \widehat{\mathrm{opt}}_k$$

for some universal constant $c > 0$. In order to give a sum-of-squares proof of the statements, we need to rewrite this in terms of polynomials of (w, l, X'). To this end, for simplicity, let $\mathrm{err}(w, l, X') = (1/n)\sum_{i=1}^{n}(y_i' - \langle l, x_i'\rangle)^2$. Then we can rewrite the inequality above as

$$(\mathrm{err}_{\hat{\mathcal{D}}}(l) - \mathrm{err}(w, l, X'))^{k/2} \le \eta^{k/2-1} \cdot 2^{\Theta(k)} C^k \mathrm{err}(w, l, X')^{k/2} + \eta^{k/2-1} \cdot 2^{\Theta(k)} C^k \cdot \widehat{\mathrm{opt}}_k^{k/2}.$$

We will give a sum-of-squares of this inequality using the polynomial system $\mathcal{P}_{\mathcal{U},\eta}$. By the sum-of-squares optimization algorithm, we can output a pseudo-distribution $\tilde{\mu}$ that satisfies the inequality above. After some rearranging and simplification, we arrive at the following inequalities satisfied by $\tilde{\mu}$,

$$\tilde{\mathbf{E}}[\mathrm{err}_{\hat{\mathcal{D}}}(l)] \le (1 + cC\eta^{1-2/k}) \cdot \widehat{\mathrm{opt}}_{SOS} + cC\eta^{1-2/k}\widehat{\mathrm{opt}}_k.$$

Finally, by convexity of the l_2 loss and Fact 5.12, we can then conclude that

$$\mathrm{err}_{\hat{\mathcal{D}}}(\tilde{\mathbf{E}}_{\tilde{\mu}}[l]) \le \tilde{\mathbf{E}}_{\tilde{\mu}}[\mathrm{err}_{\hat{\mathcal{D}}}(l)] \le (1 + cC\eta^{1-2/k}) \cdot \widehat{\mathrm{opt}}_{SOS} + cC\eta^{1-2/k}\widehat{opt}_k,$$

and hence finishing the proof.

Here we present the main component of this line of arguments, namely the sum-of-squares proof of the error bound. Interested readers can refer to section 5.2.1 for how to formally conclude Lemma 6.6 using the lemma below.

Lemma 6.9. *(SOS proof of Robust Certifiability of Regression Hypothesis) Let X be a collection of n labels in $\mathbb{R}^d \times \mathbb{R}$ such that $\hat{\mathcal{D}}$, the uniform distribution on $x_1, ..., x_n$ is k-certifiably hypercontractive and all the labels $y_1, ..., y_n$ are bounded in $[-M, M]$. Let $\mathcal{U}$ be an η-corruption of X. Let (w, l, X') satisfy the set of constraints $\mathcal{P}_{\mathcal{U},\eta}$. Let $err_{\hat{\mathcal{D}}}(l)$ be the quadratic polynomial $\mathbf{E}_{(x,y)\sim\hat{\mathcal{D}}}(y - \langle l, x\rangle)^2$ in vector-valued variable l. Let $err(w, l, X')$ be the polynomial $(1/n)\sum_{i=1}^n (y_i' - \langle l, x_i'\rangle)^2$ in vector-valued variables $w, l, x_1', ..., x_n'$.*

Then for any $l^* \in \mathbb{R}^d$ of big complexity at most $B < poly(n, d^k)$, $C = C(k/2)$ and any η such that $100C\eta^{1-2/k} < 0.9$,

$$\mathcal{P}_{\mathcal{U},\eta} \vdash_{k}^{l} (err_{\hat{\mathcal{D}}}(l) - err(w, l, X'))^{k/2} \leq \eta^{k/2-1}2^{\Theta(k)}C^k err(w, l, X')^{k/2} + \eta^{k/2-1}2^{\Theta(k)}C^k\Big(\frac{1}{n}\sum_{i=1}^n (y_i - \langle l^*, x_i\rangle)^k\Big) \tag{9}$$

