List-Decodable Sparse Mean Estimation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Robust mean estimation is one of the most important problems in statistics: given a set of samples $\{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ where an α fraction are drawn from some distribution D and the rest are adversarially corrupted, it aims to estimate the mean of D. A surge of recent research interest has been focusing on the list-decodable setting where $\alpha \in (0, \frac{1}{2}]$, and the goal is to output a finite number of estimates among which at least one approximates the target mean. In this paper, we consider that the underlying distribution is Gaussian and the target mean is k-sparse. Our main contribution is the first polynomial-time algorithm that enjoys sample complexity $O\big(\mathrm{poly}(k, \log d)\big)$, i.e. poly-logarithmic in the dimension. One of the main algorithmic ingredients is using low-degree sparse polynomials to filter outliers, which may be of independent interest.

1 Introduction

Mean estimation is arguably one of the most fundamental inference tasks in statistics and machine learning. Given a set of samples $\{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ where an α fraction are drawn from some well-behaved (e.g. Gaussian) distribution D and the rest are adversarially corrupted, the goal is to estimate the mean of D. In the noiseless case where $\alpha = 1$, the problem can be easily solved in view of the concentration of measure phenomenon [LT91]. However, this is rarely the case in practice because modern data sets are often contaminated by random noise or even by adversarial corruptions. Thus, a great deal of recent efforts are focused on efficiently and robustly estimating the target mean in the presence of outliers.

Generally speaking, there is a phase transition between $\alpha > 1/2$ and $0 < \alpha \leq 1/2$, and solving either problem in a computationally efficient manner is highly nontrivial. The problem that most of the samples are uncorrupted, i.e. $\alpha > 1/2$, has a very long history dating back to 1960s [Tuk60, Hub64], yet only until recently have computationally efficient algorithms been established [DKK+16, LRV16]. The other yet more challenging regime concerns that an overwhelming fraction of the samples are corrupted, i.e. $\alpha \leq 1/2$, which even renders learning impossible. This motivates the research on *list-decodable* mean estimation [CSV17], where in place of outputting one single estimate, the algorithm is allowed to generate a list of candidates and is considered to be successful if there exists at least one candidate in the list that is sufficiently close to the true mean.

In this work, we investigate the problem of list-decodable mean estimation, for which there has been a plethora of elegant results established in recent years. From a high level, most of them concern error guarantees and running time. For example, [CSV17] proposed the first tractable algorithm based on the ellipsoid method, which runs in polynomial time and achieves optimal error rate for variance-bounded distributions. [DKS18b] developed a multi-filtering scheme and showed that the error rate can be improved by using higher-order degree polynomials if the underlying distribution is Gaussian. The more recent works [CMY20, DKK20a, DKK+21a] further addressed the computational efficiency of this task and achieved almost linear running time.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Although all of these algorithms exhibit near-optimal guarantees on either error rate or computational complexity, it seems that less is explored to improve another yet important metric: the sample complexity. In particular, the sample complexity of all these algorithms is $O(\mathrm{poly}(d))$, hence they quickly break down for data-demanding applications such as healthcare where the number of available samples n is typically orders of magnitude less than the dimension d [Wai19]. Therefore, a pressing question that needs to be addressed in such high-dimensional regime[1] is the following:

> Does there exist a provably robust algorithm for list-decodable mean estimation that runs in polynomial time and enjoys a sample complexity bound of $O(\mathrm{polylog}(d))$?

In this paper, we answer the question in the affirmative by showing that when the true mean μ is k-sparse, i.e. it has at most k non-zero elements, it is attribute-efficiently list-decodable.

Theorem 1 (Main result)**.** *Given the parameter $\alpha \in (0, \frac{1}{2}]$, $\ell \in \mathbb{N}$, failure probability τ, and a set T of $\Omega\left(\alpha^{-7} k^{4\ell+8\ell^2} \cdot \log(\frac{d}{\tau}) \cdot \log^{4\ell}(\frac{dk}{\alpha}\log(\frac{1}{\tau}))\right)$ samples in $\mathbb{R}^d$, of which at least a (2α)-fraction are independent draws from the standard Gaussian $N(\mu, \mathbb{I}_d)$, where $\|\mu\|_0 \leq k$, there exists an algorithm that runs in time $\mathrm{poly}\left(|T|, d\right)$ and returns a list of $O(1/\alpha)$ candidate vectors such that, with probability at least $1-\tau$, the list contains at least one $\hat{\mu} \in \mathbb{R}^d$ with $\|\hat{\mu} - \mu\|_2 = \tilde{O}_\ell\left(\alpha^{-\frac{1}{2\ell}}\right)$.*

Remark 2. The key message of the theorem is that when the true mean is k-sparse, it is possible to efficiently approximate it with $O(\mathrm{poly}(k, \log d))$ samples. This is in stark contrast to existing list-decodable results [CSV17, DKS18b, CMY20, DKK20a, DKK+21a] where the sample complexity is $O(\mathrm{poly}(d))$ – it is unclear how these algorithms can be made attribute-efficient. The only attribute-efficient robust mean estimators are [BDLS17, DKK+19, CDK+21], but their results only hold for the mild corruption regime where $\alpha > 1/2$.

Remark 3. As opposed to an $O(\epsilon)$ error rate obtained for $\alpha > 1/2$, the (non-vanishing) error rate $\tilde{O}_\ell\left(\alpha^{-\frac{1}{2\ell}}\right)$ is typically what one can expect for list-decodable mean estimation under bounded (2ℓ)-th order moment condition, in light of the lower bounds established in [DKS18b].

1.1 Overview of Techniques

The existing techniques for robust mean estimation can be divided into two categories: one is to formulate and solve the problem by convex optimization [DKK+16, CSV17, CMY20, CDK+21]; the other is iteratively filtering the outliers until the samples are sufficiently concentrated [DKK+16, LRV16, DKS18b, DKK+19, DKK20a, DKK+21a]. Our techniques fall into the latter. Among all these prior works, the closest to ours are [DKS18b, DKK+19], which use iterative filtering framework and leverage low-degree polynomials to filter outliers. However, [DKS18b] requires a sample complexity polynomial in the sample dimension d, and [DKK+19] only works under the mild corruption regime. In this work, we propose an attribute-efficient algorithm which addresses overwhelming corruptions.