Proof. Let $w' \in \{0, 1\}^n$ be given by $w_i' = w_i$ if and only if the ith sample is uncorrupted in $\mathcal{U}$ and 0 otherwise. Let $\mathcal{D}'$ denote the empirical distribution weighted by w'. Notice that $\sum_i w_i' \geq (1 - 2\eta)n$. Therefore, we have

$$\vdash_{2}^{w'} \Big\{\frac{1}{n}\sum_i (1 - w_i')^2 \leq 2\eta\Big\}.$$

Let $\mathrm{err}_{w'}(l) = (1/n)\sum_{i=1}^n w_i'(v_i - \langle l, u_i\rangle)^2$. Then we have:

$$\vdash_{4}^{w',l} \mathrm{err}_{\hat{\mathcal{D}}}(l) = \frac{1}{n}\sum_{i=1}^n w_i'(y_i - \langle l, x_i\rangle)^2 + \frac{1}{n}\sum_{i=1}^n (1 - w_i')(y_i - \langle l, x_i\rangle)^2$$

On the other hand, we also have

$$\vdash_{4}^{w,l} \frac{1}{n}\sum_{i=1}^n w_i'(y_i - \langle l, x_i\rangle)^2 \leq \sum_{i=1}^n (y_i' - \langle l, x_i'\rangle)^2 = \mathrm{err}_{\mathcal{D}'}(l).$$

□

Using these two observations together with sum-of-squares Hölder's inequality, we have

$$\begin{aligned}
\vdash_{k}^{w,l} (\mathrm{err}_{\hat{\mathcal{D}}} - \mathrm{err}_{\mathcal{D}'}(l))^{k/2} &= \Big(\frac{1}{n}\sum_{i=1}^n w_i'(y_i - \langle l, x_i\rangle)^2 + \frac{1}{n}\sum_{i=1}^n (1 - w_i')(y_i - \langle l, x_i\rangle)^2 - \mathrm{err}_{\mathcal{D}'}(l)\Big)^{k/2} \\
&\leq \Big(\frac{1}{n}\sum_{i=1}^n (1 - w_i')(y_i - \langle l, x_i\rangle)^2\Big)^{k/2} \\
&\leq \Big(\frac{1}{n}\sum_{i=1}^n (1 - w_i')\Big)^{k/2-1}\Big(\frac{1}{n}\sum_{i=1}^n (y_i - \langle l, x_i\rangle)^k\Big) \\
&\leq 2^{k/2-1}\eta^{k/2-1}\Big(\frac{1}{n}\sum_{i=1}^n (y_i - \langle l, x_i\rangle)^k\Big)
\end{aligned} \tag{10}$$

where the first inequality uses the two observations, the second inequality comes from sum-of-squares Hölder's inequality and the last inequality results from the fact that $\sum_i w_i' \geq (1 - 2\eta)n$. Now, by the sum-of-squares triangle inequality, we have

$$\vdash_{k}^{l} \Big\{\Big(\frac{1}{n}\sum_{i=1}^n (y_i - \langle l, x_i\rangle)^k\Big) \leq 2^k\Big(\frac{1}{n}\sum_{i=1}^n (y_i - \langle l^*, x_i\rangle)^k\Big) + 2^k\Big(\frac{1}{n}\sum_{i=1}^n (\langle l - l^*, x_i\rangle)^k\Big)\Big\}. \tag{11}$$

By certifiably hypercontractivity of the marginal distribution $\mathcal{D}_x$, we have

$$\vdash_{k}^{l} \Big\{\Big(\frac{1}{n}\sum_{i=1}^n (\langle l - l^*, x_i\rangle)^k\Big) \leq C(k)^{k/2}\Big(\frac{1}{n}\sum_{i=1}^n (\langle l - l^*, x_i\rangle)^2\Big)^{k/2}\Big\}.$$

Applying sum-of-squares triangle inequality again, we have

$$\vdash_{k}^{l} \Big(\frac{1}{n}\sum_{i=1}^n (\langle l - l^*, x_i\rangle)^2\Big)^{k/2} \leq 2^{k/2}\Big(\frac{1}{n}\sum_{i=1}^n (y_i - \langle l, x_i\rangle)^2\Big)^{k/2} + 2^{k/2}\Big(\frac{1}{n}\sum_{i=1}^n (y_i - \langle l^*, x_i\rangle)^2\Big)^{k/2}.$$

Again, by sum-of-squares Hölder's inequality, we have

$$\vdash_k^l \Big\{ \Big(\frac{1}{n} \sum_{i=1}^n (y_i - \langle l^*, x_i \rangle)^2 \Big)^{k/2} \leq \frac{1}{n} \sum_{i=1}^n (y_i - \langle l^*, x_i \rangle)^k \Big\}.$$

Combining this with 11, we obtain

$$\vdash_k^l \Big\{ \Big(\frac{1}{n} \sum_{i=1}^n (y_i - \langle l^*, x_i \rangle)^k \Big) \leq O(C(k/2))^k \Big(\frac{1}{n} \sum_{i=1}^n (y_i - \langle l^*, x_i \rangle)^2 \Big) + O(C(k/2))^k \Big(\frac{1}{n} \sum_{i=1}^n (y_i - \langle l, x_i \rangle)^2 \Big)^{k/2} \Big\}. \quad (12)$$

Substituting this into 10, we have

$$\vdash_k^l (\mathrm{err}_{\hat{\mathcal{D}}}(l) - \mathrm{err}_{\mathcal{D}'}(l))^{k/2} \leq \eta^{k/2-1} \cdot O(C(k/2))^k (\mathrm{err}_{\hat{\mathcal{D}}}(l))^{k/2} + \eta^{k/2-1} \cdot O(C(k/2))^k \Big(\frac{1}{n} \sum_{i=1}^n (y_i - \langle l^*, x_i \rangle)^k \Big). \quad (13)$$

With the following alternate form of the sum-of-squares triangle inequality in hand $\delta^k a^k \leq (2\delta)^k (a-b)^k + (2\delta)^k b^k$ for any a, b and even k, applying this inequality with $a = \mathrm{err}_{\hat{\mathcal{D}}}(l), b = \mathrm{err}_{\mathcal{D}'}(l)$ and $\delta = \eta^{k/2-1} \cdot O(C(k/2))^k$ and rearranging, we get

$$\vdash_k^l (1-\delta)(\mathrm{err}_{\hat{\mathcal{D}}}(l) - \mathrm{err}_{\mathcal{D}'}(l))^{k/2} \leq \eta^{k/2-1} \cdot O(C(k/2))^k (\mathrm{err}_{\mathcal{D}'}(l))^{k/2} + \eta^{k/2-1} \cdot O(C(k/2))^k \Big(\frac{1}{n} \sum_{i=1}^n (y_i - \langle l^*, x_i \rangle)^k \Big). \quad (14)$$

For $\delta < 0.9$, this suggests

$$\vdash_k^l (\mathrm{err}_{\hat{\mathcal{D}}}(l) - \mathrm{err}_{\mathcal{D}'}(l))^{k/2} \leq \eta^{k/2-1} \cdot O(C(k/2))^k (\mathrm{err}_{\mathcal{D}'}(l))^{k/2} + \eta^{k/2-1} \cdot O(C(k/2))^k \Big(\frac{1}{n} \sum_{i=1}^n (y_i - \langle l^*, x_i \rangle)^k \Big). \quad (15)$$

This concludes the proof of Lemma 6.9.

6.5.2 Bounding the generalization error

In this section, we give a proof of the first part of Lemma 6.7. The second part follows from standard generalization bound which we will not go over here. As a reminder, we are trying to prove that $\widehat{\mathrm{opt}}_{\mathrm{SOS}} \leq \mathrm{opt}(\mathcal{D}) + \epsilon$.

Proof. Suppose l^* is a linear function of bit complexity bounded by B that achieves the optimum least squares regression error on $\mathcal{D}$. We first show that $\widehat{\mathrm{opt}}_{\mathrm{SOS}} \leq \mathrm{err}_{\hat{\mathcal{D}}}(l^*)$ by giving a feasible pseudo-distribution that achieves this property. Let $\tilde{\mu}$ be supported on the set of (w, l^*, X') such that $w_i = 1$ if $(x_i, y_i) = (u_i, v_i)$ and 0 otherwise and $(x_i', y_i') = (x_i, y_i)$ for all $i \in [n]$. In other words, $w_i = 0$ if an only if the i-th sample is uncorrupted. It is easy to see that $\tilde{\mu}$ satisfies the set of polynomial constraints $\mathcal{P}_{\mathcal{U},\eta}$. Moreover, we have

$$\widehat{\mathrm{opt}}_{\mathrm{SOS}}^{k/2} \leq \tilde{\mathbf{E}}_{\tilde{\mu}}[\mathrm{err}(w, l, X)^{k/2}] = \Big(\frac{1}{n} \sum_{i=1}^n (y_i - \langle l^*, x_i \rangle)^2 \Big)^{k/2} = \mathrm{err}_{\hat{\mathcal{D}}}(l^*)^{k/2}.$$