Sparse polynomials. To ensure that our algorithm is attribute-efficient, all the polynomials we use to filter are sparse, i.e. consisting of at most $\mathrm{poly}(k)$ monomials. To do this, we will measure the discrepancy between the empirical covariance matrix of our sample set and the Gaussian, and identify a subset of coordinates where the discrepancies are the most significant. We then construct a polynomial of which the monomials only include coordinates from the subset. This would be sufficient for us to remove the outliers until the samples become close to Gaussian. Moreover, we develop a set of new results which bound the function values of our sparse polynomials over an L_∞-distance bounded sample set, given that the function value is well concentrated on some Gaussian distribution. This proves to be critical for distinguishing between good samples and outliers.

Quadratic multifilter with Hermite basis. When filtering on a subset of coordinates, finding the exact set of coordinates, i.e. $\Omega \subseteq [d], [d] := \{1, \ldots, d\}$, where the samples behave mostly unlike a Gaussian is an NP-hard problem. To rectify this issue, instead of directly searching for Ω, we will embed the samples in a higher-dimensional space and search for an index set $U \subseteq [d] \times [d]$. Specifically, we will subtract the identity matrix $\mathbb{I}_d$ from the empirical covariance matrix and retrieve the largest entries to construct a *quadratic polynomial*, which should be more sensitive to a small amount of outliers than the linear projections [DKS18b, DKK+19]. This would suffice for removing enough amount of outliers without paying the price of an exhaustive search.

[1] We refer the regime $n \ll d$ as the high-dimensional regime, following convention in statistics.

The key challenge here really lies in how to estimate the variance of a quadratic polynomial on the Gaussian samples, as it depends on the unknown mean (which is actually what we are estimating). In other words, if the empirical mean lies far away from the true mean, the good samples can be scattered by applying the quadratic polynomial to them. This is problematic as the algorithm may filter out a large amount of good samples while keeping most of the outliers. Hence, we need to carefully select appropriate polynomials to filter. That is, we require the quadratic polynomials to be harmonic (consisting of only Hermite basis) in light of an observation in that estimating the variance of a harmonic polynomial is equivalent to estimating the expectation of several relevant multilinear polynomials [DKS18b].

Pruning by L_∞-norm. To bound the function value of a sparse polynomial, we leverage a key property of the Gaussian samples that they are contained in a L_∞-ball with high probability. We thus define a notion of α-good set if at least an α-fraction of it are the good samples and the L_∞-distance between the samples is bounded. Notice that we do not require the input set to be spacially bounded - it only has to include a noticeable α fraction that are drawn from $N(\mu, \mathbb{I}_d)$. We then introduce a clustering step that measures the L_∞-distance between the samples and is guaranteed to generate an initial list of subsets among which at least one is α-good. This would suffice for bounding the function value of any sparse polynomial. Reader can refer to Section 3.3 for more details.

2 Related Works

Breaking the barrier of the typical $O(\mathrm{poly}(d))$ sample complexity bound is one of the central problems across many fields of science and engineering. Motivated by real-world applications, a property termed *sparsity* is often assumed for this end, meaning that only k out of the d number of attributes contribute to the underlying inference problem. In this way, an improved bound of $O(\mathrm{poly}(k, \log d))$ can be obtained in many inference paradigms such as linear regression [CDS98, Tib96, CT05, Don06, PV13, PVY17], learning of threshold functions [Lit87, BHL95, STT12, ABHZ16, ZSA20, SZ21], principal component analysis [Ma13, DKK+19], and mean estimation [BDLS17, DKK+19, CDK+21]. Unfortunately, the success of all these attribute-efficient algorithms hinges on the presumption that the majority of the data are clean, hence they immediately fail to provide any performance guarantee in the list-decodable setting.

Learning with mild corruption ($\alpha > 1/2$). Learning with noise has been extensively studied in a broad context. In supervised learning where a sample consists of an instance (i.e. feature vector) and a label, lots of research efforts were dedicated to robust algorithms under label noise, such as random classification noise [AL87, BFKV96] and Massart noise [Slo88, MN06], where each time the adversary must flip a coin to see if it is allowed to flip the label. Recent years have witnessed significant progress towards optimal algorithms in the presence of label noise, see for example, [KKMS05, ABL17, DKTZ20, ZSA20, DKK+20b] and the many references therein.

The regime that both instances and labels are corrupted turns out to be significantly more challenging. The problem of learning halfspaces under such setting was put forward in 1990s [Val85, KV88], yet only until recently have efficient algorithms been established with dimension-independent noise tolerance [ABL17, DKS18a, She21]. In addition, [BJK15, KKM18, LSLC20] studied robust linear regression and [BDLS17] presented a set of interesting results regarding robustness under various statistical models such as generalized linear models. More in line with this work is robust mean estimation, see the breakthrough works of [DKK+16, LRV16] and many follow-up works [DKK+17, BDLS17, DKS17, SCV18, KSS18, DKK+19, HLZ20, CDK+21].

Learning with overwhelming corruption ($\alpha \leq 1/2$). The agnostic label noise of [Hau92, KSS92] seems the earliest notion that allows the adversary to arbitrarily corrupt any fraction of the data (say 70%), though it can only corrupt labels. Following [CSV17], a lot of recent works have studied the scenario that both instances and labels are grossly corrupted, and the goal is to output a list of candidate parameters among which at least one is a good approximation to the target. This includes list-decodable learning of mixture models [DKS18b, DKK+21b], regression [KKK19, RY20a], and subspace recovery [RY20b, BK21]. [DKP+21] also established a statistical query low bound for list-decodable linear regression that qualitatively matches the performance of existing algorithms. Interestingly, there are some works studying the problem under the crowdsourced learning settings, where the samples are collected from crowd workers and most of them behave adversarially [SVC16, MV18, ZS21].

3 Preliminaries

For a vector v, denote by $\|v\|_2$ its L_2-norm, $\|v\|_1$ its L_1-norm, $\|v\|_0$ its zero "norm" that counts the number of non-zeros, and $\|v\|_\infty$ its infinity norm. For a matrix M, denote by $\|M\|_2$ its spectral norm, $\|M\|_F$ its Frobenius norm and $\|M\|_*$ its nuclear norm. The hard thresholding operator $\mathrm{hard}_k : \mathbb{R}^d \to \mathbb{R}^d$ keeps the k largest elements (in magnitude) of a vector and sets the remaining to zero. Define $[d] := \{1, 2, \ldots, d\}$. For index set $\Omega \subseteq [d]$, v_Ω can be regarded as an $|\Omega|$-dimensional vector or a d-dimensional vector with non-zeros on Ω. For $U \subseteq [d] \times [d]$, denote by M_U the submatrix of M with entries restricted to U.

We reserve the capital letter G for a random draw of $N(\mu, \mathbb{I}_d)$, i.e. $G \sim N(\mu, \mathbb{I}_d)$, where $\mu \in \mathbb{R}^d$, $\|\mu\|_0 \le k$ is reserved for the target mean that we aim to estimate.