This suggests that $\widehat{\mathrm{opt}}_{\mathrm{SOS}} \leq \mathrm{err}_{\hat{\mathcal{D}}}(l^*)$. The next step is to show that $\mathrm{err}_{\hat{\mathcal{D}}}(l^*)$ is sufficiently close to $\mathrm{err}_{\mathcal{D}}(l^*)$ for n large enough. Consider the random variable $Z = (y - \langle l^*, x \rangle)^2$ where $(x, y) \sim \mathcal{D}$. Notice that $\mathrm{err}_{\hat{\mathcal{D}}}(l^*) = (1/n)Z$ and $\mathbf{E}[\mathrm{err}_{\mathcal{D}}(l^*)] = \mathrm{opt}(\mathcal{D})$. The second moment of Z is given by

$$\mathbf{E}[Z^2] = \mathbf{E}[(y - \langle l^*, x \rangle)^4] \leq 2\,\mathbf{E}[y^4] + 2\,\mathbf{E}[\langle l^*, x \rangle^4] \leq 2M^4 + 2C^2(\mathbf{E}[\langle l^*, x \rangle^2])^2,$$

where the last inequality comes from the boundedness assumption on the labels y_i and hypercontractivity of x. On the other hand

$$\mathbf{E}[\langle l^*, x \rangle^2]) \leq 2\,\mathbf{E}[(y - \langle l^*, x \rangle)^2] + 2\,\mathbf{E}[y^2] \leq 2\mathrm{opt}(\mathcal{D}) + 2M^2 \leq 4M^2.$$

The last inequality comes from the fact that $\mathrm{opt}(\mathcal{D}) \le M^2$. This is true since the 0 function satisfies that $\mathrm{opt}(\mathcal{D}) \le \mathbf{E}[(y_i - \langle 0, x\rangle)^2] \le M^2$. Substitute this into the second moment bound above, we know that $\mathbf{E}[Z^2] = O(M^4)$. By Chebyshev's inequality, we know that if we have $n \ge n_0$ independent samples where $n_0 = O(1/\epsilon^3) \cdot M^4$, then

$$\mathbf{P}\Big[\Big\|\frac{1}{n}\sum_{i=1}^{n} Z_i - \mathbf{E}[Z_i]\Big\| \ge \epsilon\Big] \le \epsilon.$$

Hence, we can then conclude that $\mathrm{err}_{\hat{\mathcal{D}}}(l^*) \le \mathrm{opt}(\mathcal{D}) + \epsilon$ with probability at least $1-\epsilon$. The lemma then follows. □

6.6 Robust algorithm for L1 regression

In this section, we state without proof the algorithm for robust l_1 regression. Interested readers can find proofs of the statements in Section 5.3 of [KKM20]. First, we need the following robust certifiability lemma for l_1 regression.

Lemma 6.10. *(Robust certifiability for l_1 regression) Let $\mathcal{D}, \mathcal{D}'$ be two distributions on $\mathbb{R}^d \times \mathbb{R}$ with marginals D, D' on $\mathbb{R}^d$ respectively. Suppose $\|\mathcal{D} - \mathcal{D}'\|_{TV} \le \eta$ and that the ratio of the largest to the smallest eigenvalue of the 2nd moment matrix of D is at most κ. Then for any $l, l^* \in \mathbb{R}^d$ such that $\|l^*\|_2 \ge \|l\|_2$, we have*

$$\mathop{\mathbf{E}}_{\mathcal{D}} |\langle l, x\rangle - y| \le \mathop{\mathbf{E}}_{\mathcal{D}'} |\langle l, x\rangle - y| + 2\kappa^{1/2}\eta^{1/2}\sqrt{\mathop{\mathbf{E}}_{\mathcal{D}} y^2} + 2\kappa^{1/2}\eta^{1/2} \cdot \sqrt{\mathop{\mathbf{E}}_{\mathcal{D}}(y - \langle l^*, x\rangle)^2}.$$

Similar as in the l_2 regression case, we need to give a low-degree sum-of-squares proof of the certifiability result. To this end, we need our variables w_i's to satisfy the following polynomial system.

$$\mathcal{A}_{\mathcal{U},\eta,Q} = \begin{cases} \sum_{i=1}^{n} w_i = (1-\eta)\cdot n & \\ w_i^2 = w_i & \forall i \in [n] \\ w_i \cdot (u_i - x_i') = 0 & \forall i \in [n] \\ w_i \cdot (v_i - y_i') = 0 & \forall i \in [n] \\ \tau_i' \ge (y_i' - \langle l, x_i'\rangle) & \forall i \in [n] \\ \tau_i' \ge -(y_i' - \langle l, x_i'\rangle) & \forall i \in [n] \\ \|l\|_2^2 \le Q^2 & \end{cases}$$

where the conditions on τ_i' serves as constraints that enforce the absolute value in l_1 regression and the last condition ensures the l_2 norm of l is bounded. Now we are ready to state the algorithm for robust l_1 regression.

Algorithm 5 Algorithm for Robust l_1 Regression via sum-of-squares

Given: An η-corruption $\mathcal{U}$ of a labeled sample X of size n from an arbitrary distribution $\mathcal{D}$. Q, the Euclidean norm of the best fitting l_1 regression hypothesis for $\mathcal{D}$.
Operation:

```
1. Find a level-4 pseudo-distribution $\tilde{\mu}$ that satisfies $\mathcal{A}_{\mathcal{U},\eta,Q}$ and minimizes $((1/n)\sum_{i=1}^{n} \tau_i)^2$.
2. Return $\hat{l} = \tilde{\mathbf{E}}_{\tilde{\mu}} l$.
```

The algorithm has the following guarantee.

Theorem 6.11. *Let $\mathcal{D}$ be an arbitrary distribution on $\mathbb{R}^d \times \mathcal{Y}$ for $\mathcal{Y} \subseteq [-M, M]$ for a positive real M. Let κ be the ratio of the maximum to the minimum eigenvalue of the covariance matrix of $\mathcal{D}$, the marginal of $\mathcal{D}$ on x. Let $opt(\mathcal{D})$ be the minimum of $\mathbf{E}_{\mathcal{D}} |y - \langle l, x\rangle|$ over all l that has bit complexity bounded by B. Let l^* be any such minimizer and $\eta > 0$ be an upper bound on the fraction of corruptions. For any $\epsilon > 0$, let X be an i.i.d. sample from $\mathcal{D}$ of size $n \ge n_0$ for some $n_0 = O(1/\epsilon^2) \cdot (M^2\|l^*\|_2^4 + d\log(d)\|\Sigma\|/\eta)$.*

Then, with probability at least $1-\epsilon$ over the draw of the sample X, given any η-corruption $\mathcal{U}$ of X and η as input, Algorithm 5 outputs a function $f : \mathbb{R}^d \to \mathbb{R}$ such that:

$$\mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}} |y - f(x)| < opt(\mathcal{D}) + O(\sqrt{\kappa\eta})\Big(\sqrt{\mathop{\mathbf{E}}_{\mathcal{D}} y^2} + \sqrt{(\mathop{\mathbf{E}}_{\mathcal{D}}(y - \langle l^*, x\rangle)^2)}\Big) + \epsilon.$$

One thing to notice is that both of the algorithms rely heavily on the fact the loss function can be written as some polynomial of the variables. In order for the algorithms to work intended, we also require the convexity of the loss function in order to pull the pseudo-expectation operator inside. As a result, whether this sum-of-squares based algorithm can be adapted to solve problems that has a either non-polynomial or non-convex loss function is still unknown.

One example of such problem is logistic regression which is a special example of generalized linear model. In this case, loss function is the cross-entropy loss which takes the form of $l(\theta) = -y\log(\theta^T x) + (1-y)\log(1-\theta^T x)$ where $\theta \in \mathbb{R}^d$ is the predicted linear function. This loss function is convex but is not a polynomial of the input parameter θ. Hence, we cannot directly apply our sum-of-squares-based algorithm to this case. Some possible solutions include low-order Taylor expansions and a different choice of loss function. These are interesting problems to consider.