3.1 Polynomials

We first define the Hermite basis for standard d-dimensional Gaussian distribution $N(0, \mathbb{I}_d)$ as complete orthonormal for $L^2(\mathbb{R}^d, N(0, \mathbb{I}_d))$ analysis.

Definition 4 (Hermite basis)**.** For any natural number $a \in \mathbb{N}$ and $x \in \mathbb{R}$, the univariate Hermite polynomial is defined as $\mathrm{He}_a(x) = (-1)^a e^{\frac{x^2}{2}} \frac{d^a}{dx^a} e^{-\frac{x^2}{2}}$, which can be extended to a multivariate Hermite polynomial by taking products. That is, for $a \in \mathbb{N}^d$, $x \in \mathbb{R}^d$, d-variate the Hermite polynomial is given by $\mathrm{He}_a(x) = \prod_{i=1}^d \mathrm{He}_{a_i} x_i$, which is of degree $\|a\|_1$.

We then define some useful polynomails. A multivariate degree-l polynomial $\mathrm{Hom}(x)$ is called homogeneous if it only contains monomials of degree exactly l. Similarly, a multivariate degree-l polynomial $h(x)$ is called harmonic if it can be written as a linear combination of multivariate Hermite polynomials of degree exactly l. Formally, we have the following definitions.

Definition 5 (Homogeneous and harmonic polynomials)**.** Let A be a symmetric order-l tensor over $\mathbb{R}^d$. We have

$$\mathrm{Hom}_A(x) = \textstyle\sum_{i_1,\ldots,i_l} A_{i_1,\ldots,i_l} \prod_j x_j^{c_j(i_1,\ldots,i_l)} \tag{3.1}$$

$$\sqrt{l!} \cdot h_A(x) = \textstyle\sum_{i_1,\ldots,i_l} A_{i_1,\ldots,i_l} \prod_j \mathrm{He}_{c_j(i_1,\ldots,i_l)}(x_j) \tag{3.2}$$

where $c_j(i_1, \ldots, i_l)$ denotes the total times that j appears in $\{i_1, \cdots, i_l\}$.

Moreover, if the polynomial has at most κ monomials and only applies on a sparse index set $\Omega \subseteq [d], |\Omega| \le \psi$ of variable x, i.e. $p(x) = p(x_\Omega)$, we say that such a polynomial is sparse.

Definition 6 ((κ, ψ)-sparse polynomial)**.** Consider a polynomial function $p : \mathbb{R}^d \to \mathbb{R}$ that has at most κ monomials and ψ effective coordinates. We call such a polynomial a (κ, ψ)-sparse polynomial.

For a polynomial function $p : \mathbb{R}^d \to \mathbb{R}$, we will denote by $p(T)$ the random variable $p(x)$ where x is drawn uniformly from T. Moreover, the L^2-norm of p reads $\|p\|_2 := \sqrt{\langle p, p \rangle} = \mathbb{E}_{N(0,\mathbb{I}_d)}[p(x)^2]^{1/2}$.

3.2 Representative and Good Sets

Let $\alpha \in (0, \frac{1}{2}]$ be the fraction of good samples in the contaminated dataset T. We say that the set of good samples is representative if it non-trivially represents the underlying distribution $N(\mu, \mathbb{I}_d)$.

Definition 7 (Representative set)**.** For $\alpha \in (0, \frac{1}{2}]$ and $\tau \in (0, 1)$, we say that a set $S \subset \mathbb{R}^d$ is representative (with respect to G) if for any degree-l, $l \le 2\ell$, $(k^{2\ell}, 2\ell k^{2\ell})$-sparse polynomial function p, it holds that

$$|\Pr[p(G) \ge 0] - \Pr[p(S) \ge 0]| \le \frac{\alpha^3}{100(k^{2\ell} \cdot \log(\frac{dk}{\alpha} \log(\frac{1}{\tau})))^l}. \tag{3.3}$$

One basic requirement for a set T being α-good is that at least an α-fraction of the samples are from G, i.e. $\in S$. Apart from that, during the process of multifiltering, we do not want to remove too many samples from S to keep $S \cap T$ representative. For technical reasons, we also require that all the samples within one subset are contained in a small L_∞ ball of radius $O\big(\sqrt{\log(d\,|S|\,/\alpha)}\big)$.

Definition 8 (α-good set)**.** For $\alpha \in (0, \frac{1}{2}]$, a multiset $T \subset \mathbb{R}^d$ is α-good with respect to G if it holds that

1. There exists a set $S \subset \mathbb{R}^n$ which is representative (with respect to G). Moreover, $|S \cap T| \geq \max\{(1-\alpha/6)\,|S|\,, \alpha\,|T|\}$.
2. $\forall x, y \in T$, $\|x-y\|_\infty \leq O\big(\sqrt{\log(d\,|S|\,/\alpha)}\big)$.

We show that it is with high probability that a sufficient large set S of independent samples drawn from $N(\mu, \mathbb{I}_d)$ is representative. The proof is deferred in the appendix.

Proposition 9 (Sample Complexity)**.** *For $\alpha \in (0, \frac{1}{2}]$ and $\tau \in (0,1)$, let S be a set of $N = O\left(\alpha^{-6}(2\ell) \cdot k^{4\ell+8\ell^2} \cdot \log(\frac{d}{\tau}) \cdot \log^{4\ell}(\frac{dk}{\alpha}\log(\frac{1}{\tau}))\right)$ samples from $N(\mu, \mathbb{I}_d)$. Then, S is representative (with respect to G) with probability $1-\tau$.*

3.3 Clustering for the Initial List

We will do a preliminary step of clustering that splits the input set T into an initial list of subsets among which at least one is α-good due to Definition 8. We first claim that the good samples, i.e. $x \in S$ drawn from $N(\mu, \mathbb{I}_d)$, have bounded L_∞ norm with high probability.

Claim 10 (Bounded L_∞-norm)**.** For any $x \in S$, we have that $\|x-\mu\|_\infty \leq \sqrt{2\log(100d\,|S|\,/\alpha)}$ with probability at least $1-\alpha/100$.

We cluster the dataset due to the bounded L_∞ norm of the Gaussian samples. Since $|S| = O\left(\mathrm{poly}(\frac{k}{\alpha}) \cdot \mathrm{poly}\log(\frac{dk}{\tau})\right)$, we requires each subset T_i has L_∞ diameter of $O\left(\log\left(\frac{dk}{\alpha}\log(\frac{1}{\tau})\right)\right)$.