7 Conclusion

In this paper, we examined two techniques that have played important roles in recent progresses in the field of robust statistics. Filtering approaches are more intuitive than sum-of-squares-based methods. It is applicable to a wider range of problems at the expense of the requirement for stronger assumptions. On the other hand, sum-of-squares methods make use of a general framework that can be employed to problems with polynomial and convex loss functions. It does not require strong assumptions on the underlying distribution and can achieve near-optimal bounds in many cases. However, it relies heavily on the polynomial and the convexity properties of the loss function, making it hard to adapt to problems such as logistic regression. Below we post some potentially interesting open questions in this area.

- Can we improve SEVER so that it requires weaker assumptions on the data?
- Both SEVER and sum-of-squares robust linear regression method has sub-optimal dependence on the corruption constant η, namely $O(\sqrt{\epsilon})$. Is it possible to achieve $O(\eta)$-dependence on the corruption level?
- Can we adapt sum-of-squares methods to problems with non-polynomial loss functions?
- Can we apply filtering-based methods to nonparametric regression for certain structured function classes (e.g. monotone, Lipschitz, piece-wise)?
- Can we give some theoretical guarantees of SEVER on single-layer or two-layer neural networks?

References

[And08] Robert Andersen, *Modern methods for robust regression*, Sage Publications, 2008.

[Ber06] T. Bernholt, *Robust estimators are hard to compute*, 2006.

[BJK15] Kush Bhatia, Prateek Jain, and Purushottam Kar, *Robust regression via hard thresholding*, Jun 2015.

[BJKK18] Kush Bhatia, Prateek Jain, Parameswaran Kamalaruban, and Purushottam Kar, *Consistent robust regression*, Oct 2018.

[BKNS00] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander, *Lof: identifying density-based local outliers*, ACM SIGMOD Record **29** (2000), no. 2, 93–104.

[BKS14] Boaz Barak, Jonathan A. Kelner, and David Steurer, *Dictionary learning and tensor decomposition via the sum-of-squares method*, Nov 2014.

[BS16] Boaz Barak and David Steurer, *Proofs, beliefs, and algorithms through the lens of sum-of- squares*, 2016, Lecture notes in preparation, available on `http://sumofsquares.org`.

[CDG18] Yu Cheng, Ilias Diakonikolas, and Rong Ge, *High-dimensional robust mean estimation in nearly-linear time*, Nov 2018.

[CDGS20] Yu Cheng, Ilias Diakonikolas, Rong Ge, and Mahdi Soltanolkotabi, *High-dimensional robust mean estimation via gradient descent*, May 2020.

[CGR17] Mengjie Chen, Chao Gao, and Zhao Ren, *Robust covariance and scatter matrix estimation under huber's contamination model*, 2017.

[CHK$^+$19] Yeshwanth Cherapanamjeri, Samuel B. Hopkins, Tarun Kathuria, Prasad Raghavendra, and Nilesh Tripuraneni, *Algorithms for heavy-tailed statistics: Regression, covariance estimation, and beyond*, 2019.

[Din21] Peng Ding, *Linear model and extensions*, 2021, Lecture notes from Stat 230A Linear Models.

[DK19] Ilias Diakonikolas and Daniel M. Kane, *Recent advances in algorithmic high-dimensional robust statistics*, Nov 2019.

[DKK$^+$16] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Ankur Moitra, and Alistair Stewart, *Robust estimators in high dimensions without the computational intractability*, 2016.

[DKK$^+$17] Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart, *Being robust (in high dimensions) can be practical*, 2017.

[DKK$^+$19] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart, *Sever: A robust meta-algorithm for stochastic optimization*, May 2019.

[DKS17] Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart, *List-decodable robust mean estimation and learning mixtures of spherical gaussians*, Nov 2017.

[DMR22] Luc Devroye, Abbas Mehrabian, and Tommy Reddad, *The total variation distance between high-dimensional gaussians with the same mean*, Feb 2022.

[FB87] Martin A. Fischler and Robert C. Bolles, *Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography*, Readings in Computer Vision (1987), 726–740.

[GKKW02] Laszalo Gyorfi, Michael Kohler, Adam Krzyzak, and Harro Walk, *A distribution-free theory of nonparametric regression*, 2002.

[GLS81] M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica **1** (1981), no. 2, 169–197.

[Ham86] Frank R. Hampel, *Robust statistics: The approach based on influence functions*, Wiley, 1986.