Algorithm 1 CLUSTER

Input: dataset $T \subset \mathbb{R}^d$, parameter $\alpha > 0$.
1. $\mathcal{C} \leftarrow \emptyset$, $\gamma \leftarrow \log\left(\frac{dk}{\alpha}\log(\frac{1}{\tau})\right)$.
2. For each $x \in T$, proceed as follows: If there are $\alpha \cdot |T|$ samples y in T that satisfy
$$\|x-y\|_\infty \leq 2C \cdot \sqrt{\gamma},$$
and no sample $x' \in \mathcal{C}$ has $\|x-x'\|_\infty \leq 6C \cdot \sqrt{\gamma}$, then $\mathcal{C} \leftarrow \mathcal{C} \cup x$.
3. For each $x_i \in \mathcal{C}$, let $T_i = \{y \in T : \|x_i - y\|_\infty \leq 6C \cdot \sqrt{\gamma}\}$
4. Return the list of T_i's.

We show that if T is of size at least $1/\alpha$ times the sample complexity in Proposition 9, by feeding it to Algorithm 1 we obtain at least one α-good subset T_i in the initial list in view of Definition 8.

Lemma 11 (Initial List)**.** *Given $\alpha \in (0, \frac{1}{2}]$ and a set T of $O\left(\alpha^{-7}(2\ell) \cdot k^{4\ell+8\ell^2} \cdot \log(\frac{d}{\tau}) \cdot \log^{4\ell}(\frac{dk}{\alpha}\log(\frac{1}{\tau}))\right)$ samples in $\mathbb{R}^d$ of which a 2α-fraction are independent samples from $N(\mu, \mathbb{I}_d)$, Algorithm 1 returns a list of at most $1/\alpha$ many subsets of T, such that with probability at least $1-\tau$, there exists at least one of them is α-good with respect to G.*

The motivation of bounding the L_∞-distance is to make sure the function value of any $(k^{2\ell}, 2\ell k^{2\ell})$-sparse p is bounded for samples in the α-good subset T_i (technically, also assuming $\mathrm{Var}[p(G)] \leq 1$). This is because when there exist a significant fraction of good samples in T_i, we want to efficiently distinguish the inliers from the outliers. A value-bounded polynomial function will facilitate our analysis of the function variance. Specifically, we have the following guarantee.

Lemma 12. *Suppose T is α-good and for some $(k^{2\ell}, 2\ell k^{2\ell})$-sparse p of degree l, we have $\mathrm{Var}[p(G)] \leq 1$. Then, it holds that $\max_{x,y \in T} |p(x) - p(y)| \leq O\left(\left(2\ell \cdot k^{2\ell} \cdot \log\left(\frac{dk}{\alpha}\log\left(\frac{1}{\tau}\right)\right)\right)^{l/2}\right)$.*

4 The Multifiltering Algorithm

We start with introducing the multifiltering framework that is broadly used in prior works [DKS18b, DKK20a, DKK+21b], followed by our new techniques and algorithmic designs for learning the sparse true mean with attribute-efficient performance.

The multifiltering framework, i.e. Algorithm 2, includes three main steps. The first one is our initial clustering step that generates the intial list L with guarantees in Lemma 11. We then consider a tree of subsets, of which the root is the original contaminated sample set T and each child node of the root represents a subset in L. The multifilter then iterates through the subsets and does one of the following: (1) filter the set by removing outliers and create one child node for it, (2) further split the set into two potentially overlapping subsets and create two children nodes, (3) certify that the set can't be good and delete branch. In the end, if all leaves of the tree can not be further filtered, split or deleted, the mean of the subsets on leaf nodes will be returned as a list M. The final step is a black-box list reduction algorithm that reduces the list size of M from $\mathrm{poly}(1/\alpha)$ to $O(1/\alpha)$.

Algorithm 2 Attribute-Efficient List-Decodable-Multifiltering

Input: Multiset $T \in \mathbb{R}^d$, parameters $\alpha \in (0, 1/2]$, confidence $\tau > 0$.
1. Run CLUSTER(T, α).
2. Let L be the list $\{T_i, \alpha/2\}$, for T_i returned by CLUSTER and let M be the empty list.
3. While L is non-empty:
   (a) Get the first element (T', α') from L and remove it from the list.
   (b) Run ATTRIBUTE-EFFICIENT-MULTIFILTER$(T', \alpha', \tau/|T|)$, and do one of the following:
      (i) If the algorithm returns a vector, add it to M.
      (ii) If it returns a list of (T_i, α_i), remove those with $\alpha_i > 1$, and append the remainings to L.
      (iii) If it returns "NO", proceeds to the next iteration.
4. Run LISTREDUCTION to the list M and return its output.

In the following sections, we will elaborate on our design of the ATTRIBUTE-EFFICIENT-MULTIFILTER, which admits the desired guarantees as in Theorem 1. The black-box list reduction algorithm follows directly from [DKS18b]. We will include in the appendix for reader's reference.

4.1 The Attribute-efficient Multifilter

To achieve attribute-efficient multifiltering, the main multifilter only works on a few attributes of the samples at one time. That is, it fixes an $\Omega \subseteq [d]$ and then only looks at $x_\Omega, \forall x \in T$. For this purpose, Algorithm 3 searches for a sufficient condition of the samples being concentrated such that the empirical mean is a close estimation. Precisely, observing that the eigenvalue of an empirical covariance matrix is exactly the variance of the samples along with its corresponding eigenvector, by searching the largest eigenvalue over all k-sparse support sets we can certify whether there still exist outliers in the sample set.

Algorithm 3 ATTRIBUTE-EFFICIENT-MULTIFILTER

Input: multiset $T \subset \mathbb{R}^d$, parameter α.
1. Let $P_{d,\ell}(x)$ be the vector of all degree-ℓ Hermite basis, i.e. $\prod_{i=1}^d \mathrm{He}_{a_i}(x_i)/\sqrt{a_i!}$. Calculate sample mean $\tilde{\mu} = \mathbb{E}(T)$, and the degree-$\ell$ "covariance matrix" $\tilde{\Sigma} = \mathbb{E}[P_{d,\ell}(T - \tilde{\mu})P_{d,\ell}(T - \tilde{\mu})^\top]$. Let $C > 0$ be a sufficiently large universal constant.
2. Find $k^{2\ell}$ entries in $\tilde{\Sigma}$ with the largest magnitude, of which k^ℓ are on the diagonal and $k^{2\ell} - k^\ell$ are off-diagonal. Let $U \subseteq [d^\ell] \times [d^\ell]$ be the index set of these $k^{2\ell}$ entries. Since $\tilde{\Sigma}$ is symmetric, U will be selected such that the $(\tilde{\Sigma})_U$ is symmetric as well.
3. **If** $\left\|(\tilde{\Sigma})_U\right\|_F \leq C \cdot ((\ell + C\log(\frac{1}{\alpha}))\log^2(2 + \log(\frac{1}{\alpha})))^{2\ell}$, **then** return $\hat{\mu} = \mathrm{hard}_k(\tilde{\mu})$.
4. Set $U' = \{(i, j) \in [d^\ell] \times [d^\ell] : \exists (i, j') \in U \text{ and } (i', j) \in U\}$.
5. Compute largest eigenvalue λ^* of $(\tilde{\Sigma})_{U'}$, and its corresponding eigenvector v^*.
6. **If** $\lambda^* \geq C \cdot ((\ell + C\log(\frac{1}{\alpha}))\log^2(2 + \log(\frac{1}{\alpha})))^{2\ell}$, **then** apply (i) BASICMULTIFILTER if $\ell = 1$, or (ii) HARMONICMULTIFILTER if $\ell > 1$, on the degree-ℓ polynomial $p_1(x) = v^* \cdot P_{d,\ell}(x)$.
7. **Otherwise**, let $A' = (\tilde{\Sigma})_U$, and the normalized $A = \frac{A'}{\|A'\|_F}$. Let $p_2(x) = \frac{1}{\|A'\|_F} \cdot \left(P_{d,\ell}(x)^\top A' P_{d,\ell}(x)\right) = \frac{1}{\|A'\|_F} \cdot p_{A'}(x)$ be the degree-2ℓ harmonic function, run HARMONICMULTIFILTER on $p_2(x)$.

Unfortunately, searching for the largest eigenvalue over k-sparse support sets is a NP-hard problem, which requires iterating over all possible combinations. We instead use the Frobenius norm which

upper bounds the largest eigenvalue, i.e. the spectral norm, of the submatrices. That is, we will simply find an index set U which contains the largest $k^{2\ell}$ entries of $\tilde{\Sigma}$ (in magnitude). If all these $k^{2\ell}$ entries are small, then the Frobenius and spectral norms of any $k^\ell \times k^\ell$ submatrices of $\tilde{\Sigma}$ are also bounded, meaning that the samples behave like a standard Gaussian. As a result, if at Step 3, the algorithm returns a vector $\hat{\mu}$, we can certify that it is sufficiently close to the true mean μ.

Lemma 13. *If T is α-good, when Algorithm 3 returns a vector $\hat{\mu} = \mathrm{hard}_k(\tilde{\mu})$ at Step 3., we have that $\|\mu - \hat{\mu}\|_2 \leq O\left(\alpha^{-\frac{1}{2\ell}}\sqrt{\ell}(\ell + \log(\frac{1}{\alpha}))\log^2(2+\log(\frac{1}{\alpha}))\right)$.*

On the other hand, if Algorithm 3 does not return at Step 3, we show that either Step 6 or Step 7 would make progress such that the algorithm can proceed. First, if the largest eigenvalue λ^* calculated in Step 5 is sufficiently large, the algorithm finds a direction in which the outliers are distinguishable from the inliers. This suffices for the basic multifilter to proceed, i.e. it does not return "YES". In other words, the basic multifilter must either certify that T is not α-good, or return a list of subsets of which at least one is cleaner than the input set. In either way, the algorithm can proceed.

Lemma 14 (Degree-ℓ Harmonic Polynomial)**.** *If Algorithm 3 finds that $\mathrm{Var}[v^* \cdot T] = \lambda^* \geq C \cdot ((\ell + C\log(\frac{1}{\alpha}))\log^2(2+\log(\frac{1}{\alpha})))^{2\ell}$, Algorithm 4 must not return "YES" at Step 4(a).*

The step of filtering by the degree-ℓ harmonic polynomial essentially relies on the condition that the algorithm can find a sufficiently large eigenvalue λ^*. However, the condition can be broken if λ^* is much less than $\left\|(\tilde{\Sigma} - \mathbb{I})_U\right\|_F$, since we use a sufficient but not necessary condition at Step 3. Fortunately, if the algorithm does not find a large λ^*, we show that it can still proceed by calling the harmonic multifilter on a degree-2ℓ polynomial. This is because the degree-2ℓ (originally quadratic when $\ell = 1$) polynomials are more sensitive to a small amount of outliers than the degree-ℓ polynomials [DKS18b, DKK+19]. As a result, we will have the following guarantee.

Lemma 15 (Degree-2ℓ Harmonic Polynomial)**.** *If Algorithm 3 neither returns at Step 3, nor filters at Step 6, then it must proceed with the harmonic multifilter in Step 7.*

Notice that p_1 and p_2 can both be of degree greater than 1. The algorithmic design for higher-degree polynomials is more subtle due to two factors. The first is that the basic multifilter requires the input polynomial p have a small variance on G, i.e. $\mathrm{Var}[p(G)] \leq 1$, when T is α-good. For this reason, Algorithm 5 utilizes a set of multilinear polynomials to verify the variance of quadratic harmonic polynomials. This leads out our second idea of design that the polynomial function p used to filter must consist of only Hermite basis. More details can be found in Section 4.3.

4.2 Basic Multifilter for Sparse Polynomials

The basic multifilter, which is a key ingredient in the multifiltering framework, takes as input a $(k^{2\ell}, 2\ell k^{2\ell})$-sparse polynomial p and use it to certify whether T is α-good and sufficiently concentrated. The central idea is to measure how $p(T)$ distributed and compare it to that of the Gaussian variable G. Therefore, we require the input p has small variance on G, i.e. $\mathrm{Var}[p(G)] \leq 1$, as otherwise, it could filter away a large quantity of the good samples. Notice that when p is linear, this requirement is satisfied due to the property of Guassian. When p is of higher degree, we will invoke the harmonic multifilter which includes additional verification steps.

Given that p is sufficiently concentrated on Gaussian variable G, it remains to find the appropriate threshold at Step 4(b)(i) for filtering. An appropriate threshold is which the algorithm can use it to filter a large amount of the outliers in T, but keeps most of the samples from the representative set S. For this purpose, the parameter in Step 4(b)(i) is set to be $8/\alpha$ times larger than the concentration bound of S (due to Definition 7 and the concentration of Gaussian variables). In other words, each time Step 4(b) is executed, the number of the samples removed from T is $8/\alpha$ times more than that of the good samples in S, which means most of the removed samples are outliers.