[HL17] Samuel B. Hopkins and Jerry Li, *Mixture models, robustness, and sum of squares proofs*, Nov 2017.

[HM13] Moritz Hardt and Ankur Moitra, *Algorithms and hardness for robust subspace recovery*, Dec 2013.

[Hop18] Sam Hopkins, *Clustering and sum of squares proofs*, 2018.

[HR11] Peter J. Huber and eauthor Ronchetti, Elvezio M., *Robust statistics*, Wiley, 2011.

[Hub64] Peter J. Huber, *Robust estimation of a location parameter*, The Annals of Mathematical Statistics **35** (1964), no. 1, 73–101.

[JLM08] H. Tang A.M. Southwick A.M. Casto S. Ramachandran H.M. Cann G.S. Barsh M. Feldman L.L. Cavalli-Sforza J.Z. Li, D.M. Absher and R.M. Myers, *Worldwide human relationships inferred from genome-wide patterns of variation*, Science (2008).

[JP78] D.S. Johnson and F.P. Preparata, *The densest hemisphere problem*, Theoretical Computer Science **6** (1978), no. 1, 93–107.

[KKM20] Adam Klivans, Pravesh K. Kothari, and Raghu Meka, *Efficient algorithms for outlier-robust regression*, Jun 2020, `https://arxiv.org/abs/1803.03241`.

[KLS09] Adam R. Klivans, Philip M. Long, and Rocco A. Servedio, *Learning halfspaces with malicious noise*, Automata, Languages and Programming (2009), 609–621.

[KS17] Pravesh K. Kothari and Jacob Steinhardt, *Better agnostic clustering via relaxed tensor norms*, Nov 2017.

[Las01] Jean B. Lasserre, *New positive semidefinite relaxations for nonconvex quadratic programs*, Nonconvex Optimization and Its Applications (2001), 319–331.

[Li19] Jerry Li, *Slides from the efficient algorithms for high dimensional robust learning talk*, Microsoft Research (2019).

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

[Mor07] Stephan Morgenthaler, *A survey of robust statistics*, 2007.

[MSS16] Tengyu Ma, Jonathan Shi, and David Steurer, *Polynomial-time tensor decompositions with sum-of-squares*, 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS) (2016).

[Nes00] Yurii Nesterov, *Squared functional systems and optimization problems*, Applied Optimization (2000), 405–440.

[NRF02] J. Weber H. Cann K. Kidd L.A. Zhivotovsky N. Rosenberg, J. Pritchard and M.W. Feldman, *Genetic structure of human populations*, Science (2002).

[Owe07] Art B. Owen, *A robust hybrid of lasso and ridge regression*, Prediction and Discovery (2007), 59–71.

[PAL17] M. F. Balcan P. Awasthi and P. M. Long, *The power of localization for efficiently learning linear separators with noise*, Feb 2017.

[Par00] *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*, Ph.D. thesis (2000).

[RD99] Peter J. Rousseeuw and Katrien Van Driessen, *A fast algorithm for the minimum covariance determinant estimator*, Technometrics **41** (1999), no. 3, 212–223.

[SBS17] J. Li S. Balakrishnan, S. S. Du and A. Singh, *Computationally efficient robust sparse estimation in high dimensions*, 2017, pp. 169–212.

[SCV17] Jacob Steinhardt, Moses Charikar, and Gregory Valiant, *Resilience: A criterion for learning in the presence of arbitrary outliers*, 2017.

[Sho87] N. Z. Shor, *Quadratic optimization problems*, Izv. Akad. Nauk SSSR Tekhn. Kibernet (1987), no. 1, 128–139, MR 939596.

[Ste21] Jacob Steinhardt, *Lecture notes for stat240 (robust and nonparametric statistics)*.

[Tuk60] J. W. Tukey, *A survey of sampling from contaminated distributions*, Contributions to probability and statistics (1960), 2:448–485.

[Tuk75] *Mathematics and picturing of data*, 1975, pp. 523–531.

[Ver18] Roman Vershynin, *High-dimensional probability: An introduction with applications in data science*, Cambridge University Press, 2018.

[Wai19] Martin Wainwright, *High-dimensional statistics: A non-asymptotic viewpoint*, Cambridge University Press, 2019.

[ZJS20] Banghua Zhu, Jiantao Jiao, and Jacob Steinhardt, *When does the tukey median work?*, Mar 2020.