As a result, the very novel analysis here is showing that if Algorithm 4 reaches Step 4(b)(i), there must exist a threshold t such that the algorithm can proceed. The idea is to show by contradiction that if such threshold does not exist, the set T must be sufficiently concentrated such that the algorithm will return at Step 4(a). This essentially relies on our result of the initial clustering of Algorithm 1, which guarantees that each subset T is bounded in a small ℓ_∞ ball and the function value of p on the α-good T does not change drastically (Lemma 12). As a result, we have the following guarantees for Algorithm 4. The detailed proof is deferred to the appendix.

Algorithm 4 BasicMultifilter

Input: $(k^{2\ell}, 2\ell k^{2\ell})$-sparse polynomial $p : \mathbb{R}^n \to \mathbb{R}$ of degree l and has $\mathrm{Var}[p(G)] \le 1$, multiset $T \subset \mathbb{R}^d$, parameter α, confidence τ.

1. Let $C > 0$ be the same constant as in Algorithm 3.
2. $R \leftarrow (C \cdot \log(1/\alpha))^{l/2}$, $\gamma \leftarrow \log\left(\frac{dk}{\alpha}\log(\frac{1}{\tau})\right)$.
3. **If** $\max_{x,y\in T} |p(x) - p(y)| \ge (C(1 + 2\ell \cdot k^{2\ell}\gamma))^{l/2}$, then return "NO".
4. **If** there is an interval $[a, b]$ of length $L = C \cdot R \cdot \log(2 + \log(1/\alpha))$ that contains at least $(1 - \alpha/2)$-fraction of samples in $\{p(x) : x \in T\}$, **then**
   (a) **if** $\mathrm{Var}[p(T)] \le C \cdot \left(\ell + C\log(\frac{1}{\alpha})\right)^l \log^2(2 + \log(\frac{1}{\alpha}))$, **then** return "YES".
   (b) **Otherwise**,
      (i) Find a threshold $t > 2R$ such that
$$\Pr_{x\in T}[\min\{|p(x) - a|, |p(x) - b|\} \ge t] > \frac{32}{\alpha}\exp(-(t - 2R)^{2/l}) + \frac{2\alpha^2}{(k^{2\ell}\gamma)^l}.$$
      (ii) Let $T' \leftarrow \{x \in T : \min\{|p(x) - a|, |p(x) - b|\} \le t\}$ and $\alpha' \leftarrow \alpha \cdot \left(\frac{(1-\alpha/8)|T|}{|T'|} + \frac{\alpha}{8}\right)$.
      (iii) Return $\{(T', \alpha')\}$.
5. **Else**
   (i) Find $t \in \mathbb{R}$, $R > 0$ such that the sets $T_1 = \{x \in T : p(x) > t - R\}$ and $T_2 = \{x \in T : p(x) < t + R\}$ satisfy
$$|T_1|^2 + |T_2|^2 \le |T|^2 (1 - \alpha/100)^2$$
$$\text{and } |T| - \max(|T_1|, |T_2|) \ge \alpha |T|/4.$$
   (ii) Let $\alpha_i = \alpha \cdot (1 - \alpha^2/100) \cdot |T| / |T_i|$, for $i = 1, 2$.
   (iii) Return $\{(T_1, \alpha_1), (T_2, \alpha_2)\}$.

Proposition 16 (Basic Multifilter)**.** *Given a set $T \subset \mathbb{R}^d$, parameter $0 < \alpha \le 1$, and a degree-l, $(k^{2\ell}, 2\ell k^{2\ell})$-sparse polynomial $p : \mathbb{R}^d \to \mathbb{R}$ for which it holds that T is α-good then $\mathrm{Var}[p(G)] \le 1$, Algorithm 4 returns one of the following with guarantees: (1) "YES", and we have that (i) $\mathrm{Var}[p(T)] \le O\left((2\ell + \log(\frac{1}{\alpha}))^l \cdot \log^2(2 + \log(\frac{1}{\alpha}))\right)$, and (ii) If T is α-good, then $|\mathbb{E}[p(G)] - \mathbb{E}[p(T)]| \le O\left((2\ell + \log(\frac{1}{\alpha}))^{\frac{l}{2}} \cdot \log(2 + \log(\frac{1}{\alpha}))\right)$. (2) "NO", then T is not α-good. (3) A list of pairs $\{(T_i, \alpha_i)\}$, $T_i \subset T$, of length one or two satisfying the multifilter condition for (T, α).*

4.3 Harmonic Multifilter with Hermite Basis

The purpose of having Step 7 in the main multifilter is to proceed even when no degree-ℓ polynomial is found to have large variance on the set of samples. This step is necessary because even though the largest eigenvalue λ^* on the current index set U is small, it does not necessarily mean that it is small on all $k^{2\ell}$-sparse index sets. Thereby, it is of most importance that Step 7 of Algorithm 3 inevitably filters the set T or certify that T is not α-good given the Frobenius norm of $(\tilde{\Sigma})_U$ is large. For this reason, we construct another sparse harmonic polynomial $p_2(x)$ which can be seens as "projecting" $P_{d,\ell}(x - \tilde{\mu})P_{d,\ell}(x - \tilde{\mu})^\top$ on $(\tilde{\Sigma})_U$ for each $x \in T$. Intuitively, the algorithm must find some x that cause the large deviation of $\tilde{\Sigma}$ from that of Gaussian. Thus, cleaner sample sets must be returned, except that T is verified to be not α-good.

Recall that applying the basic multifilter on a polynomial p requires $\mathrm{Var}[p(G)] \le 1$. This does not come for free for a higher-degree polynomial p (say p_2, which is at least quadratic), as the variance of higher-degree polynomials on the Gaussian variable essentially depends on the distribution mean, i.e. $\mu - \tilde{\mu}$ in this case, which is unfortunately unknown. As a remedy, notice that our p_1 and p_2 are designed to be harmonic in light of an observation in [DKS18b]. Roughly speaking, for any harmonic polynomial $h_A(x)$, $\mathbb{E}_{x\sim N(\mu',\mathbb{I})}[h_A(x)^2]$ is equal to the summation of homogeneous polynomials of μ', which can also be seen as the expectation of multilinear polynomials over independent variables $X_{(i)} \sim N(\mu', \mathbb{I})$. Thus, we only need to verify the expectation of these corresponding multilinear polynomials, whose variance on G does not grow with μ'. The harmonic multifilter is presented in Algorithm 5.

Lemma 17 (Variance of p)**.** *Given multilinear polynomail $A(x_1, \ldots, x_l), x_1, \ldots, x_l \in \mathbb{R}^d$ with $\|A\|_2 = 1$, multiset $T \subset \mathbb{R}^d$, parameter α, and confidence τ, if* MULTILINEAR-MULTIFILTER *returns "YES" correctly and if T is α-good, then we have $\mathbb{E}[p(G - \tilde{\mu})^2] = O\left((2\ell + \log(1/\alpha)) \cdot \log^2(2 + \log(1/\alpha))\right)^l$.*

Due to Lemma 17, we have that $\mathrm{Var}[p(G - \tilde{\mu})/\beta] \leq 1$, for which we can invoke the basic multifilter. This gives our design of Algorithm 5.

Algorithm 5 HarmonicMultifilter

Input: degree-l harmonic polynomial $p(x) = h_A(x)$ with $(k^{2\ell}, 2\ell k^{2\ell})$-sparse A, multiset T, parameter α, confidence τ.

1. For $l' = 0$ to l:

(a) Let $B^{(l')}$ by the order-$2l'$ tensor with

$$B^{(l')}_{i_1,\ldots,i_{l'},j_1,\ldots,j_{l'}} = \sum_{k_{l'+1},\ldots,k_l} A_{i_1\ldots,i_{l'},k_{l'+1},\ldots,k_l} A_{j_1\ldots,j_{l'},k_{l'+1},\ldots,k_l}.$$

(b) Consider $B^{(l')}$ as an $d^{k'} \otimes d^{k'}$ symmetric matrix by grouping each of the $i_1, \ldots, i_{l'}$ and $j_1, \ldots, j_{l'}$ coordinates together. Do eigendecomposition to $B^{(l')}$ such that

$$B^{(l')} = \sum_i \lambda_i V_i \otimes V_i \tag{4.1}$$

(c) Run MULTILINEARMULTIFILTER on each $(T, V_i, l', \alpha, \tau/(ld^l))$. If it returns "NO" or a list of (T, w_i, α), then return the same output.

2. Run BASICMULTIFILTER on $p(x - \tilde{\mu})/\beta$, where $\beta := \left(C \cdot (1 + \log(\frac{1}{\alpha})) \cdot \log^2(2 + \log(\frac{1}{\alpha}))\right)^{\frac{l}{2}}$. If it returns "NO" or a list of (T_i, α_i), return the same output. If it returns "YES", also return "NO".

Lemma 18 (Harmonic Multifilter Must Filter)**.** *If Algorithm 5 reaches Step 2 for p_2, it either certifies that T is not α-good, or returns a list of (T_i, α_i) such that all T_i constain less samples than T.*

To prove Lemma 18, we only need to show that if T is α-good, the basic multifilter at Step 4 must not return "YES". This can be implied from the guarantee 1(ii) of Proposition 16. Specifically, if T is α-good, returning "YES" means that $\mathbb{E}[p(G)]$ is close to $\mathbb{E}[p(T)]$. In light of Lemma 17, $|\mathbb{E}[p(G)]|$ is small implies that $|\mathbb{E}[p(T)]|$ is also small. However, by construction we have that $\left\|(\tilde{\Sigma})_U\right\| = \mathbb{E}[p(T)]$ is large, which is a contradiction. Thus, if the basic multifilter return "YES", we can conclude that T is not α-good and return "NO". This completes the proof for Lemma 15.

5 Conclusion and Future Work

In this paper, we introduce an attribute-efficient mean estimation algorithm which achieves sample complexity polylogarithmic in dimension with low-degree sparse polynomials under the list-decodable learning settings. There is evidence showing that smaller error rates are achievable under the Gaussian assumption by leveraging higher-degree polynomials, which can be an interesting future direction. Another natural question is whether the current techniques could be utilized to attribute-efficiently solve the other interesting learning problems such as learning of halfspaces, polynomial threshold functions, and regression.

References

[ABHZ16] Pranjal Awasthi, Maria-Florina Balcan, Nika Haghtalab, and Hongyang Zhang. Learning and 1-bit compressed sensing under asymmetric noise. In *Proceedings of the 29th Conference on Learning Theory*, pages 152–192, 2016.

[ABL17] Pranjal Awasthi, Maria-Florina Balcan, and Philip M. Long. The power of localization for efficiently learning linear separators with noise. *Journal of the ACM*, 63(6):50:1–50:27, 2017.

[AL87] Dana Angluin and Philip D. Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1987.

[BDLS17] Sivaraman Balakrishnan, Simon S. Du, Jerry Li, and Aarti Singh. Computationally efficient robust sparse estimation in high dimensions. In *Proceedings of the 30th Annual Conference on Learning Theory*, pages 169–212, 2017.

[BFKV96] Avrim Blum, Alan M. Frieze, Ravi Kannan, and Santosh S. Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 330–338, 1996.

[BHL95] Avrim Blum, Lisa Hellerstein, and Nick Littlestone. Learning in the presence of finitely or infinitely many irrelevant attributes. *Journal of Computer and System Sciences*, 50(1):32–40, 1995.

[BJK15] Kush Bhatia, Prateek Jain, and Purushottam Kar. Robust regression via hard thresholding. In *NIPS*, pages 721–729, 2015.

[BK21] Ainesh Bakshi and Pravesh K. Kothari. List-decodable subspace recovery: Dimension independent error in polynomial time. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms*, pages 1279–1297, 2021.

[CDK+21] Yu Cheng, Ilias Diakonikolas, Daniel M. Kane, Rong Ge, Shivam Gupta, and Mahdi Soltanolkotabi. Outlier-robust sparse estimation via non-convex optimization. *CoRR*, abs/2109.11515, 2021.

[CDS98] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[CMY20] Yeshwanth Cherapanamjeri, Sidhanth Mohanty, and Morris Yau. List decodable mean estimation in nearly linear time. In *61st IEEE Annual Symposium on Foundations of Computer Science*, pages 141–148. IEEE, 2020.

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

[CT05] Emmanuel J. Candès and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[DKK+16] Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. In *Proceedings of the 57th Annual IEEE Symposium on Foundations of Computer Science*, pages 655–664, 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. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 999–1008. PMLR, 2017.

[DKK+19] Ilias Diakonikolas, Daniel Kane, Sushrut Karmalkar, Eric Price, and Alistair Stewart. Outlier-robust high-dimensional sparse estimation via iterative filtering. In *NeurIPS*, pages 10688–10699, 2019.

[DKK20a] Ilias Diakonikolas, Daniel Kane, and Daniel Kongsgaard. List-decodable mean estimation via iterative multi-filtering. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems*, 2020.

[DKK+20b] Ilias Diakonikolas, Daniel M. Kane, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. A polynomial time algorithm for learning halfspaces with Tsybakov noise. *CoRR*, abs/2010.01705, 2020.

[DKK+21a] Ilias Diakonikolas, Daniel Kane, Daniel Kongsgaard, Jerry Li, and Kevin Tian. List-decodable mean estimation in nearly-pca time. In *Proceedings of the 35th Annual Conference on Neural Information Processing Systems*, pages 10195–10208, 2021.

[DKK+21b] Ilias Diakonikolas, Daniel M. Kane, Daniel Kongsgaard, Jerry Li, and Kevin Tian. Clustering mixture models in almost-linear time via list-decodable mean estimation. *CoRR*, abs/2106.08537, 2021.

[DKP+21] Ilias Diakonikolas, Daniel Kane, Ankit Pensia, Thanasis Pittas, and Alistair Stewart. Statistical query lower bounds for list-decodable linear regression. In *Processings of the 35th Annual Conference on Neural Information Processing Systems*, pages 3191–3204, 2021.

[DKS17] Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures. In *Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science*, pages 73–84, 2017.

[DKS18a] Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. Learning geometric concepts with nasty noise. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing*, pages 1061–1073, 2018.

[DKS18b] Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. List-decodable robust mean estimation and learning mixtures of spherical gaussians. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1047–1060. ACM, 2018.

[DKTZ20] Ilias Diakonikolas, Vasilis Kontonis, Christos Tzamos, and Nikos Zarifis. Learning halfspaces with massart noise under structured distributions. In *Proceedings of 33rd the Conference on Learning Theory*, volume 125, pages 1486–1513, 2020.

[Don06] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[Hau92] David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

[HLZ20] Sam Hopkins, Jerry Li, and Fred Zhang. Robust and heavy-tailed mean estimation made simple, via regret minimization. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems*, volume 33, pages 11902–11912, 2020.

[Hub64] Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73 – 101, 1964.

[KKK19] Sushrut Karmalkar, Adam R. Klivans, and Pravesh Kothari. List-decodable linear regression. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems*, pages 7423–7432, 2019.

[KKM18] Adam R. Klivans, Pravesh K. Kothari, and Raghu Meka. Efficient algorithms for outlier-robust regression. In *Proceedings of the 31st Annual Conference on Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 1420–1430. PMLR, 2018.

[KKMS05] Adam Tauman Kalai, Adam R. Klivans, Yishay Mansour, and Rocco A. Servedio. Agnostically learning halfspaces. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 11–20, 2005.

[KSS92] Michael J. Kearns, Robert E. Schapire, and Linda Sellie. Toward efficient agnostic learning. In *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory*, pages 341–352, 1992.

[KSS18] Pravesh K. Kothari, Jacob Steinhardt, and David Steurer. Robust moment estimation and improved clustering via sum of squares. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1035–1046, 2018.

[KV88] M. Kearns and L. G. Valiant. Learning boolean formulae or finite automata is as hard as factoring. Technical Report TR 14-88, Harvard University Aiken Computation Laboratory, 1988.

[Lit87] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. In *Proceedings of the 28th Annual IEEE Symposium on Foundations of Computer Science*, pages 68–77, 1987.

[LRV16] Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *Proceedings of the 57th Annual Symposium on Foundations of Computer Science*, pages 665–674, 2016.

[LSLC20] Liu Liu, Yanyao Shen, Tianyang Li, and Constantine Caramanis. High dimensional robust sparse regression. In *The 23rd International Conference on Artificial Intelligence and Statistics,*, volume 108 of *Proceedings of Machine Learning Research*, pages 411–421. PMLR, 2020.

[LT91] Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. Springer-Verlag Berlin Heidelberg, 1991.

[Ma13] Zongming Ma. Sparse principal component analysis and iterative thresholding. *The Annals of Statistics*, 41(2):772–801, 2013.

[MN06] Pascal Massart and Élodie Nédélec. Risk bounds for statistical learning. *The Annals of Statistics*, 34:2326–2366, 2006.

[MV18] Michela Meister and Gregory Valiant. A data prism: Semi-verified learning in the small-alpha regime. In *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 1530–1546. PMLR, 2018.

[PV13] Yaniv Plan and Roman Vershynin. Robust 1-bit compressed sensing and sparse logistic regression: A convex programming approach. *IEEE Transactions on Information Theory*, 59(1):482–494, 2013.

[PVY17] Yaniv Plan, Roman Vershynin, and Elena Yudovina. High-dimensional estimation with geometric constraints. *Information and Inference: A Journal of the IMA*, 6(1):1–40, 2017.

[RY20a] Prasad Raghavendra and Morris Yau. List decodable learning via sum of squares. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms*, pages 161–180. SIAM, 2020.

[RY20b] Prasad Raghavendra and Morris Yau. List decodable subspace recovery. In *Proceedings of the 33rd Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 3206–3226. PMLR, 2020.

[SCV18] Jacob Steinhardt, Moses Charikar, and Gregory Valiant. Resilience: A criterion for learning in the presence of arbitrary outliers. In *Proceedings of the 9th Innovations in Theoretical Computer Science Conference*, pages 45:1–45:21, 2018.

[She21] Jie Shen. Sample-optimal PAC learning of halfspaces with malicious noise. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[Slo88] Robert H. Sloan. Types of noise in data for concept learning. In *Proceedings of the First Annual Workshop on Computational Learning Theory*, pages 91–96, 1988.

[STT12] Rocco A. Servedio, Li-Yang Tan, and Justin Thaler. Attribute-efficient learning and weight-degree tradeoffs for polynomial threshold functions. In *Proceedings of the 25th Annual Conference on Learning Theory*, pages 1–19, 2012.

[SVC16] Jacob Steinhardt, Gregory Valiant, and Moses Charikar. Avoiding imposters and delinquents: Adversarial crowdsourcing and peer prediction. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems*, pages 4439–4447, 2016.

[SZ21] Jie Shen and Chicheng Zhang. Attribute-efficient learning of halfspaces with malicious noise: Near-optimal label complexity and noise tolerance. In *ALT*, volume 132 of *Proceedings of Machine Learning Research*, pages 1072–1113. PMLR, 2021.

[Tib96] Robert Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[Tuk60] John W. Tukey. A survey of sampling from contaminated distributions. *Contributions to probability and statistics*, pages 448–485, 1960.

[Val85] Leslie G. Valiant. Learning disjunction of conjunctions. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 560–566, 1985.

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

[ZS21] Shiwei Zeng and Jie Shen. Semi-verified learning from the crowd with pairwise comparisons. *CoRR*, abs/2106.07080, 2021.

[ZSA20] Chicheng Zhang, Jie Shen, and Pranjal Awasthi. Efficient active learning of sparse halfspaces with arbitrary bounded noise. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [No]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See the appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]